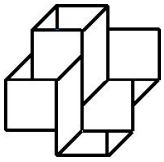

Laboratório Nacional de Computação Científica

Programa de Pós-Graduação em Modelagem Computacional

Generalized Lambda Distribution for Uncertainty Quantification of Large-scale Spatio-temporal Models

Noel Moreno Lemus

Petrópolis, RJ - Brasil

June 2018

Noel Moreno Lemus

Generalized Lambda Distribution for Uncertainty Quantification of Large-scale Spatio-temporal Models

Thesis submitted to the examining committee in partial fulfillment of the requirements for the degree of Doctor of Sciences in Computational Modeling.

Laboratório Nacional de Computação Científica

Programa de Pós-Graduação em Modelagem Computacional

Supervisor: Fábio André Machado Porto

Petrópolis, RJ - Brasil

June 2018

L562g

Lemus, Noel Moreno

Generalized lambda distribution for uncertainty quantification of large-scale spatio-temporal models/ Noel Moreno Lemus. Petrópolis, RJ. : Laboratório Nacional de Computação Científica, 2018.

108 p. : il. ; 29 cm.

Orientador: Fábio André Machado Porto

Tese (Doutorado) – Laboratório Nacional de Computação Científica, 2018.

1. Big data 2. Quantificação de incerteza 3. Modelos espaço-temporais a grande escala I. Porto, Fábio André Machado. II. LNCC/MCTI. III. Título.

CDD: 004.35

Noel Moreno Lemus

Generalized Lambda Distribution for Uncertainty Quantification of Large-scale Spatio-temporal Models

Thesis submitted to the examining committee in partial fulfillment of the requirements for the degree of Doctor of Sciences in Computational Modeling.

Approved by:

Prof. Fábio André Machado Porto, D.Sc.
(Presidente)

Prof. Fernando Alves Rochinha, D.Sc.

Prof. Hugo de La Cruz, D.Sc.

Prof. Artur Ziviani, D.Sc.

Petrópolis, RJ - Brasil
June 2018

Dedication

to my children Leandro and Claudia.

Acknowledgements

My special thanks are due to Professor Dr. Fábio André Machado Porto, who besides being my advisor and mentor, was my teacher in several subjects. His organization, dedication and his great capacity of work have represented for me a constant inspiration source during all this time. I thank him for his guidance, encouragement and for providing me excellent ideas during the development of this work. Without his guidance and persistent help, this dissertation would not have been possible. Actually, I am very pleased to have done my PhD under his mentoring.

I want to express sincere gratitude to all the members of the DEXL LAB at LNCC for its friendship and unconditional support.

To all the professors of the LNCC, to the post-graduation team, for its support; because of the LNCC is a big family.

To my family, to my uncles, my grandmothers and to my eternal protector, to my grandfather, wherever you are this thesis is for you.

To my fathers, they are the real artificers of this job. While the rest of my friends are there in the same place, doing the same things I am here because they always support us to be better. As my mother always says, go ahead, because anyone can go back.

For my brother Edel and my sister Karina, this thesis is also for you, because you are my support, my motivations. In the moments when I think I cannot keep going, I always look at your attitude, how you fight each day, and from there I draw the strength to continue.

My sincere thanks go to my friend, the PhD Kathrin Rodriguez Llanes, who is always available to help, discuss and collaborate, for his very helpful feedback and advice during the development of this work.

To my very best friends Hugo and Yoisell, without its support this thesis is not be possible.

To Mayte, I have no words to express my gratitude. For more than five years, I've always been calm because I know you're in charge of our children. Your support was determinant to this work. I will always be grateful to you for this, you are a Ph.D. too.

To my children, Leandro and Claudia, because they have lived without their father for more than five years, but I always felt their love towards me immensely great.

"Essentially, all models are wrong, but some are useful."
(George Edward Pelham)

Resumo

Simulações espaço-temporais em grande escala com tratamento de quantificação de incerteza permitem que cientistas avaliem com precisão o grau de confiança de suas previsões. Tal incerteza pode ser quantificada ou caracterizada de diferentes maneiras, desde o uso de momentos estatísticos de baixa ordem (o mais comumente usado) até a avaliação de uma PDF (uma abordagem mais completa). Esta última fornece uma descrição mais abrangente da incerteza, levando à decisões conscientes. No entanto, a estimação de PDFs é uma tarefa computacionalmente intensiva. Além disso, devido à heterogeneidade de distribuições em pontos dos espaço-tempo, torna-se difícil o cálculo da incerteza nas regiões do conjunto de dados quando esta é representação por diferentes tipos de PDF.

Nesta tese, propomos um novo método para quantificar a incerteza em modelos espaço-temporais de larga escala baseados na Distribuição Generalizada de Lambda (GLD). O GLD é uma família de PDFs que modela bem a heterogeneidade da incerteza, conforme discutido acima. A GLD é especificada por 4 parâmetros, o que simplifica as comparações entre PDFs, facilitando o processamento analítico, como o *clustering*. Mostramos como o conjunto de dados modelado através de GLDs pode ser usado para responder à consultas, tais como: (*i*) como agrupa a saída do processo de quantificação de incerteza com base na semelhança entre as PDFs?, (*ii*) qual é a incerteza em algumas localizações espaço-temporais não previamente analisadas?, (*iii*) qual é a incerteza de uma região espaço-temporal específica?, (*iv*) como comparar duas regiões em função da sua incerteza?, e (*v*) qual o menos incerto de um conjunto de modelos? O método proposto foi testado em casos de uso reais de várias áreas científicas. Adicionalmente, um pacote R foi implementado com todas as funcionalidades discutidas ao longo da tese.

Keywords: Quantificação de Incerteza, Modelos Espaço-temporais a Grande Escala, Big Data, Generalized Lambda Distribution

Abstract

Large-scale spatio-temporal simulations with quantified uncertainty enable scientists/decision-makers to precisely assess the degree of confidence of their simulation-based predictions. This uncertainty could be quantified or characterized in different ways, from the use of low order statistical moments (the most commonly used), to the evaluation of a complete PDF (a most complete approach). The latter provides a more comprehensive description of the uncertainty leading to aware decisions. However, fitting PDFs to the data is computational intensive. Moreover, due to heterogeneity the uncertainty computed in regions of the dataset is hampered by the representation with different PDF types.

In this thesis, we propose a new method to quantify the uncertainty in large-scale spatio-temporal models based on the Generalized Lambda Distribution (GLD). GLD is a family of PDFs that nicely models the heterogeneity of uncertainty as discussed above. It is specified by 4 parameters that simplifies PDFs comparisons easing analytical processing, such as clustering. We show how the dataset modeled through GLDs can be used to answer queries, such as: (*i*) how to group the output of the UQ process based on the uncertainty similarity?, (*ii*) what is the uncertainty in some spatio-temporal locations not previously analysed?, (*iii*) what is the uncertainty of an specific spatio-temporal region?, (*iv*) how to compare two regions as a function of their uncertainty?, and (*v*) what is the less uncertain model from a set of models? The proposed method has been tested in realistic use cases from various scientific areas. Additionally, an R package has been implemented with all the functionalities discussed in the thesis.

Keywords: Uncertainty Quantification, Large-scale spatio-temporal models, Big Data, Generalized Lambda Distribution

List of Figures

List of Tables

List of abbreviations and acronyms

UQ	Uncertainty Quantification
PDF	Probability Density Function
FP	Forward Problem
QoI	Quantity of Interest
GLD	Generalized Lambda Distribution
p.f.	percentile function
r.v.s	random variables
GLDEX	r package to compute the GLD
M&S	Modeling and Simulation
LSSTM	Large-scale spatio-temporal model
KDE	Kernel Density Estimation
IE	Information Entropy

List of symbols

σ	standard deviation
Θ	random space
θ	random (input) variable
$\mathcal{M}$	mathematical/computational model

Contents

1 Introduction

"A measurement result is complete only when accompanied by a quantitative statement of its uncertainty. The uncertainty is required in order to decide if the result is adequate for its intended purpose and to ascertain if it is consistent with other similar results"

The rapid growth of high-performance computing and the advances in numerical techniques in the last two decades have provided an unprecedented opportunity to explore complex physical phenomena using large-scale spatio-temporal modeling and simulation. At the same time, scientific community is leaving behind the traditional deterministic approach, which offers point predictions with no associated uncertainty (JOHNSTONE et al., 2016); to include Uncertainty Quantification (*UQ*) as a common practice in their researches.

Large-scale spatio-temporal simulations with quantified uncertainty enable scientists to make precise statements about the degree of confidence they have in their simulation-based predictions. These approaches find practical applicability in models for predicting the behavior of weather, hurricane forecasts (TOBERGTE; CURTIS, 2013), subsurface hydrology (BARONI; TARANTOLA, 2014), geology (GUERRA et al., 2016), nuclear reactor design, financial portfolios (CHEN; FLOOD; SOWERS, 2008), and biological phenomena, just to name a few. They also allow to study physical phenomena that are impossible to assess experimentally, for example: simulate nuclear accidents, or the conditions that some spatial vehicle will find at landing in Mars, and so on. The success of these techniques has made them increasingly important tools for high impact predictions and decision making.

UQ includes different aspects that warranty the predictive fidelity of a numerical simulation, such as the uncertainty in the experimental data, which is used for defining the parameter values of a model; the propagation of uncertain parameters through the model; and the choice of the model itself. *UQ* is a complex process that covers the following main tasks: (i) uncertainty characterization (CRESPO; KENNY; GIESY, 2014), also called model calibration (FARRELL, 2015) or statistical inverse problem (ESTACIO-HIROMS; PRUDENCIO, 2012); (ii) sensitivity analysis; (iii) forward problem or uncertainty propagation; and (iv) model selection.

This thesis is focused on *forward propagation*, whose objective is to quantify the uncertainties in model output(s) propagated from uncertain inputs. The targets of *forward propagation* analysis can be: (i) evaluate low-order moments (i.e. mean and variance) of

the outputs, (ii) evaluate the reliability of the outputs, and/or (iii) assess the complete probability distribution (*PDF*) of the outputs.

When dealing with large-scale spatio-temporal models, a huge among of data is generated as a result of the simulation process. Indeed, on each spatio-temporal location $(s_i, t_j) \in \mathcal{S} \times \mathcal{T} \subseteq \mathbb{R}^3 \times \mathbb{R}$, usually more than 10^4 simulations are performed. Then, the size of the output dataset is in the order of $N_s \times N_t \times N_{sim}$, where: N_s is the number of spatial locations, N_t is the number of time steps, and N_{sim} is the number of simulations. An example of the volume of data generated by these simulations is given in the first case study of the Chapter 6, where the output dataset is about 2.4 TB. This turn *forward propagation* in a data intensive problem.

Another important aspect, which is often not taken into account, is that the uncertainty need to be quantified in some way that can be used after, to answer questions that arise in the *UQ* context. In that sense, assess the complete *PDF* could be the best way to quantify uncertainty, because if you can find the *PDF* that best fit the dataset with reasonably accurately, you can get all the statistical properties under one roof. At the same time, we can substitute the original data by the *PDFs*, which represents a huge reduction in the volume of data to manipulate.

Contradictorily, statistical moments (e.g. mean and standard deviation) are possibly the most used ways to quantify the uncertainty, despite the fact that they doesn't have information about the manner in which the data are distributed (LAMPASI; Di Nicola; PODESTA, 2006). This is because of the difficulty to find the *PDF* that best fit a dataset (KARIAN; DUDEWICZ, 2011), even more, when dealing with large-scale spatio-temporal models where the *PDF* needs to be derived on each spatio-temporal location, and therefore the *forward propagation* problem becomes time consuming and computationally intensive too.

However, the use of low order moments alone prevents us from making accurate analysis with respect to the uncertainty. They are not enough neither for the characterization nor for the quantification of the uncertainty, and questions such as:

- What is the uncertainty in the spatio-temporal region $\mathcal{S}_i \times \mathcal{T}_j$ associated to the *QoI* q_k and a computational model $\mathcal{M}_m$?
- How to compare different spatio-temporal regions $\mathcal{S}_i \times \mathcal{T}_j$ with respect to the uncertainty?
- What is the less uncertain model from the set of models $\mathcal{M} = \mathcal{M}_1, \mathcal{M}_2, ...\mathcal{M}_m$, to predict the value of a QoI q_k, over a spatio-temporal region $\mathcal{S}_i \times \mathcal{T}_j$?

can be poorly answered. So, we emphasize that only the characterization of the uncertainty by using the *PDF* allows aware decisions.

A first effort to try to estimate the *PDFs* on large-scale spatio-temporal simulations was done by (LIU; NIU; LIAO, 2018) Ji et. al. in ***Parallel Computation of PDFs on Big Spatial Data Using Spark***. They propose a new solution to efficiently compute the *PDFs* in parallel using Spark, through three methods: data grouping, machine learning prediction and sampling. The main drawback of the proposed approach is that you should try many different distributions, to find the PDF that best fits the dataset at each specific spatio-temporal location. Another drawback is that, as we mentioned above, the uncertainty needs to be quantified in the way that facilitates its further use; and the heterogeneity of the functions used in the approach doesn't facilitate it.

To face these challenges, in this thesis we propose a general framework to quantify the uncertainty in large-scale spatio-temporal models. It uses a data-driven approach and combines the generalized lambda distribution (*GLD*), clusters algorithms and information entropy, for helping researchers to answer the above questions and many others that arise in *UQ* context. Our proposal provides a generally applicable and easy-to-use tool that supports the representation and analysis of uncertainty, as was suggested in the "*Workshop on Quantification, Communication, and Interpretation of Uncertainty in Simulation and Data Science*" (TOBERGTE; CURTIS, 2013).

1.1 Research Objectives

The main objective of this thesis is a new method to quantify the uncertainty in large-scale spatio-temporal models based on the Generalized Lambda Distribution (GLD).

To achieve that goal the following research questions need to be answered:

RQ1. how to group the output of the UQ process based on the similarity of the uncertainty?

RQ2. what is the uncertainty in some spatio-temporal locations not previously analyzed?

RQ3. what is the uncertainty at a specific spatio-temporal region?

RQ4. how to compare two regions as a function of their uncertainty?

RQ5. what is the least uncertain from a set of models?

RQ1 is answered in Chapter 4, while **RQ2**, **RQ3**, **RQ4** and **RQ5** are answered in Chapter 5. In Chapter 6 all the questions are answered again on the context of real use cases.

1.2 Organization of the Dissertation

The structure of the remainder of this thesis is outlined for reference.

Chapter 2 [Uncertainty Quantification Background]. In this Chapter, we review the UQ specialized literature, comment some interesting results and highlight the remainder challenges we are interested in to solving in the present thesis.

Chapter 3 [The Generalized Lambda Distribution]. In this Chapter, we review the principal features of the *GLD* that motivated us to select it as a distribution to characterize the uncertainty in our proposed approach. We highlight some interesting aspects of the shapes of the *GLD*, its flexibility to fit a variate well known distributions and datasets in many different fields. Finally we summarize all the characteristics of the *GLD* that make it the perfect candidate to be used in UQ.

Chapter 4 [Clustering Uncertain Data Based on GLD Similarity]. This chapter explore the how the *GLD* can be used to cluster uncertain data. An algorithm was proposed an tested in two synthetic datasets with excellent results.

Chapter 5 [Our Approach]. Based on the challenges highlighted in Chapter 2, the characteristics of the *GLD* that make it a good candidate to be used in UQ as summarized in Chapter 3, and the results of Chapter 4, in this Chapter we present our approach. All the algorithm of the approach are discuss.

Chapter 6 [Use Cases]. In order to illustrate the use of the proposed framework, two case studies are discussed. The main results obtained are: (i) the *GLD* is a good fits for more than the 80% of the datasets, (ii) the use of the *GLD* allows to include clustering algorithms to group the spatio-temporal locations with similar uncertainty, (iii) the centroids of the clusters can be used as a faithful representation of the rest of the spatio-temporal locations, which significantly reduces the data corresponding to the simulation outputs, (iv) with the use of these centroids we can characterize the uncertainty in any spatio-temporal region as a mixture of *GLDs*.

Chapter 7 [Conclusions and Future Works]. Finally, in this chapter the main contributions of the thesis are highlighted, we revisit the research questions formulated in the Introduction and remark the future directions of this work.

2 Uncertainty Quantification Background

"UQ cannot tell you that your model is 'right' or 'true', but only that, if you accept the validity of the model (to some quantified degree), then you must logically accept the validity of certain conclusions (to some quantified degree)"
(SULLIVAN, 2015)

Uncertainty Quantification (UQ) is a topic of great importance and hence widespread interest in computational analyses that are used to support important societal decisions on issues related to climate change (ALLEN et al., 2000; PATT; KLEIN; VEGA-LEINERT, 2005), hurricane forecasts (TOBERGTE; CURTIS, 2013), subsurface hydrology (BARONI; TARANTOLA, 2014), geology (GUERRA et al., 2016), reactor safety [20-26], radioactive waste disposal [27-34], nuclear weapon safety [35-38], financial portfolios (CHEN; FLOOD; SOWERS, 2008), environmental degradation [44- 47], and many additional areas of concern and challenge.

In this Chapter, we review the UQ specialized literature, comment some interesting results and highlight the challenges we are interested in the present thesis. The rest of the Chapter is organized as follow: in Section 2.1 we define what we understand as uncertainty, the differences between uncertainties and errors, some classifications of the uncertainties, and finally what is uncertainty quantification. In Section 2.2 we introduce the mathematical formalisms used to represent uncertainty. Section 2.3 presents the main problems that UQ covers. Next, in Section 2.4 the two principal *forward propagation* methods are referenced. Section 2.5 reviews the UQ challenges when we are in the presence of large-scale spatio-temporal models; and finally Section 2.6 summarizes the Chapter.

2.1 Definitions

2.1.1 Errors vs Uncertainties

The mismatch between the true physical phenomena and the prediction obtained by modeling and simulation (M&S) process can arise from the mathematical representation of a real problem, a physical problem (are the values of the parameters a good representation of the reality?), a computational problem (translation of a mathematical formulation into a numerical algorithm and a computational code) (MELOROSE et al., 2015). Uncertainty and error can be considered as the broad categories that are normally associated to this mismatch. Until recently the terms uncertainty and error have commonly been used

interchangeably. It is believed, however, that failure to distinguish between these terms is detrimental to the quantification of credibility in M&S. According to (ALVIN et al., 1998) we can classify errors and uncertainties as follow:

- **errors:** recognizable deficiencies of the model or the algorithms employed. Errors are associated to: physical approximations to simplify the modeling of a physical process, translation of the mathematical to computational model, numerical approximations (truncation or roundoff), etc. When the errors are known there are reasonable means of estimating its magnitude of the introduced error.
- **uncertainties:** potential deficiency that is due to lack of knowledge. The different sources of uncertainty can be:
  - **parameter uncertainty**, which comes from the model parameters that are inputs to the computer model (mathematical model) but whose exact values are unknown to experimentalists and cannot be controlled in physical experiments, or whose values cannot be exactly inferred by statistical methods. Examples are the local free-fall acceleration in a falling object experiment, various material properties in a finite element analysis for engineering, and multiplier uncertainty in the context of macroeconomic policy optimization (KENNEDY; O'HAGAN, 2001).
  - **model inadequacy**, no model is perfect. Even if there is no parameter uncertainty, so that we know the true values of all the inputs required to make a particular prediction of the process being modeled, the predicted value will not equal the true value of the process. This discrepancy is considered as model inadequacy. Since the real process may itself exhibit random variability, we define model inadequacy to be the difference between the true mean value of the real world process and the code output at the true values of the inputs.
  - **parametric variability**, which comes from the variability of input variables of the model. For example, the dimensions of a work piece in a process of manufacture may not be exactly as designed and instructed, which would cause variability in its performance.
  - **structural uncertainty**, aka model inadequacy, model bias, or model discrepancy, which comes from the lack of knowledge of the underlying true physics. It depends on how accurately a mathematical model describes the true system for a real-life situation, considering the fact that models are almost always only approximations to reality. One example is when modeling the process of a falling object using the free-fall model; the model itself is inaccurate since there always exists air friction. In this case, even if there is no unknown parameter in the model, a discrepancy is still expected between the model and true physics.

- **algorithmic uncertainty**, aka numerical uncertainty, which comes from numerical errors and numerical approximations per implementation of the computer model. Most models are too complicated to solve exactly. For example, the finite element method or finite difference method may be used to approximate the solution of a partial differential equation, which, however, introduces numerical errors. Other examples are numerical integration and infinite sum truncation that are necessary approximations in numerical implementation.
- **experimental uncertainty**, aka observation error, which comes from the variability of experimental measurements. The experimental uncertainty is inevitable and can be noticed by repeating a measurement for many times using exactly the same settings for all inputs/variables.
- **interpolation uncertainty**, which comes from a lack of available data collected from computer model simulations and/or experimental measurements. For other input settings that don't have simulation data or experimental measurements, one must interpolate or extrapolate in order to predict the corresponding responses.

A more elegant definition of what uncertainty is, is enunciated in (HELTON, 2009) as:

Definition 2.1. Uncertainty is a best estimate of the range of a particular metric which may derive from one or two broad sources. Uncertainties that reflect a lack of knowledge about the appropriate value to use for a quantity that is assumed to have (missing modifier: a fixed?) value in the context of a particular analysis are termed ***epistemic***. Uncertainties that arise from an inherent randomness in the behavior of the system under study are termed ***aleatoric***.

2.1.2 Aleatoric vs Epistemic Uncertainty

It is sometimes assumed that uncertainty can be classified into those two categories, ***aleatoric*** and ***epistemic***, although the validity of this categorization is open to debate (KIUREGHIAN; DITLEVSEN, 2009).

Aleatoric uncertainty arises from an inherent randomness in the properties or behavior of the system under study. For example, the weather conditions at the time of a reactor accident are inherently random with respect to our ability to predict the future. Other examples include the variability in the properties of a population of weapon components and the variability in the possible future environmental conditions that a weapon component could be exposed to. Alternative designations for **aleatory uncertainty** include variability, stochastic, irreducible and type A. (HELTON, 2009)

Epistemic uncertainty derives from a lack of knowledge about the appropriate value to use for a quantity that is assumed to have a fixed value in the context of a particular analysis. For example, the pressure at which a given reactor containment would fail for a specified set of pressurization conditions is fixed but not amenable to being unambiguously defined. Other examples include minimum voltage required for the operation of a system and the maximum temperature that a system can withstand before failing. Alternative designations for **epistemic uncertainty** include state of knowledge, subjective, reducible and type B. (HELTON, 2009)

While **epistemic uncertainty** can be reduced through experiments, improvement of the numerical methods and so on, **aleatory uncertainty** can not be reduced.

2.1.3 Uncertainty Quantification

UQ is not a mature field like linear algebra or single-variable complex analysis, with stately textbooks containing well-polished presentations of classical theorems. Both because of its youth as a field and its very close engagement with applications, UQ is much more about problems, methods and 'good enough for the job'. There are some very elegant approaches within UQ, but as yet no single, general, overarching theory (SULLIVAN, 2015).

UQ neither have a unique and globally accepted definition. In the reviewed literature we find some definitions that, from our point of view, are those that better describe what UQ is.

In the Wikipedia we find the following definition:

Definition 2.2. Uncertainty Quantification is the science of quantitative characterization and reduction of uncertainties in applications. It tries to determine how likely certain outcomes are if some aspects of the system are not exactly known.

This definition is very general and may ignore some important aspects, but as a first approach is a good one.

In October of 2009 the U.S. Department of Energy organized a comission to study the impact of Extreme Scale computing in its National Security. One of the aspects analysed by the comission was UQ. In the report *"Scientific Grand Challenges in National Security: The Role of Computing at the Extreme Scale"* (U.S. Department of Energy, 2009), the authors define UQ as:

Definition 2.3. UQ studies all sources of error and uncertainty, including the following: systematic and stochastic measurement error; ignorance; limitations of theoretical models; limitations of numerical representations of those models; limitations of the accuracy and reliability of computations, approximations, and algorithms; and human error. A more

precise definition is "UQ is the end-to-end study of the reliability of scientific inferences". (U.S. Department of Energy, 2009)

A more recent definition was introduced by Higdon et al. (HIGDON, 2017) in the *"Handbook of Uncertainty Quantification"*:

Definition 2.4. UQ is the rational process by which proximity between predictions and observations is characterized. It can be thought of as the task of determining appropriate uncertainties associated with model-based predictions. More broadly, it is a field that combines concepts from applied mathematics, engineering, computational science, and statistics, producing methodology, tools, and research to connect computational models to the actual physical systems they simulate. In this broader interpretation, UQ is relevant to a wide span of investigations. These range from seeking detailed quantitative predictions for a well-understood and accurately modeled engineering systems to exploratory investigations focused on understanding trade-offs in a new or even hypothetical physical system. (HIGDON, 2017)

Just to remark, in this definition the sentence: **a field that combines concepts from applied mathematics, engineering, computational science, and statistics, producing methodology, tools, and research to connect computational models to the actual physical systems they simulate**, illustrates the multidisciplinar nature of UQ and the main objectives of this research field.

2.2 Uncertainty Representation

An immediate challenge in the development of an appropriate treatment of uncertainty is the selection of a mathematical structure to be used in its representation (HELTON et al., 2010). Traditionally, probability theory has provided this structure. However, in the last several decades, additional mathematical structures for the representation of uncertainty such as evidence theory, possibility theory, fuzzy set theory, and interval analysis have been introduced. This introduction has been accompanied by a lively discussion of the strengths and weaknesses of the various mathematical structures for the representation of uncertainty. For perspective, several comparative discussions of these different approaches to the representation of uncertainty are available.

This section briefly summarizes some of this approaches, and discuss in more details probability theory as this is the main one used in the rest of the thesis.

2.2.1 Interval Analysis

Interval analysis is the simplest way to represent the uncertainty used when nothing more can be said about some unknown quantity than a range of it possible values. All the values have the same probability. For example, in the case of some unknown variable x, such information may be expressed as: $x \in [a, b]$, where a and b represent the left and right limit of the interval. This is a very basic form of uncertainty (SULLIVAN, 2015).

2.2.2 Variance

Suppose that, for a random variable $X \in \mathcal{X}$, the knowledge is summarized by a probability function $p \in P(\mathcal{X})$. The probability measure p is a very rich and high-dimensional object, but sometimes we need to summarize the uncertainty in p, with just one number. Variance is an obvious statistic to summarize this. The formal definition of Variance is:

Definition 2.5. Variance is the expectation of the squared deviation of a random variable from its mean. The variance measures how far each number in the set is from the mean.

If we know the mean m of p, then the Variance can be computed as:

$$\mathcal{V}(p) = \int_{\mathcal{X}} \|x - m\|^2 dp(x) = E_{X\ p}[\|X - m\|^2] \tag{2.1}$$

If $\mathcal{V}(p)$ is small, then we are **relatively certain** that the values of X are quite close to the mean m, and if $\mathcal{V}(p)$ is large, then we are more uncertain about X.

Variance and standard deviation $\sigma = \sqrt{\mathcal{V}(p)}$ are the standard way of quantify the uncertainty of a random variable. This is due to its simplicity of interpretation and easy to compute. But, there are some interesting questions that arise in UQ context that can't be answered correctly by the use of low order statistical moments, as Lampasi et al. (LAMPASI; Di Nicola; PODESTA, 2006) say, only assessing the complete PDF allow aware decisions.

It is clear that assessing the complete PDF of each random variables at each spatio-temporal location, and worse yet, when we are in the presence of large-scale models, is a computational intensive task. But, this is exactly the main objective of this thesis, to present a new workflow to compute the PDF at each spatio-temporal location, and demonstrate its practical use answering the research questions enunciated in Chapter 1.

2.2.3 Information Entropy

The concept of information entropy was first defined by Shannon (1948) in a study performed to identify the amount of information required to transmit English text.

The underlying idea was that, given the probabilities of letters occurring in the English alphabet, it is possible to derive a measure describing the missing information to determine the full text of a partially transmitted message, where information is understood as the one required to identify the message, not the information of the message itself. Based on several theoretical considerations, Shannon derived the following equation to classify a measure of the missing information, often referred to as information entropy:

$$H = -\sum_{i}^{N} p_i \log p_i \tag{2.2}$$

The information entropy H is defined as the sum of the product of the probability p for each possible outcome i of N, total possible outcomes, with its logarithm. The minimum value is 0, because $\log 1 = 0$.

2.2.3.1 Information entropy in a spatio-temporal context

For each spatio-temporal region, the information entropy can be described as:

$$H(s,t) = -\sum_{m=1}^{M} p_m(s,t) \log p_m(s,t) \tag{2.3}$$

where s denotes the location of the subregion, M represents the number of possible (exclusive) members the subregion may contain, and t is the physical time.

2.2.3.2 Information entropy as a meause of uncertainty

Based on 2.2.3 and 2.2.3.1, if the possible outcomes of the model and the probability of each outcome on each (s,t), are known, then the information entropy could be used as a qualitative measure of the uncertainty of the model output (WELLMANN; REGENAUER-LIEB, 2012). For example, in a spatio-temporal region (s,t) where the outcome is always the same, the information entropy is 0, because the outcome is known. On the other hand, in the worse case where all the outcomes have the same probability in (s,t), the entropy is maximum and the uncertainty too.

2.2.4 Probability Theory

Probability theory is based on the specification of a triple $(\Omega, \mathcal{F}, P)$, where Ω is the set of all possible outcomes, $\mathcal{F}$ is a suitably restricted collection of subsets of Ω, and P defines the probability of the elements of Ω.

The probability measure P is a function returning an event's probability, with the properties that $0 \leq P \leq 1$ and $P(\Omega) = 1$

One way to characterize the probability is through the probability density function (*PDF*). It is a mathematical function that, stated in simple terms, can be thought of as providing the probabilities of occurrence of different possible outcomes in an experiment.

Probability density function: for a continuous random variable X, we can define the probability that X is in $[a, b]$ as:

$$P(a <= X <= b) = \int_a^b f(x)dx \tag{2.4}$$

where $f(x)$ is a probability density function, which satisfies two properties:

$$f(x) >= 0$$
$$\int_{-\infty}^{+\infty} f(x)dx = 1$$

a, b are real numbers. The *PDF* defines the probability that $X <= a$ as $P(X <= a) = \int_{-\infty}^{a} f(x)dx$

2.3 Some Typical UQ Problems

Many typical UQ problems can be illustrated in the context of a system F, that maps input X in some space $\mathcal{X}$ to outputs $Y = \mathcal{M}(X)$ in some space $\mathcal{Y}$, through a mathematical/computational model $\mathcal{M}$. Some common UQ objectives include: *forward propagation or push-forward problem*, Section 2.3.1; *reliability or certification problem*, Section 2.3.2; *prediction problem*, Section 2.3.3; *inverse problem*, Section 2.3.4; *sensitivity analysis*, Section 2.3.5; and *model reduction or model calibration problem*, Section 2.3.6. Roughly all of this objectives are summarized in Figure 1.

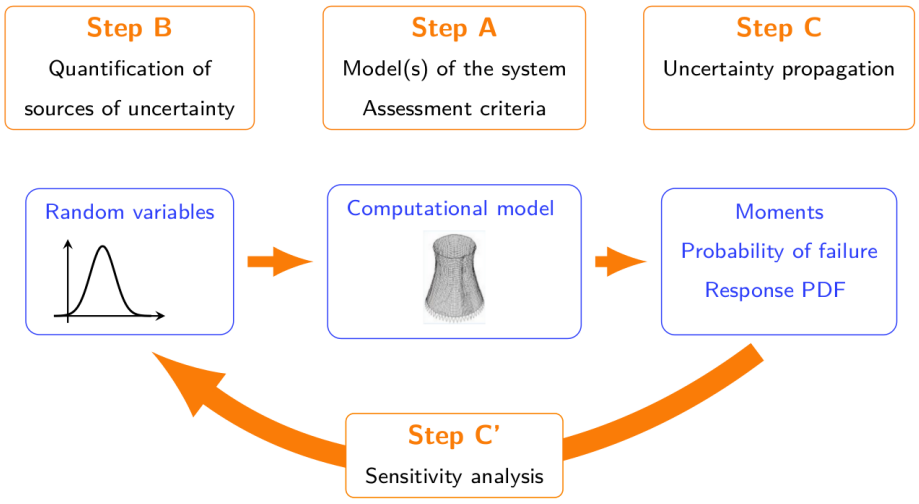

Figure 1 – Uncertainty Quantification workflow. Taken from UQLab.

2.3.1 Forward propagation or push-forward problem

Given the equation $\boldsymbol{Y} = \mathcal{M}(\boldsymbol{X})$ where:

- $\boldsymbol{X} \in \mathcal{X}$ is a vector of input parameters of the model,
- $\mathcal{M}$ is a computational model, and
- $\boldsymbol{Y} \in \mathcal{Y}$ is a vector that represents quantities of interest (QoI).

Suppose that the uncertainty about the inputs of $\mathcal{M}$ can be summarized in a probability distribution P on $\mathcal{X}$. Then in a *forward propagation*, the objective is to quantify the uncertainty of $\boldsymbol{Y}$, induced by $\boldsymbol{X}$ through $\mathcal{M}$.

The main objective of this thesis is a new method to quantify the uncertainty of the output of large-scale spatio-temporal models, that is a *forward propagation problem*. In Section 2.4 some methods for *forward propagation* are exposed; while in Section 2.5 we discus about *forward propagation* in large-scale spatio-temporal models.

2.3.2 Reliability or certification problem

Suppose that some set $\mathcal{Y}_{fail} \subseteq \mathcal{Y}$ is identified as a 'failure set'. Then in a reliability analysis we are interesting in, given appropriate information about the input X and a process F, determine the failure probability

$$\mathcal{P}[\mathcal{M}(X) \in \mathcal{Y}_{fail}] \tag{2.5}$$

Furthermore, how large will the deviation from acceptable performance be, and what are the consequences? (SULLIVAN, 2015)

2.3.3 Prediction problem

Similar to the reliability problem, given a maximum acceptable probability error $\epsilon > 0$, find a set $\mathcal{Y}_\epsilon \subseteq \mathcal{Y}$ such that

$$\mathcal{P}[\mathcal{M}(X) \in \mathcal{Y}_\epsilon] \geq 1 - \epsilon \tag{2.6}$$

in other works, the prediction $\mathcal{M}(X) \in \mathcal{Y}_\epsilon$ is wrong with probability at most ϵ.

2.3.4 Inverse problem or parameter estimation

Given some experimental measurements of the output Y of the system and some computer simulation results from its mathematical model $\mathcal{M}$, inverse uncertainty quantification estimates the discrepancy between the experiment and the mathematical model

(which is called *bias correction*), and estimates the values of unknown parameters in the model if there are any (which is called *parameter calibration*) (Gharib Shirangi, 2014). Generally this is a much more difficult problem than forward uncertainty propagation; however it is of great importance since it is typically implemented in a model updating process.

2.3.5 Sensitivity Analysis

Sensitivity analysis refers to the determination of the contributions of individual uncertain analysis inputs to the uncertainty in analysis results. The goal in sensitivity analysis is to apportion the uncertainty in Y to the uncertainty in inputs X, (SANKARARAMAN, 2012).

2.3.6 Model reduction or model calibration problem

Construct another model $\mathcal{M}_h$ such that $\mathcal{M}_h \approx \mathcal{M}$ in an appropriate sense.

2.3.7 Model selection

If, for the system F we have a set of models $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_n\}$, then a model selection problem consist in the selection of the most plausible model $\mathcal{M}_i$ that best fits the experimental data.

Sometimes a UQ problem consists of several of these problems coupled together, for example, one might have to solve an ***inverse problem*** to produce or improve some model parameters, and then use those parameters to propagate some other uncertainties ***forwards***, and hence produce a ***prediction*** that can be used for decision support in some ***certification problem*** (SULLIVAN, 2015).

In this thesis we focus on ***forward propagation problem*** although in chapter 6 we introduce some queries that help to solve ***reliability or certification*** and ***prediction problem***.

2.4 Methods for Forward Propagation

Two different methods are used to study how the uncertainty is propagated through a computational model, *intrusive* and *non-intrusive*. *Intrusive methods* require the modification of the mathematical/computational model. On the other hand, *non-intrusive methods* consider the mathematical/computational model as a black-box, and therefore the simulation codes don't need to be rewritten. For this reason, non-intrusive methods are attractive (KAWAI; SHIMOYAMA, 2014). The most popular methods for non-intrusive uncertainty propagation are sampling methods, such as Monte Carlo (*MC*) and Latin

hypercube sampling (*LHS*). To date, the MC simulation is the most powerful method for the uncertainty propagation, (RAJAN et al., 2016).

In this thesis we are not interested into the methods themselves because our principal objective is to process the uncertainty data generated as an output of a forward propagation process. To illustrate in a better way the problem, lets consider a *non-intrusive method* as an example.

To estimate a stochastic behavior of the output solution $\boldsymbol{q}$ in terms of input uncertainties $\boldsymbol{\theta}$, the sampling methods analyze the values of $\mathcal{M}(\boldsymbol{\theta})$ at multiple sampled conditions in the Θ space (called stochastic space or parameters space) directly from numerical simulations. Basically, *MC* and *LHS* methods randomly sample in the stochastic space, and hence both require many sample calculations to achieve a convergence of stochastic estimations (although the *LHS* method is more efficient than the *MC* method). As a result, the method returns multiple realizations of $\boldsymbol{q}$, and then other methods to measure the uncertainty, as those proposed in Section 2.2, need to be applied, (BAXTER; COOL, 2016; ESTACIO-HIROMS; PRUDENCIO, 2012; FARRELL; ODEN; FAGHIHI, 2015).

The choise of one method to measure the uncertainty or other depends on the characteristics of each problem and the accuracy we are interesting in. But, as all results of interest can be derived from the **PDF** (COX et al., 2012), our objective in this thesis is to use **PDFs** as the representation of the uncertainty.

Derive a PDF from uncertain data is a fitting process. Given a random sample $q_1, q_2, q_3, ..., q_n$, the basic problem in fitting a statistical distribution to these data is that of approximating the distribution from which the sample was obtained.

Until now we review what uncertainty is, some definitions of uncertainty quantification, how to quantify the uncertainty, and some typical UQ problems. In the next section we explore what happens when we are in presence of a large-scale spatio-temporal model, which is the problem that really motivates this thesis.

2.5 UQ in Large-scale Spatio-temporal models

First to all lets define what is a large-scale spatio-temporal model (**LSSTM**). Although it is an extremely used term in the area, we don't find any exact definition in the reviewed literature. According to the Cambridge Dictionary, the mean of large-scale is:

Definition 2.6. Large-scale: involving many people or things, or happening over a large area.

In the simulation context **happening over a large area** is the most appropriated

term. The spatio-temporal part of the concept is straightforward. Join both ideas the definition of a large-scale spatio-temporal model could be considered as:

Definition 2.7. Large-scale spatio-temporal model: a mathematical/computational model that study the spatio-temporal evolution of a physical system over a large area.

In this context, a computational model of the form: $\boldsymbol{q} = \mathcal{M}(\boldsymbol{\theta})$ represents the spatio-temporal evolution of a complex systems, and the *QoI* $\boldsymbol{q}$ can be represented as:

$$\mathbf{Q} = (\mathbf{q}(s_1, t_1), \mathbf{q}(s_2, t_2), ..., \mathbf{q}(s_n, t_n)) \tag{2.7}$$

where:

- $(s_1, t_1), (s_2, t_2), ..., (s_n, t_n) \in \mathcal{S} \times \mathcal{T} \subseteq \mathbb{R}^3 \times \mathbb{R}$ represent a set of distinct spatio-temporal locations, and
- $\mathbf{q}(s_i, t_j)$ represents a value of the *QoI* at the spatio-temporal location (s_i, t_j)

Many *QoI* can be analyzed, but for simplicity in this thesis we consider only one.

When dealing with **LSSTMs**, a huge amount of data is generated as a result of the simulation process. Indeed, at each spatio-temporal location $(s_i, t_j) \in \mathcal{S} \times \mathcal{T} \subseteq \mathbb{R}^3 \times \mathbb{R}$, usually more than 10^4 simulations are performed. Then, the size of the output dataset is in the order of $N_s \times N_t \times N_{sim}$, where: N_s is the number of spatial locations, N_t is the number of time steps, and N_{sim} is the number of simulations. An example of the volume of data generated by these simulations is given in the first case study of the Chapter 6, where the output dataset is about 2.4 TB. From a computational point of view, this classifies *forward propagation* as a data intensive problem.

This information can be modeled as:

$$S(s_i, t_j, simId, q(s_i, t_j)) \tag{2.8}$$

where $simId$ represents the *id* of one simulation (realization).

The emerging field of data science, nevertheless, is largely lacking in generalizable methods for quantifying the uncertainty in the output of analyzed systems. As a result, a major new research initiative needs to be initiated in this area (TOBERGTE; CURTIS, 2013).

Remember, we are interested in characterizing the uncertainty at each spatio-temporal location through a PDF. If we know, by theoretical considerations, that the distribution at each location is of certain type (e.g. normal, gamma, beta), then through moment matching, one can determine a specific distribution that fits the data at that location, (KARIAN; DUDEWICZ, 2011; Mustafa Inchasi, 2016). This is usually not the

case, even worse in **LSSTM**, because it is impossible to know what could be a distribution type at each location. In those cases it makes sense to use a flexible family of distributions and choose a specific member of that family.

By a **flexible family** we mean one whose members can:

(i) assume a large variety of shapes: skewness in either direction, tails that are truncated or extend to infinity on either or both sides, bell-shaped distributions as well as inverted bell-shaped ones,

(ii) be able to represent a wide range of distributional characteristics such as moments (or combination of moments) or percentiles (or combinations of percentiles),

(iii) to have convenient expressions for at least one of their p.d.f., c.d.f., and quantile function,

(iv) no prior knowledge is needed to fit the distribution to a dataset, which is practical and suitable for automatic and software procedures.

Another challenge that emerges from **LSSTM** is related to the amount of data. It is not always convenient to retain the $N_s \times N_t \times N_{sim}$, (vector) values produced by MC and use them subsequently (COX et al., 2012). This fact introduces another desirable characteristic of the distribution family.

(v) reduce the amount of data to use in subsequent UQ analysis, after the **PDF** is obtained.

In Chapter 3 we present one of this distribution families an discus why this family is suitable to our proposes.

2.6 Summary

Summarizing, in this Chapter we present a review about UQ, showing what is uncertainty quantification; the principal mathematical tools used to represent it; and the typical UQ problems. Immediately we talk about some methods for uncertainty propagation and the problems we are interested in. Basically, we are interested in proposing a new Big Data approach that would allow to compute a **PDF** at each spatio-temporal location of the output of a forward propagation problem.

3 The Generalized Lambda Distribution

"There are good reasons for using the GLD distribution methods... GLD fits have been used successfully in many fields ... Try the GLD first and stop there if the results are acceptable."
(Karian and Dudewicz, 2011)

Fitting statistical distribution to data (real or simulated) is an important task in uncertainty quantification forward problem. When fitting data, one typically first selects a general class, or family, of distributions and then finds values for the distributional parameters that best match the observed data (LAKHANY; MAUSSER, 2000). One of this families is the Generalized Lambda Distribution (*GLD*), originally proposed by Ramberg and Schmeiser in 1974, as a generalization of the Tukey's distribution (1947). The *GLD* has been used in for years to fit data in the most diverse areas such us: analysis of brain MRI scan data, human twin data for quantifying genetic (vs. environmental) variance, rainfall distributions, radiation in soil, velocities within galaxies, exchange rate data for Japanese yen, and so on (KARIAN; DUDEWICZ, 2011). This is due to its flexibility to describe a variety of shapes and to provide good fits to many well known distributions.

Contradictorily, the *GLD* is not extensively used in UQ, despite the features that make it attractive for these purposes. In the present Chapter we show the features that motivated us to select the *GLD* as a distribution to characterize the uncertainty in our proposed approach.

The rest of the Chapter is organized as follow: Section 3.1 presents the different parameterizations of the *GLD*, highlighting the advantages and drawbacks of those parameterizations and justifying the selected one. In Section 3.2 the shapes of the *FMKL* parametrization of the *GLD* are investigated. Next, in Section 3.3 the principal references to the numerical method to estimate the parameters of the *GLD* are shown, with some considerations and comments about future works in this area. Section 3.4 shows how the *GLD* fits many well know distributions. Section 3.5 describes the flexibility of the *GLD* to fit mixture of distributions (bimodal and multimodals). The ability of the *GLD* as a random variate generator is presented in Section 3.6, that reinforces the idea of using it in UQ context. In Section 3.7 we resume all the characteristics of the *GLD* that makes it suitable to UQ. In Section 3.8 we present **GLDEX** an R package that represents the state-of-the-art of the algorithms to work with the *GLD*. Finally in Section 3.9 we summarize the principal results of the Chapter.

3.1 The Generalized Lambda Distribution

The generalized lambda distribution is a continuous distribution defined in terms of its quantile function. Various parameterizations exist (see Section 3.1.3), but the most popular are the proposed by Ramberg and Schmeiser in 1974, Section 3.1.1; and the proposed by Freimer et al. in 1988, Section 3.1.2.

3.1.1 The Ramberg and Schmeiser Parameterization

The Generalized Lambda Distribution (GLD) was proposed by Ramberg and Schmeiser in 1974 as an extention of the Tukey's distribution. It is represented by the quantil function:

$$Q_{RS}(y|\lambda) = Q_{RS}(y|\lambda_1, \lambda_2, \lambda_3, \lambda_4) = \lambda_1 + \frac{y^{\lambda_3} - (1-y)^{\lambda_4}}{\lambda_2} \tag{3.1}$$

where $Q_{RS} = F^{-1}$ is the quantile function for probabilities y, with $y \in [0, 1]$; λ_1 and λ_2 are the location and scale parameters, and λ_3 and λ_4 determine the skewness and kurtosis of the $GLD(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$.

The probability density function of the GLD at the point $x = Q_{RS}(y)$ is given by:

$$f(x) = f(Q_{RS}(y)) = \frac{\lambda_2}{\lambda_3 y^{\lambda_3 - 1} + \lambda_4 (1-y)^{\lambda_4 - 1}} \tag{3.2}$$

In order to have a valid distribution function, the probability density function $f(x)$ need to be positive for all x and integrates to one over the allowed domain:

$$f(x) \geqslant 0 \tag{3.3}$$

$$\int f(x)dx = 1 \tag{3.4}$$

This imposes complex constraints on the parameters and support regions of the RS parameterization, as summarized in table 1 and figure 2.

Table 1 – Support regions of the GLD and conditions on the parameters given by the RS parameterization to define a valid distribution function (KARIAN; DUDEWICZ, 2011). The support regions are displayed in Fig. 2. Note that there are no conditions on λ_1 to obtain a valid distribution.

Region	λ_2	λ_3	λ_4	Q(0)	Q(1)
1	< 0	$<= -1$ $-1 < \lambda_3 < 0$ $\frac{(1-\lambda_3)^{1-\lambda_3}(\lambda_4-1)^{\lambda_4-1}}{(\lambda_4-\lambda_3)^{\lambda_4-\lambda_3}} = \frac{-\lambda_3}{\lambda_3}$	$>= 1$ > 1	$-\infty$	$\lambda_1 + \frac{1}{\lambda_2}$
2	< 0	$>= 1$ > 1 $\frac{(1-\lambda_4)^{1-\lambda_4}(\lambda_3-1)^{\lambda_3-1}}{(\lambda_3-\lambda_4)^{\lambda_3-\lambda_4}} = \frac{-\lambda_4}{\lambda_3}$	$<= -1$ $-1 < \lambda_4 < 0$	$\lambda_1 - \frac{1}{\lambda_2}$	∞
3	> 0	> 0 $= 0$ > 0	> 0 > 0 $= 0$	$\lambda_1 - \frac{1}{\lambda_2}$ λ_1 $\lambda_1 - \frac{1}{\lambda_2}$	$\lambda_1 + \frac{1}{\lambda_2}$ $\lambda_1 + \frac{1}{\lambda_2}$ λ_1
4	< 0	< 0 $= 0$ < 0	< 0 < 0 $= 0$	$-\infty$ λ_1 $-\infty$	∞ ∞ λ_1

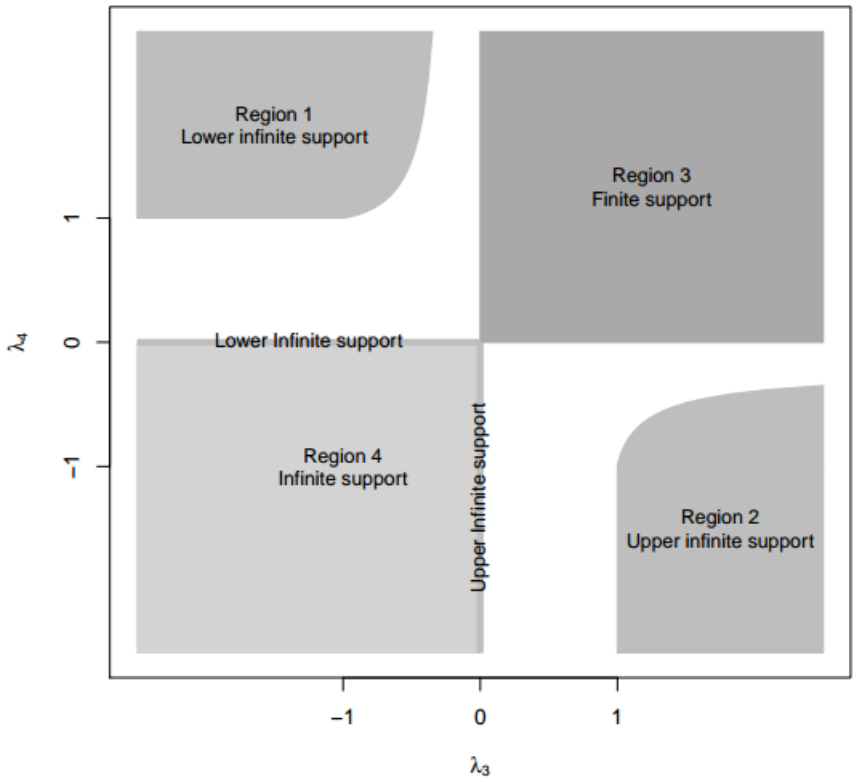

Figure 2 – Support regions of the GLD in the RS parameterization that produce valid statistical distributions.

3.1.2 The FMKL Parameterization

To circumvent the constraints on the *RS* parameter values, Freimer et al. (FREIMER; LIN; MUDHOLKAR, 1988) introduced a new parameterization called *FKML*, equation 3.5.

$$Q_{FMKL}(y|\lambda_1, \lambda_2, \lambda_3, \lambda_4) = \lambda_1 + \frac{1}{\lambda_2}\left[\frac{y^{\lambda_3} - 1}{\lambda_3} - \frac{(1-y)^{\lambda_4} - 1}{\lambda_4}\right] \tag{3.5}$$

As in the previous parameterization, λ_1 and λ_2 are the location and scale parameters, but in this one λ_3 and λ_4 are the tail index parameters. The advantage over the previous parameterization is that the only constraint on the parameters is that λ_2 must be positive. Figure 3 displays the support regions of the *GLD* in the *FKML* parameterization, table 2.

Table 2 – Support regions of the *GLD* given by the *FMKL* parameterization (MARCONDES; PEIXOTO; MAIA, 2017).

λ_3	λ_4	Q(0)	Q(1)
> 0	> 0	$\lambda_1 - \frac{1}{\lambda_2\lambda_3}$	$\lambda_1 + \frac{1}{\lambda_2\lambda_4}$
> 0	≤ 0	$\lambda_1 - \frac{1}{\lambda_2\lambda_3}$	∞
≤ 0	> 0	$-\infty$	$\lambda_1 + \frac{1}{\lambda_2\lambda_4}$
≤ 0	≤ 0	$-\infty$	∞

The probability density function of the *FMKL-GLD* at the point $x = Q_{FMKL}(y)$ is given by (SU, 2015):

$$f(x) = f(Q_{FMKL}(y)) = \frac{\lambda_2}{y^{\lambda_3 - 1} + (1-y)^{\lambda_4 - 1}} \tag{3.6}$$

Although both the *RS* and *FMKL GLD* are generalizations of Tuckey's Lambda Distribution, they are not equivalent, so that the distribution fitted by one parametrization to a dataset differs in general from the one fitted by the other (MARCONDES; PEIXOTO; MAIA, 2017). Both of these representations can present a wide variety of shapes and therefore are utilized in practice; however, generally the *FMKL GLD* is preferred due to the ease in its use (CORLU; METERELLIYOZ, 2016). In this thesis, we also use the *FMKL GLD* representation.

3.1.3 Other Parameterizations

One of the criticisms of the *GLD* is that its skewness is expressed in terms of both tail indices λ_3 and λ_4. In one approach addressing this concern, a five-parameter *GLD* was introduced by Joiner et al. (JOINER; ROSENBLATT, 1971), which, expressed in the *FKML* parameterization, can be written as,

$$Q_{JR}(y|\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5) = \lambda_1 + \frac{1}{2\lambda_2}\left[(1-\lambda_5)\frac{y^{\lambda_3} - 1}{\lambda_3} - (1+\lambda_5)\frac{(1-y)^{\lambda_4} - 1}{\lambda_4}\right] \tag{3.7}$$

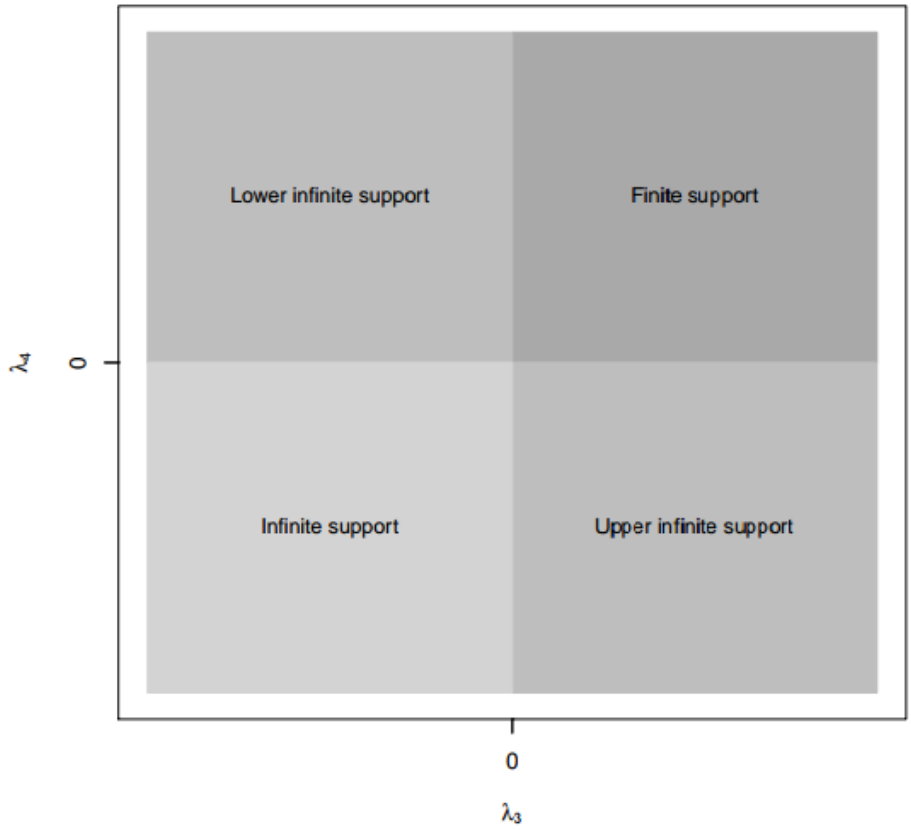

Figure 3 – Support regions of the *GLD* in the *FMKL* parameterization that produce valid statistical distributions.

It has λ_1 and λ_2 as the location and scale parameters, and an asymmetry parameter, λ_5, which weights each side of the distribution and the two tail indices, λ_3 and λ_4. The conditions on the parameters are $\lambda_2 > 0$ and $-1 < \lambda_5 < 1$. The drawback of this parameterization is that the additional parameter can make the estimation of the parameter values even more difficult.

In (CHALABI; DIETHELM; SCOTT, 2012) the authors introduce a new parameterization of the *GLD* that transforms the *FMKL* parameterization, equation 3.5 in terms of an asymmetry and steepness parameter without adding a new variable. Its major advantage is that it provides an intuitive interpretation of its parameters. A new **R** package called **gldist** was implemented with the new *GLD* parameterization, along with the parameter estimation methods they present in their work. The problem with this parameterization is that the **R** package was removed from the official repository because the code is out of date.

3.2 FMKL GLD Shapes

Both *RS* and *FMKL GLD* can describe a variety of shapes, such as: U-shaped, bell shaped, triangular, and exponentially (SU, 2007). At the same time they also provide good fits to many well know distributions. In the case of *RS GLD* an extensive study can be found in (KARIAN; DUDEWICZ, 2011), for the *FMKL GLD* see (FREIMER; LIN; MUDHOLKAR, 1988).

Those properties of the *GLD* are important to our purpose for two reasons: first we don't need previous knowledge to use the *GLD* to fit any dataset, that is exactly the case in large-scale spatio-temporal models, and second the *GLD* can be compared and grouped based on its shapes, that allows us to answer **RQ1** as we show in Chapter 4.

As the *FMKL GLD* parameterization is the one selected to be used in this thesis, in the next sub-sections we present a brief review of its shapes and how this parameterization fits some well-known distributions.

The shape of the *GLD* are dependent on its λ values. In the case of the *FMKL GLD* parameterization, Freimer et al. (FREIMER; LIN; MUDHOLKAR, 1988) classify the shapes into five categories depending on the variety of distributions that can be represented by the several combinations of the shape parameters λ_3 and λ_4. In particular, Class-I ($\lambda_3 < 1, \lambda_4 < 1$) represents unimodal densities with continuous tails. This class is subdivided in I_a ($\lambda_3, \lambda_4 \leq 1/2$), I_b ($1/2 < \lambda_3 < 1, \lambda_4 \leq 1/2$) and I_c ($1/2 < \lambda_3 < 1, 1/2 < \lambda_4 < 1$). In I_a we find distributions such us $Gaussian(Normal)$, $Beta(2.3)$ and $\Gamma(\alpha = 5)$; in I_b $\Gamma(\alpha = 3)$ and $Lognormal(\sigma = 0.5)$; and in I_c distributions as the example of Class-I in Figure 4.

Class-II ($\lambda_3 > 1$, $\lambda_4 < 1$) represents monotone pdfs similar to the *Exponential* distribution, $Beta(1.2)$ or $Lognormal(\sigma = 1.0)$. Class-III ($1 < \lambda_3 < 2$, $1 < \lambda_4 < 2$) represents U-shaped densities with truncated tails, Class-IV ($\lambda_3 > 2$, $1 < \lambda_4 < 2$) represents S-shaped densities, and Class-V ($\lambda_3 > 2$, $\lambda_4 > 2$) represents unimodal densities with truncated tails. Figure 4 provides the shapes that are represented by the parameters indicated in Table 3.

Table 3 – Examples of the five categories of distributions the *FMKL GLD* can represent.

	λ_1	λ_2	λ_3	λ_4
Class-I	0	1	0.5	0.6
Class-II	0	1	2	0.5
Class-III	0	1	1.5	1.5
Class-IV	0	1	2.5	1.5
Class-V	0	1	3	3

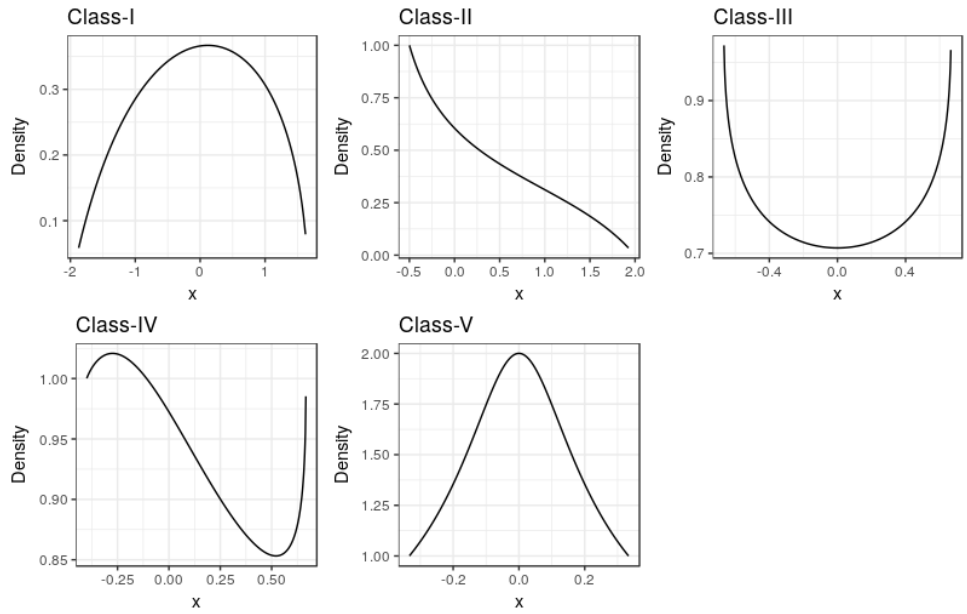

Figure 4 – Examples of the five categories of shapes the *FMKL GLD* can represent.

Figure 5 shows the five categories of the *FMKL GLD* shapes in (λ_3, λ_4) space. There are two regions in this figure that were left out of the analysis, the regions with $(\lambda_3 < 1,\ \lambda_4 > 1)$ and $(1 < \lambda_3 < 2,\ \lambda_4 > 2)$. Those regions are symmetric to region II and IV respectively, see Figure 6.

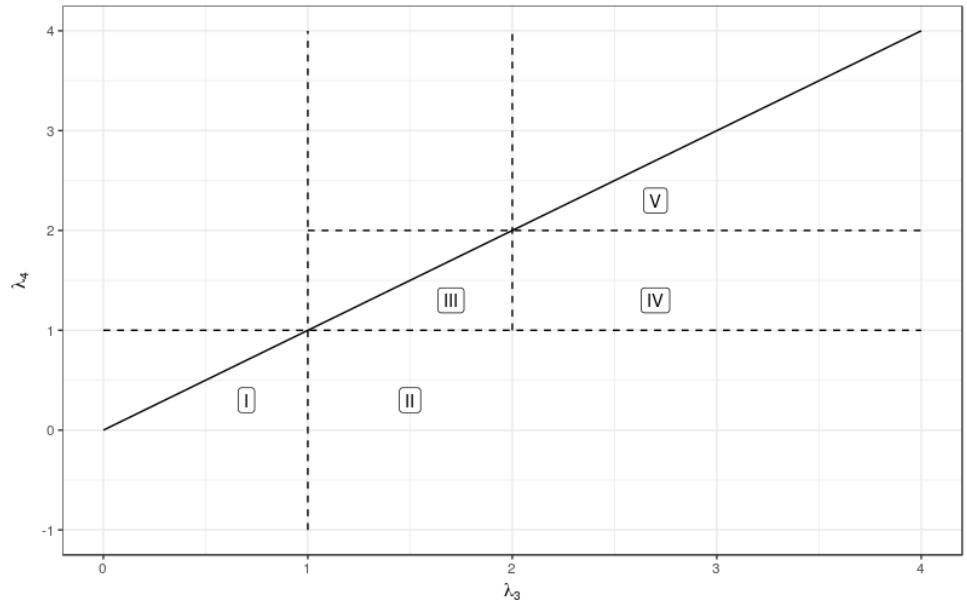

Figure 5 – The five categories of shapes of the *FMKL GLD* in the (λ_3, λ_4) space.

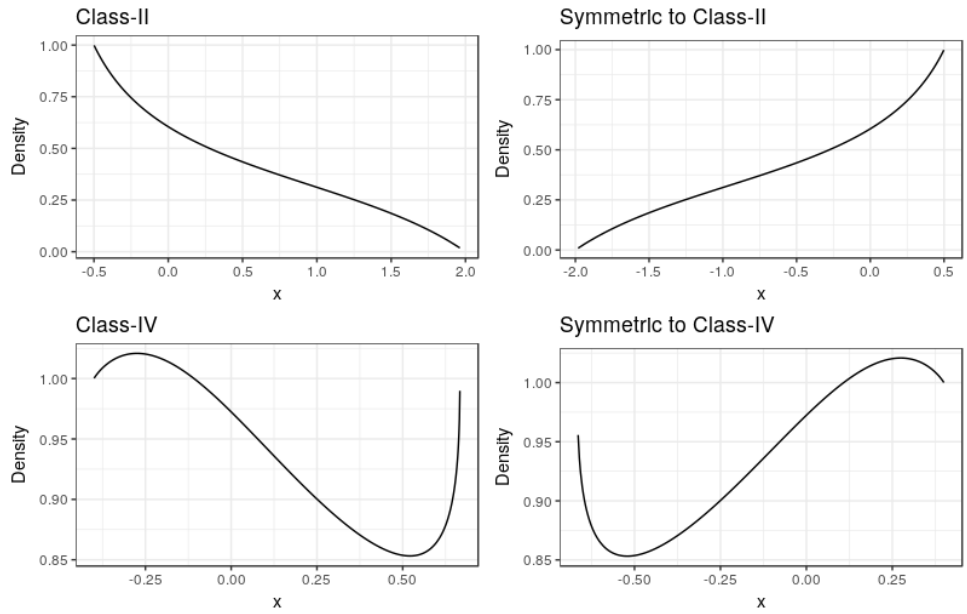

Figure 6 – Symmetry of the regions ($\lambda_3 < 1$, $\lambda_4 > 1$) and ($1 < \lambda_3 < 2$, $\lambda_4 > 2$) with respect to region II and IV.

The results presented in this section are extremely important to our approach because they are the basis for the clustering algorithms we propose in Chapter 4.

3.3 Numerical Methods to Fit the GLD to Data

Given a random sample $x_1, x_2, x_3, ...x_n$, the basic problem in fitting a statistical distribution to this data is that of approximating the distribution from which the sample was obtained. If it is known, because of theoretical considerations, that the distribution is of a certain type (e.g., a gamma distribution with unknown parameters), then through moment matching, or some other means, one can determine a specific distribution that fits the data. This, however, is generally not the case and, in the absence of any knowledge regarding the distribution, it makes sense to appeal to a flexible family of distributions and choose a specific member of that family (KARIAN; DUDEWICZ, 2011).

There are two different parameter estimation philosophies, **direct estimation methods**, such as least-squares estimation with order statistics and with percentiles (FOURNIER et al., 2007; KARIAN; DUDEWICZ, 2011); the methods of moments (LODZIENSIS, 2013), L-moments (KARVANEN; NUUTINEN, 2008), and trimmed L-moments (FOURNIER et al., 2007); and the goodness-of-fit method with histograms (**??**) and with maximum likelihood estimation (SU, 2007). On the other side, **stochastic methods** have been introduced with various estimators such as goodness-of-fit (LAKHANY; MAUSSER, 2000) or the starship method [King and MacGillivray, 1999].

Without doubts the major contributions in the implementation of parameter

estimation algorithms are due to Steve Su (SU, 2007; SU, 2011; SU, 2015; SU, 2016), that besides the theoretical contributions is the author of the state-of-the-art R package to work with the *GLD*. A brief review of this package is presented in Section 3.8, as this package is the one we use in this thesis to solve many task related to the *GLD*.

Out of the two estimation philosophies presented above, (CORLU; METEREL-LIYOZ, 2016) a genetic algorithms approach to estimate the parameters of the *GLD* was introduced.

Recently Marcondes at al. (MARCONDES; PEIXOTO; MAIA, 2017) present a new parameterization of the *GLD* with its respective numerical methods. The main contribution of this paper is that the new parameterization allows fitting the *GLD* to highly skewed data, with a great number of zeros and heavy tails.

The methods to fit the *GLD* to data are out of the scope of this thesis, as we are interested in demonstrating its usability in UQ. Nevertheless it is important to remark that, fit the *GLD* to data is computationally intensive but suitable to parallelization, we haven't found any work in the literature to explore the possibility of increasing the performance of the fitting process by means of parallelization. This is an open problem we are interested in exploring in the future.

3.4 GLD Approximations of Some Well-Known Distributions

For the $GLD(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ to be useful for fitting distributions to data, it should be able to provide good fits to many distributions. In (KARIAN; DUDEWICZ, 2011), the authors explore how the *GLD* fit sixteen well-known distributions using the *RS GLD* parameterization. Here we explore how the *GLD* fit eight distributions, but using the *FMKL* parameterization.

In Table 4 we show the original distribution, the four λ values of the fit and the result of applying a Kolmogorov-Smirnov test to validate the good of the fit.

Table 4 – GLD Approximations of 8 Well-Known Distributions

Distribution	Parameters	λ_1	λ_2	λ_3	λ_4	KS-test
Normal	$N(0, 1)$	-0.04263	1.49039	0.13787	0.12571	951
Uniform	$U(0, 1)$	0.46250	2.16223	1.00008	0.8614	912
Exponential	$\theta = 1$	0.49150	1.40546	1.44813	-0.10419	923
Chi-Square	$nu = 5$	4.141853	0.486702	0.508298	-0.045440	911
Gamma	$(\alpha = 5, \theta = 3)$	1.535078	1.846312	0.410183	0.027492	885
Weibull	$(\alpha = 1, \beta = 5)$	2.684657	0.263865	1.413826	-0.067658	940
Lognormal	$(\mu = 0, \sigma = 1/3)$	0.984696	4.516254	0.324879	-0.074348	903
Beta	$(\beta_3 = \beta_4 = 1)$	0.50092	2.00702	0.99505	1.00060	906

As expected the results are slightly different to those presented by Karian et al., as we use a different parameterization. The KS-test value was over 900 in seven cases and near 900 in the case of the Gamma distribution. This result suggests that the fit was good. All fits provide (λ_3, λ_4) values that match the shape region each distribution belongs to.

3.5 Fitting Mixture Distributions Using a Mixture of Generalized Lambda Distributions

In general, a *mixture distribution* is the probability distribution of a random variable that is derived from a collection of other random variables. Mathematically, given a finite set of *PDFs* $p_1(x), p_2(x), \ldots, p_n(x)$, and weights $w_1, w_2, \ldots, w_n$ such that $w_i \geq 0$ and $\sum w_i = 1$, the mixture distribution can be represented by writing the density $f(x)$ as a sum (which is a convex combination):

$$f(x) = \sum_{i}^{n} w_i p_i(x) \tag{3.8}$$

Since its introduction by Karl Pearson in 1894 *mixture distribution* are extensively used, but in the majority of the cases using normal mixtures. The advantage of using the *GLD* family is that the *GLD* can fit the normal distribution well. Hence whenever a mixture of normals would fit data well, so would a mixture of at most the same number of *GLDs*. Meanwhile, the *GLD* family is a much broader family, and can do well in cases where the normal cannot (NING; GAO; DUDEWICZ, 2008). Due to the versatile and rich shapes of the *GLD*, they are particularly suited for mixture modeling as they eliminate the need to choose between a wide range of different distributions on the same data set (SU, 2007).

The *GLD* mixture distribution can be represented as:

$$f(x) = \sum_{i}^{n} w_i GLD_i(\lambda_1, \lambda_2, \lambda_3, \lambda_4) \tag{3.9}$$

Some studies that compare the use of *GLD* mixtures with normal mixtures are presented by Ning et al. (NING; GAO; DUDEWICZ, 2008), where the author shows that the mixture of *GLDs* performs as well as the mixture of normal distributions, and sometimes even better. In (SU, 2011) the author presents examples of fitting bimodal and trimodal data with mixture of *GLDs*, again with excellent results. Numerical methods to fit mixture of *GLDs* to data are discused in (SU, 2007; SU, 2011), while the implementations are part of the **GLDEX** R package (SU, 2007).

We return to *GLD* mixtures in Chapter 5, where we present how to answer the **RQ.3** by the use of a mixture of *GLDs*.

3.6 GLD Random Variate Generation

An important thing to take into account when we substitute the raw data produced as an output of a simulation process, by its *PDF* is that the latter needs to allow us to reproduce the original data as close as possible. The outcome produced by a particular *PDF* is known as random variate, its definition is:

Definition 3.1. A **random variate** is a particular outcome of a random variable. The random variates which are other outcomes of the same random variable might have different values.

Random variates are used on simulating processes driven by random influences. One of the important applications of the *GLD* has been the generation of random variables for Monte Carlo studies (Mustafa Inchasi, 2016).

This fact is justified by the following theorem, enunciated by Karian and Dudewicz (2010).

Theorem 3.2. *If $Q_X(y)$ is the percentile function of a random variable X, and U is a uniform random variable on $(0, 1)$ then $Q_X(U)$ has the same PDF as does X.*

For a proof, also see p. 156 of Karian and Dudewicz (1999). The percentile function is not available in a closed (or easy-to-work-with) form for many of the most important distributions, such as the normal distribution. However, the GLD is (see sections **??** and **??**) defined by its p.f., which is a simple-to-calculate expression.

Thus, **r.v.s for a simulation study can easily be generated from any distribution that can be modeled by a GLD.**

Example 3.3. Suppose we have modeled an important **r.v.** by an approximate standard normal distribution X. We show in Section 3.4 that a close fit to the standard normal is available via the *RS-GLD* with

$$(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (0, 0.1975, 0.1349, 0.1349) \tag{3.10}$$

and this *GLD* has **p.f.**

$$Q(y) = \frac{y^{0.1349} - (1-y)^{0.1349}}{0.1975} \tag{3.11}$$

Thus, if $U_1, U_2, ...$ are independent uniform **r.v.s** on $(0, 1)$, then

$$Q(U_1), Q(U_2), ... \tag{3.12}$$

are independent and (approximately) $N(0, 1)$ **r.v.s** for the simulation study at hand.

This theorem means that, independently of the nature of the dataset (Normal, Exponential, etc.), when we fit a GLD to it, we can proceed similarly to the example

above. That is, we just need to generate a stream of independent uniform **r.v.s** on $(0, 1)$, and then evaluate the equation 3.12. There are a number of good sources of independent uniform r.v.s on $(0, 1)$ (KARIAN; DUDEWICZ, 2011).

This is an important property of the *GLD* that allows us to substitute the raw data by the four lambdas of the *GLD* that best fit it (if the fit is a good one), with the warranty that if we need to go back, the *GLD* would generate a good representation of the original data.

It is evident that the *GLD* allows easy generation of random variables from every kind of distribution, because featuring an explicit and accessible $Q_X(y)$ reduces it to a uniform generation in $[0, 1]$, (LAMPASI; Di Nicola; PODESTA, 2006).

3.7 GLD and Uncertainty Quantification

From the best of our knowledge, the first effor to use the *GLD* in UQ is due to Lampasi et al. (LAMPASI; Di Nicola; PODESTA, 2006) in the paper *"Generalized Lambda Distribution for the Expression of Measurement Uncertainty"*. In this paper the authors argue why the *GLD* is suitable for UQ, (i) how the use of the *GLD* to represent the uncertainty both at the input and the output of the models does help to homogenize the information we are processing in UQ worflows, (ii) the *GLD* allows easy generation of random variables from every kind of distribution (see Section 3.6), (iii) the amount of data we need to store is extremely smaller than required by histograms or by the approximation presented in (JCGM, 2008), as a *GLD* is fully described by its four parameters, and (iv) how their introduction is practical and suitable for automatic and software procedures, as required by the industrial standards.

Later Cox et al. (COX et al., 2012) comment that as the *GLD* is defined with respect to its quantile function, drawing random samples from the resulting model approximation is straightforward, and then its use is suitable because it is not always convenient to retain the among of values produced by MC simulations and use them subsequently. Then we can substitute the raw data generated in MCS by the *GLDs* that represent this raw data.

A more general reference of the use of the *GLD* in UQ is due to Hack et al. in (**??**). They do a literature review about UQ and include the *GLD* as a very interesting option to characterize the uncertainty.

In (MOVAHEDI; LOTFI; NAYYERI, 2013), a solution to determine the reliability of products using the *GLD* is presented. The novelty here is because of the variability of distributions of the products they need a flexible distribution family that allows to fit those distributions without previous knowledge.

Finally, in (RAJAN et al., 2016) the authors present a benchmark test distributions

for expanded uncertainty evaluation algorithms. Between many other distributions, the *GLD* is included because of its potential use in UQ.

3.7.1 Relevance of GLD in Uncertainty Quantification

The use of the *GLD* to quantify the uncertainty is justified because:

- the *GLD* fits the *PDF* of a wide variety of datasets, including those that follow distributions such as normal, uniform, Student's t, U-shaped, exponential, etc;
- no prior knowledge is needed to fit the *GLD* to a dataset, which is practical and suitable for automatic and software procedures;
- the *PDF* is completely characterized by the four parameters of the *GLD*, which represents a reduction in the amount of data that must be stored for post-processing;
- the shape of the *GLD* is governed by its parameters, so the *GLDs* can be grouped based on their shapes, which is especially useful for further queries;
- in cases where mixture of distributions are needed, *GLD* mixtures could be a very good option; and
- the *GLD* allows easy generation of random variables from every kind of distribution.

3.8 The GLDEX R package

In the implementation of our approach, we use the ***GLDEX***[1] ***R*** package (SU, 2007). The GLDEX R package provides fitting algorithms with two objectives: (i) to provide a smoothing device to fit distributions to data using the weight and unweighted discretised approach based on the bin width of the histogram; (ii) to provide a definitive fit to the data set using the maximum likelihood estimation.

The GLDEX package also provides diagnostic tests to examine the quality of fit through the resample Kolmogorov-Smirnoff test, quantile plots and comparison of the mean, variance, skewness and kurtosis between the empirical data and the fitted distribution.

The GLDEX package is used in this thesis to: (i) fit the *GLD* distribution to a dataset on each spatio-temporal location; (ii) examine the quality of the fit; (iii) sampling any spatio-temporal location based on its *GLD*.

[1] https://cran.r-project.org/web/packages/GLDEX/index.html

3.9 Summary

The main contribution of this Chapter is to summarize the relevance of *GLD* in UQ. Besides that, we select the *FMKL* parameterization as the one we use in this thesis, because it is more general than the *RS-GLD*. The shapes of the *FMKL-GLD*, the capacity to describe many well know distributions and the fact that *GLD* mixtures outperform normal mixtures, allow us to conclude that the *GLD* is a good candidate to be used in our approach.

The *GLD* also fits the five requirements enunciated in Section 2.5 of Chapter 2, as the desirable characteristics of a flexible family, to be used in the quantification of uncertainty in **LSSTM**.

4 Clustering Uncertain Data Based on GLD Similarity

In Chapter 3 we exposed the two most important parameterizations of the *GLD* and select the *FMKL* as the one to be used for the rest of the thesis. In this parameterization λ_1 represent the location of the *GLD* and is directly related to the mean of the distribution. λ_2 is the scale, directly related to the standard deviation; and λ_3 and λ_4 represent the left and right tails of the distribution. Combinations of λ_3 and λ_4 can be used to estimate the skewness and kurtosis of the distribution.

As λ_2 define the dispersion, and λ_3 and λ_4 the shape of a *GLD*, so the combination of those parameters are the responsible of the quantification of the uncertainty, from the *GLD* point of view.

The **RQ.1** we formulate in the Introduction (see Chapter 1) is:

> **RQ1.** how to group the output of the UQ process based on the similarity of the uncertainty?

If the uncertainty in a *GLD* is characterized by λ_2, λ_3 and λ_4 and we are interesting into group the uncertainty based on the similarity, is clear that we need to explore how to group the *GLDs* based on its λ values. This is the main objective of this Chapter, that is organized as follow: in Section 4.1 a brief review of some related works is performed, and the advantage and drawbacks of those works are highlighted. Some considerations about the possibilities of the use of the *GLD* to solve some of the drawbacks are commented. Next, in section 4.2 our hypothesis about the use of the *GLD* to clustering uncertain data, is presented. Sections 4.3 and 4.4 present two synthetics datasets and the results of the clustering technique. Those results help us to validate our hypothesis. Finally, section 4.5 summarize and discuss the main results of the Chapter.

4.1 Related Works

Clustering uncertain data is recognized as an essential tasks in mining uncertain data (JIANG et al., 2011). It impose significant challenges in both modeling similarities between uncertain objects and developing efficient computational methods. Many methods are implemented as extensions of k-means and density-based clustering methods like DBSCAN, but using geometric distances between objects. Such methods cannot handle uncertain objects that are geometrically indistinguishable, such as products with the same

mean but very different variances, as the uniform and normal distributions in Figure 7.

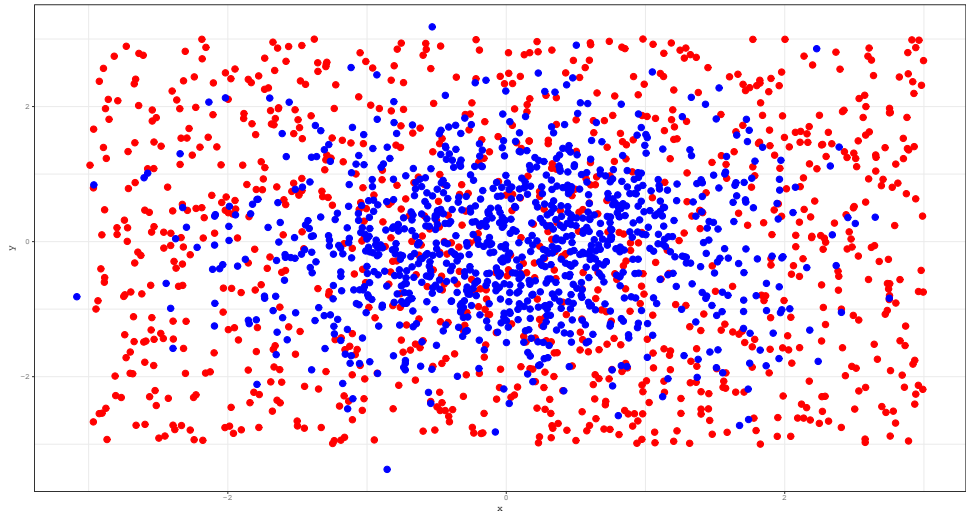

Figure 7 – Two geometrically indistinguishable distributions. Both distributions have the same mean but different variances. In color blue a bivariate Gaussian distribution, in red a bivariate uniform distribution.

Until recently, probability distributions, which are essential characteristics of uncertain objects, had not been considered in measuring the similarity between uncertain objects. In the last years, a plethora of new methods to clustering uncertain data emerge, based in the use of statistical metrics like Kullback-Leibler divergence (KL-divergence). From the best of our knowledge, the first effort in this direction was the work of Jiang et al. (JIANG et al., 2011), where they propose a new clustering method based on the use of Kernel Density Estimation (KDE) to fit a *PDF* to the uncertain data, and modifications to the k-means algorithm to use the KL-divergence as a distance function. Some improvements to the computational cost of the previous method were introduced by the same authors in (JIANG et al., 2013).

More recently Lui et al. (LIU; NIU; LIAO, 2018) present some contributions to the k-means algorithm to clustering uncertain objects but with unsatisfactory results. The proposed approach could be useful just to find the optimal number of clusters.

Basically, those methods are two steps methods: (i) first a method to find the *PDF* that best fit the uncertain objects, and (ii) the clustering algorithm to group the uncertain objects using an statistical metric as a distance function. Two major drawbacks can be remarked here: first, as we mention in Chapter 3, find the *PDF* that best fit an uncertain object is not so easy. And second the computation of any statistical distance between many objects is a time consuming and computationally intensive task. Other restrictions are also imposed to some methods, for example in (JIANG et al., 2011) all the *PDF* need to be defined in the same domain.

Some of those drawbacks can be solved by the use of the *GLD*. For example, the *GLD* is defined in the same domain, we don't need any previous knowledge to fit the *GLD* to an uncertain object. In the next section we formulate an hypothesis about how the *GLD* can be used to clustering uncertain data.

4.2 Clustering Based on GLD

According to Lampasi et al. (LAMPASI; Di Nicola; PODESTA, 2006), a particular $GLD(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ can be rewrite as:

$$GLD(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = \lambda_1 + \frac{1}{\lambda_2} GLD(0, 1, \lambda_3, \lambda_4) \tag{4.1}$$

Based on the second term of Equation 4.1, our hypothesis is that we can group the uncertainty using clustering algorithms above λ_2, λ_3 and λ_4, looking to λ_2 first and then refine the groups with the values of λ_3 and λ_4. As λ_2 characterize the dispersion (variance or standard deviation) of the *GLD*, in the first step of the algorithm we group all the *GLDs* with similar dispersion. Dispersion alone don't tell us how the data is distributed, then to refine the clusters we look at λ_3 and λ_4, that are the parameters that define the shape of the distribution.

Intuitively, suppose we have two distributions, one normal and one exponential, and both distributions have the same standard deviation. The value of λ_2 of both distributions is similar. In the first step of the algorithm both distributions are grouped together. But, as the normal distribution is symmetric it has left and right tails, differently of the exponential that only have tail in one direction. The values of λ_3 and λ_4 of both distributions are dissimilar, then in the second step of the algorithm both distributions are separated in different clusters.

To test our hypothesis, we generate two synthetic datasets using 4 different probability density functions: Gaussian, Exponential, Uniform and Gamma. The structure of the datasets is represented in 4.2.

$$S(x_i, < v_j >) \quad i = 1.....n, j = 1.....m \tag{4.2}$$

where:

- n represent the number of objects of the dataset and,
- m represent the size of each object.

The datasets are described in details in Sections 4.3 and 4.4.

4.2.1 Fit the GLD to a dataset

When we generate a synthetic dataset, the next step is to find the *GLD* that best fit $< v_j >$ on each x_i. As the fitting process is computationally intensive we implement a parallel algorithm using **R**. The pseudo-code is shown in Algorithm 1.

Algorithm 1 Fitting the GLD to a synthetic dataset

1: **function** GLDFIT($S(x_i, < v_1, v_2, ..., v_n >)$)
2: $< \lambda_1, \lambda_2, \lambda_3, \lambda_4 > \leftarrow$ FIT.GLD.LM($< v_1, v_2, ..., v_n >$)
3: $isValid_{(x_i)} \leftarrow$ VALIDITYCHECK($< \lambda_3, \lambda_4 >$)
4: **if** $isValid_{(x_i)}$ **then**
5: $[pvalue, D]_{(x_i)} \leftarrow$ KS($< \lambda_1, \lambda_2, \lambda_3, \lambda_4 >_{(x_i)}$)
6: **if** $pvalue_{(x_i)} > 0.05$ **then**
7: STORELAMBDAS($< \lambda_1, \lambda_2, \lambda_3, \lambda_4 >, x_i$)

The algorithm receive a dataset represented by 4.2 and, for each position x_i, call a function ***fit.gld.lm*** from the **R** package **GLDEX** presented in section 3.8, line 2 of Algorithm 1. In line 3 we check the validity of the *GLD* returned by the function (remember from Chapter 3 that the *GLD* is not always valid). In line 5 a good-of-fit test is perform to be sure that each *GLD* is a good representation for the dataset in x_i. Finally all the *GLD* with $pvalue > 0.05$ are stored to be used in the next section.

The final result of this process is a new dataset with the form:

$$S(x_i, < \lambda_1, \lambda_2, \lambda_3, \lambda_4 >) \quad i = 1.....n \tag{4.3}$$

4.2.2 Clustering the GLD

The clustering algorithm follow the hypothesis mentioned above, see Algorithm 2. The dataset 4.3 is modified to remove λ_1 as we don't use it in the clustering process.

Algorithm 2 Clustering the GLD based on its $\lambda_{(2,3,4)}$ values.

1: **function** GLDCLUSTERING($S(x_i, < 0, \lambda_2, \lambda_3, \lambda_4 >)$)
2: $S(x_i, clusterID_I) \leftarrow$ FIRSTCLUSTERSTEP($S(x_i, \lambda_2)$)
3: **for each** $clId_I$ **do**
4: $S(x_i, clusterID_{II}) \leftarrow$ SECONDCLUSTERSTEP($S(x_i, < \lambda_3, \lambda_4 >), S(x_i, clId_I)$)

To test the influence of the three parameters in the clustering process, we first run the Algorithm 2 as it, over $< \lambda_2, \lambda_3, \lambda_4 >$. But in the second experiment we run it over $< \lambda_3, \lambda_4 >$. The results are discuses in sections 4.3 and 4.4.

4.3 Synthetic Data I

To generate the first synthetic data set we use 11 probability density functions, where 5 are Gaussian, 5 Exponential, and one Uniform, figures 8, 9 and 10. The standard deviation of the 5 Gaussian distributions is $0.05 * i$, with $i = 1, 2, 3, 4, 5$, and we generate 90 samples of each distribution. This is, the first 90 objects where generated from a Gaussian distribution with standard deviation 0.05, and so on. Similarly, the rate of the 5 Exponential distributions is i, with $i = 1, 2, 3, 4, 5$, and again we generate 90 samples of each one. Finally, 100 samples of a Uniform distribution between $[0, 1]$ were generated. In resume, we have 1000 objects, where the first 450 were sampled from a Gaussian distributions, the next 450 from an Exponential and the last 100 from a Uniform distribution. As we generate a synthetic dataset in this way, we have the ground truth of the clustering in the dataset. This ground truth is used to evaluate the clustering quality of our algorithms.

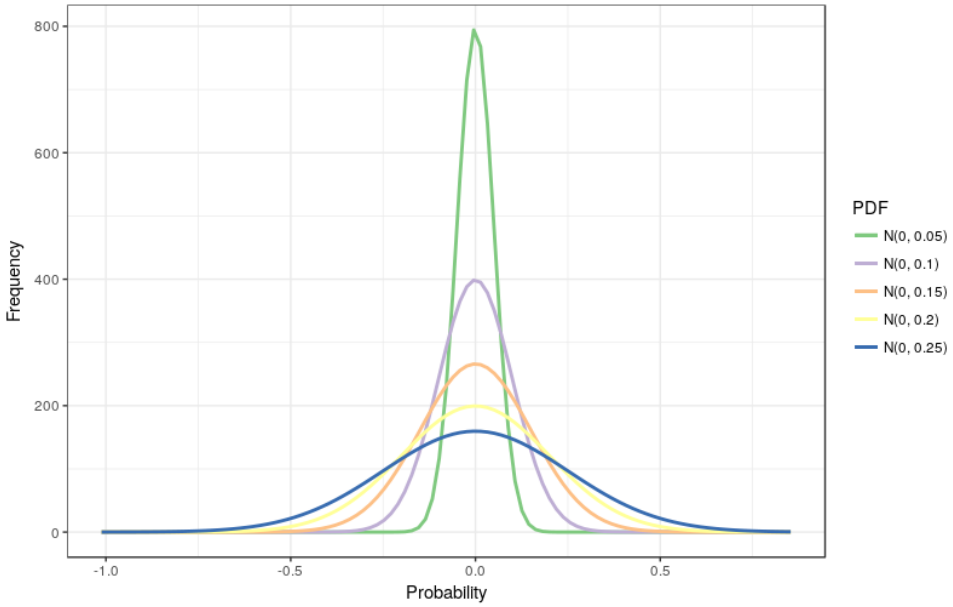

Figure 8 – Gaussian (Normal) distributions used to generate the synthetic dataset.

This dataset could be represented as a multidimensional array where for each position x_i, we have 1000 values v_j, equation 4.4. In this case i and j vary from 1 to 1000 casually.

$$S(x_i, < v_j >) \quad i, j = 1, 2.....1000 \tag{4.4}$$

The fitting algorithm proposed in subsection 4.2.1 is applied over 4.4. The good-of-fit test return that all the *GLDs* are good fit for its corresponding distribution. As a result the dataset 4.5 is genarated.

$$S(x_i, GLD(\lambda_1, \lambda_2, \lambda_3, \lambda_4)) \quad i = 1.....1000 \tag{4.5}$$

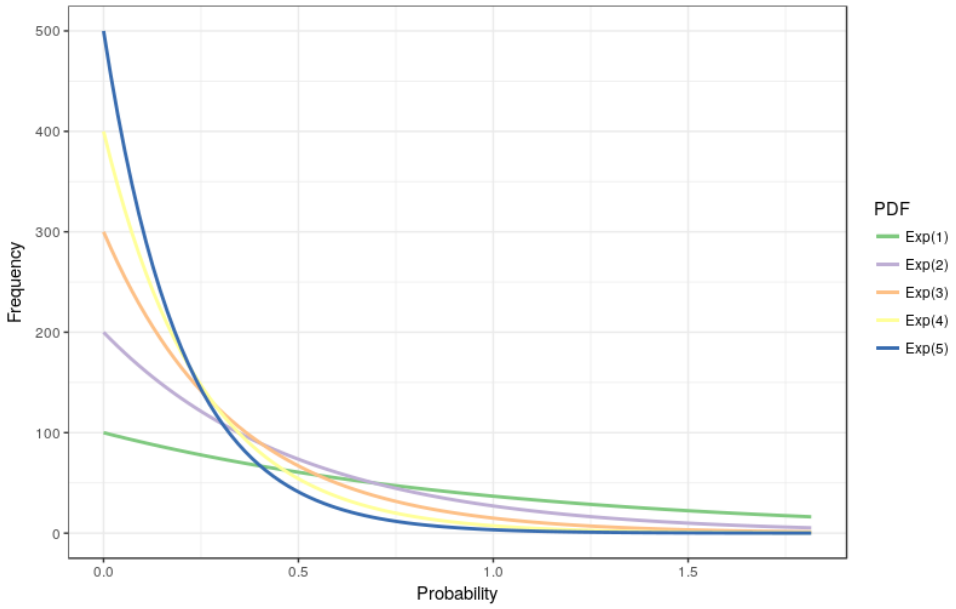

Figure 9 – Exponential distributions used to generate the synthetic dataset.

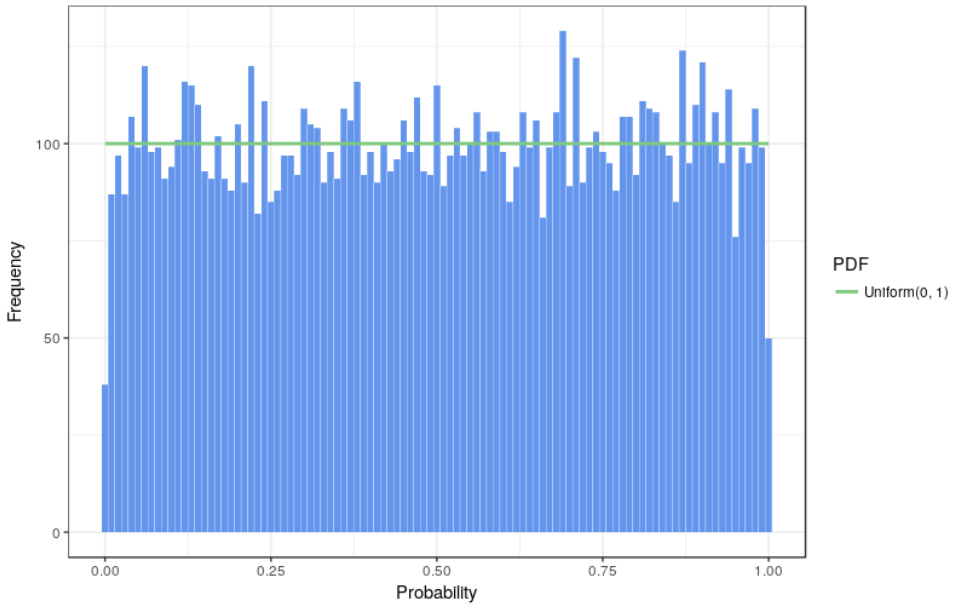

Figure 10 – Uniform distribution used to generate the synthetic dataset.

4.3.1 Clustering using λ_2, λ_3 and λ_4

As we mention above, our idea is to test what happen if we use a simple proposed algorithm over the λ_2, λ_3 and λ_4 values of the *GLDs*. Similar to the paper (JIANG et al., 2011), as we use 11 *PDFs* to generate the synthetic dataset I, we expect that the algorithm will return 11 clusters as well (one for each distribution).

In Figure 11 and table 5 the distribution of the clusters returned by the algorithm is shown.

Table 5 – Distribution of the clusters using k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*.

Cluster	Type of Distribution	No. of Elements
1	Exponential	81
2	Normal	93
3	Normal	96
4	Normal	104
5	Exponential	96
6	Exponential	83
7	Exponential	87
8	Uniform	91
9	Normal	90
10	Exponential	83
11	Normal	90

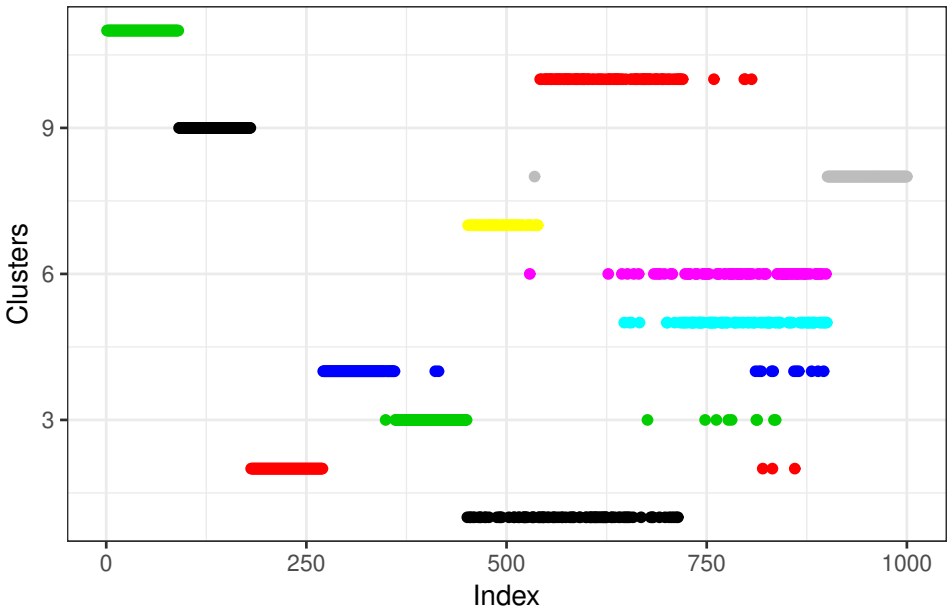

Figure 11 – Distribution of the clusters using k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*.

Remembered, the first 450 elements are normal distributions, the next 450 are exponential and the last 100 are uniform. Looking to those regions in general, the first observation is that we have 19 false positives, three in cluster II, 7 in cluster III, 7 in cluster IV and one in cluster VIII. The second observation is that, the normal distributions were grouped in 5 clusters (II, III, IV, IX and XI), clusters II, III and IV group perfectly it 90 elements with 17 false positives, while clusters IX and XI group exactly its 90 elements each.

The Uniform distribution was grouped totally in cluster VIII, with one false positives as was mention above. The last observation is that the algorithm can't separate

the 5 Exponential distributions, but this is not a bad result as we will show soon.

In Figure 12 we show the *PDFs* of all the distributions that belongs to the same cluster. If we take a look at clusters I, V, VI, VII and X we see that the exponential distribution was well grouped. Really the problem is that the rate value of $0.05 * i$ used to generate the exponential distribution does not have such a big difference between one and another.

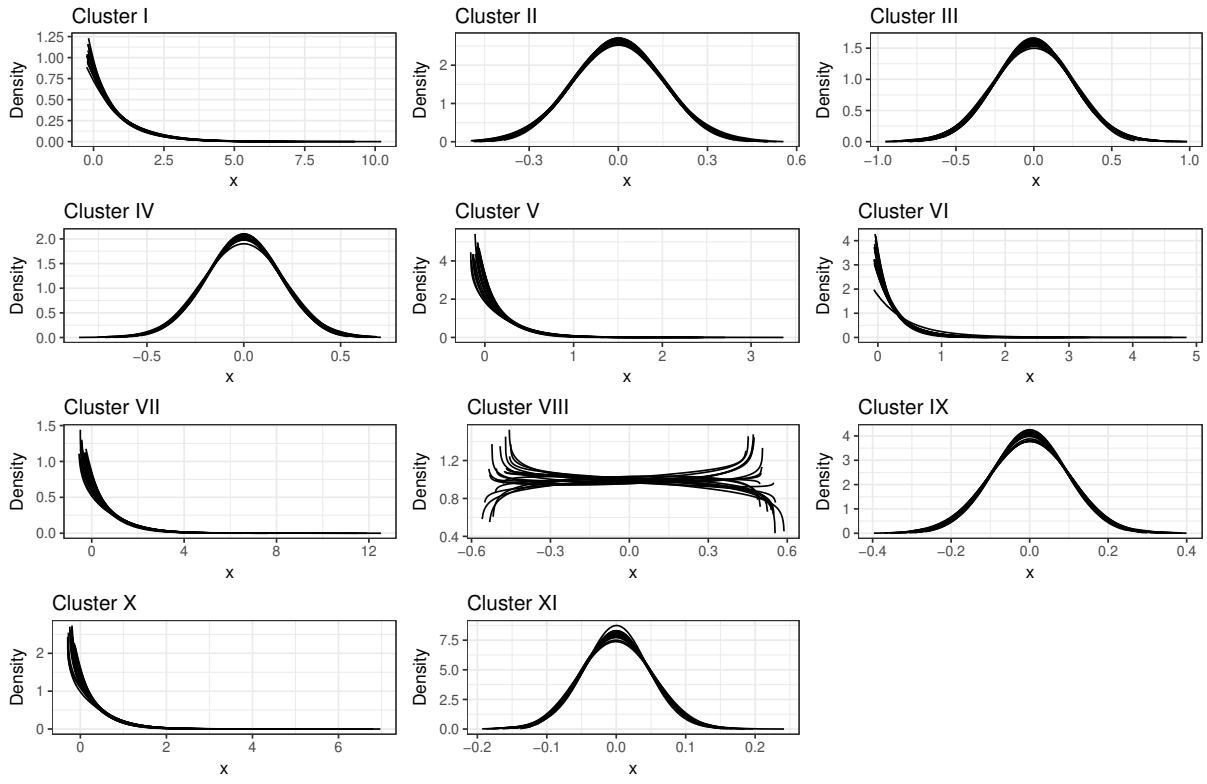

Figure 12 – *PDFs* of 60 members of the 11 clusters obtained using the clustering proposed algorithm over $(\lambda_2, \lambda_3, \lambda_4)$ values.

Another interesting result is show in Figure 13. As we can see, clusters II, III, IV, IX and XI that represent the Normal distribution are all at the same region over the λ_3 and λ_4 space, near $\lambda_3 = 0$ and $\lambda_4 \in [0, 0.3]$. Similarly cluster VIII, that represent the Uniform distribution is on the top left of the λ_3 and λ_4 space, $\lambda_3 \in [0, 0.3]$ and $\lambda_4 \in [0.7, 1.5]$. And finally the rest of the clusters that represent the Exponential distribution are distributed in the bottom of the λ_3 and λ_4 space, $\lambda_3 \in [0.2, 7]$ and $\lambda_4 \in [-0.1, 0.1]$. As we see in the rest of the thesis, this result is repeated in all the use cases.

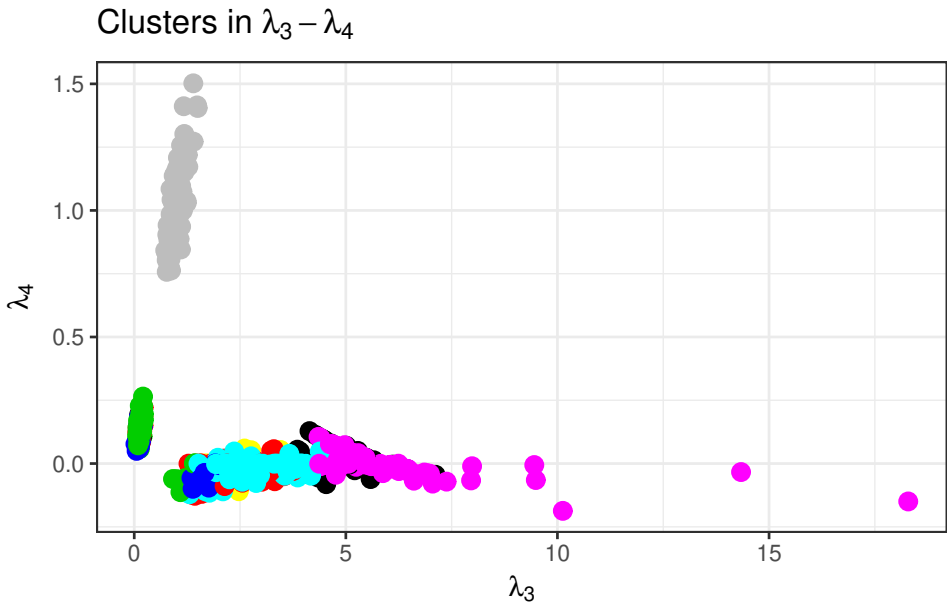

Figure 13 – Distribution of the clusters over the λ_3 and λ_4 space.

The responsible of separate the overlapping distributions in different clusters in this case is λ_2, as we can see in Figure 14.

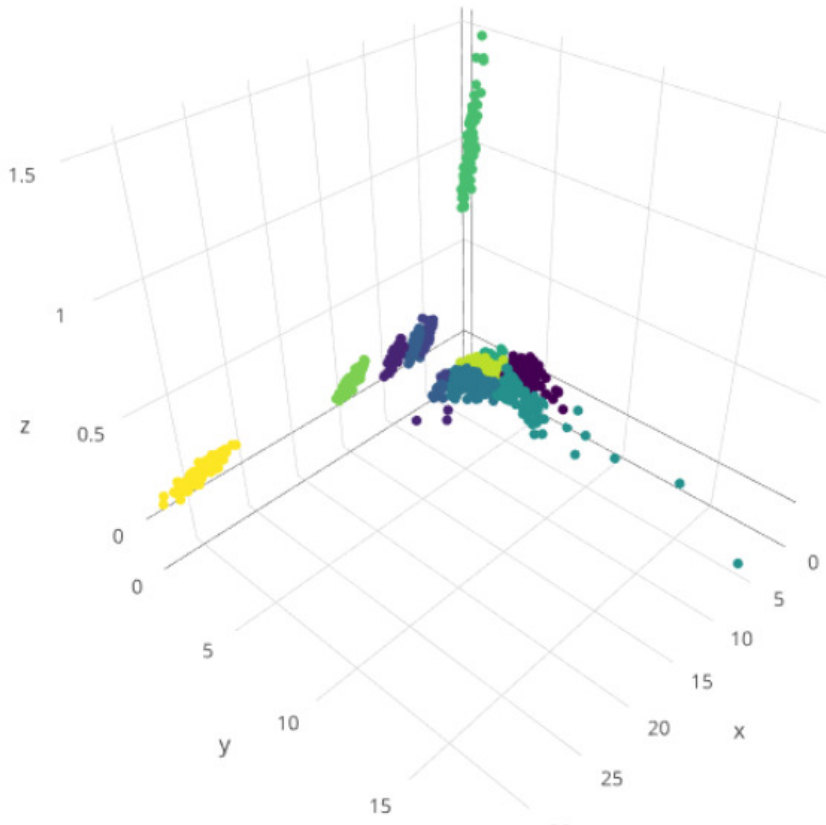

Figure 14 – Distribution of the clusters over the λ_2, λ_3 and λ_4 space.

Table 6 – Distribution of the clusters using k-means over the λ_3 and λ_4 values of the *GLDs*.

Cluster	Type of Distribution	No. of Elements
1	Uniform	100
2	Exponential	118
3	Exponential	110
4	Exponential	35
5	Exponential	74
6	Exponential	2
7	Normal	197
8	Exponential	41
9	Exponential	105
10	Normal	131
11	Normal	122

4.3.2 Clustering using λ_3 and λ_4

In this section we proceed similar to section 4.3.1, but the algorithm run over λ_3 and λ_4. The distribution of the clusters is shown in figure 15 and table 6.

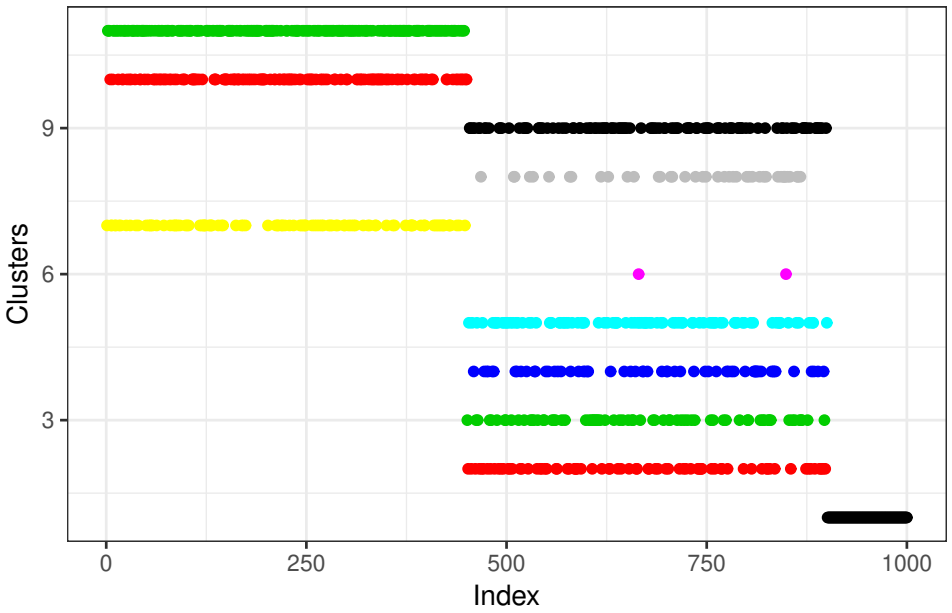

Figure 15 – Distribution of the clusters using k-means over the λ_3 and λ_4 values of the *GLDs*.

As we don't use λ_2 here, is clear that the algorithm can't distinguish the distributions by its standard deviation. But, as the shape of the *GLD* is defined by λ_3 and λ_4, what we expect is that the algorithm can separate the objects by distribution types. As we see in figure 15 this is exactly what we get, there is no any false positive in this case, the three regions (Normal, Exponential and Uniform) are identified by the algorithm.

Clusters VII, X and XI group all the Normal distributions, cluster I group the Uniform and the rest group the Exponential.

In the λ_3 and λ_4 space the behavior is very similar at the one we get in subsection 4.3.1, Figure 16. Again the distributions are concentrated near the same (λ_3, λ_4) values.

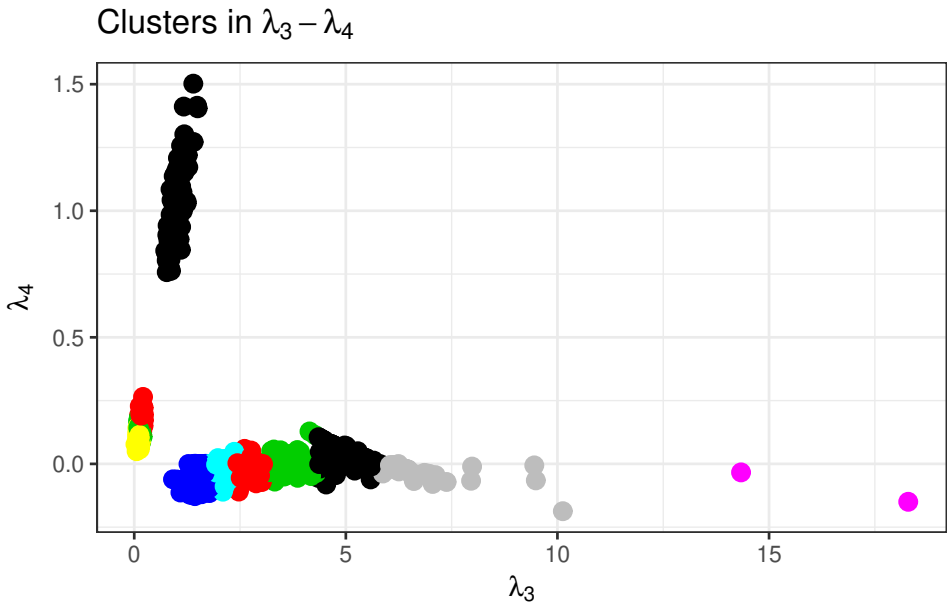

Figure 16 – Distribution of the clusters over the λ_3 and λ_4 space.

In Figure 17 we show the PDFs of all the distributions that belongs to the same cluster.

4.4 Synthetic Data II

The second synthetic dataset is similar to the first one, here we include 5 Gamma distributions, between the Exponential and the Uniform, figure 18. The shape of the Gamma distribution is i, with $i = 1, 2, 3, 4, 5$. This dataset have 1450 objects, where the first 450 were sampled from a Gaussian distributions, the next 450 from an Exponential, the next 450 are Gamma, and the last 100 from a Uniform distribution. As we use 16 different distributions, this is the number of clusters to be used with the k-means algorithm.

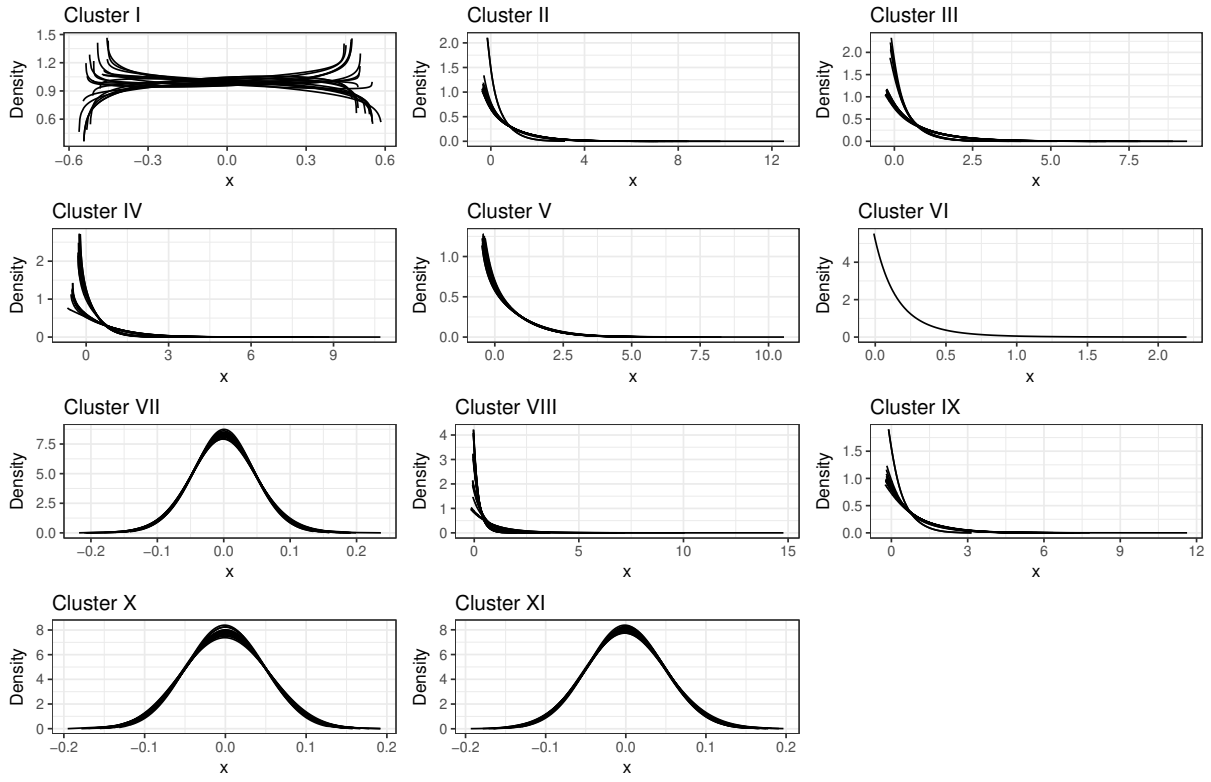

Figure 17 – *PDFs* of 60 members of the 11 clusters obtained using the clustering proposed algorithm over $(\lambda_2, \lambda_3, \lambda_4)$ values.

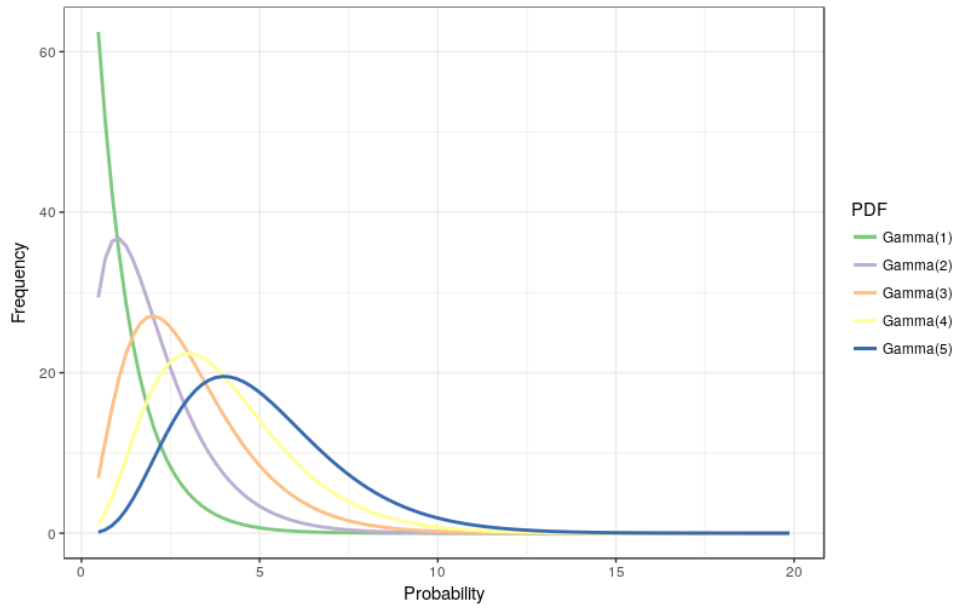

Figure 18 – Gamma distributions used to generate the synthetic dataset.

Similar to the dataset I, the fitting algorithm proposed in subsection 4.2.1 is applied over dataset II. The good-of-fit test return that all the *GLDs* are good fit for its

corresponding distribution.

4.4.1 Clustering using λ_2, λ_3 and λ_4

The distribution of the clusters returned by the k-means algorithm is shown in figure 19 and table 7.

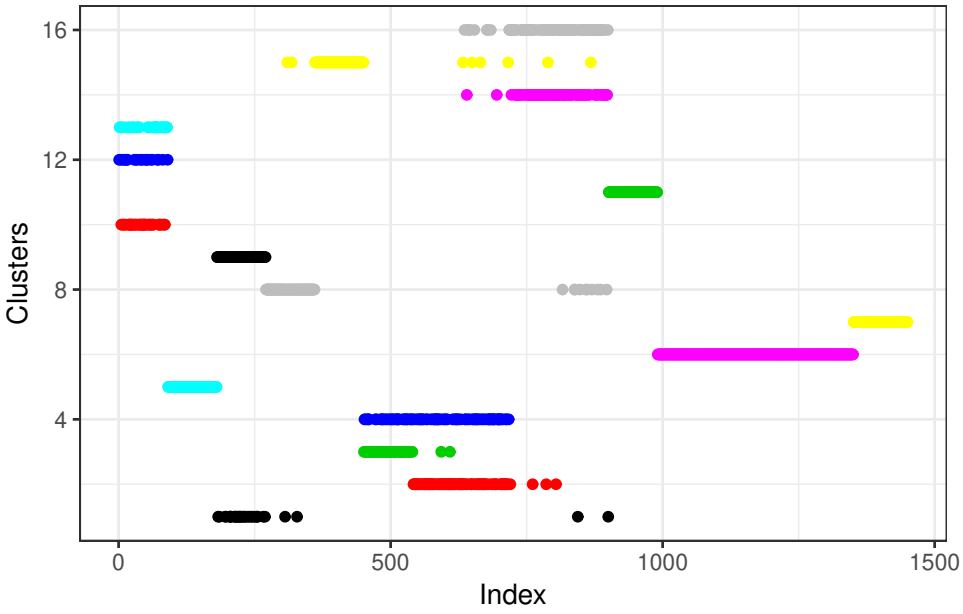

Figure 19 – Distribution of the clusters using k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*.

In general the results are very similar to the results of the section 4.3, but we get less false positives, 9 in total. Two false positives in cluster I and 7 false positives in cluster VIII. The normal distribution was groping in 7 clusters: I, V, VIII, IX, X, XII and XIII. The uniform distribution was grouping in cluster VII without false positives. The gamma distribution introduced here was grouped in clusters VI and XI, without false positives. And finally the rest of the clusters are for the exponential distribution.

The projection of the clusters over the λ_3 and λ_4 space is show in figure 20. The cluster of the uniform distribution is located again in the top-left region of the figure. The normal distribution is located in the same place, near $\lambda_3 = 0$ and $\lambda_4 \in [0, 0.3]$. The exponential distribution is distributed in the bottom of the λ_3 and λ_4 space. The gamma distribution is overlapped together with the normal distribution.

Table 7 – Distribution of the clusters using k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*.

Cluster	Type of Distribution	No. of Elements
1	Normal	51
2	Exponential	44
3	Exponential	45
4	Exponential	179
5	Normal	90
6	Gamma	360
7	Uniform	100
8	Normal	102
9	Normal	42
10	Normal	31
11	Gamma	90
12	Normal	34
13	Normal	25
14	Exponential	90
15	Exponential	30
16	Exponential	360

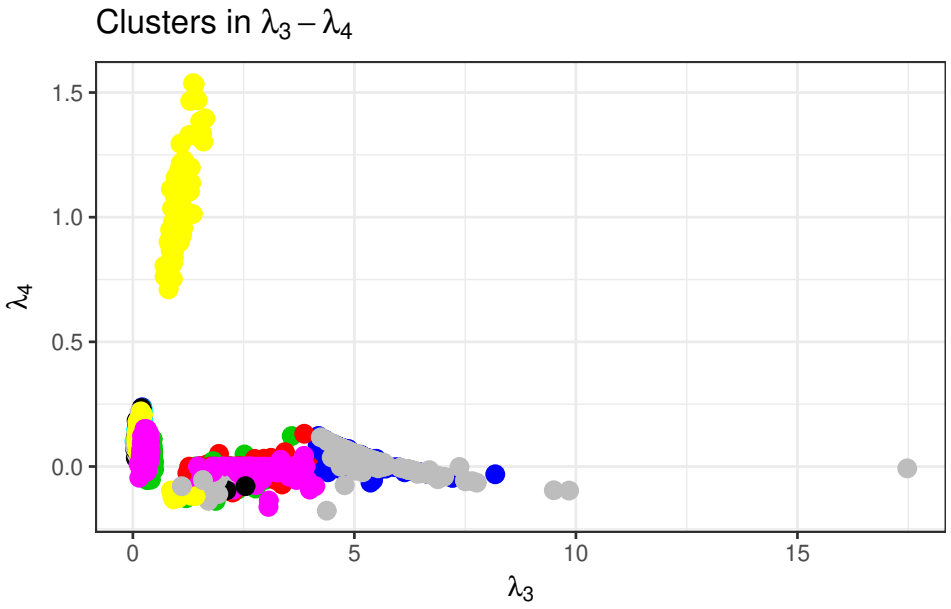

Figure 20 – Distribution of the clusters over the λ_3 and λ_4 space.

In Figures 21 and 22 we show the PDFs of all the distributions that belongs to the same cluster.

The responsible of separate the overlapping distributions in different clusters is λ_2, as we can see in Figure 23.

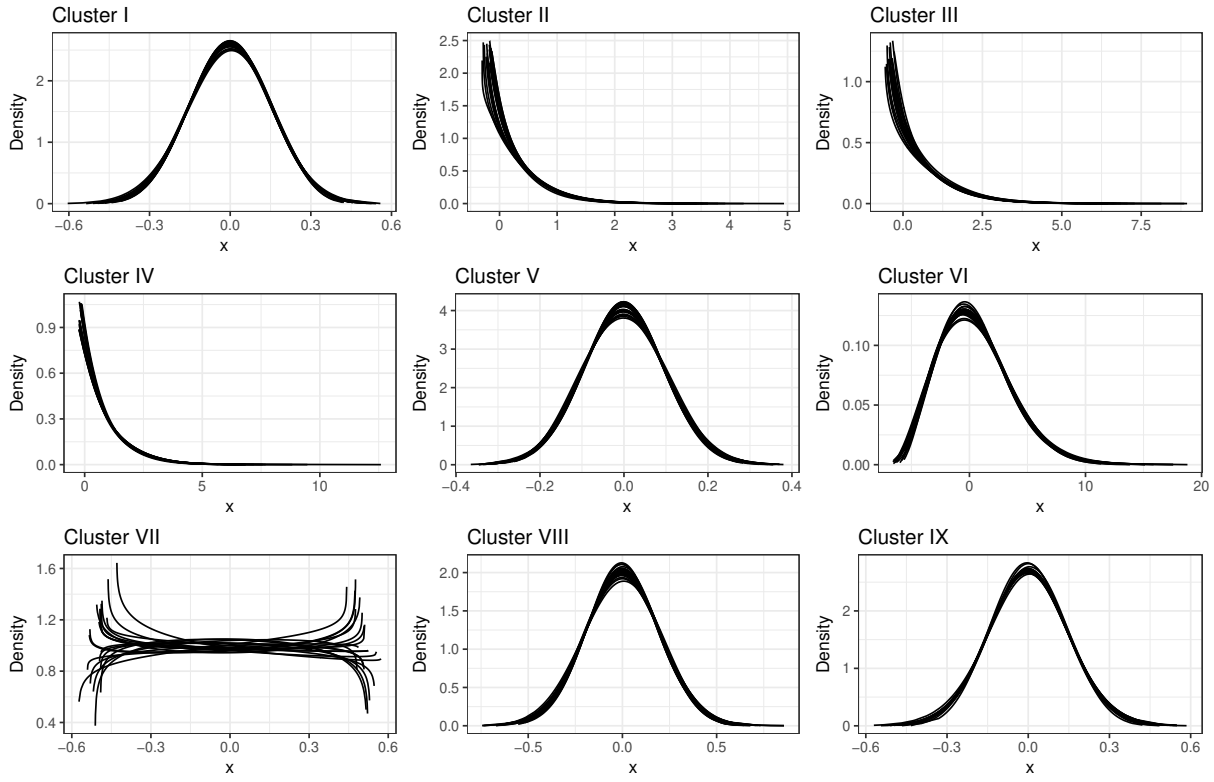

Figure 21 – *PDFs* of 60 members of the first 9 clusters obtained using the clustering proposed algorithm over $(\lambda_2, \lambda_3, \lambda_4)$ values.

4.4.2 Clustering using λ_3 and λ_4

The distribution of the clusters returned by the k-means when using the values of λ_3 and λ_4 to group the second synthetic dataset are shown in figure 24 and table 8.

Two false positives are observed in clusters I and XIII, but nothing to worry about. Again the regions of the four distribution families are perfectly separated by the algorithm.

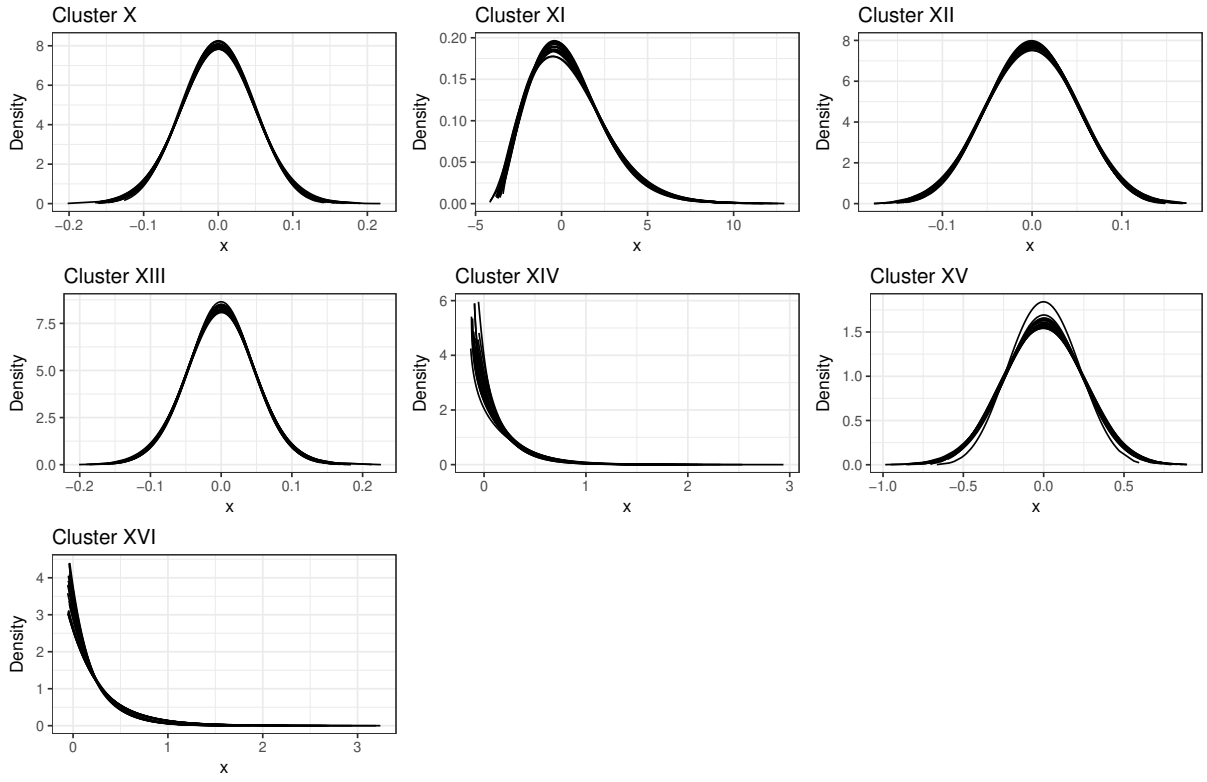

Figure 22 – *PDFs* of 60 members of the last 7 clusters obtained using the clustering proposed algorithm over $(\lambda_2, \lambda_3, \lambda_4)$ values.

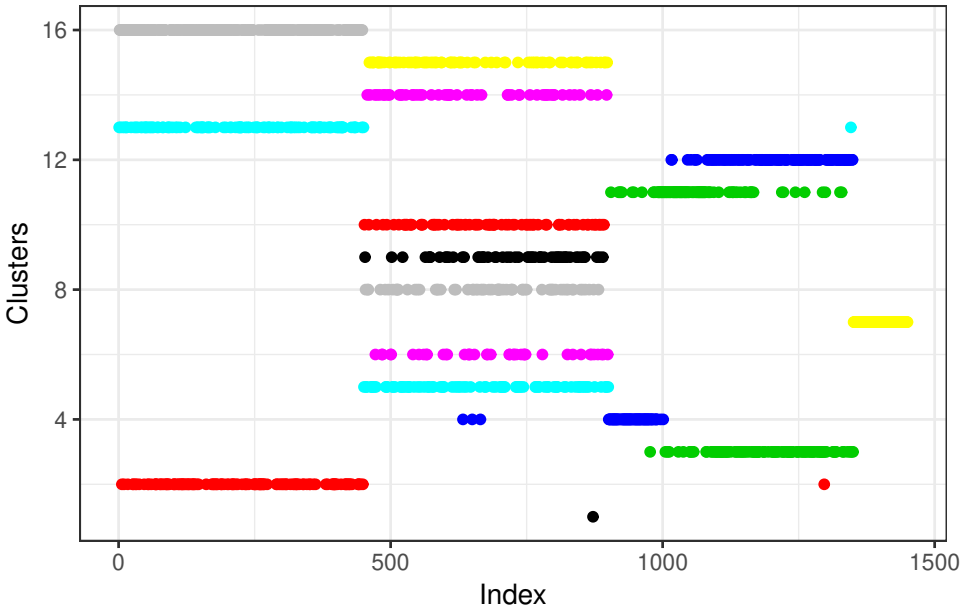

Figure 24 – Distribution of the clusters using k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*.

The projection of the clusters over the λ_3 and λ_4 space is show in figure 25.

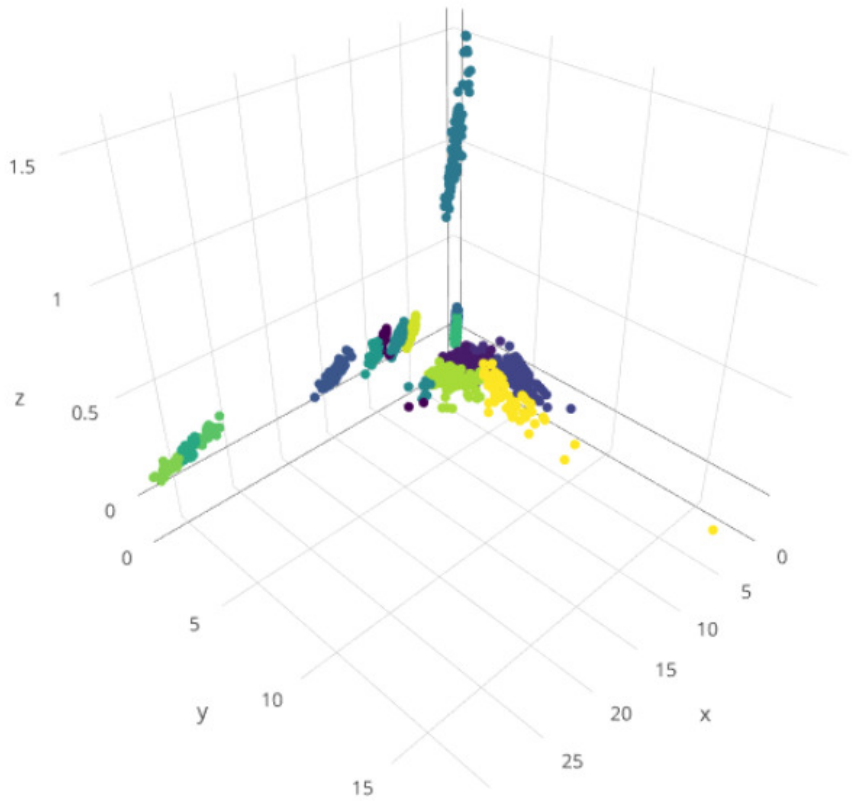

Figure 23 – Distribution of the clusters over the λ_2, λ_3 and λ_4 space.

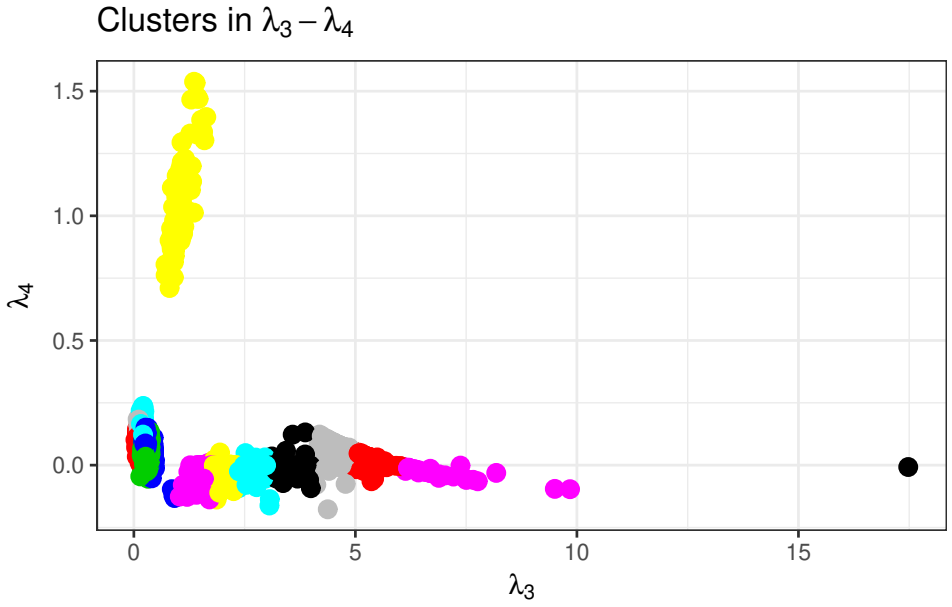

Figure 25 – Distribution of the clusters over the λ_3 and λ_4 space.

In Figures 26 and 27 we show the PDFs of all the distributions that belongs to the same cluster.

Table 8 – Distribution of the clusters using k-means over the λ_3 and λ_4 values of the *GLDs*.

Cluster	Type of Distribution	No. of Elements
1	Exponential	1
2	Normal	139
3	Gamma	112
4	Gamma	63
5	Exponential	66
6	Exponential	60
7	Uniform	100
8	Exponential	64
9	Exponential	57
10	Exponential	75
11	Gamma	128
12	Gamma	148
13	Normal	112
14	Exponential	80
15	Exponential	44
16	Normal	201

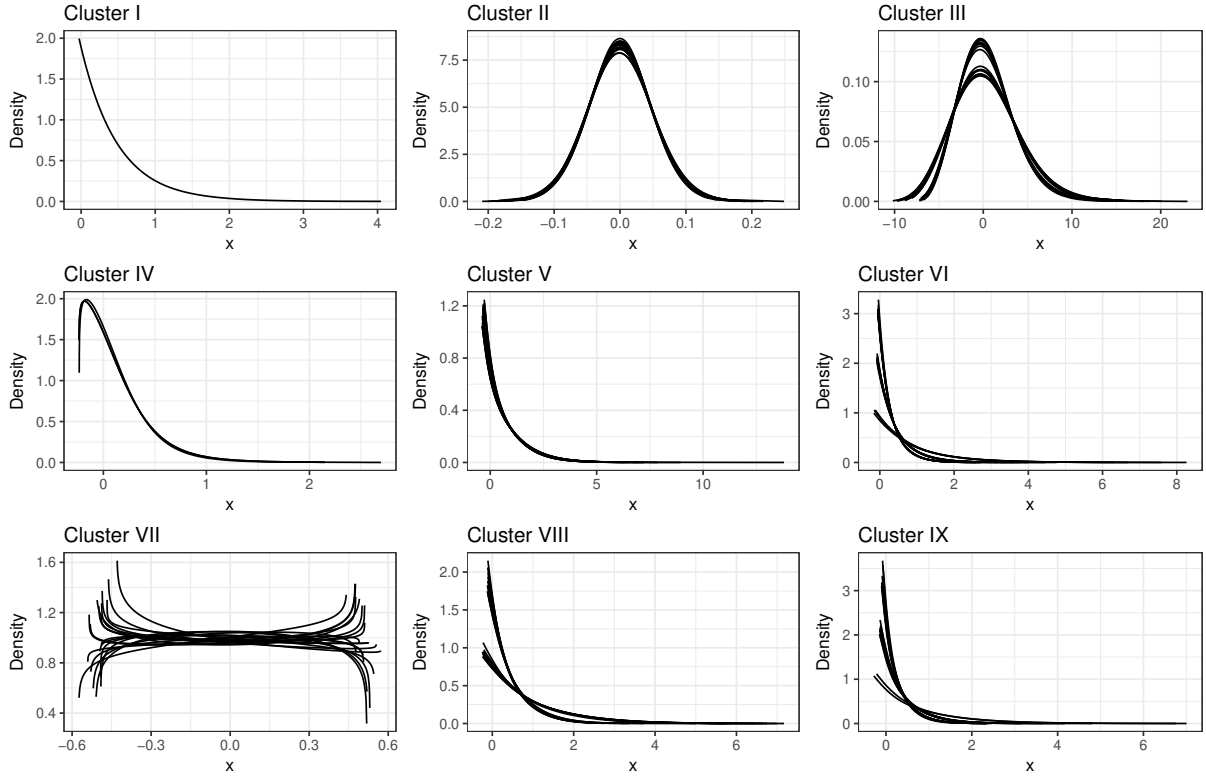

Figure 26 – *PDFs* of 60 members of the first 9 clusters obtained using the clustering proposed algorithm over $(\lambda_2, \lambda_3, \lambda_4)$ values.

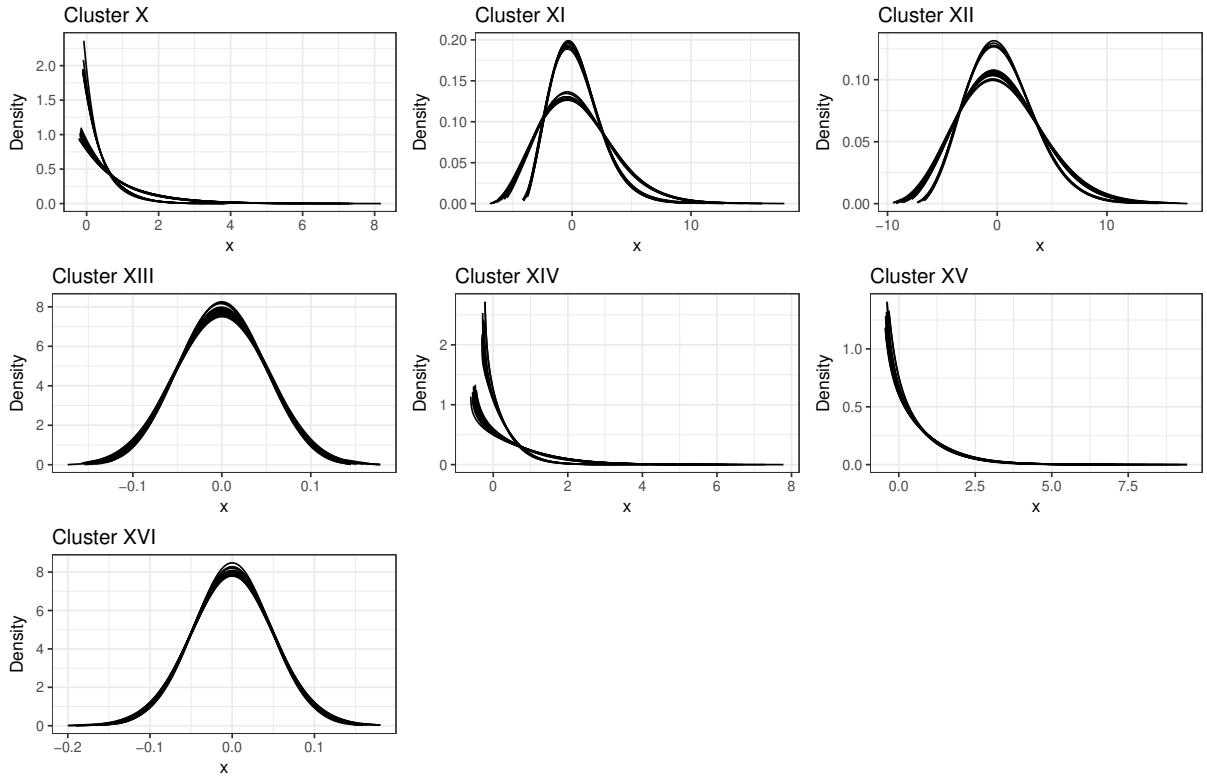

Figure 27 – *PDFs* of 60 members of the last 7 clusters obtained using the clustering proposed algorithm over $(\lambda_2, \lambda_3, \lambda_4)$ values.

4.5 Summary

In this Chapter, we explore clustering uncertain data based on the similarity between their distributions. The idea is to answer the **RQ.1** *"how to group the output of the UQ process based on the similarity of the uncertainty?"*. The hypothesis enunciated at the beginning of the Chapter was tested against two synthetic datasets, and the results corroborate that we can group uncertain data based in the λ_i values of the *GLDs* that describe the data.

As a second result we test that, when using λ_2, λ_3 and λ_4 the separations of the different distributions of the dataset was almost perfect with a few false positives; and when using λ_3 and λ_4 all the elements of the same family are grouping together without consider the difference in the standard deviation. Both results are exactly what we expect.

Another important result of this Chapter, is that if we look at the clustering technique proposed here and compare it with the state-of-the-art, our approach is a competitive one. For example, in the approach proposed by (JIANG et al.,) the computational cost of its algorithm depend of two factors, the fit of the distributions using Kernel Density Estimation (KDE) and the computation of the KL-divergence (the distance measure used

in its approach). Both factors are computationally intensive. In our approach we substitute KDE by *GLD* fit, that is most costly form the computational point of view; but at the same time we substitute the KL-divergence by simple distance comparisons in $\mathbf{R}$ and $\mathbf{R^2}$. On the other hand, some limitations of the KL-divergence approach as: (i) the *PDF* of every uncertain object to be clustered need to be defined in the same domain and (ii) the needs to select an appropriate kernel to fit the data using KDE, are solved with the use of the *GLD*.

All of this observation join with the rest of the advantage of the use of the *GLD* to quantify the uncertainty, make our approach far superior to those reported in the literature.

5 Our Approach

In Chapter 3 we present the *GLD* and argue why this distribution family is suitable to be used in UQ, while in Chapter 4 we explore the possibility to use the λ values of the *GLD* to clustering uncertain data. Now in this Chapter we present a workflow to quantify the uncertainty in large-scale spatio-temporal models using the *GLD*. In Chapter 4 we present a solution to the **RQ.1**, in this Chapter we present a solution to the remaining four research questions.

The rest of the Chapter is organized as follow: Section 5.1 describes the proposed workflow and comment briefly the motivations to propose it and its steps. Section 5.2 presents the fitting step, that is divided into three sub-steps: the fitting process, *GLD* validity check and the fitting process evaluation quality. Section 5.3 presents how to use spatio-temporal interpolation over the λ values of the *GLD* to estimate the uncertainty in spatio-temporal locations not previously analyzed (**RQ.2**). Section 5.4 discusses the integration of the clustering algorithm proposed in Chapter 4 into the workflow. Section 5.5 presents how to use the previous results to answer queries that arise in the UQ context, such us those queries we formulate in Chapter 1 (**RQ.3, RQ.4 and RQ.3**). Section 5.6 presents an implementation of the proposed workflow in an R package named Simulation Uncertainty Quantification Querying (SUQ^2). Finally Section 5.7 summarize the Chapter.

5.1 UQ Proposed Dataflow

The proposed workflow to uncertainty quantification in **LSSTM** based on the *GLD*, is divided into four steps, Figure 28. The first step is the **fitting process**, where we implement algorithms to estimate the parameters of the *GLD* that best fit the dataset at each spatio-temporal location. The second step is the **spatio-temporal interpolation (kriging)**. This step is included because in UQ is common to estimate the uncertainty (e.g. low order statistical moments, such as the standard deviation) in some points and then interpolate the uncertainty to other points. Analogously, we propose to do the same but using the λ values of the *GLD*. The third step is c**lustering uncertain data**, using the algorithm proposed and tested in Chapter 4. And finally the fourth is the **queries** step, where we expose how the results produced in the previous steps help us to answer queries that arise in the UQ context.

In the next sections we detail each of the UQ processing steps.

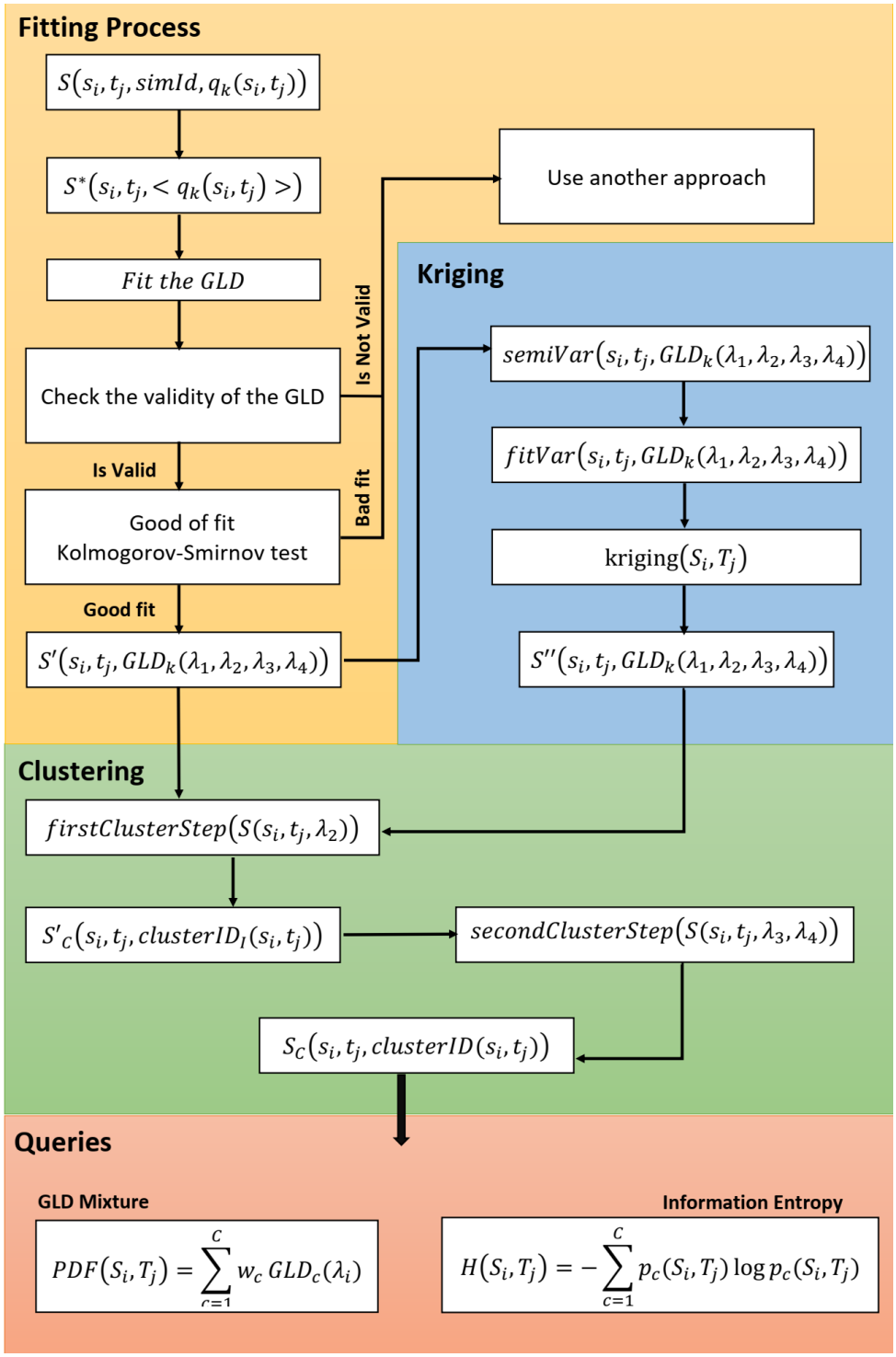

Figure 28 – Proposed workflow. The workflow was divided in four steps, (i) the fitting process, (ii) the spatio-temporal interpolation (kriging), (iii) the clustering of the GLDs and, (iv) the queries over the results of the clustering process.

5.2 Fitting a GLD to a spatio-temporal dataset

In the more general case, the computational model $\boldsymbol{q} = \mathcal{M}(\boldsymbol{\theta})$ represents the spatio-temporal evolution of a complex systems, and the *QoI* $\boldsymbol{q}$ could be represented as:

$$\mathbf{Q} = (\mathbf{q}(s_1, t_1), \mathbf{q}(s_2, t_2), ..., \mathbf{q}(s_n, t_n)) \tag{5.1}$$

where:

- $(s_1, t_1), (s_2, t_2), ..., (s_n, t_n) \in \mathcal{S} \times \mathcal{T} \subseteq \mathbb{R}^3 \times \mathbb{R}$ represents a set of distinct spatio-temporal locations, and
- $\mathbf{q}(s_i, t_j)$ represents a value of the *QoI* at the spatio-temporal location (s_i, t_j)

In the presence of a stochastic problem, at each spatio-temporal location (s_i, t_j), we have many realizations of $q(s_i, t_j)$. A data set schema to represent this information can be modeled as (see also 2.8):

$$S(s_i, t_j, simId, q(s_i, t_j)) \tag{5.2}$$

where $simId$ represents the *id* of one simulation (realization).

The first step of our approach consists in find the *GLD* that best fits our simulations on each spatio-temporal location. This step is divided in three minor tasks:

- Fit the *GLD* to the data.
- Evaluate the validity of the resulting *GLD* on each spatio-temporal location.
- Perform a ks-test to evaluate the quality of the fit on each spatio-temporal location.

The fitting process has been implemented following Algorithm 3. Before starting the fitting process, we group all the simulations that correspond to the same spatio-temporal location (s_i, t_j). As a result, a new dataset with the following data schema is created: $S^*(s_i, t_j, < q_1, q_2, .., q_n >)$, where $q_i, 1 \leq i \leq n$, represents a vector holding all the values of q at point (s_i, t_j).

5.2.1 The Fitting process

Having prepared the dataset with the distribution at each spatio-temporal location, we are in conditions to process the GLD fitting step. Thus,for each spatio-temporal location $(s_i, t_j) \in \mathcal{S} \times \mathcal{T}$ we apply a fitting function provided by the GLDEX R package described in section 3.8. The lattter fits the *GLD* to a vector $< q_1, q_2, .., q_n >$, line 2 of Algorithm 3.

As a result of this task, we get the λ values of the *GLD* that best fit the dataset at each spatio-temporal location, Equation 5.3 (see also 4.5).

$$S'(s_i, t_j, GLD(\lambda_1, \lambda_2, \lambda_3, \lambda_4)) \tag{5.3}$$

5.2.2 GLD validity check

As we mention in section 3.1 the *GLD* is not always valid, it depends on the λ_3 and λ_4 values. The evaluation of the validity of the *GLD* is straightforward, if λ_3 and λ_4 are in the gray regions of Figure 2 the *GLD* is not valid, on the other case is valid.

The validity check is performed in line 3 of Algorithm 3, and as a result we get:

$$S_{validity}(s_i, t_j, valid(s_i, t_j)), \tag{5.4}$$

where:

$$valid(s_i, t_j) = \begin{cases} 1 & \text{if GLD is valid in } (s_i, t_j) \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

5.2.3 Quality of the fit

Now at the remaining points, where the *GLD* is valid, we need to evaluate how good is the fit. That is, we evaluate whether the *GLD* (PDF) correctly describes the dataset. We use here the Kolmogorov-Smirnov test (KS-test). The KS-test determines if two datasets differ significantly. In this case, these datasets are: the original dataset and a second one generated using the fitted *GLD*. As a result, this test returns two values: a Kolmogorov-Smirnoff Distance (D); and a p-value, line 5 of Algorithm 3. The distance D is the maximum distance between both cumulative density functions (CDF), as shown in Figure 29. A small distance means that both, the dataset and the fitted PDF, are similar.

The second value, the p-value, is a more robust test, as it helps us to determine the significance of our results. Suppose we have two hypotheses, the null hypothesis is that our PDF is a good fit to our dataset, and the alternative hypothesis is that it is not. Then, a small p-value (typically ≤ 0.05) indicates strong evidence against the null hypothesis, so you reject the null hypothesis. A large p-value (> 0.05) indicates weak evidence against the null hypothesis, so you fail to reject the null hypothesis. p-values very close to the cutoff (0.05) are considered to be marginal (could go either way).

At the end of this task we have two new multidimensional arrays with the values of $\mathcal{D}$ and p-value at each spatio-temporal locations.

$$S_{\mathcal{D}}(s_i, t_j, \mathcal{D}(s_i, t_j)) \tag{5.6}$$

$$S_{p_{value}}(s_i, t_j, p_{value}(s_i, t_j)) \tag{5.7}$$

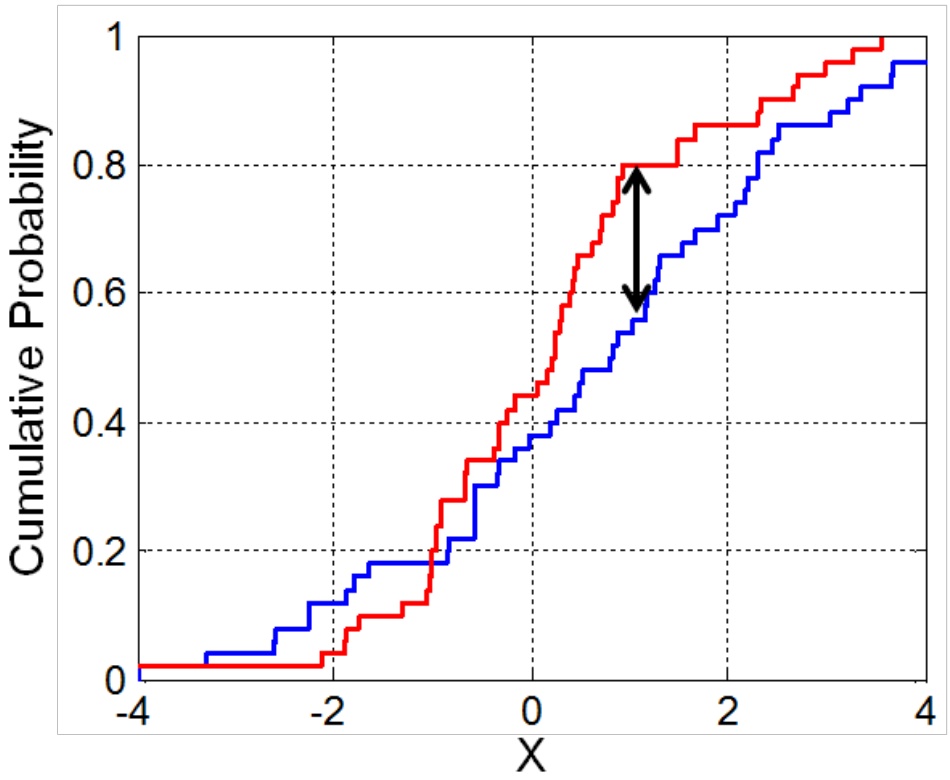

Figure 29 – Illustration of the two-sample Kolmogorov–Smirnov statistic. Red and blue lines each correspond to an empirical distribution function, and the black arrow is the two-sample KS statistic.

Finally, in line 7 of Algorithm 3, we store the λ values of those *GLDs* that are valid and return p-values greater than 0.05.

Algorithm 3 Fitting the GLD to a spatio-temporal dataset

1: **function** GLDFIT($S(s_i, t_j, < q_1, q_2, ..., q_n >)$)
2: $< \lambda_1, \lambda_2, \lambda_3, \lambda_4 > \leftarrow$ FIT.GLD.LM($< q_1, q_2, ..., q_n >$)
3: $isValid_{(s_i,t_j)} \leftarrow$ VALIDITYCHECK($< \lambda_3, \lambda_4 >$)
4: **if** $isValid_{(s_i,t_j)}$ **then**
5: $[pvalue, D]_{(s_i,t_j)} \leftarrow$ KS($< \lambda_1, \lambda_2, \lambda_3, \lambda_4 >_{(s_i,t_j)}$)
6: **if** $pvalue_{(s_i,t_j)} > 0.05$ **then**
7: STORELAMBDAS($< \lambda_1, \lambda_2, \lambda_3, \lambda_4 >, s_i, t_j$)

5.3 Spatio-Temporal Interpolation

Although our interest is to quantify the uncertainty at each spatio-temporal locations, this is a timely and computationally prohibitive task. Neither the **GLD** or simpler approaches, as the evaluation of low order statistical moments, can be computed over the whole output space. Usually researchers place control points at particular points of interest, and then interpolate the uncertainty to other points as needed.

Spatial or temporal interpolation independently, are very well studies, and dozens of algorithms exist to compute new values by means of those methods. However, working with spatio-temporal domain implies that variability in space and time must be modelled, which is more complicated than modelling purely spatial or purely temporal variability (GRALER; PEBESMA; HEUVELINK, 2016).

Our purpose here is not to provide a new interpolation method over the **GLDs** λ values, but show how spatio-temporal interpolation can be used to answer the **RQ.2**:

> **RQ2.** what is the uncertainty in some spatio-temporal locations not previously analyzed?

For this reason we select the state-of-the-art spatio-temporal interpolation method, from the best of our knowledge, proposed by Graler et al. in (GRALER; PEBESMA; HEUVELINK, 2016), and implemented it in the R package **gstat**. The selection obeys the fact that this implementation includes time as an extra dimension, and allows us to interpolate in space and time at the same time.

5.3.1 Kriging over *GLD*

First to all, we define some concepts that are important to understand our proposal. The first concept is **what is Kriging?**.

Definition 5.1. Optimal interpolation based on regression against observed values of surrounding data points, weighted according to spatial covariance values.

An the definition of **interpolation** is:

Definition 5.2. Estimation of a variable at an unmeasured location from observed values at surrounding locations.

For example, suppose we have a measure of the porosity at an spacial region, Figure 30, and we want to estimate the porosity value in an unmeasured point marker with + in the figure, based on porosity values at nearest six data points, Figure 31.

All interpolation algorithms (inverse distance squared, splines, radial basis functions, triangulation, etc.) estimate the value at a given location as a weighted sum of data values at surrounding locations. Almost all assign weights according to functions that give a decreasing weight with increasing separation distance. Kriging assigns weights according to a (moderately) **data-driven weighting function**, rather than an arbitrary function, but it is still just an interpolation algorithm.

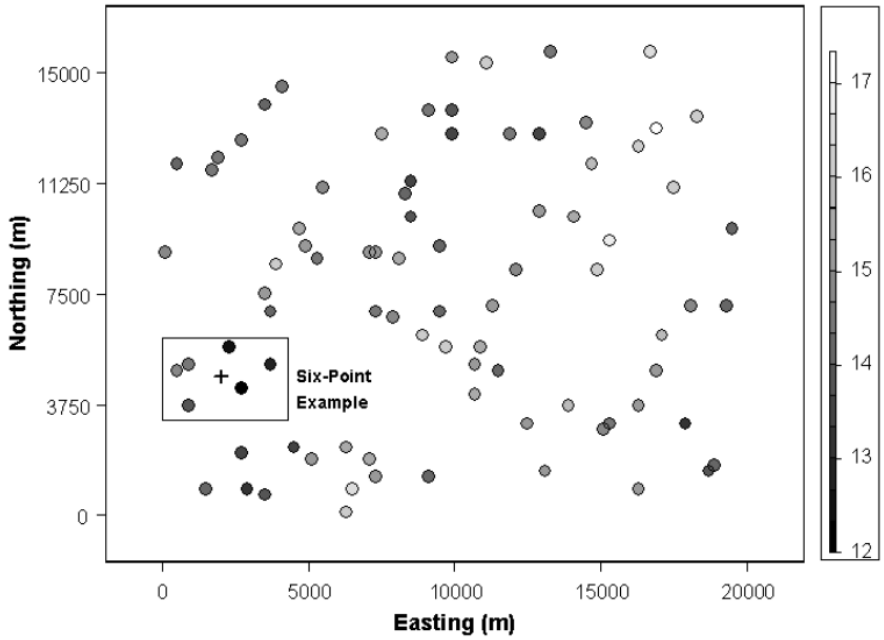

Figure 30 – Porosity measure over an spatial region. We want to estimate the porosity value in an unmeasured point marker with +.

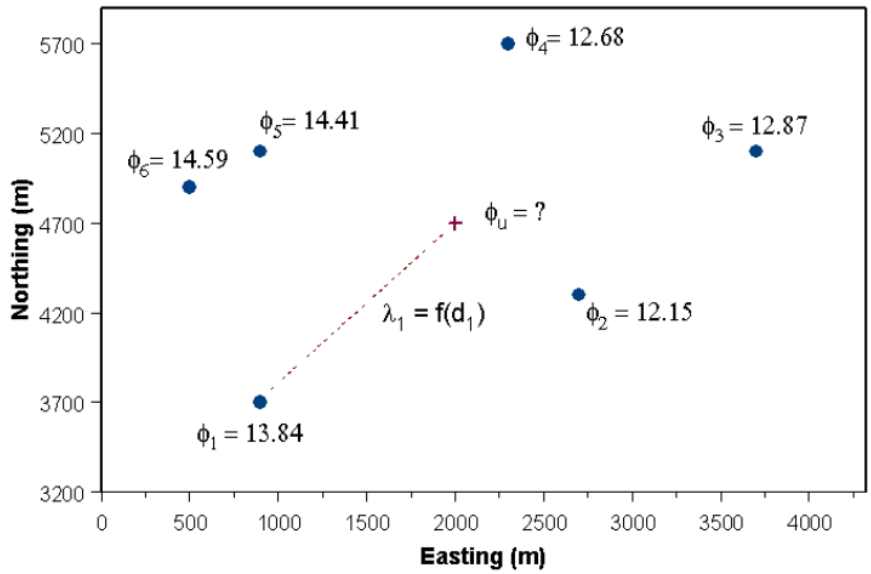

Figure 31 – Nearest six data points surrounding the point where we want to estimate the porosity.

All kriging estimators are but variants of the basic linear regression estimator defined as:

$$Z(u*) - m(u) = \sum_{\alpha=1}^{n(u)} \lambda_\alpha \left[Z(u_\alpha - m(u_\alpha)) \right] \tag{5.8}$$

with:

- $\boldsymbol{u}, \boldsymbol{u_\alpha}$ location vectors for estimation point and one of the neighboring data points, indexed by α,
- $\boldsymbol{n(u)}$ number of data points in local neighborhood used for estimation of $Z(u*)$,
- $\boldsymbol{m(u)}, \boldsymbol{m(u_\alpha)}$ expected values (means) of $Z(u)$ and $Z(u*)$,
- $\boldsymbol{\lambda_\alpha}$ kriging weight assigned to datum $Z(u_\alpha)$ for estimation location u; same datum will receive different weight for different estimation location

The goal is to determine weights, $\boldsymbol{\lambda_\alpha}$, that minimize the variance of the estimator

$$\sigma_E^2 = Var\{Z*(u) - Z(u)\} \tag{5.9}$$

Kriging weights, $\boldsymbol{\lambda_\alpha}$, are derived from covariance function or semivariogram. The semivariogram is defined as:

$$\gamma(u_i, u_j) = \frac{1}{2} Var\{Z(u_i) - Z(u_j)\} \tag{5.10}$$

If two locations, u_i and u_j, are close to each other, you expect them to be similar, so the difference in their values, $Z(u_i) - Z(u_j)$, will be small. As u_i and u_j get farther apart, they become less similar, so the difference in their values, $Z(u_i) - Z(u_j)$, will become larger.

There are different types of semivariograms, such as: circular, spherical, tetraspherical, exponential, Gaussian, etc.

The semivariogram provide information on the spatial autocorrelation of datasets. However, they do not provide information for all possible directions and distances. For this reason, and to ensure that kriging predictions have positive kriging variances, it is necessary to fit a model (in other words, a continuous function or curve) to the empirical semivariogram.

After you have uncovered the dependence or autocorrelation in your data and have finished with the first use of the data -using the spatial information in the data to compute distances and model the spatial autocorrelation- you can make a prediction using the fitted model.

Summarizing, the kriging method is a process with three main steps (i) estimate the semivariogram, (ii) fit a model to the empirical semivariogram, and (iii) make predictions.

Those steps are implemented into the **gstat** R package. The implementation of this workflow over the **GLD** using **gstat** is shown in Algorithm 4. In line 2 the semivariogram is created, **gstat** provide different functions to create different semivariograms. In line 3 we

fit a model using the dataset and the semivariogram. Finally in line 4 we make predictions of the λ values of the *GLDs* over the unmeasured regions.

Algorithm 4 Spatio-temporal interpolation over the $\lambda_{(2,3,4)}$ values of the GLD.

1: **function** GLDKRIGING($S(s_i, t_j, < 0, \lambda_2, \lambda_3, \lambda_4 >)$)
2: $gldModel \leftarrow$ VGMST($S(s_i, t_j, < 0, \lambda_2, \lambda_3, \lambda_4 >)$)
3: $fitGldVariogram \leftarrow$ FIT.STVARIOGRAM($gldModel$)
4: $S''(s_i, t_j) \leftarrow$ KRIGEST($fitGldVariogram, (\mathcal{S}_i, \mathcal{T}_j)$)

As a result of this algorithm a new dataset with the interpolated values of the **GLD** in the spatio-temporal locations not previously analyzed is returned, Equation 5.11. This strategy answers query **RQ2.** for the given spatio-temporal location.

$$S''(s_i, t_j, GLD(\lambda_1, \lambda_2, \lambda_3, \lambda_4)) \tag{5.11}$$

5.4 GLD Clustering

In Chapter 4 we discussed how to use a *GLD* to group uncertain data. Algorithm 5, proposed in that chapter, is used here to provide the clustering of *GLDs*.

Algorithm 5 Clustering the GLD based on its $\lambda_{(2,3,4)}$ values.

1: **function** GLDCLUSTERING($S(s_i, t_j, < 0, \lambda_2, \lambda_3, \lambda_4 >)$)
2: $S(s_i, t_j, clusterID_I) \leftarrow$ FIRSTCLUSTERSTEP($S(s_i, t_j, \lambda_2)$)
3: **for each** $clId_I$ **do**
4: $S(s_i, t_j, clusterID_{II}) \leftarrow$ SECONDCLUSTERSTEP($S(s_i, t_j, < \lambda_3, \lambda_4 >), S(s_i, t_j, clusterID_I)$)

Algorithm 5 receives as an input the results of the previous steps, could be the outcome of the fitting process, Equation 5.3 or that of the kriging process Equation 5.11. The output of the algorithm is a dataset with schema where the cluster ids are associated to each GLD:

$$S_{\mathcal{C}}(s_i, t_j, GLD_k, clusterID) \tag{5.12}$$

where: $clusterID$ represents the ID of the cluster to which the *GLD* at the spatio-temporal location (s_i, t_j) belongs to.

Once the dataset has been clusterized according to the *GLD*, we can use this result to characterize the uncertainty in a particular spatio-temporal region, or to measure numerically the corresponding uncertainty. In subsections 5.5.1 and 5.5.2, we describe how those approaches are implemented (see Figure 28).

5.5 UQ Queries

In this section we discuss how UQ questions can be answered with the use of the UQ workflow. The following queries motivated in chapter 1 are discussed:

RQ3. what is the uncertainty at a specific spatio-temporal region?
RQ4. how to compare two regions as a function of their uncertainty?
RQ5. what is the least uncertain from a set of models?

5.5.1 Use of GLD mixture to characterize the uncertainty in an spatio-temporal region

Lets start with **RQ.3**. The naive algorithm to answer this question could be something as follow: given a spatio-temporal region $(\mathcal{S}_i \times \mathcal{T}_j)$, read and analyze all the data generated during the simulation process on that region. This is a process current used by many researchers and proposed for example in provenance softwares. It is clear that this approach is not so efficient. Now, in our approach we first find the *GLD* that best fits the dataset at each (s_i, t_j), then we test whether the fit is a good one. As a *GLD* is proved to be a good random variate generator (see Section 3.6), then we can substitute the raw data by the associated *GLD*. Next, in the clustering step, we group all the *GLDs* by their similarities, and then we test if the centroid of each cluster statistically represent the remaining members of their cluster. If this condition is met, then is possible to substitute each *GLD* by the centroid of the cluster it belongs to. At the end of this process we substitute the raw data by a few centroids of the clusters.

Now, coming back to the question: *"what is the uncertainty at a specific spatio-temporal region* $(\mathcal{S}_i \times \mathcal{T}_j)$*?"*, we can answer it looking to the centroids of the clusters.

In $(\mathcal{S}_i \times \mathcal{T}_j)$ each cluster may be qualified with a weight given by (see also 5.5):

$$w_k = \frac{1}{N} \sum_{i=1}^{S} \sum_{j=1}^{T} w(s_i, t_j) \tag{5.13}$$

where:

$$w(s_i, t_j) = \begin{cases} 1 & \text{if } clusterID(s_i, t_j) = k \\ 0 & \text{otherwise} \end{cases} \tag{5.14}$$

and N is the number of points in the region $(\mathcal{S}_i \times \mathcal{T}_j)$.

The weight w_k is the frequentist probability of occurrence of the cluster k in the region, and complies with the conditions outlined in section 3.5 that $w_k \geq 0$ and $\sum w_k = 1$.

Remember that the mixture of the *GLDs* can be written as:

$$f(x) = \sum_{k=1}^{K} w_k GLD(\lambda_1, \lambda_2, \lambda_3, \lambda_4) \tag{5.15}$$

So, if we have the weights and a representative *GLD* for each cluster, we have the mixture of *GLD* that characterizes the uncertainty in the spatio-temporal region $(\mathcal{S}_i \times \mathcal{T}_j)$.

The process to generate the mixture of *GLDs* that characterizes the uncertainty on a region $(\mathcal{S}_i \times \mathcal{T}_j)$ is presented in Algorithm 6.

Algorithm 6 GLD mixture in a region $(\mathcal{S}_i \times \mathcal{T}_j)$

1: **function** GLDMIXTURE($\mathcal{S}_i \times \mathcal{T}_j, C_{\mathcal{S}_i \times \mathcal{T}_j}$)
2: **for each** p_i in $(\mathcal{S}_i \times \mathcal{T}_j)$ **do**
3: $c \leftarrow cluster(p_i)$
4: $w_c = w_c + 1$
5: $N = N + 1$
6: **end for**
7: **return** $\frac{1}{N} \sum_{c}^{C_{(\mathcal{S}_i \times \mathcal{T}_j)}} w_c * c.getGLD()$

5.5.2 Information Entropy as a measure of the uncertainty in an spatio-temporal region

Another way to answer query **RQ.3** is using the Information Entropy (see Section 2.2.3.2). In that section we highlight a limitation related to the fact that we need to know the possible outcomes of the system to use Information Entropy. In the cases where we can use the clusters obtained in 5.4 as the different outcomes of the system, Information Entropy could be used as a measure of the uncertainty in an spatio-temporal region.

The equation 2.3 can be rewritten as follows:

$$H(s,t) = -\sum_{c=1}^{C} p_c(s,t) \log p_c(s,t) \tag{5.16}$$

where c represents a particular cluster of the total number of clusters C, and $p_c(s,t)$ represents the probability of occurences of the cluster c in the spatio-temporal region (s,t).

Algorithm 7 computes the Information Entropy in a region $C_{(\mathcal{S}_i \times \mathcal{T}_j)}$. In lines 2 to 7, we compute the probability of each cluster in the region, similar to section 5.5.1. Using this probability we compute the Information Entropy $H(s,t)$, line 8, and finally we return the result in line 9.

The advantage of the Information Entropy, with respect to the GLD mixture approach, is that it synthesizes the uncertainty in a single comparable value.

5.5.3 Information Entropy and regions comparison

Information Entropy can be used to answer other questions, such as: *"**RQ4.** how to compare two regions as a function of their uncertainty?"*. Its application here is straightforward, as it has been mentioned in Section 2.2.3, the Information Entropy is

Algorithm 7 Information Entropy in a region $(\mathcal{S}_i \times \mathcal{T}_j)$

function GLDMIXTURE($\mathcal{S}_i \times \mathcal{T}_j, C_{\mathcal{S}_i \times \mathcal{T}_j}$)
  for each p_i in $(\mathcal{S}_i \times \mathcal{T}_j)$ **do**
    $c \leftarrow cluster(p_i)$
    $w_c = w_c + 1$
    $N = N + 1$
  end for
  $p_c(s,t) = \frac{w_c}{N}$
  $H(s,t) \leftarrow -\sum_{c=1}^{C} p_c(s,t) \log p_c(s,t)$
  return $H(s,t)$

zero when we are certain and reaches a maximum value when the uncertainty is maximal, respectively. Then, if we want to compare two regions we compute the $H(s,t)$ for each one, and the region with the smaller value of $H(s,t)$ is the taken as the one presenting less uncertainty.

5.5.4 Information Entropy and model selection

Similarly to the previous section, we can use Information Entropy to compare two models. Suppose we have a set of models $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_n\}$ and we want to know what is the less uncertain in $(\mathcal{S}_i \times \mathcal{T}_j)$. We proced to compute $H(s,t,\mathcal{M}_i)$ for each model, and the one with smaller value of $H(s,t,\mathcal{M}_i)$ is taken as the least uncertain in that region.

5.5.5 Other queries

Although in this section we present four different queries to answer three different questions (**RQ.3**, **RQ.4** and **RQ.5**), our approach leaves open the option to develop new queries to solve questions that arise in UQ context. In Section 6.2.3 we show how new queries can be answered quickly, thanks to the flexibility of this approach.

5.6 SUQ^2 R package

The proposed approach has been implemented as an R package named SUQ^2, an acronym of **S**imulation **U**ncertainty **Q**uantification **Q**uerying, freely available at SUQ^2. As this package is in a development stage, to install it you need to install the R **devtools** package first and then use a function **install_github** to install SUQ^2.

```
> install.packages("devtools")
> install_github("nmlemus/suq2")
```

The package has been divided into five sub-package, one for each step or the workflow and other one to show the results. The name of the sub-package are: **fit**, **kriging**, **clustering**, **queries** and **plot**.

As R is an interactive language, to warrant the easy to use of the functions inside the package we use the following pattern: (i) the name of all package fucntions start with **suq2.**; (ii) after **suq2.** we add a name that identifies a sub-package, for example the name of all the functions of the sub-package **plot** start with **suq2.plot.**; (iii) and finally a name that identifies the functionality of the function. For example a function to plot a gld based on its λ values is named **suq2.plot.gld()**, while a function to cluster the *GLDs* based on λ_2 is **suq2.clustering.lambda2()**.

The full documentation of the package is available inside the installation and in Appendix A.

5.7 Summary

In this Chapter we present the workflow of the proposed approach. The workflow is divided into four steps: fitting, kriging, clustering and queries. The algorithms of all the steps are presented and commented. In Section 5.5 the queries to answer the research questions raised in the introduction were discuss. Finally, in Section 5.6 we present SUQ^2, an R package that implements our approach.

6 Use Cases

In this Chapter we test our approach in two different case studies. Section 6.1 presents a case study from the seismic field, where we test the full workflow proposed in Chapter 5. Section 6.2 presents an environmental problem. This second case requires answering a question, for which the proposed queries are not sufficient, then we illustrate how to extend the proposed framework to include the queries we need to answer the question. Finally, Section 6.3 summarize the main results of the current Chapter.

6.1 Case Study: HPC4E Seismic Test Suite

As a first case study we use the "HPC4E Seismic Test Suite", a collection of four 3D models and sixteen associated tests that can be downloaded freely at the project's website (<https://hpc4e.eu/downloads/datasets-and-software>) (Josep de la Puente, 2015). Those models have been designed as a set of 16 layers with constant physical properties. The top layer delineates the topography and the other 15 delineate different layer interface surfaces or horizons. A Matlab script is provided that generates 3D gridded volumes and 2D gridded layer surfaces for any desired spacing and for three different variables $v_p(m/s)$, $v_s(m/s)$ and $density(Kg/m^3)$. For example, to generate a 3D volume with dimensions $250 \times 501 \times 501$ in the $v_p(m/s)$ variable we can use the values of Table 9, and run the Matlab script *generate_hpc4e_grid.m*.

The first slice of this 3D volume generated by the script is shown in Figure 32.

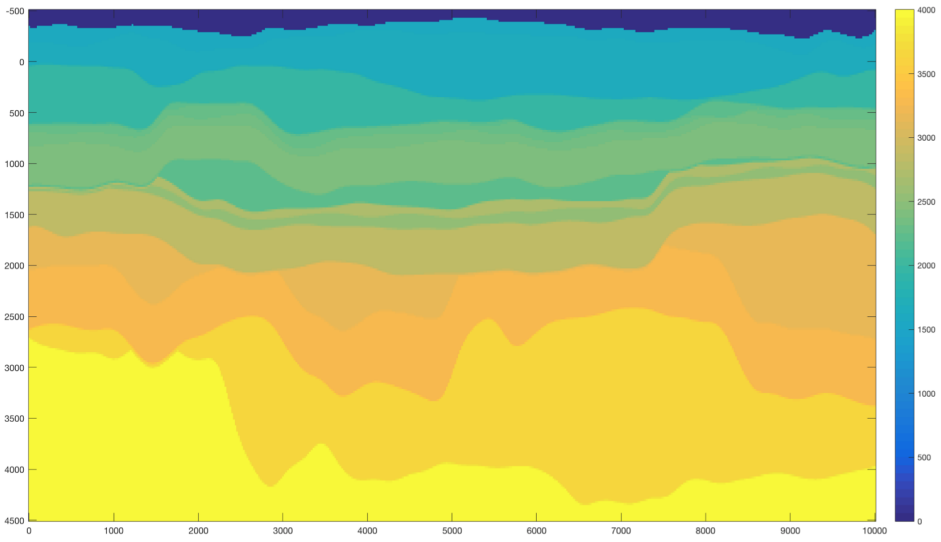

Figure 32 – One slice of the $250 \times 501 \times 501$ cube. In the slice we can distinguish between the different layers.

Layer	$v_p(m/s)$
1	1618.92
2	1684.08
3	1994.35
4	2209.71
5	2305.55
6	2360.95
7	2381.95
8	2223.41
9	2712.06
10	2532.22
11	2841.03
12	3169.31
13	3252.35
14	3642.28
15	3659.22
16	4000.00

Table 9 – Values of v_p used in the generation of a single velocity field cube.

6.1.1 The Dataset

The Matlab script *generate_hpc4e_grid.m* could be consider as a mathematical model (black box) $v_p(x, y, z) = \mathcal{M}(v_p)$ that receive v_p as an input variable and generate a $v_p(x, y, z)$ as an output. In the benchmark the values of v_p are fixed (Table 9), so we cannot use it as is. We need to take the input, $v_p(m/s)$ in this case, as a random variable. To achieve this, we designate a *PDFs* to each v_p for each layer as shown in Table 10.

Now, using a Monte Carlo method we can sampling the input variable v_p and run as many simulations as we want using the model $v_p(x, y, z) = \mathcal{M}(v_p)$, to posteriorly analyze the uncertainty in the output $v_p(x, y, z)$. An example of 1000 samplings of v_p is shown in Figure 33. Is clear that this is a not a real problem as a benchmark was not conceived as an uncertainty problem, but the datasets generated here are useful for our purposes.

Using the 1000 samplings of Figure 33, we run the model $v_p(x, y, z) = \mathcal{M}(v_p)$ times, to generate 1000 3D cubes with dimensions of $250 \times 501 \times 501$. The full dataset is a multidimensional array of $250 \times 501 \times 501 \times 1000$ with a size og 245 GB.

To simplify the computational process and visualize the results, we select the slice 200 of each 3D cubes, then our dataset is now a multidimensional array of $250 \times 501 \times 1000$. This case study is a the spatial domain only, then the equation 2.8 can be rewrited as $S(x_i, y_j, simId, v_p(x_i, y_j))$, where $i = 1, 2, ..., 250$, $j = 1, 2, ..., 501$ and $simId = 1, 2, ..., 1000$. In this new representation (x_i, y_j) are the 2D coordinates and

Layer	PDF Family	Parameters
1	Gaussian	[1619, 711.2]
2	Gaussian	[3368, 711.2]
3	Gaussian	[8839, 711.2]
4	Gaussian	[7698, 301.5]
5	Lognormal	[7723, 294.7]
6	Lognormal	[7733, 292.2]
7	Lognormal	[7658, 312.1]
8	Lognormal	[3687, 368.7]
9	Exponential	[3949, 394.9]
10	Exponential	[5983, 711.2]
11	Exponential	[3520, 352.0]
12	Exponential	[3155, 315.5]
13	Uniform	[2541, 396.4]
14	Uniform	[2931, 435.3]
15	Uniform	[2948, 437.0]
16	Uniform	[3289, 471.1]

Table 10 – PDFs and its parameters used to sampling the v_p, to generate n velocity models.

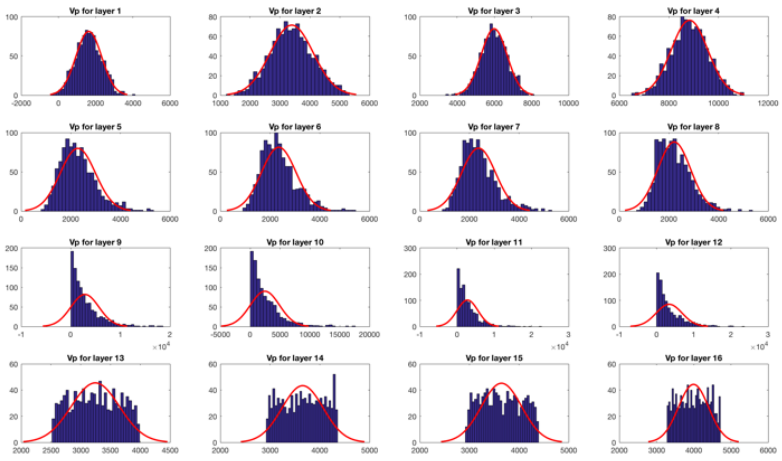

Figure 33 – Histograms of the 1000 samplings generated using Monte Carlo method and the PDFs reported in Table 10.

$v_p(x_i, y_j)$ is the velocity value at point (x_i, y_j). *simId* still represents the Id of the simulation.

Now that we have an experimental dataset we can apply the proposed workflow step by step.

6.1.2 Fitting the GLD

The first step is to find the *GLD* that best fits the dataset at each spatial location. Running the algorithm proposed in Section 5.2.1 we get as a result a new 2D array:

$$S'(x_i, y_j, GLD(\lambda_1, \lambda_2, \lambda_3, \lambda_4)) \tag{6.1}$$

The raw data is reduced and our dataset is characterized by four lambda values at each spatial location. Now we need to assess the validity of the *GLDs* and how well they fit the dataset. Those analyses are described in sections 6.1.2.1 and 6.1.2.2.

6.1.2.1 GLD validity check

Once the algorithm to check the validity of the *GLD* is run on the experimental dataset, we obtain as a result that the *GLD* is valid in all the (x_i, y_j) space.

6.1.2.2 Quality of the fit

The next step is to check how good is the fit. To do this we use an algorithm that returns the D and *p-value* for the KS-test at each spatial location. As we show in figure 34, and remember that with a *p-value* > 0.05 we cannot reject the null hypothesis, we conclude that the fit of the GLD is acceptable in most cases. To be more exact, the p-value was greater than 0.05 in 82 % of the spatial locations, figure 35.

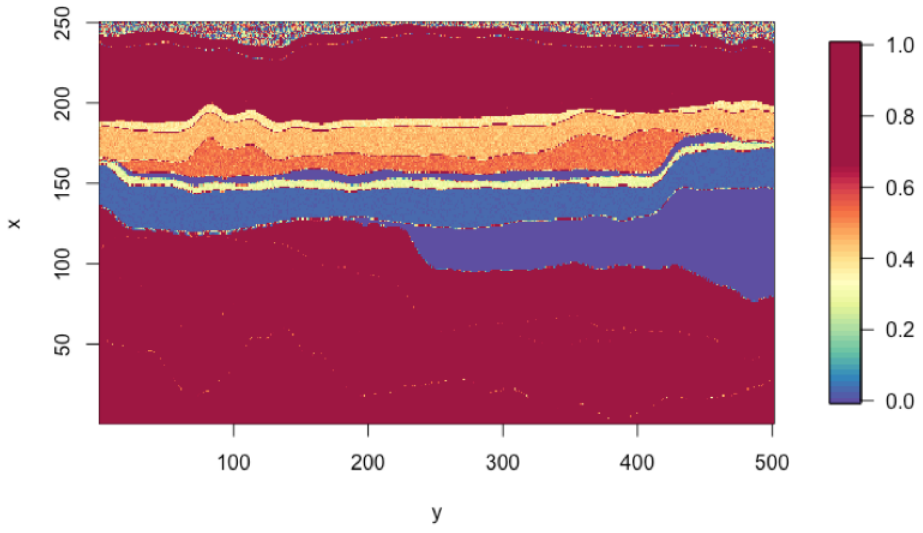

Figure 34 – Goodness of the fit based on the p-value returning by the KS-test. p-value $>$ 0.05 represent a good fit of the GLD to the dataset at (x_i, y_j).

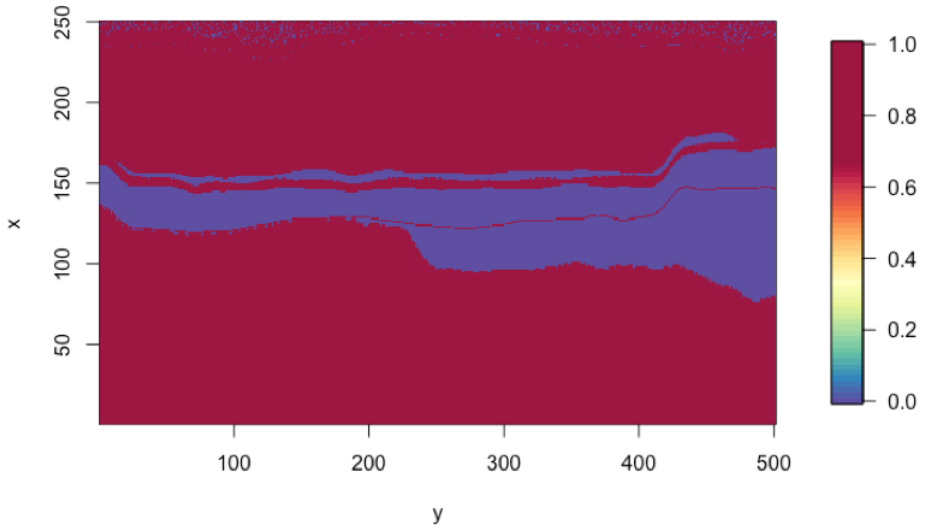

Figure 35 – The red color shows where the p-value was greater than 0.05.

If we consider the distance D, returned by the KS-test, the result is similar, figure 36. We can see a blue region that is common in figures 35 and 36. This region is where the quality of the GLD fit is below a threshold. On those cases, some *GLD* extensions proposed in (KARIAN; DUDEWICZ, 2011) could be used.

As the main purpose of this thesis is to demonstrate the utility of the use of the *GLD* in *UQ*, then we are not going to deep in other algorithms to solve this particular problem.

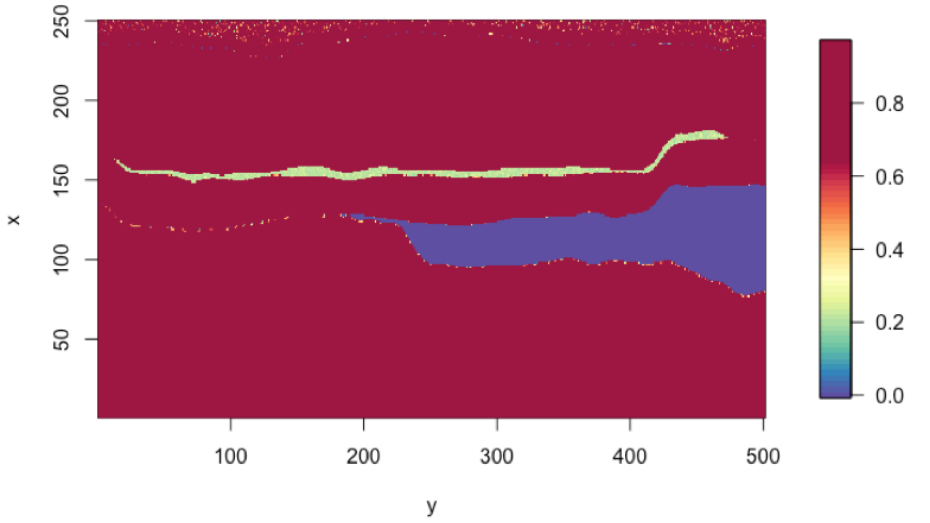

Figure 36 – Kolmogorov-Smirnoff Distance (D). The red regions represent where the GLD fits well.

6.1.3 Clustering

At this point we have our dataset characterized by the *lambda* values of the *GLDs* on each (x_i, y_j), then using the clustering algorithm proposed in Section 5.4, over $(\lambda_2, \lambda_3, \lambda_4)$ values, we get a result shown in Figure 37. A new dataset is produced, where for each spatial location we have a label that indicates the cluster the GLD at each position belongs to (see the schema at Equation 6.2). Note that, in Figure 37, the blue region corresponding to cluster 11 is not a cluster itself. It is rather the region where the *GLD* is not valid, see section 6.1.2.2.

$$S_{\mathcal{C}}(x_i, y_j, clusterID, GLD_{x_i,y_j}) \tag{6.2}$$

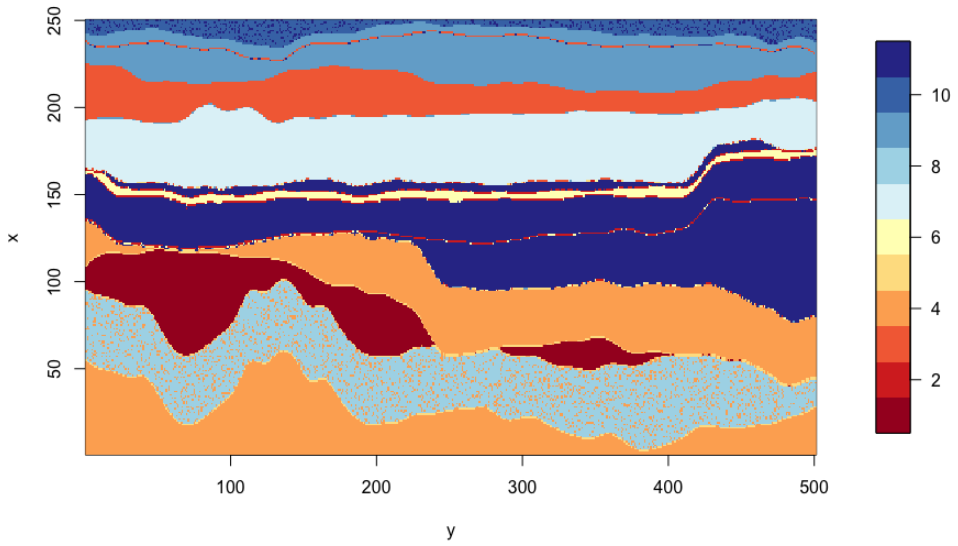

Figure 37 – Result of the clusterization using the clustering algorithm proposed in Section 5.4, over $(\lambda_2, \lambda_3, \lambda_4)$ values with $k = 10$.

If we compare visually Figures 32 and 37, we observe a close similarity. It is clear that they can not be equal because we are talking about a slice of a deterministic model, and the result of clustering 1000 realizations of a stochastic model, but as the model used here is very linear, this is the result we expect.

Another interesting result is shown in Figure 38, where we plot the clusters in (λ_3, λ_4) space. As we mention in section 3.2, the shape of the *GLD* depends on the values of λ_3 and λ_4. In this scenario, the expected result is that the members of the same cluster share similar values of λ_3 and λ_4, that is exactly the result we can observe in this figure.

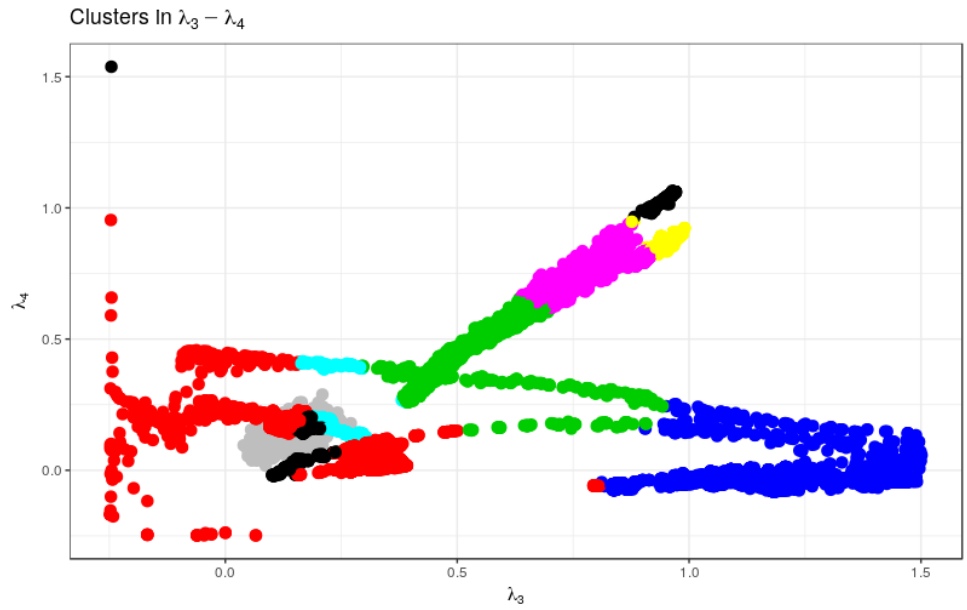

Figure 38 – Distribution of the clusters in the (λ_3, λ_4) space. The points that belongs to a same cluster are one near the others, as was expected.

To further corroborate this fact, in Figure 39 we show the *PDFs* of 60 members of the 10 clusters. Visually assessing the figures we have an idea of how similar are the shapes of the members of the same cluster and how dissimilar are the shapes of the members of different clusters. This suggests that our approach is valid. A product of these observations is that we can pick one member of each cluster (the centroid) as a representative of all the members of the cluster, Table 11. The selected member is used to answer the queries in the next sections.

Cluster	λ_2	λ_3	λ_4
1	0.0013937313	0.9585829	1.04696461
2	0.0005291388	1.1633978	-0.07162550
3	0.0020630696	0.1349486	0.17305941
4	0.0016238358	0.8653824	0.83857646
5	0.0027346929	0.5084664	0.39199164
6	0.0003894541	1.4076354	-0.01925743
7	0.0021972784	0.3253562	0.01493809
8	0.0015421749	0.9491101	0.86699555
9	0.0018672401	0.2176002	0.17862024
10	0.4856397733	0.1404140	0.14011298

Table 11 – Centers of the clusters obtained using the clustering algorithm proposed in Section 5.4, over $(\lambda_2, \lambda_3, \lambda_4)$ values.

The 125250 points of the slice are distributed through the clusters following the histogram of the figure 40 and Table 12.

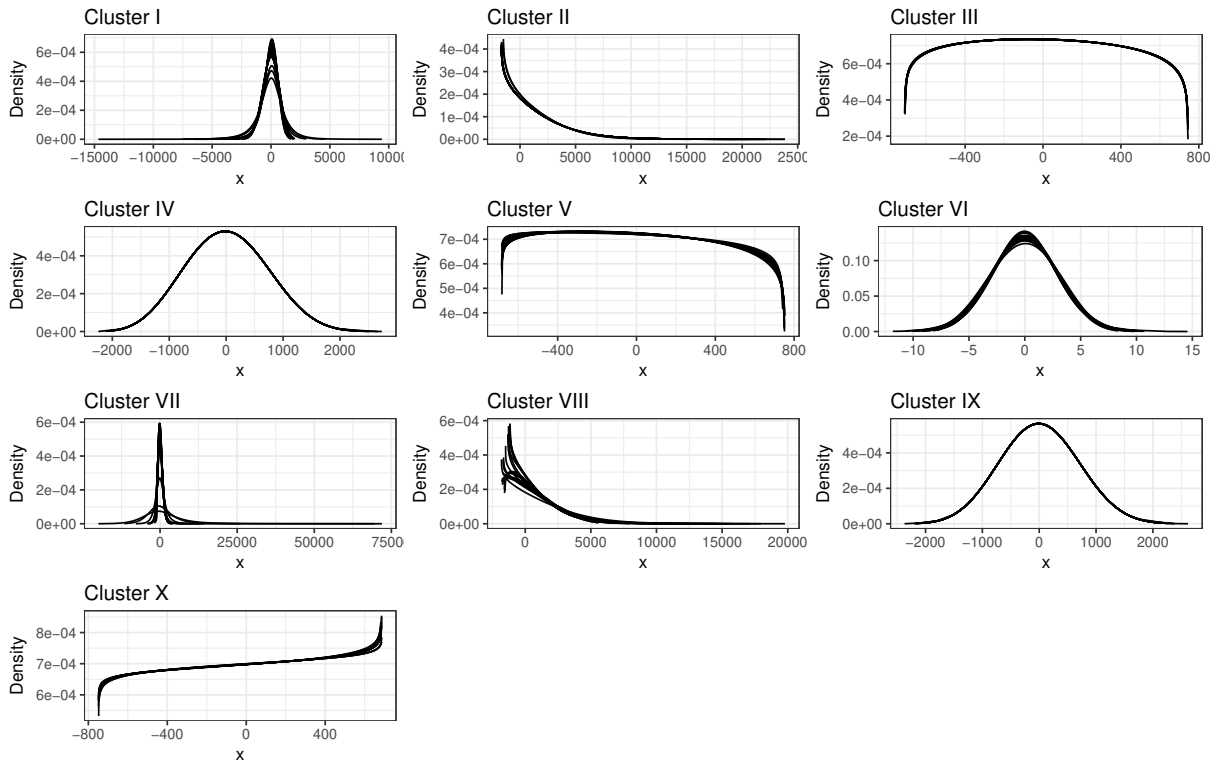

Figure 39 – *PDFs* of 60 members of the 10 clusters obtained using the clustering algorithm proposed in Section 5.4, over $(\lambda_2, \lambda_3, \lambda_4)$ values.

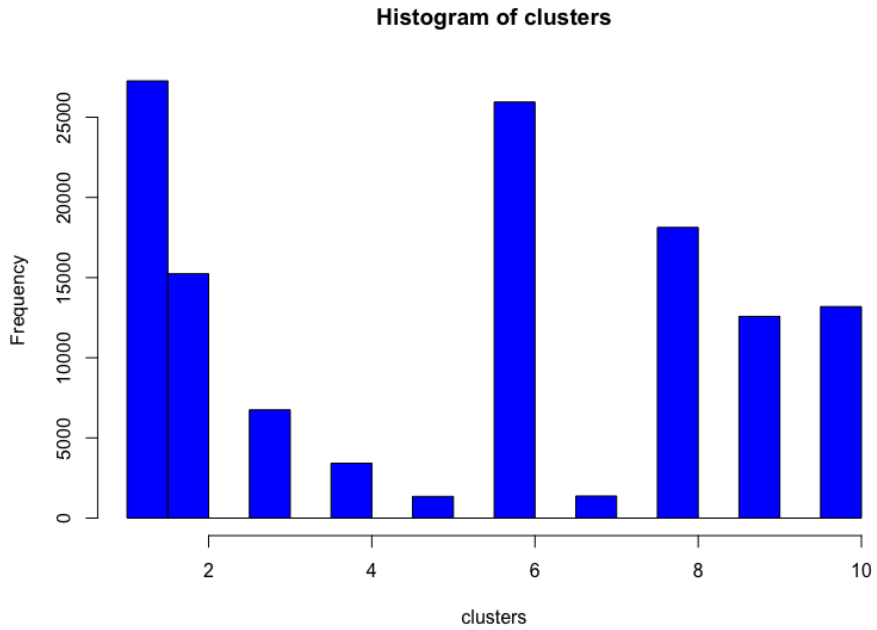

Figure 40 – Distribution of the clusters.

Cluster	No. of members
1	27217
2	15223
3	6749
4	3421
5	1353
6	25853
7	1374
8	18103
9	12051
10	13156

Table 12 – Distribution of the clusters.

6.1.4 Spatio-temporal queries

At this point, the initial dataset is summarized as depicted by the schema in equation 6.2. It can be used to answer queries and to validate our approach, comparing the results with the raw data.

First of all we select four spatio-temporal regions of the dataset where the clusters suggest us different behaviors. The regions are shown in Figure 41 and the values of $[x_1, x_2], [y_1, y_2]$ that define the regions are shown in Table 13.

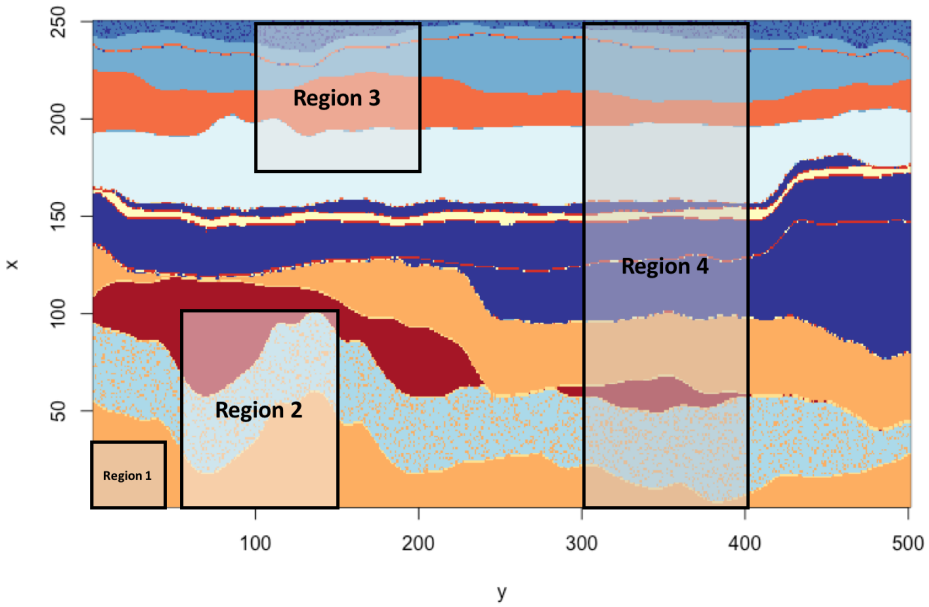

Figure 41 – Analysis Regions.

With these four regions we assess the adoption of the *GLD* mixture to obtain the *PDF* that characterizes the uncertainty in an specific region, Section 6.1.4.1. In Section 6.1.4.2 we use the Information Entropy to assign a value that measures the uncertainty at

Region	x_1	x_2	y_1	y_2
Region 1	210	250	0	40
Region 2	150	250	50	150
Region 3	0	75	100	200
Region 4	0	250	300	400

Table 13 – Analysis Regions.

each region. The expected result is that in section 6.1.4.1 the GLD mixture characterize correctly the raw data; and in 6.1.4.2 we expect a zero as a value of the Information Entropy in region 1 and increased values between regions 2, 3 and 4.

6.1.4.1 GLD mixture

The experiment here is to use the representative *GLDs* at each cluster and the weight associated to it in the region. Using these parameters we can build a *GLD mixture* that characterizes the uncertainty on that region. Here we use the algorithm described in section 5.5.1.

First of all we query the region to find the clusters represented inside it, and how are they distributed. Below we show the R codes to query the four regions. The retrieved results are shown in Table 14.

$$> clRegion1 = clByRegion(210, 250, 0, 40)$$

$$> clRegion2 = clByRegion(150, 250, 50, 150)$$

$$> clRegion3 = clByRegion(0, 75, 100, 200)$$

$$> clRegion4 = clByRegion(0, 250, 300, 400)$$

Cluster	**Region 1**	**Region 2**	**Region 3**	**Region 4**
1	0	2250	0	979
2	0	0	0	268
3	0	0	2596	1468
4	1640	4467	0	5173
5	0	149	0	269
6	0	0	0	416
7	0	0	1967	3920
8	0	3335	0	3432
9	0	0	1918	3280
10	0	0	901	583

Table 14 – Distribution of the clusters by regions.

If we divide the columns of Table 14 by the sum of the elements of each column we get the weight needed to formulate the *mixed GLDs*. It is clear that the *GLD* in region 1 is represented by the *GLD* of cluster 4. On the other 3 cases we get:

$$
\begin{aligned}
&GLD_{region1} = GLD_{c4} \\
&GLD_{region2} = 0.22GLD_{c1} + 0.44GLD_{c4} + 0.014GLD_{c5} \\
&+ 0.33GLD_{c8} \\
&GLD_{region3} = 0.34GLD_{c3} + 0.26GLD_{c7} + 0.25GLD_{c9} \\
&+ 0.12GLD_{c10} \\
&GLD_{region4} = 0.22GLD_{c1} + 0.44GLD_{c4} + 0.014GLD_{c5} \\
&+ 0.33GLD_{c8}
\end{aligned}
$$

Now we need to evaluate if the *mixture of GLDs* describes well the uncertainty in the regions. To do this we perform the same *ks-test* used to evaluate the goodness of the fit and described in Section 5.2.3.

Metrics	**Region 1**	**Region 2**	**Region 3**	**Region 4**
p-value	0.73	0.56	0.34	0.08

Table 15 – p-values by regions.

Based on the *p-value*, Table 15, we can conclude that in all 4 regions the *mixture of GLDs* is a good fit to the raw data.

6.1.4.2 Information Entropy

Now we are going to evaluate what happens with the information entropy. Based on the distribution of clusters inside the regions, table 14; we can compute the entropy. In this case we use an R function called *entropy*, implemented in the r-package of the same name (HAUSSER; STRIMMER, 2008).

entropy	**Region 1**	**Region 2**	**Region 3**	**Region 4**
value	0	1.122243	1.41166	2.024246

Table 16 – Information Entropy by regions.

As we expect, Table 16, the entropy in region 1 is zero, because the region contains only members of the cluster 4. On the other regions the entropy increases from region 2 to region 4, as we expected.

It is clear that the information entropy is a very good and simple measure of the uncertainty, and here it is demonstrated its utility combined with the *GLD*.

6.2 Case Study: Distribution of the C/N Ratio

As a second case study we select an example included in an R package **spus** (K. Sawicka; SOIL, 2016). This package implement a methodology for uncertainty propagation analysis in spatial environmental. The **spus** package includes functions for uncertainty model specification, propagation of uncertainty using Monte Carlo (MC) techniques, and uncertainty visualization functions. The case study describe the spatial distribution of organic carbon (OC) and nitrogen (N), variables that are used to derive C/N ratio, vital information to evaluate soil management and to increase the crop productivity. Maps of OC and N are approximations encumbered with errors. These errors will propagate through the calculation into the C/N prediction.

6.2.1 The Dataset

The example data for C/N calculations are a 250m resolution mean OC and TN (total N) of a $33km \times 33km$ area adjacent to lake Alaotra in Madagascar.

The Madagascar dataset contains four spatial objects: a mean OC and TN of the area and their standard deviations. It also has a saved function that calculates C/N using OC and TN that will be used later.

In Figure 42 we present the mean of OC and TN in the area.

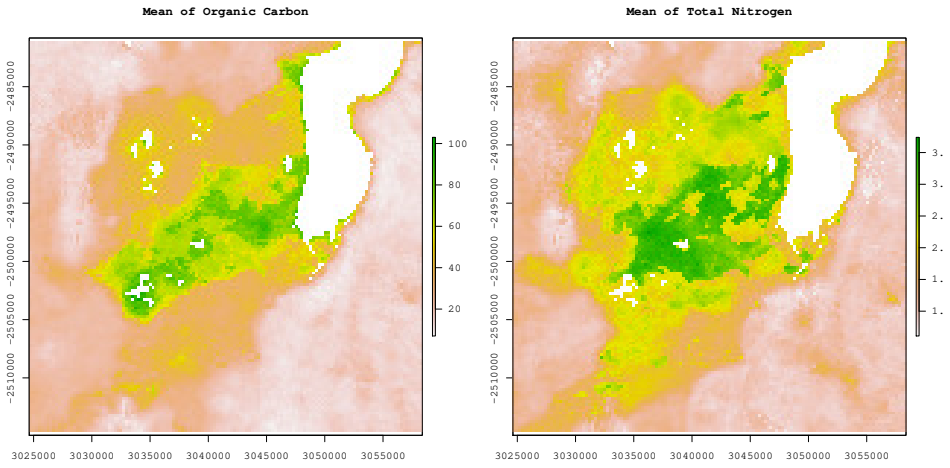

Figure 42 – Mean of organic carbon (OC) and total nitrogen (TN) of a $33km \times 33km$ area adjacent to lake Alaotra in Madagascar.

OC and TN are the input variables of a model $C/N = \mathcal{M}(OC, TN)$ defined by an R function as follow:

```
function (OC, TN)
{
    OC/TN
}
```

This R function receive OC and TN as inputs and return the C/N ratio. The original Madagascar dataset as it, only contain information about the mean and standard deviation of OC and TN. To propagate the uncertainty through a model, the authors of **spus** package define OC and TN as uncertain variables. In both cases the uncertainty is characterized by a normal distribution with mean OC and TN and standard deviation OC_sd and TN_sd respectively. Using a Monte Carlo method, the author sampling the input space to obtain 500 realizations.

The propagation of uncertainty occurs when the model is run with the uncertain inputs. Running the model with a sample of realizations of uncertain input variable(s) yields an equally large sample of model outputs that can be further analyzed, in this case 500 samples of 134×135.

6.2.2 Fitting the GLD

Now that we have the output dataset, lets proceed to use our approach. The first step is the fitting process, find the *GLD* that best fit the dataset on each location. Similar to the previous use case, we use the fitting functions of the **suq**[2] package. In all the spatial locations the algorithm return a valid *GLD* with $p_{value} > 0.05$.

6.2.3 Queries

The main question the researchers of the **spus** package are interesting in is identify all locations where the C/N ratio is smaller than 24, with 90% probability. This information might be used by farmers to identify which plots require action on improving soil quality.

This question is not part of the queries we propose in our approach. So, we show now, how queries like this one are simple to answer with our approach, demonstrating how easy is to extend it.

The *GLD* is defined by it quantile function (see Section 3.1). The question of *"locations where the C/N ratio is smaller than 24, with 90% probability"* can be translated as: *"locations where the 90% percentile of the distribution is less or equal to 24"*.

To answer this question two new algorithms are proposed (could be one algorithm but to warranty that the functions are sufficiently generic, we propose a two algorithms),

Algorithm 8 to compute the value of the quantile q of the *GLD* over a spatio-temporal region (s_i, t_j), and Algorithm 9 to compare if the quantile value returned by the previous algorithm is less or equal to the desired value.

Algorithm 8 This Algorithm return the value of the quantile q for each *GLD* in (s_i, t_j).

```
1: function GLDQUANTILE(S(s_i, t_j, < λ_1, λ_2, λ_3, λ_4 >), q)
2:     for each (s_i, t_j) do
3:         S(s_i, t_j, q_value) ← SUQ2.UTILS.QGL(S(s_i, t_j, < λ_1, λ_2, λ_3, λ_4 >), q)
```

Algorithm 9 This Algorithm assigns a 1 to $S(s_i, t_j, q_{result})$ if the value at this position meets the condition, and 0 otherwise.

```
1: function GLDQUANTILECOMPARE(S(s_i, t_j, q_value), q_compare)
2:     for each (s_i, t_j) do
3:         if q_value ≤ q_compare then
4:             S(s_i, t_j, q_result) ← 1
5:         else
6:             S(s_i, t_j, q_result) ← 0
```

In lines 4 and 6 of Algorithm 9 we assign a value 1 or 0 if the quantile at position (s_i, t_j) meets the condition or not. Figures 43 and 44 show the resultant images of the **spus** and **suq**[2] respectively. The results is not exactly the same as we can see by simple observation, but is extremely similar. To be more precise, the comparison of both image matrix return a 89% of similarity.

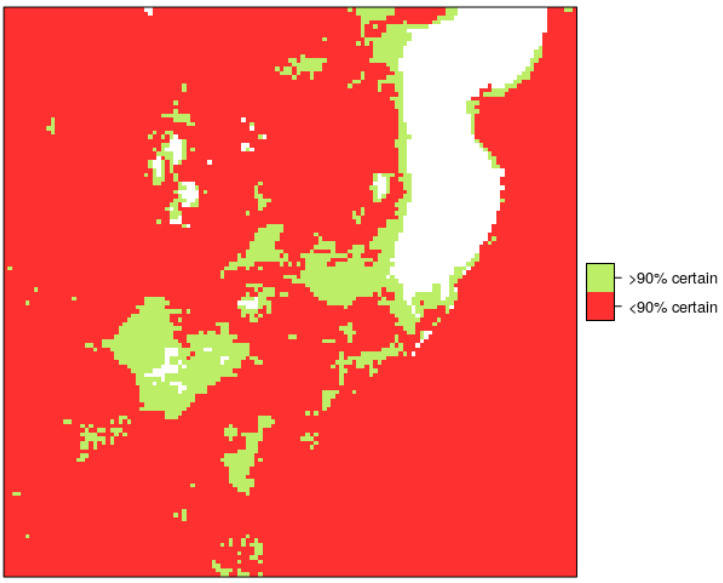

Figure 43 – Locations where the C/N ratio is smaller than 24, with 90% probability, **spus** package.

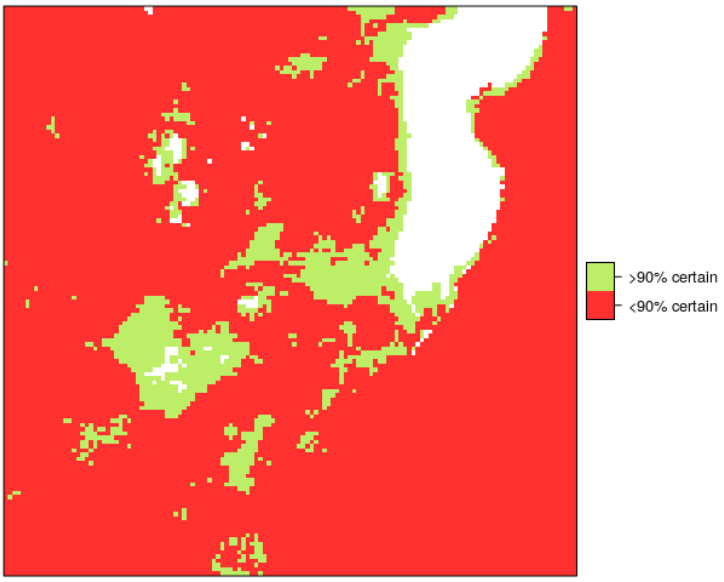

Figure 44 – Locations where the C/N ratio is smaller than 24, with 90% probability, **suq**[2] package.

6.2.4 Clustering

Although the main objective of this case study was accomplished, we test the proposed workflow with the current dataset. First to all is important to note that, as the mathematical model is $C/N = \mathcal{M}(OC, TN)$, and we use normal distributions to describe the uncertainty in both OC and TN, then we expect that the uncertainty of the output C/N follow a normal distribution too.

In Figure 45 we show the values of (λ_3, λ_4) over its space. All the values correspond to Class-I ($\lambda_3 < 1$, $\lambda_4 < 1$), sub-class I_a: $(\lambda_3, \lambda_4 \leq 1/2)$ of the *FMKL-GLD* parameterization (see Section 3.2), where we find distributions such us $Gaussian(Normal)$, $Beta(2.3)$ and $\Gamma(\alpha = 5)$. This mean that the shape of the distributions inside the dataset are similar (Normal) and (λ_3, λ_4) don't determine so much in the clustering process.

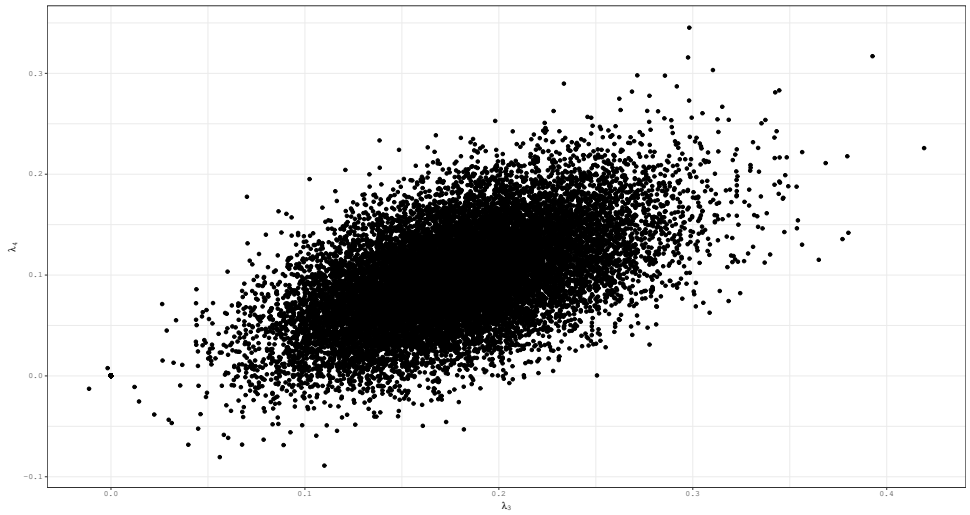

Figure 45 – Distribution of the values of (λ_3, λ_4). All the values are in the Class-I, sub-class I_a of the *FMKL-GLD* parameterization.

Using the following R code we select the optimal number of clusters in this case, based in the values of λ_2:

```
#Elbow Method for finding the optimal number of clusters
set.seed(123)
# Compute and plot wss for k = 2 to k = 15.
k.max <- 15
data <- lambda_2
wss <- sapply(1:k.max,
       function(k){kmeans(data, k, nstart=50,
               iter.max = 15 )$tot.withinss})
```

The result is shown in Figure 46. Therefore for k=4 the between_ss/total_ss ratio tends to change slowly and remain less changing as compared to other k's. So for this data k=4 should be a good choice for number of clusters.

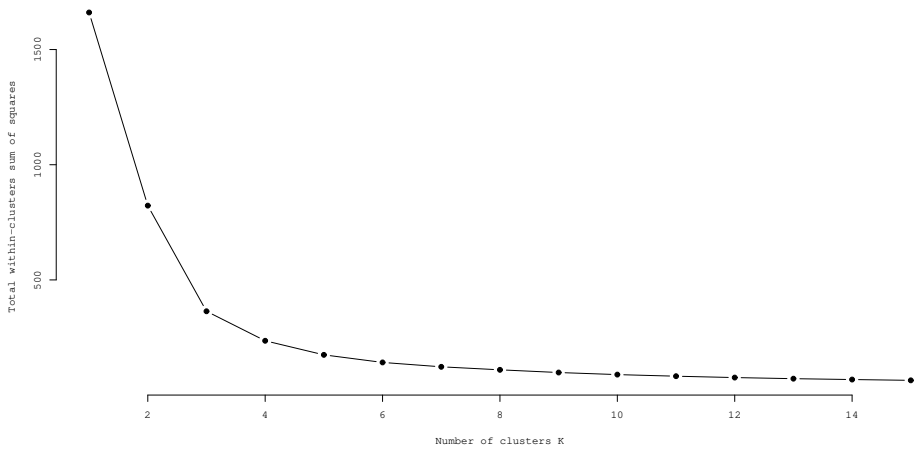

Figure 46 – Distribution of the values of (λ_3, λ_4). All the values are in the Class-I, sub-class I_a of the *FMKL-GLD* parameterization.

Using the clustering algorithm with $k = 4$ we get the image of Figure 47. In the four clusters is include the lake region where we don't have measures.

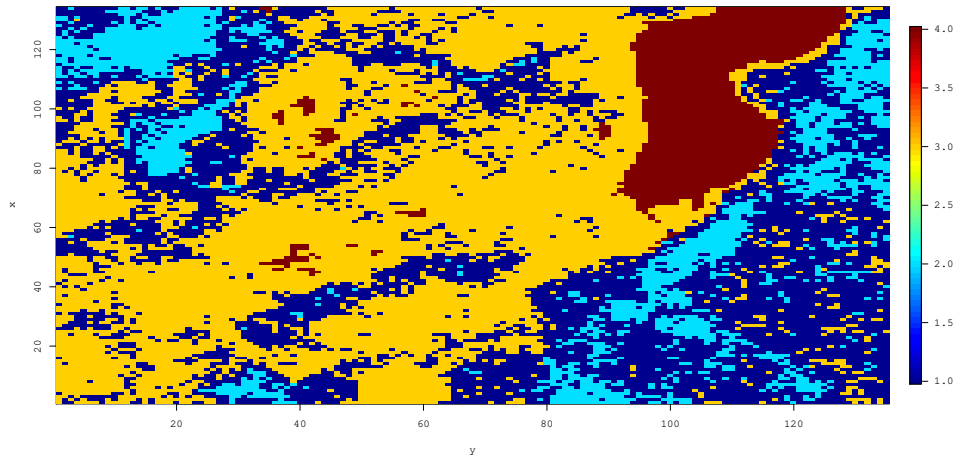

Figure 47 – Result of the clusterization using the clustering algorithm proposed in Section 5.4, over $(\lambda_2, \lambda_3, \lambda_4)$ values with $k = 4$.

In Figure 48 the clusters are projected over the (λ_3, λ_4) space. As was expected, the clusters are mixed in this figure because λ_3 and λ_4 don't determine the clusters in this case.

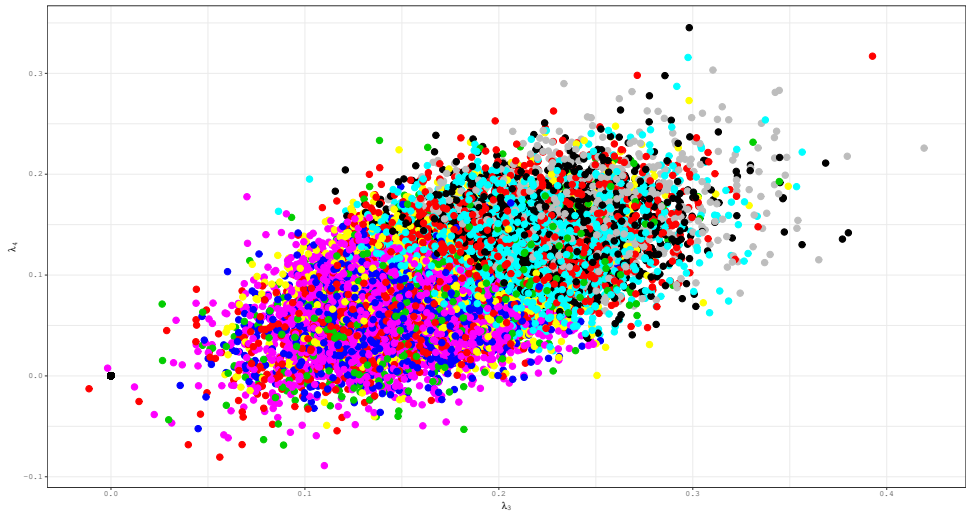

Figure 48 – Distribution of the clusters in the (λ_3, λ_4) space. The homogeneity in the image suggest that all the clusters have similar shapes.

Very different is the result we show in Figure 49 as λ_2 is the responsible of the differences in the distributions. First to all, we observe a yellow point near $(0, 0, 0)$, this point correspond to all the values of the lake region. The other three clusters are well defined in the figure.

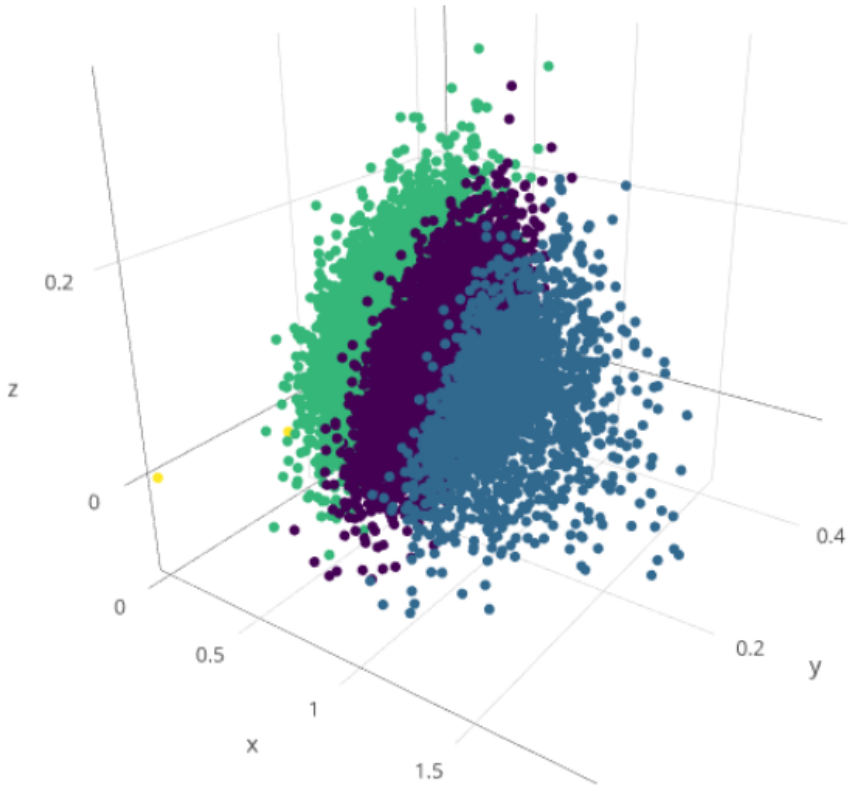

Figure 49 – Distribution of the clusters in the $(\lambda_2, \lambda_3, \lambda_4)$ space. The difference in λ_2 determine the different variances of the clusters.

Finally, in Figure 50 we show 60 members of the 4 clusters returned by the algorithm. The last figure is empty because it correspond to the lake region. On the other three figures we can see the differences in the variance, the first one is located between $[-4, 4]$, the second between $[-5, 5]$ and the last one between $[-5, 10]$.

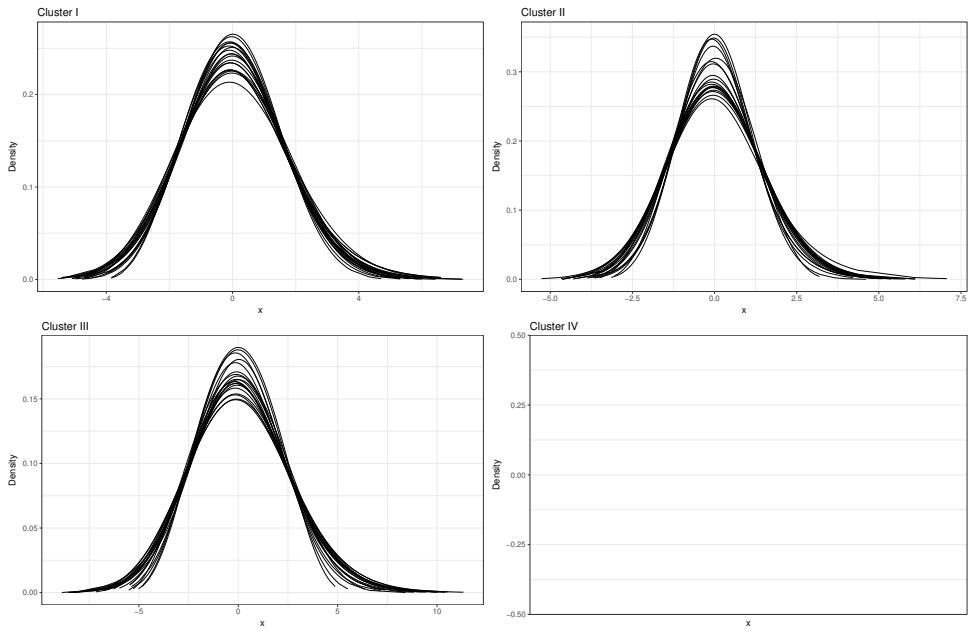

Figure 50 – *PDFs* of 60 members of the 4 clusters obtained using the clustering algorithm proposed in Section 5.4, over $(\lambda_2, \lambda_3, \lambda_4)$ values. The last cluster is empty because is the lake region, where we don't have measures of the C/N ratio.

6.3 Summary

In the present Chapter we present two case study, one from the seismic area and the second form the environmental field. In both cases the proposed approach return very good results. In the second case we need to implement two new queries to answer a question not previously analyzed, which showed the possibility of extending our approach, as necessary.

7 Conclusions and Future Works

Large-scale spatio-temporal simulations produce a huge amount of data that need to be interpreted in order to assess the simulation quality in different regions of space-time. Querying these data poses a great challenge due to their volume and different data distributions. In order to solve this problem, in this thesis we propose a general approach to answer uncertainty quantification queries.

The approach uses *GLD* that enables the representation of a spatio-temporal simulation output using a single functional formalism. By modeling each spatio-temporal point by a *GLD* instance, we can synthesize the region in a number of clusters, represented by their centroid *GLD* function. From this basis, queries can be answered by combining the centroids in a spatio-time region into *GLD*-mixture functions. Moreover, by using information entropy techniques, a value can be assigned that represents the uncertainty in a region. The proposed approach is implemented in a workflow that can be extended to solve new UQ queries.

We ran extensive experiments using two use cases. The results showed that *GLD* representation of the data is valid on 85% of the dataset. Other extensions of the GLD formalism, such as EGLD (KARIAN; DUDEWICZ, 2011), can be evaluated to improve the GLD dataset coverage. Moreover, we showed that the computed centroid function is a good representation of the function instances in its cluster. Additionally, we use the Kolmogorov-Smirnov test to evaluate the quality of the GLD mixture. The p-value, larger than 0.05, assures that the results of the mixture is a good representation of the raw data in the region. Finally, the adoption of the Information Entropy technique was validated by showing the correspondence of the computed values with the uncertainty in the spatio-temporal regions.

To the best of our knowledge, this is the first work to use GLD as the basis for answering UQ queries in spatio-temporal regions and to compile a series of techniques to produce a query answering workflow.

7.1 Revisiting the Research Questions

Let us now revisit the research questions.

RQ1. *how to group the output of the UQ process based on the similarity of the uncertainty?* In Chapter 4 we demonstrate how the *GLD* λ values can be used to cluster uncertain data. The proposed approach was tested against two synthetic datasets and the results we got were exactly what we expected.

RQ2. *what is the uncertainty in some spatio-temporal locations not previously analyzed?* To answer this research question, we propose in Chapter 5 the use of the state-of-the-art methods for spatio-temporal interpolation (kriging).

RQ3. *what is the uncertainty at a specific spatio-temporal region?* Also in Chapter 5 we propose two algorithms to deal with this research question. The first one proposes the use of *GLD* mixtures to characterize the uncertainty in a region as a **PDF**. The second algorithm proposes the use of Information Entropy (IE) to quantify the uncertainty of the region in a single number. The second approach can be used when the different clusters that represent the *GLDs* of the regions, can be interpreted as the possible outcomes of the system.

RQ4. *how to compare two regions as a function of their uncertainty?* Using the second algorithm proposed to answer **RQ3.** the comparison of the uncertainty in two regions is straightforward, see Chapter 5. We just need to compare the values of the IE in the two regions.

RQ5. *what is the least uncertain from a set of models?* The comparison of what is the least uncertain model from a set of models can be performed in the same way that we compare the uncertainty in two regions, by means of the IE.

7.2 Open Problems and Future Work

Some of the future directions we are interested in pursuing were mentioned above. For example, in Section 6.1.2.2, there is a region where the *GLD* does not fit well the dataset. If we want to provide a general purpose computational approach for *forward propagation* we need to further investigate this issue.

The use of Information Entropy to quantify the uncertainty is very powerful. However, when applied on clusters of PDFs, such as the GLD, it observes the information variation as a function of the PDF definition, in the case of GLD this is given by its for λ parameters. In this context, a complete region modeled by a single GLD function would have a very low information entropy value. This, however would not express the uncertainty modeled by the GLD function, which could be very high. The outcome of the information entropy evaluation must be interpreted by the user.

The methods to fit the *GLD* to data are out of the scope of this thesis, as we are interested in demonstrating its usability in UQ. Nevertheless it is important to remark that,to fit the *GLD* to data is computationally intensive but suitable to parallelization. We have not found any work in the literature that explores the possibility of improving the performance of the fitting process by means of parallelization. This is an open problem we are interested in exploring in the future.

Acknowledgments

This work has been funded by CNPq, CAPES, FAPERJ, Inria (SciDISC project) and the European Commission (HPC4E H2020 project) and performed (for E. Pacitti and P. Valduriez) in the context of the Computational Biology Institute (www.ibc-montpellier.fr) and for (F. Porto, H. Lustosa and N. Lemus) in the context of the DEXL Laboratory (dexl.lncc.br) at LNCC.

Bibliography

ALLEN, M. R. et al. Quantifying the uncertainty in forecasts of anthropogenic climate change. *Nature*, Nature Publishing Group, v. 407, n. 6804, p. 617, 2000. Citado na página 22.

ALVIN, K. F. et al. Uncertainty quantification in computational structural dynamics: a new paradigm for model validation. *Society for Experimental Mechanics, Inc, 16 th International Modal Analysis Conference.*, v. 2, p. 1191–1198, 1998. Citado na página 23.

BARONI, G.; TARANTOLA, S. A General Probabilistic Framework for uncertainty and global sensitivity analysis of deterministic models: A hydrological case study. *Environmental Modelling and Software*, Elsevier Ltd, v. 51, p. 26–34, 2014. ISSN 13648152. Disponível em: <http://dx.doi.org/10.1016/j.envsoft.2013.09.022>. Citado 2 vezes nas páginas 18 and 22.

BAXTER, M. J.; COOL, H. E. M. Reinventing the wheel? Modelling temporal uncertainty with applications to brooch distributions in Roman Britain. *Journal of Archaeological Science*, Elsevier Ltd, v. 66, p. 120–127, 2016. ISSN 10959238. Citado na página 32.

CHALABI, Y.; DIETHELM, W.; SCOTT, D. J. Flexible Distribution Modeling with the Generalized Lambda Distribution. 2012. Disponível em: <https://pdfs.semanticscholar.org/6b34/5bfa8ca3e73fadc11359155c2c5f33e63a7b.pdf>. Citado na página 39.

CHEN, J.; FLOOD, M. D.; SOWERS, R. B. Measuring the Unmeasurable: An Application of Uncertainty Quantification to Financial Portfolios Measuring the Unmeasurable An application of uncertainty quantification to financial portfolios. *Quantitative Finance*, v. 7688, n. January, p. 1–18, 2008. ISSN 14697696. Disponível em: <http://dx.doi.org/10.1080/14697688.2017.1296176>. Citado 2 vezes nas páginas 18 and 22.

CORLU, C. G.; METERELLIYOZ, M. Estimating the Parameters of the Generalized Lambda Distribution: Which Method Performs Best? *Communications in Statistics: Simulation and Computation*, v. 45, n. 7, p. 2276–2296, 2016. ISSN 15324141. Citado 2 vezes nas páginas 38 and 43.

COX, M. et al. Numerical aspects in the evaluation of measurement uncertainty. *IFIP Advances in Information and Communication Technology*, v. 377 AICT, p. 180–192, 2012. ISSN 18684238. Citado 3 vezes nas páginas 32, 34, and 46.

CRESPO, L. G.; KENNY, S. P.; GIESY, D. P. The NASA Langley Multidisciplinary Uncertainty Quantification Challenge. *16th AIAA Non-Deterministic Approaches Conference*, n. January, p. 1–9, 2014. Disponível em: <http://arc.aiaa.org/doi/abs/10.2514/6.2014-1347>. Citado na página 18.

ESTACIO-HIROMS, K. C.; PRUDENCIO, E. E. User's Manual: Quantification of Uncertainty for Estimation, Simulation, and Optimization (QUESO). 2012. Citado 2 vezes nas páginas 18 and 32.

FARRELL, K.; ODEN, J. T.; FAGHIHI, D. A Bayesian framework for adaptive selection, calibration, and validation of coarse-grained models of atomistic systems. *Journal of Computational Physics*, Elsevier Inc., v. 295, p. 189–208, 2015. ISSN 10902716. Disponível em: <http://dx.doi.org/10.1016/j.jcp.2015.03.071>. Citado na página 32.

FARRELL, K. A. Selection , Calibration , and Validation of Coarse-Grained Models of Atomistic Systems. 2015. Citado na página 18.

FOURNIER, B. et al. Estimating the parameters of a generalized lambda distribution. *Computational Statistics and Data Analysis*, v. 51, n. 6, p. 2813–2835, 2007. ISSN 01679473. Citado na página 42.

FREIMER, M.; LIN, C. T.; MUDHOLKAR, G. S. A Study Of The Generalized Tukey Lambda Family. *Communications in Statistics - Theory and Methods*, v. 17, n. 10, p. 3547–3567, 1988. ISSN 1532415X. Citado 2 vezes nas páginas 38 and 40.

Gharib Shirangi, M. History matching production data and uncertainty assessment with an efficient TSVD parameterization algorithm. *Journal of Petroleum Science and Engineering*, v. 113, p. 54–71, 2014. ISSN 09204105. Citado na página 31.

GRALER, B.; PEBESMA, E.; HEUVELINK, G. Spatio-Temporal Interpolation using gstat. *Wp*, v. 8, p. 1–20, 2016. ISSN 20734859. Citado na página 74.

GUERRA, G. M. et al. Uncertainty quantification in numerical simulation of particle-laden flows. *Computational Geosciences*, v. 20, n. 1, p. 265–281, 2016. ISSN 1420-0597. Disponível em: <http://link.springer.com/10.1007/s10596-016-9563-6>. Citado 2 vezes nas páginas 18 and 22.

HAUSSER, J.; STRIMMER, K. Entropy inference and the James-Stein estimator, with application to nonlinear gene association networks. n. October 2008, p. 1–18, 2008. ISSN <null>. Disponível em: <http://arxiv.org/abs/0811.3579>. Citado na página 92.

HELTON, J. Conceptual and computational basis for the quantification of margins and uncertainty. n. June, 2009. Disponível em: <http://www.osti.gov/energycitations/product.biblio.jsp?osti{_}id=958>. Citado 2 vezes nas páginas 24 and 25.

HELTON, J. C. et al. Representation of analysis results involving aleatory and epistemic uncertainty. *International Journal of General Systems*, v. 39, n. 6, p. 605–646, 2010. ISSN 0308-1079. Citado na página 26.

HIGDON, D. *Handbook of Uncertainty Quantification*. [s.n.], 2017. ISBN 978-3-319-12384-4. Disponível em: <http://link.springer.com/10.1007/978-3-319-12385-1>. Citado na página 26.

JCGM, J. C. F. G. I. M. Evaluation of measurement data — Guide to the expression of uncertainty in measurement. *International Organization for Standardization Geneva ISBN*, v. 50, n. September, p. 134, 2008. ISSN 00099147. Disponível em: <http://www.bipm.org/en/publications/guides/gum.html>. Citado na página 46.

JIANG, B. et al. CLUSTERING UNCERTAIN DATA BASED ON PROBABILITY DISTRIBUTION SIMILARITY. Disponível em: <https://pdfs.semanticscholar.org/e172/2c8911b7db1a3114fbd38b3ea5a9e93d1290.pdf>. Citado na página 67.

JIANG, B. et al. Clustering Uncertain Data Based on Probability Distribution Similarity. *IEEE Transactions on Knowledge and Data Engineering*, p. 1–14, 2011. ISSN 1041-4347. Disponível em: <https://pdfs.semanticscholar.org/e172/2c8911b7db1a3114fbd38b3ea5a9e93d1290.pdf>. Citado 3 vezes nas páginas 49, 50, and 54.

JIANG, B. et al. Clustering Uncertain Data Based on Probability Distribution Similarity. *IEEE Transactions on Knowledge and Data Engineering*, v. 25, n. 4, p. 751–763, apr 2013. ISSN 1041-4347. Disponível em: <http://ieeexplore.ieee.org/document/6051435/>. Citado na página 50.

JOHNSTONE, R. H. et al. Uncertainty and variability in models of the cardiac action potential: Can we build trustworthy models? *Journal of Molecular and Cellular Cardiology*, The Authors, v. 96, p. 49–62, 2016. ISSN 10958584. Disponível em: <http://dx.doi.org/10.1016/j.yjmcc.2015.11.018>. Citado na página 18.

JOINER, B. L.; ROSENBLATT, J. R. Some Properties of the Range in Samples from Tukey's Symmetric Lambda Distributions. *Journal of the American Statistical Association*, v. 66, n. 334, p. 394–399, jun 1971. ISSN 0162-1459. Disponível em: <http://www.tandfonline.com/doi/abs/10.1080/01621459.1971.10482275>. Citado na página 38.

Josep de la Puente, A. C. *Website deploying a suite of geophysical tests for wave propagation problems on extreme scale machines*. 2015. 1–9 p. Citado na página 82.

K. Sawicka, G. H.; SOIL. spup- an R package for uncertainty propagation in spatial environmental modelling. *International symposium on "Spatial Accuracy Assessment in Natural Resources and Environmental Sciences"*, v. 53, n. 9, p. 1689–1699, 2016. ISSN 1098-6596. Disponível em: <http://spatial-accuracy.org/Accuracy2016>. Citado na página 93.

KARIAN, Z. A.; DUDEWICZ, E. J. *Handbook of fitting statistical distributions with R*. [S.l.: s.n.], 2011. ISSN 1098-6596. ISBN 9788578110796. Citado 11 vezes nas páginas 12, 19, 33, 35, 37, 40, 42, 43, 46, 86, and 101.

KARVANEN, J.; NUUTINEN, A. Characterizing the generalized lambda distribution by L-moments. *Computational Statistics and Data Analysis*, v. 52, n. 4, p. 1971–1983, 2008. ISSN 01679473. Disponível em: <https://ia801001.us.archive.org/27/items/arxiv-math0701405/math0701405.pdf>. Citado na página 42.

KAWAI, S.; SHIMOYAMA, K. Kriging-model-based uncertainty quantification in computational fluid dynamics. *32nd AIAA Applied Aerodynamics Conference*, n. June, p. 1–16, 2014. Disponível em: <http://arc.aiaa.org/doi/10.2514/6.2014-2737>. Citado na página 31.

KENNEDY, M. C.; O'HAGAN, A. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Blackwell Publishers Ltd., v. 63, n. 3, p. 425–464, aug 2001. ISSN 1369-7412. Disponível em: <http://doi.wiley.com/10.1111/1467-9868.00294>. Citado na página 23.

KIUREGHIAN, A. D.; DITLEVSEN, O. Aleatory or epistemic? Does it matter? *Structural Safety*, v. 31, n. 2, p. 105–112, mar 2009. ISSN 01674730. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0167473008000556>. Citado na página 24.

LAKHANY, A.; MAUSSER, H. Estimating the Parameters of the Generalized Lambda Distribution. *ALGO RESEARCH QUARTERLY*, v. 3, n. 3, 2000. Disponível em: <https://pdfs.semanticscholar.org/0f9d/1848671969232d58cb7cf3d2d06d9c4c347e.pdf?{_}ga=2.56724463.474793085.1524894682-1121088995.1524894>. Citado 2 vezes nas páginas 35 and 42.

LAMPASI, D. A.; Di Nicola, F.; PODESTA, L. Generalized lambda distribution for the expression of measurement uncertainty. *IEEE Transactions on Instrumentation and Measurement*, v. 55, n. 4, p. 1281–1287, 2006. ISSN 00189456. Citado 4 vezes nas páginas 19, 27, 46, and 51.

LIU, C. M.; NIU, Z.; LIAO, K. T. *Mechanisms to improve clustering uncertain data with UKmeans*. North-Holland, 2018. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0169023X16303287>. Citado 2 vezes nas páginas 20 and 50.

LODZIENSIS, A. U. Generalizations of tukey-lambda distributions. 2013. Citado na página 42.

MARCONDES, D.; PEIXOTO, C.; MAIA, A. C. FITTING A HURDLE GENERALIZED LAMBDA DISTRIBUTION TO HEALTHCARE EXPENSES. *Annals of Applied Statistics*, 2017. Disponível em: <https://arxiv.org/pdf/1712.02183.pdf>. Citado 3 vezes nas páginas 12, 38, and 43.

MELOROSE, J. et al. *A PROBABILISTIC FRAMEWORK FOR UNCERTAINTY QUANTIFICATION IN LARGE-SCALE SIMULATIONS: APPLICATION IN SEISMIC IMAGING*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 2015. Citado na página 22.

MOVAHEDI, M. M.; LOTFI, M. R.; NAYYERI, M. A solution to determining the reliability of products Using Generalized Lambda Distribution. *Research Journal of Recent Sciences Res.J.Recent Sci*, v. 2, n. 10, p. 41–47, 2013. Disponível em: <http://www.isca.in/rjrs/archive/v2/i10/7.ISCA-RJRS-2013-227.pdf>. Citado na página 46.

Mustafa Inchasi, E. A. *The Generalized Lambda Distribution and Its Use in Fitting Distributions to Data*. Tese (Doutorado), 2016. Citado 2 vezes nas páginas 33 and 45.

NING, W.; GAO, Y.; DUDEWICZ, E. J. Fitting mixture distributions using generalized lambda distributions and comparison with normal mixtures. *American Journal of Mathematical and Management Sciences*, v. 28, n. 1-2, p. 81–99, 2008. ISSN 01966324. Citado na página 44.

PATT, A.; KLEIN, R. J. T.; VEGA-LEINERT, A. de la. Taking the uncertainty in climate-change vulnerability assessment seriously. *Comptes Rendus Geoscience*, Elsevier, v. 337, n. 4, p. 411–424, 2005. Citado na página 22.

RAJAN, A. et al. Benchmark Test Distributions for Expanded Uncertainty Evaluation Algorithms. *IEEE Transactions on Instrumentation and Measurement*, v. 65, n. 5, p. 1022–1034, 2016. ISSN 00189456. Citado 2 vezes nas páginas 32 and 46.

SANKARARAMAN, S. *Uncertainty Quantification and Integration*. Tese (Doutorado) — Vanderbilt University, 2012. Citado na página 31.

SU, S. Fitting Single and Mixture of Generalized Lambda Distributions to Data via Discretized and Maximum Likelihood Methods: GLDEX in R. *Journal of Statistical Software*, v. 21, n. 9, 2007. Citado 5 vezes nas páginas 40, 42, 43, 44, and 47.

SU, S. Maximum Log Likelihood Estimation using EM Algorithm and Partition Maximum Log Likelihood Estimation for Mixtures of Generalized Lambda Distributions. *Journal of Modern Applied Statistical Methods*, v. 10, n. 2, p. 599–606, 2011. ISSN 1538-9472. Disponível em: <http://digitalcommons.wayne.edu/jmasm/vol10/iss2/17>. Citado 2 vezes nas páginas 43 and 44.

SU, S. Flexible parametric quantile regression model. *Statistics and Computing*, v. 25, n. 3, p. 635–650, 2015. ISSN 15731375. Citado 2 vezes nas páginas 38 and 43.

SU, S. Fitting Flexible Parametric Regression Models with GLDreg in R. *Journal of Modern Applied Statistical Methods*, v. 15, n. 2, p. 768–787, 2016. ISSN 1538-9472. Disponível em: <http://digitalcommons.wayne.edu/jmasmhttp://digitalcommons.wayne.edu/jmasm/vol15/iss2/46>. Citado na página 43.

SULLIVAN, T. J. *Introduction to Uncertainty Quantification*. Springer, 2015. ISBN 9783319233949. Disponível em: <http://www.springer.com/series/1214>. Citado 5 vezes nas páginas 22, 25, 27, 30, and 31.

TOBERGTE, D. R.; CURTIS, S. Workshop on Quantification, Communication, and Interpretation of Uncertainty in Simulation and Data Science. *Journal of Chemical Information and Modeling*, v. 53, n. 9, p. 1689–1699, 2013. ISSN 1098-6596. Citado 4 vezes nas páginas 18, 20, 22, and 33.

U.S. Department of Energy. *Scientific Grand Challenges in National Security: The Role of Computing at the Extreme Scale*. [S.l.], 2009. 255 p. Citado 2 vezes nas páginas 25 and 26.

WELLMANN, J. F.; REGENAUER-LIEB, K. Uncertainties have a meaning: Information entropy as a quality measure for 3-D geological models. *Tectonophysics*, Elsevier B.V., v. 526-529, p. 207–216, 2012. ISSN 00401951. Disponível em: <http://dx.doi.org/10.1016/j.tecto.2011.05.001>. Citado na página 28.