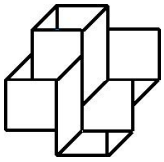

Laboratório Nacional de Computação Científica

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

Abril de 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

Abril de 2018

XXXX

Moreno Lemus, Noel

Generalized Lambda Distribution for Uncertainty Quantification of Large-scale Spatio-temporal Models / Noel Moreno Lemus. – Petrópolis, RJ - Brasil, Abril de 2018-

88 p. : il. ; 30 cm.

Orientador(es): Fábio André Machado Porto e

Thesis (D.Sc.) – Laboratório Nacional de Computação Científica
Programa de Pós-Graduação em Modelagem Computacional, Abril de 2018.

1. Uncertainty Quantification. 2. Big Data. 3. Information Entropy. I. Machado Porto, Fábio André. II. LNCC/MCTI. III. Title

CDD: XXX.XXX

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, Ph.D.

Prof. Antonio Tadeu, Ph.D.

Petrópolis, RJ - Brasil
Abril de 2018

Dedication

To my little and special family.

Acknowledgements

O autor manifesta reconhecimentos às pessoas e instituições que colaboraram para a execução de seu trabalho.

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

Abstract

Segundo a **??**, 3.1-3.2), o resumo deve ressaltar o objetivo, o método, os resultados e as conclusões do documento. A ordem e a extensão destes itens dependem do tipo de resumo (informativo ou indicativo) e do tratamento que cada item recebe no documento original. O resumo deve ser precedido da referência do documento, com exceção do resumo inserido no próprio documento. (. . .) As palavras-chave devem figurar logo abaixo do resumo, antecedidas da expressão Palavras-chave:, separadas entre si por ponto e finalizadas também por ponto.

Keywords: latex. abntex. editoração de texto.

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 qu antified 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 intesive. Moreover, due to heterogeneity the uncertainty computed in regions of the dataset is hampered by the existence of 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 simillarity of the uncertainty?, (*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 its 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

Figure 1 – Gaussian (Normal) distributions used to generate the synthetic dataset. 33
Figure 2 – Exponential distributions used to generate the synthetic dataset. 34
Figure 3 – Uniform distribution used to generate the synthetic dataset. 34
Figure 4 – Distribution of the clusters using k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*. 35
Figure 5 – Cluster 1 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I. 36
Figure 6 – Cluster 2 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I. 37
Figure 7 – Cluster 3 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I. 37
Figure 8 – Cluster 4 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I. 38
Figure 9 – Cluster 5 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I. 38
Figure 10 – Cluster 6 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I. 39
Figure 11 – Cluster 7 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I. 39
Figure 12 – Cluster 8 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I. 40
Figure 13 – Cluster 9 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I. 40
Figure 14 – Cluster 10 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I. 41
Figure 15 – Cluster 11 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I. 41
Figure 16 – Distribution of the clusters over the λ_3 and λ_4 space. 42
Figure 17 – Distribution of the clusters over the λ_3 and λ_4 space. In the top left corner: clusters 1, 2 and 3. Top right corner: clusters 4, 5 and 6. Bottom left: clusters 7, 8 and 9. Bottom right: clusters 10 and 11. 42
Figure 18 – Distribution of the clusters using k-means over the λ_3 and λ_4 values of the *GLDs*. 43
Figure 19 – Distribution of the clusters over the λ_3 and λ_4 space. 44

Figure 20 – Distribution of the clusters over the λ_3 and λ_4 space. In the top left corner: clusters 1, 2 and 3. Top right corner: clusters 4, 5 and 6. Bottom left: clusters 7, 8 and 9. Bottom right: clusters 10 and 11. 44
Figure 21 – Gamma distributions used to generate the synthetic dataset. 45
Figure 22 – Distribution of the clusters using k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*. 46
Figure 23 – Distribution of the clusters over the λ_3 and λ_4 space. 47
Figure 24 – Distribution of the clusters using k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*. 48
Figure 25 – Distribution of the clusters over the λ_3 and λ_4 space. 49
Figure 26 – 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. 53
Figure 27 – Proposed workflow. The workflow was divided in three steps, (a) the fitting process, (b) the clustering of the GLDs and, (c) the queries over the results of the clustering process. 57
Figure 28 – One slice of the $250 \times 501 \times 501$ cube. In the slice we can distinguish between the different layers. 60
Figure 29 – Histograms of the 1000 samplings generated using Monte Carlo method and the PDFs reported in Table 7. 61
Figure 30 – 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). 62
Figure 31 – The red color shows where the p-value was greater than 0.05. 62
Figure 32 – Kolmogorov-Smirnoff Distance (D). The red regions represent where the GLD fits well. 63
Figure 33 – Result of the clusterization using k-means with $n = 10$. 64
Figure 34 – 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. 64
Figure 35 – *PDFs* of 60 members of the cluster 1. 65
Figure 36 – *PDFs* of 60 members of the cluster 2. 66
Figure 37 – *PDFs* of 60 members of the cluster 3. 66
Figure 38 – *PDFs* of 60 members of the cluster 4. 67
Figure 39 – *PDFs* of 60 members of the cluster 5. 67
Figure 40 – *PDFs* of 60 members of the cluster 6. 68
Figure 41 – *PDFs* of 60 members of the cluster 7. 68
Figure 42 – *PDFs* of 60 members of the cluster 8. 69
Figure 43 – *PDFs* of 60 members of the cluster 9. 69
Figure 44 – *PDFs* of 60 members of the cluster 10. 70
Figure 45 – Distribution of the clusters. 71

Figure 46 – Analysis Regions. 72

List of Tables

Table 1 – The range of the GLD parameters and the minimum and maximum values corresponding to the labeling of the regions given in Figure . . . 29
Table 2 – Distribution of the clusters using k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*. . . . 36
Table 3 – Distribution of the clusters using k-means over the λ_3 and λ_4 values of the *GLDs*. . . . 43
Table 4 – Distribution of the clusters using k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*. . . . 46
Table 5 – Distribution of the clusters using k-means over the λ_3 and λ_4 values of the *GLDs*. . . . 48
Table 6 – Values of v_p used in the generation of a single velocity field cube. . . . 59
Table 7 – PDFs and its parameteres used to sampling the v_p, to generate n velocity models. . . . 60
Table 8 – Centers of the clusters. . . . 70
Table 9 – Distribution of the clusters. . . . 71
Table 10 – Analysis Regions. . . . 72
Table 11 – Distribution of the clusters by regions. . . . 73
Table 12 – p-values by regions. . . . 73
Table 13 – Information Entropy by regions. . . . 74
Table 14 – Layer constant properties and their depth range. “Star” layers are only used in the flat case, in substitution of their non-star equivalents . . . 75

List of abbreviations and acronyms

UQ	Uncertainty Quantification
FP	Forward Problem
QoI	Quantity of Interest
GLD	Generalized Lambda Distribution
GLDEX	r package to compute the GLD

List of symbols

Γ	Letra grega Gama
Λ	Lambda
ζ	Letra grega minúscula zeta
$\in$	Pertence

Contents

1 Introduction . . . 18
1.1 Research Objectives . . . 20
1.2 Highlights of the Dissertation . . . 21
1.3 Organization of the Dissertation . . . 21
2 Uncertainty Quantification Background . . . 22
2.1 Definitions . . . 22
2.1.1 Errors vs Uncertainties . . . 22
2.1.2 Aleatoric vs Epistemic Uncertainty . . . 22
2.2 Uncertainty Representation . . . 23
2.2.1 Interval Analysis . . . 23
2.2.2 Variance . . . 23
2.2.3 Information Entropy . . . 23
2.2.3.1 Information entropy in a spatio-temporal context . . . 24
2.2.3.2 Information entropy as a meause of uncertainty . . . 24
2.2.4 Probability Theory . . . 24
2.3 Methods for Uncertainty Propagation . . . 24
2.3.1 Sampling Methods . . . 24
2.3.1.1 Monte Carlo . . . 24
2.4 Software and Tools for UQ . . . 24
2.5 Summary . . . 25
2.6 Concepts . . . 25
2.7 Ideas a usar . . . 25
3 Parallel Computation of PDFs on Large-scale Spatio-temporal Models . . . 27
3.1 Introduction . . . 27
3.2 Architecture for Computing PDFs in Spark . . . 27
3.3 Experimental Evaluation . . . 27
4 The Generalized Lambda Distribution . . . 28
4.1 The Generalized Lambda Distribution . . . 28
4.1.1 The Ramberg and Schmeiser Parametrization . . . 28
4.1.2 The FMKL Parameterization . . . 29
4.1.3 Other Parameterizations . . . 29
4.2 GLD Shapes . . . 29
4.3 Numerical Methods to Fit the GLD to Data . . . 29
4.4 Fitting Mixture Distributions Using a Mixture of Generalized Lambda Distributions . . . 29
4.5 GLD and UQ . . . 29

4.6 The GLDEX R package . . . 30
4.7 Conclusions . . . 30
5 Clustering Uncertain Data Based on GLD Similarity . . . 31
5.1 Related Works . . . 31
5.2 Clustering Based on GLD . . . 31
5.2.1 Fit the GLD to a dataset . . . 32
5.2.2 Clustering the GLD . . . 33
5.3 Synthetic Data I . . . 33
5.3.1 Clustering using λ_2, λ_3 and λ_4 . . . 35
5.3.2 Clustering using λ_3 and λ_4 . . . 42
5.4 Synthetic Data II . . . 44
5.4.1 Clustering using λ_2, λ_3 and λ_4 . . . 45
5.4.2 Clustering using λ_3 and λ_4 . . . 47
5.5 Conclusions . . . 49
6 Kriging of the GLD parameters . . . 50
6.1 Spatio-temporal Interpolation . . . 50
6.2 Kriging over GLD . . . 50
6.3 Use Case . . . 50
6.4 Conclusions . . . 50
7 Our Approach . . . 51
7.1 Fitting a GLD to a spatio-temporal dataset . . . 51
7.1.1 Fitting process . . . 52
7.1.2 GLD validity check . . . 52
7.1.3 Quality of the fit . . . 52
7.2 Clustering the GLD based on its lambda values . . . 53
7.3 Use of GLD mixture to characterize the uncertainty in an spatio-temporal region . . . 54
7.4 Information entropy as a measure of the uncertainty in an spatio-temporal region . . . 55
7.5 Information entropy and model selection . . . 56
7.6 UQ Proposed Dataflow . . . 56
7.7 LaSST-UQ R package . . . 58
7.8 Conclusions . . . 58
8 Use Cases . . . 59
8.1 Case Study: Wave Propagation Problem . . . 59
8.1.1 The Dataset . . . 59
8.1.2 Fitting the GLD . . . 61
8.1.3 GLD validity check . . . 61
8.1.4 Quality of the fit . . . 62

8.1.5 Clustering . . . 63
8.1.6 Spatio-temporal queries . . . 71
8.1.6.1 GLD mixture . . . 72
8.1.6.2 Information Entropy . . . 74
8.1.7 Mathematical Formulation . . . 74
8.1.8 Model and Dataset Description . . . 74
8.1.9 Adding uncertainty into the model . . . 75
8.2 Case Study: Austin, queso library . . . 76
8.3 Case Study: Multidisciplinary System (NASA) . . . 76
8.4 Case Study: Spatio-temporal Nicholson-Bailey model . . . 76
9 Conclusions and Future Works . . . 77
9.1 Revisiting the Research Questions . . . 77
9.2 Significance and Limitations . . . 77
9.3 Open Problems and Future Work . . . 77
9.4 Final Considerations . . . 77

Bibliography . . . 78

Appendix 81
APPENDIX A uqms R package . . . 82
A.1 Título da seção . . . 82
APPENDIX B Ideas . . . 83
B.0.1 Variance, Information and Entropy . . . 83
B.0.2 Information Gain, Distances and Divergences . . . 83
B.1 Sensitivity Analysis . . . 83
APPENDIX C Título do apêndice C . . . 84

Annex 85
ANNEX A Título do anexo A . . . 86
A.1 Título da seção . . . 86
ANNEX B Título do anexo B . . . 87
ANNEX C Título do anexo C . . . 88

1 Introduction

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 paper 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 experimental section **??** of this paper, 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 et al.,) 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 on 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 paper 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).

In order to illustrate the use of the proposed framework, a case study is discussed. The main results obtained are: (i) the *GLD* good fits for more than the 80 % of the dataset, (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*.

The rest of the paper is organized as follows: Section **??** gives the theoretical foundations of *UQ* and highlights some interesting aspects included in our proposal. Section **??** describes the principal characteristics of the *GLD* that make it suitable for this proposal. Section **??** presents the proposed approach, the workflow we implement and some considerations of the implementation. Section **??** presents a use case and discusses the results. This use case allows us to explain our approach in the context of a real problem, which facilitates its understanding. Section **??** covers the related works and finally, section 9 concludes the paper and proposes some future works.

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 analysed?

RQ3. what is the uncertainty of an specific spatio-temporal region?

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

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

RQ1 and **RQ2** are answered in chapter 5, while **RQ3**, **RQ4** and **RQ5** are answered in chapter 7. In chapter 8 all the questions are answered again for all the use cases.

1.2 Highlights of the Dissertation

1.3 Organization of the Dissertation

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

Chapter 2 background of UQ.

Chapter 3 Ji paper.

Chapter 4 GLD.

Chapter 5 GLD clustering and kriging.

Chapter 6 Workflow.

Chapter 7 Use cases.

Chapter 8 Conclusions and future works.

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)

In this chapter, we summarize some definitions in UQ context that are important to understand the rest of the document. Also, different ways of representation of the uncertainty are discussed, with a brief justification of those we are going to use in the thesis. A general workflow of the UQ process is presented where we contextualize our contributions. A detailed discussion about forward propagation is presented. And finally, we summarize the chapter.

Hablar de todos los tipos de incertaza, de incerteza en datos, en modelos, parametrica, hasta llegar a lo que a nosotros nos interesa.

2.1 Definitions

2.1.1 Errors vs Uncertainties

2.1.2 Aleatoric vs Epistemic Uncertainty

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

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 designa- tions 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 amena- ble 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)

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 [48-55]. However, in the last several decades, additional mathematical structures for the representation of uncertainty such as evidence theory [56-63], possibility theory [64- 70], fuzzy set theory [71-75], and interval analysis [76-81] 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 [82-90]. For perspective, several comparative discussions of these different approaches to the representation of uncertainty are available [72; 91-98]

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

2.2.2 Variance

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 information 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:

$$H = -\sum_{i}^{N} p_i \log p_i \tag{2.1}$$

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:

$$H(s,t) = -\sum_{m=1}^{M} p_m(s,t) \log p_m(s,t) \tag{2.2}$$

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. 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

2.3 Methods for Uncertainty Propagation

2.3.1 Sampling Methods

2.3.1.1 Monte Carlo

Monte Carlo simulations (MCS) provide the most robust and straightforward way to solve PDEs with random coefficients. In the case of (22.2), for instance, they consist of (i) generating multiple realizations of the input parameters a and b, (ii) solving deterministic PDEs for each realization, and (iii) evaluating ensemble statistics or PDFs of these solutions. MCS do not impose limitations on statistical properties of input parameters, entail no modifications of existing deterministic solvers, and are ideal for parallel computing (HIGDON, 2017).

2.4 Software and Tools for UQ

Currently, advances in uncertainty propagation and assessment have been paralleled by a growing number of software tools for uncertainty analysis, but none has gained recognition for a universal applicability, including case studies with spatial models and spatial model inputs. (K. Sawicka; SOIL, 2016)

These include both free software, like OpenTURNS (Andrianov et al., 2007), DACOTA (Adams et al., 2009) and DUE (Brown and Heuvelink, 2007), commercial, like COSSAN (Schuëller and Pradlwarter, 2006), or free, but written for a licenced software, e.g. SAFE (Pianosi et al., 2015) or UQLab (Marelli and Sudret, 2014) toolboxes for MATLAB. A broad review of existing software packages is available in Bastin et al. (2013). To the best of our knowledge, however, none of the existent software is specifically designed to be extended by the environmental science community. The use of powerful but complex languages like C++ (e.g. Dakota), Python (e.g. OpenTURNS) or Java (e.g. DUE) often discourages relevant portions of the non-highly-IT trained scientific community from the adoption of otherwise powerful tools. spup-R package (K. Sawicka; SOIL, 2016). De aqui saque lo de arriba tambien, aunque lo de arriba lo puedo buscar en sus respectivos papers y hablar un poco de cada uno de ellos.

2.5 Summary

2.6 Concepts

high-dimensional parameter spaces computationally demanding forward models nonlinearity and/or complexity in the forward model

2.7 Ideas a usar

HPC and computational modeling play a dominant role in shaping the methodological developments and research in uncertainty qualification. Depending on the complexity of the uncertainty qualification investigation, anywhere from 10^2 to 10^8 runs of the computational model may be required. Thus, uncertainty qualification investigations may require extreme-computing environments (e.g., exascale) to obtain results in a useful time frame, even if a single run of the computational model does not require such resources.

Advances in computing over the past few decades—both in availability and power—have led to an explosion in computational models available for simulating a wide variety of complex physical (and social) systems. These complex models—which may involve millions of lines of code, and require extreme-computing resources—have led to numerous scientific discoveries and advances. This is because these models allow simulation of physical processes in environments and conditions that are difficult or even impossible to access experimentally. However, scientists' abilities to quantify uncertainties in these model-based predictions lag well behind their abilities to produce these computational models. This is largely because such simulation-based scientific investigations present a set of challenges that is not present in traditional investigations.

Until recently, the original approach of describing model parameters using single values has been retained, and consequently the majority of mathematical models in use today provide point predictions, with no associated uncertainty. (JOHNSTONE et al., 2016)

a 'typical' UQ problem involves one or more mathematical models for a process of interest, subject to some uncertainty about the correct form of, or parameter values for, those models.

Often, though not always, these uncertainties are treated probabilistically.

but how will you actually go about evaluating that expected value when it is an integral over a million-dimensional parameter space? Practical problems from engineering and the sciences can easily have models with millions or billions of inputs (degrees of freedom).

the language of probability theory is a powerful tool in describing uncertainty

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

"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 ∞erences."

UQ is not a mature field like linear algebra or single-variable complex analysis, with stately textbooks containing well-polished presentations of classical theorems bearing August names like Cauchy, Gauss and Hamilton. 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, over-arching theory of UQ.

In

Probability theorists usually denote the sample space of a probability space by Ω; PDE theorists often use the same letter to denote a domain in $\Re^n$ on which a partial differential equation is to be solved. In UQ, where the worlds of probability and PDE theory often collide, the possibility of confusion is clear. Therefore, this book will tend to use Θ for a probability space and $\mathbf{X}$ for a more general measurable space, which may happen to be the spatial domain for some PDE.

3 Parallel Computation of PDFs on Large-scale Spatio-temporal Models

3.1 Introduction

3.2 Architecture for Computing PDFs in Spark

3.3 Experimental Evaluation

4 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 tested

In this chapter a review of the principal characteristics of the *GLD*

4.1 The Generalized Lambda Distribution

4.1.1 The Ramberg and Schmeiser Parametrization

The Generalized Lambda Distribution (GLD) was defined by Ramberg and Schmeiser in 1974 by the quantil function:

$$F^{-1}(p|\lambda) = F^{-1}(p|\lambda_1, \lambda_2, \lambda_3, \lambda_4) = \lambda_1 + \frac{p^{\lambda_3} - (1-p)^{\lambda_4}}{\lambda_2} \tag{4.1}$$

where p are the probabilities, $p \in [0, 1]$, λ_1 and λ_2 are the location and scale parameteres, and λ_3 and λ_4 determine the skewness and kurtosis of the $GLD(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$.

Some restrictions in the values of $\lambda_1, \lambda_2, \lambda_3$ and λ_4 define if the *GLD* is valid. Those restrictions define 6 regions as is shown in table

The probability density function of the *GLD* at the point $x = F^{-1}(p)$ is given by:

$$f(x) = f(F^{-1}(p)) = \frac{\lambda_2}{\lambda_3 p^{\lambda_3 - 1} + \lambda_4 (1-p)^{\lambda_4 - 1}} \tag{4.2}$$

Note that the valid parameteres of λ guaranty that:

$$f(x) \geqslant 0 \tag{4.3}$$

$$\int f(x)dx = 1 \tag{4.4}$$

Table 1 – The range of the GLD parameters and the minimum and maximum values corresponding to the labeling of the regions given in Figure

Region	λ_1	λ_2	λ_3	λ_4	Minimum	Maximum
1 and 5	all	< 0	< -1	> 1	$-\infty$	$\lambda_1 + \frac{1}{\lambda_2}$
2 and 6	all	< 0	> 1	< -1	$\lambda_1 - \frac{1}{\lambda_2}$	∞
3	all	> 0	> 0	> 0	$\lambda_1 - \frac{1}{\lambda_2}$	$\lambda_1 + \frac{1}{\lambda_2}$
	all	> 0	$= 0$	> 0	λ_1	$\lambda_1 + \frac{1}{\lambda_2}$
	all	> 0	> 0	$= 0$	$\lambda_1 - \frac{1}{\lambda_2}$	λ_1
4	all	< 0	< 0	< 0	$-\infty$	∞
	all	< 0	$= 0$	< 0	λ_1	∞
	all	< 0	< 0	$= 0$	$-\infty$	λ_1

4.1.2 The FMKL Parameterization

Freimer, M., G. Mudholkar, G. Kollia and C. Lin (1988)

4.1.3 Other Parameterizations

(LODZIENSIS, 2013) (CHALABI; DIETHELM; SCOTT, 2012)

4.2 GLD Shapes

4.3 Numerical Methods to Fit the GLD to Data

(LAMPASI; Di Nicola; PODESTA, 2006) Numerical maximum log likelihood estimation for generalized lambda distributions (LAKHANY; MAUSSER, 2000) (FOURNIER et al., 2007) (MARCONDES; PEIXOTO; MAIA, 2017)

4.4 Fitting Mixture Distributions Using a Mixture of Generalized Lambda Distributions

Esto esta en (TOBERGTE; CURTIS, 2013)

Fitting the GLD and compare with the normal mixture (NING; GAO; DUDEWICZ, 2008)

4.5 GLD and UQ

A solution to determining the reliability of products Using Generalized Lambda Distribution (MOVAHEDI; LOTFI; NAYYERI, 2013)

Fundamental Reference (LAMPASI; Di Nicola; PODESTA, 2006)

4.6 The GLDEX R package

4.7 Conclusions

5 Clustering Uncertain Data Based on GLD Similarity

In chapter 4 we exposed the two most important parametrizations of the *GLD* and select the *FMKL* as the one to use 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.

The uncertainty can be characterized in many different ways as we mention in chapter 2, but from the *GLD* point of view, λ_2, λ_3 and λ_4 are the responsables of this. So, in this chapter we try to answer the research question 1, we formulate in the introduction:

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

First of all, in section 5.1 a brief review of some related works is performed. In this section, some advantage and drawbacks are highlighted, and some considerations about the possibilities of the use of the *GLD* to solve some of the drawbacks are commented. Next, in section 5.2 our hypothesis about the use of the *GLD* to clusterized uncertain data, is presented. Sections 5.3 and 5.4 present two synthetics datasets and the results of the clustering technique. Those results help us to validate our hypothesis. Finally, section 5.5 summarize and discuss the main results of the chapter.

5.1 Related Works

(JIANG et al., 2011)

5.2 Clustering Based on GLD

Our hypothesis is that, as the *GLD* shape is characterized by λ_2, λ_3 and λ_4, and this shape change slowly with the change in the λ_i values, we can group the uncertainty using clustering algorithms above λ_2, λ_3 and λ_4.

To test our hypothesis, we generate two synthetic datasets using 4 different probabiltiy density functions: Gaussian, Exponential, Uniform and Gamma. The structure of

the datasets is represented in 5.1.

$$S(x_i, < v_j >) i = 1.....n, j = 1.....m \tag{5.1}$$

where:

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

For example, the first object could be a Gaussian distribution with size = 1000, mean = 0 and standard deviation = 2, figure **??**. The datasets are described in details in sections 5.3 and 5.4.

5.2.1 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 seudo-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:     < λ_1, λ_2, λ_3, λ_4 > ← FIT.GLD.LM(< v_1, v_2, ..., v_n >)
3:     isValid_(x_i) ← VALIDITYCHECK(< λ_3, λ_4 >)
4:     if isValid_(x_i) then
5:         [pvalue, D]_(x_i) ← KS(< λ_1, λ_2, λ_3, λ_4 >_(x_i))
6:     if pvalue_(x_i) > 0.05 then
7:         STORELAMBDAS(< λ_1, λ_2, λ_3, λ_4 >, x_i)
```

The algorithm receive a dataset represented by 5.1 and, for each position x_i, call a function ***fit.gld.lm*** from the **R** package **GLDEX** presented in section 4.6, line 2 of the algorithm 1. In line 3 we check the validity of the *GLD* returned by the function (remember from chapter 4 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:

$$S(x_i, < \lambda_1, \lambda_2, \lambda_3, \lambda_4 >) i = 1.....n \tag{5.2}$$

5.2.2 Clustering the GLD

The clustering algorithm is trivial because the idea we try to test is that we can clusterize the uncertain data, using a simple k-means with a Euclidean distance over the λ_i space.

The dataset 5.2 is modified to remove λ_1. Then, a k-means algorithm is used first over $< \lambda_2, \lambda_3, \lambda_4 >$ and second over $< \lambda_3, \lambda_4 >$. The results are discused in sections 5.3 and 5.4.

5.3 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 1, 2 and 3. 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.

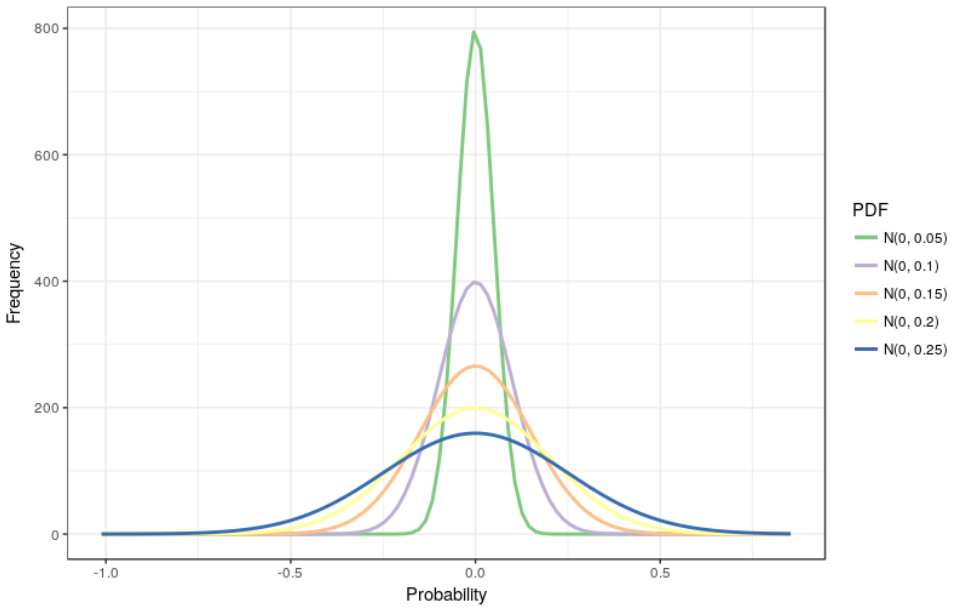

Figure 1 – Gaussian (Normal) distributions used to generate the synthetic dataset.

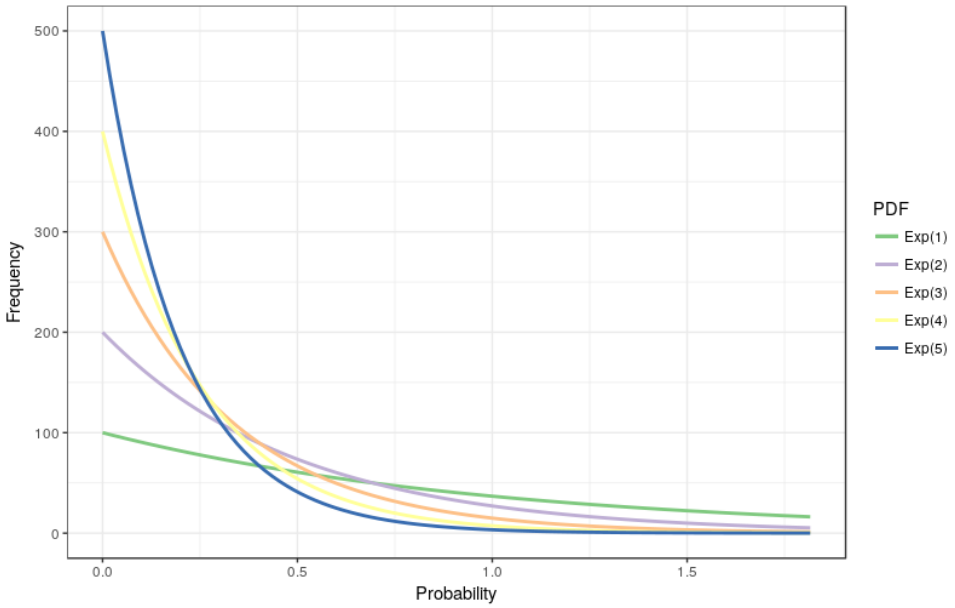

Figure 2 – Exponential distributions used to generate the synthetic dataset.

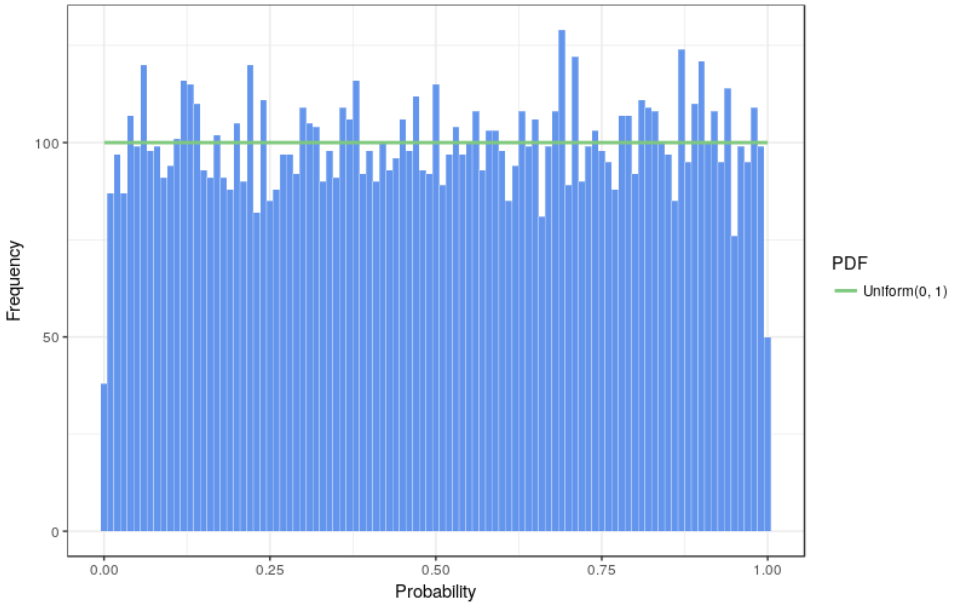

Figure 3 – Uniform distribution used to generate the synthetic dataset.

This dataset could be represented as a multidimensional array where for each position x_i, we have 1000 values v_j, equation 5.3. In this case i and j vary from 1 to 1000 casually.

$$S(x_i, < v_j >)i, j = 1, 2.....1000 \tag{5.3}$$

The fitting algorithm proposed in subsection 5.2.1 is applied over 5.3. The good-of-fit test return that all the *GLDs* are good fit for its corresponding distribution. As a result

the dataset 5.4 is genarated.

$$S(x_i, < \lambda_1, \lambda_2, \lambda_3, \lambda_4 >) i = 1.....1000 \tag{5.4}$$

5.3.1 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 k-means with euclidean distance 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 k-mean algorithm will return 11 clusters as well (one for each distribution). Then 11 is the number we use with the k-means algorithm.

In figure 4 and table 2 the distribution of the clusters returned by the k-means is shown.

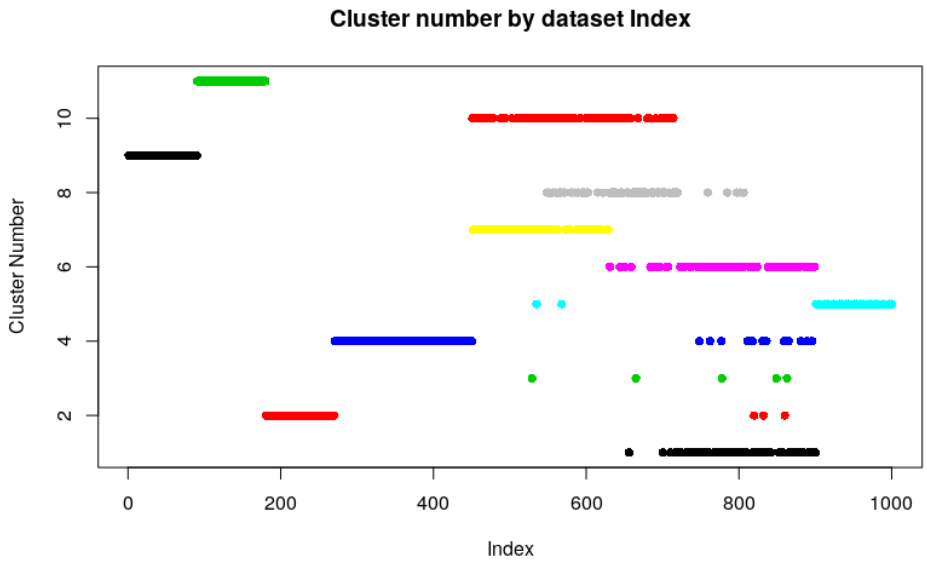

Figure 4 – Distribution of the clusters using k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*.

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 23 false positives, three in cluster 2, 18 in cluster 4 and two in cluster 5. The second observation is that, the normal distributions were grouped in 4 clusters (2, 4, 9 and 11), cluster 2 group perfectly it 90 elements with 2 false positives, clusters 9 and 11 group exactly its 90 elements each. The cluster 4 group the last 180 elements of the Normal distribution, with 18 false positives

The Uniform distribution was grouped totally in cluster 5, with two false positives as was mention above. The last observation is that the algorithm can't separate the 5

Table 2 – 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	82
2	Normal	93
3	Exponential	5
4	Normal	198
5	Uniform	102
6	Exponential	83
7	Exponential	91
8	Exponential	82
9	Normal	90
10	Exponential	84
11	Normal	90

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

In figures between 5 and 15 we show the *PDFs* of all the distributions that belongs to the same cluster. If we take a look at figures 5, 6, 7, 13, 14 and 15 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.

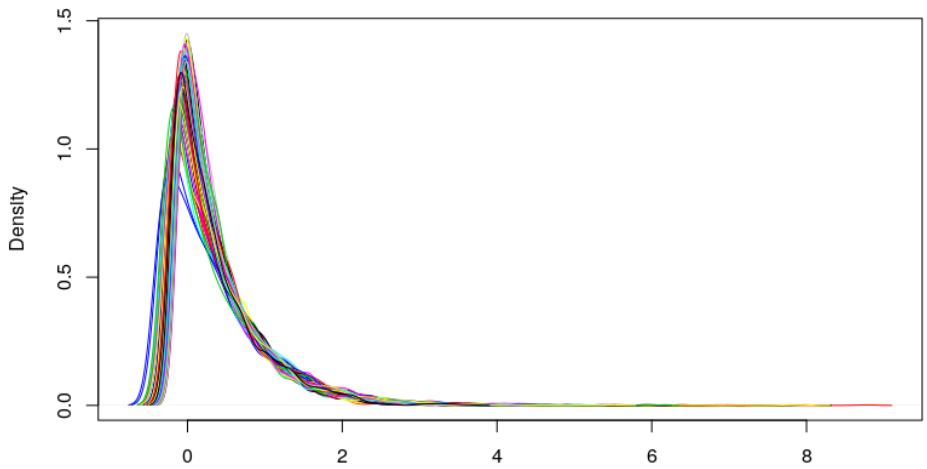

Figure 5 – Cluster 1 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I.

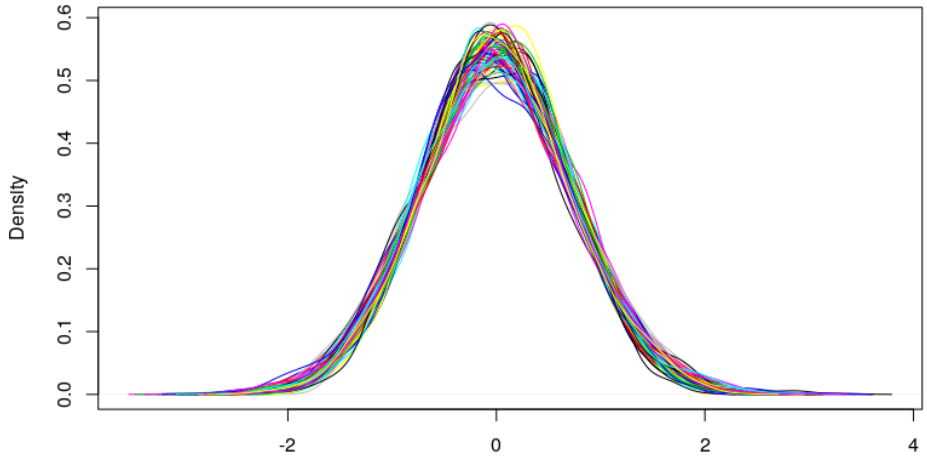

Figure 6 – Cluster 2 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I.

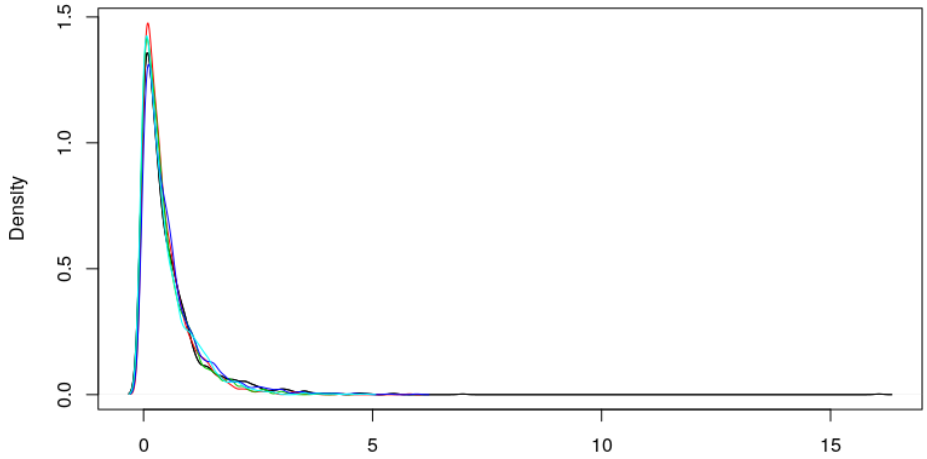

Figure 7 – Cluster 3 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I.

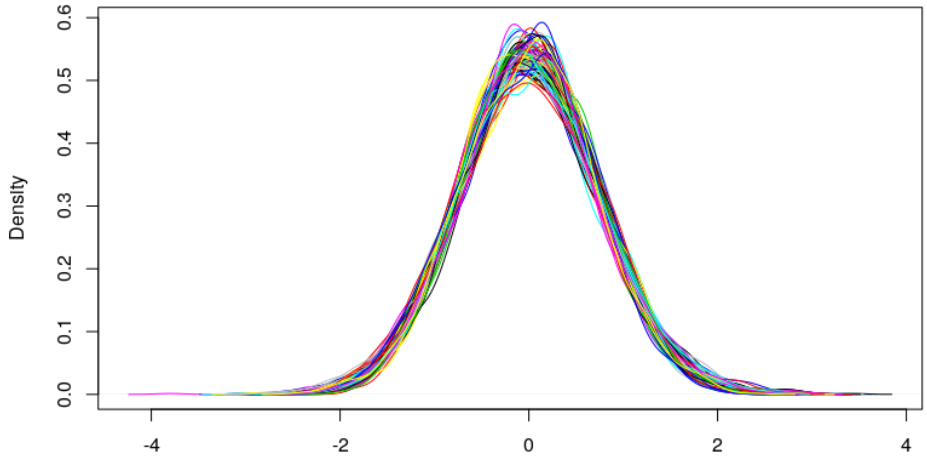

Figure 8 – Cluster 4 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I.

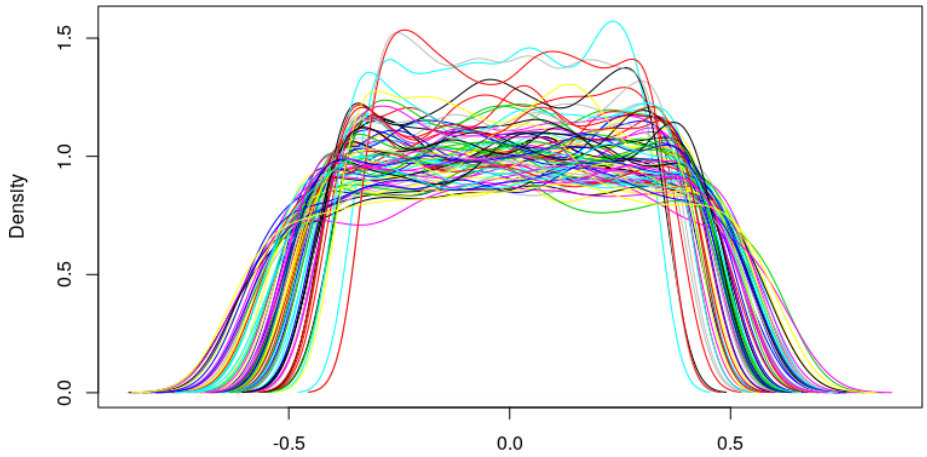

Figure 9 – Cluster 5 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I.

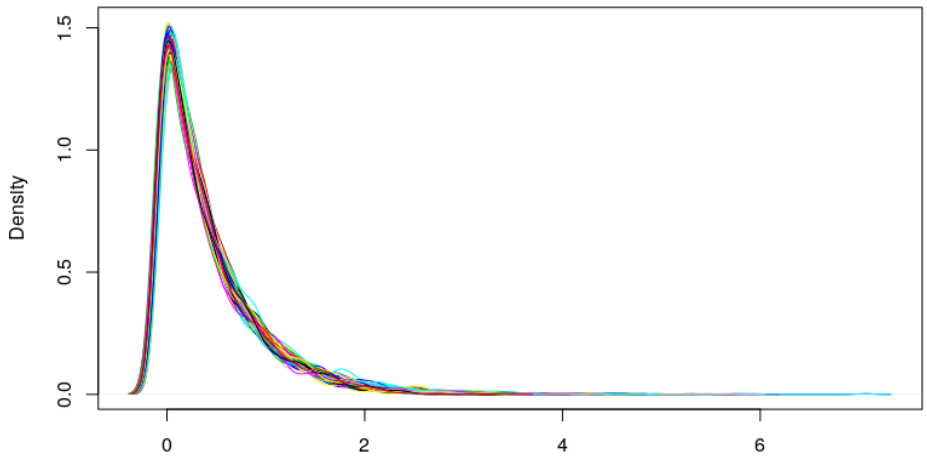

Figure 10 – Cluster 6 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I.

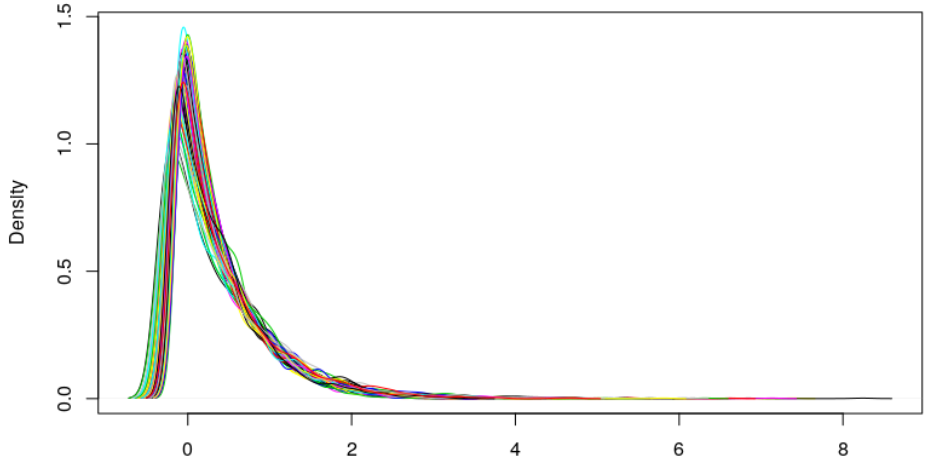

Figure 11 – Cluster 7 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I.

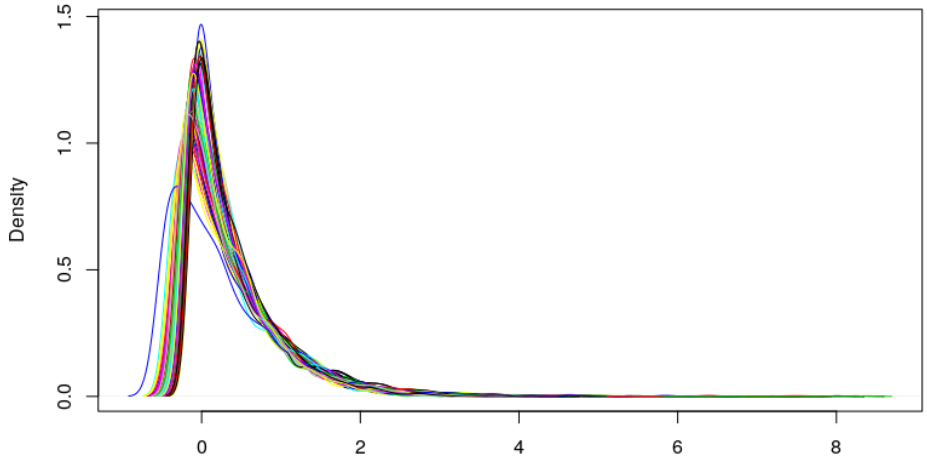

Figure 12 – Cluster 8 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I.

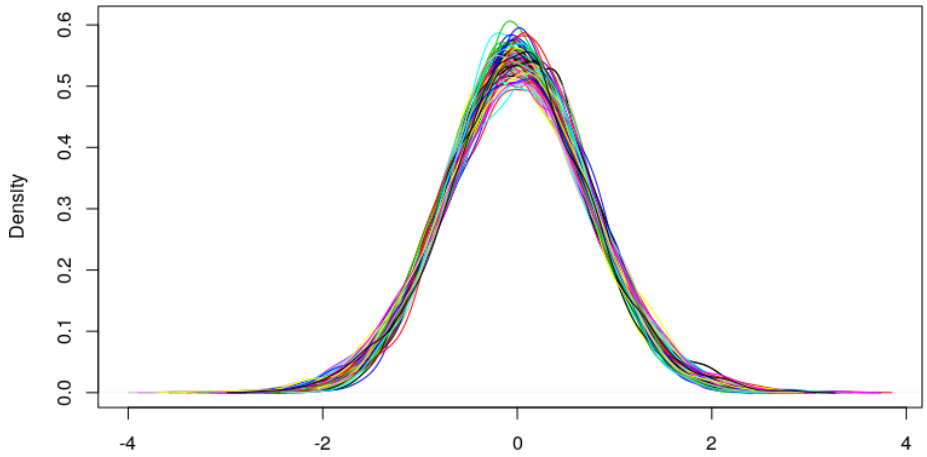

Figure 13 – Cluster 9 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I.

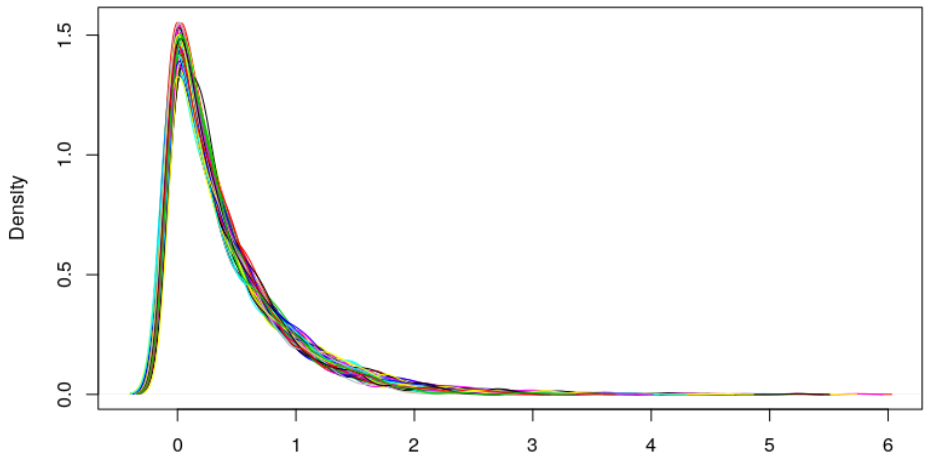

Figure 14 – Cluster 10 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I.

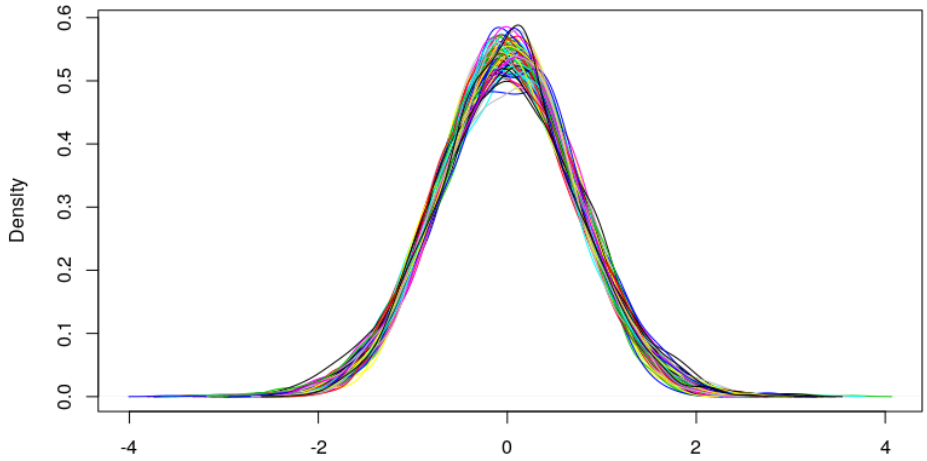

Figure 15 – Cluster 11 returned by the k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*, synthetic dataset I.

Another interesting result is show in figures 16 and 17. As we can see, clusters 2, 4, 9 and 11 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 5, 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.

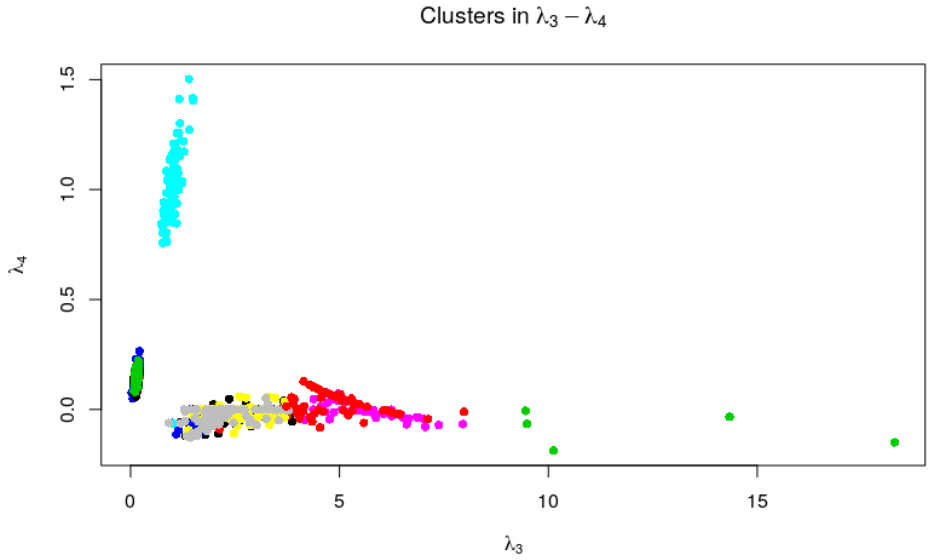

Figure 16 – Distribution of the clusters over the λ_3 and λ_4 space.

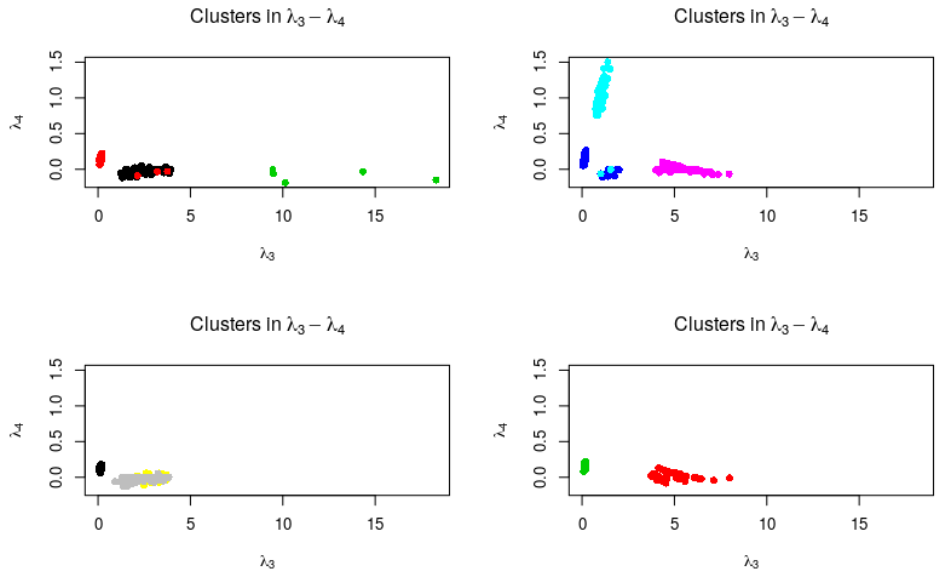

Figure 17 – Distribution of the clusters over the λ_3 and λ_4 space. In the top left corner: clusters 1, 2 and 3. Top right corner: clusters 4, 5 and 6. Bottom left: clusters 7, 8 and 9. Bottom right: clusters 10 and 11.

5.3.2 Clustering using λ_3 and λ_4

In this section we proceed similar to section 5.3.1, but the k-means algorithm run over λ_3 and λ_4. The distribution of the clusters is shown in figure 18 and table 3.

Table 3 – 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	Normal	197
2	Exponential	118
3	Exponential	110
4	Uniform	35
5	Exponential	41
6	Uniform	65
7	Exponential	2
8	Normal	131
9	Exponential	74
10	Exponential	105
11	Normal	122

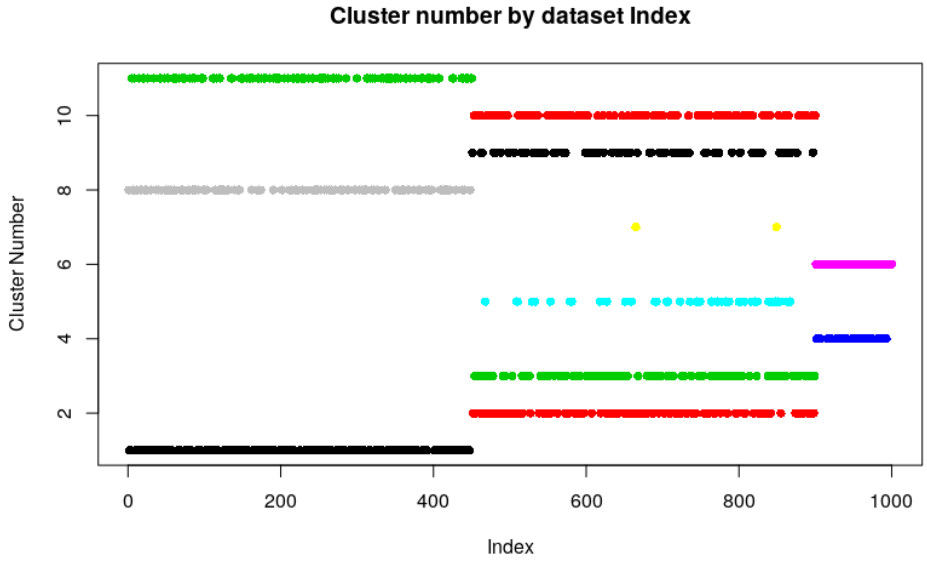

Figure 18 – Distribution of the clusters using k-means over the λ_3 and λ_4 values of the *GLDs*.

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 type of distribution. As we see in figure 18 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 k-means.

Clusters 1, 8 and 11 group all the Normal distributions, clusters 4 and 6 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 5.3.1, figures 19 and 20. Again the distributions are concentrated near the same (λ_3, λ_4)

values.

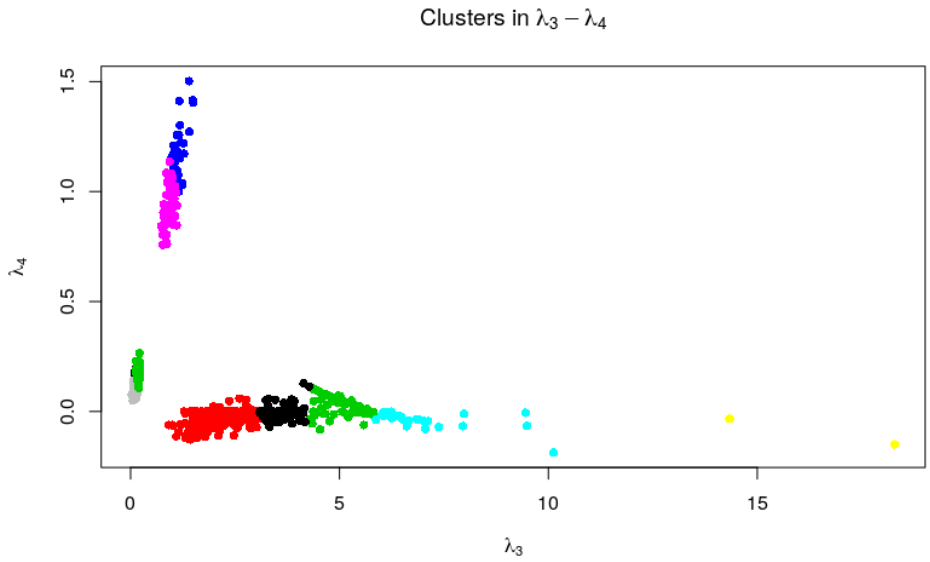

Figure 19 – Distribution of the clusters over the λ_3 and λ_4 space.

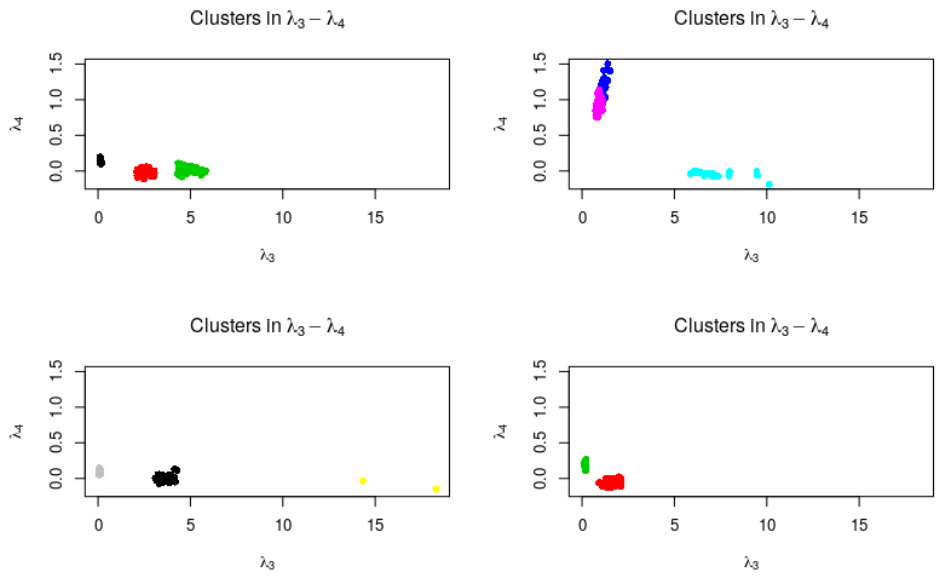

Figure 20 – Distribution of the clusters over the λ_3 and λ_4 space. In the top left corner: clusters 1, 2 and 3. Top right corner: clusters 4, 5 and 6. Bottom left: clusters 7, 8 and 9. Bottom right: clusters 10 and 11.

5.4 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 21. 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.

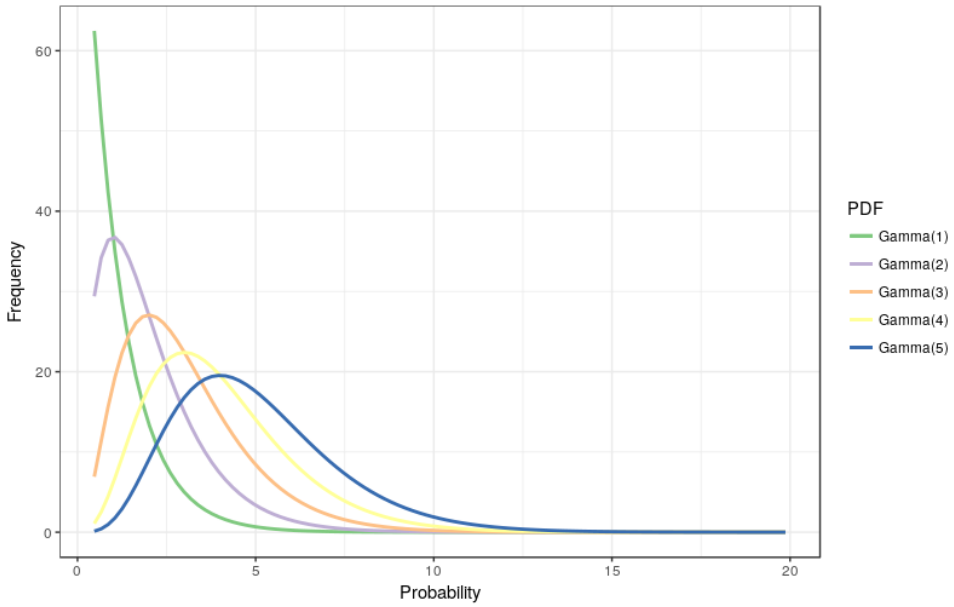

Figure 21 – Gamma distributions used to generate the synthetic dataset.

Similar to the dataset I, the fitting algorithm proposed in subsection 5.2.1 is applied over dataset II. The good-of-fit test return that all the *GLDs* are good fit for its corresponding distribution.

5.4.1 Clustering using λ_2, λ_3 and λ_4

The distribution of the clusters returned by the k-means algorithm is shown in figure 22 and table 4.

Table 4 – 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	90
2	Exponential	44
3	Uniform	45
4	Normal	179
5	Gamma	60
6	Uniform	55
7	Exponential	87
8	Exponential	58
9	Exponential	67
10	Exponential	74
11	Exponential	25
12	Exponential	90
13	Normal	96
14	Normal	90
15	Gamma	30
16	Gamma	360

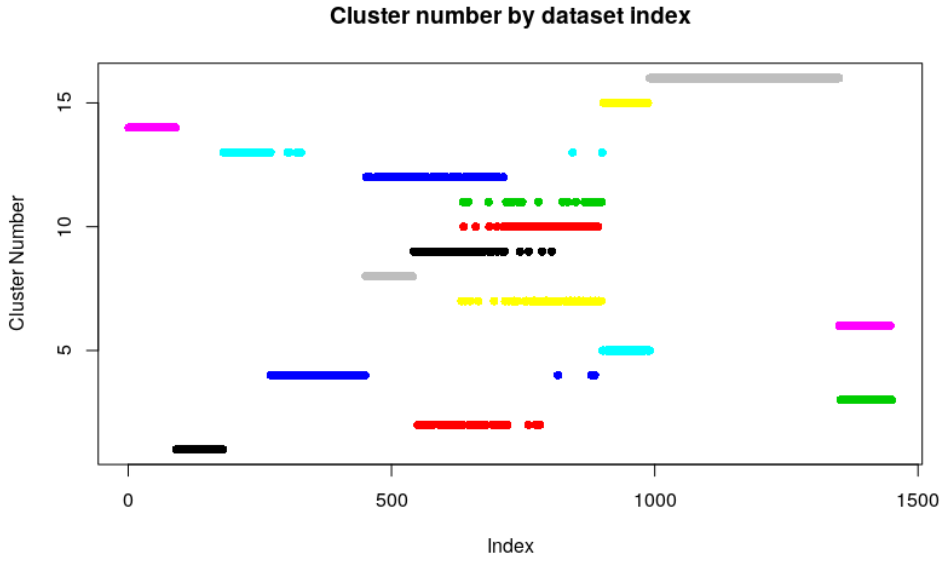

Figure 22 – Distribution of the clusters using k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*.

In general the results are very similar to the results of the section 5.3, but we get less false positives, 5 in total. 3 false positives in cluster 4 and 2 false positves in cluster 13. The normal distribution was groping again in for clusters: 1, 4, 13 and 14. The uniform distribution was grouping in clusters 3 and 6 without false positives. The gamma distribution introduced here was grouped in clusters 5, 15 and 16, 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 23. The two clusters of the uniform distribution are 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.

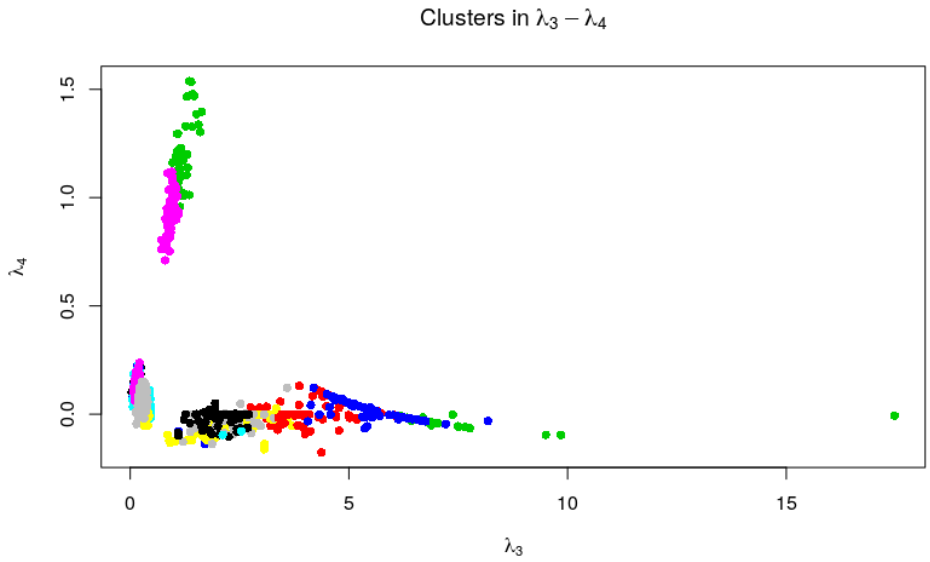

Figure 23 – Distribution of the clusters over the λ_3 and λ_4 space.

5.4.2 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 5.

A few false positives are observed in clusters 5, 6 and 12, but nothing to worry about. Again the regions of the four distribution families are perfectly separated by the algorithm.

Table 5 – 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	64
2	Exponential	126
3	Exponential	1
4	Exponential	57
5	Gamma	83
6	Normal	139
7	Uniform	67
8	Gamma	148
9	Gamma	108
10	Exponential	75
11	Exponential	80
12	Normal	112
13	Exponential	44
14	Normal	201
15	Gamma	112
16	Uniform	33

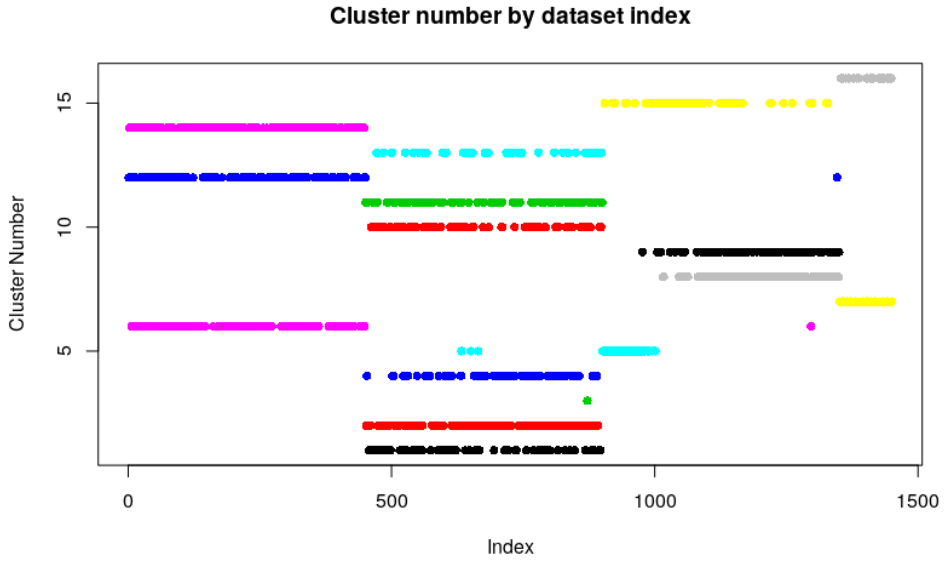

Figure 24 – Distribution of the clusters using k-means over the λ_2, λ_3 and λ_4 values of the *GLDs*.

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

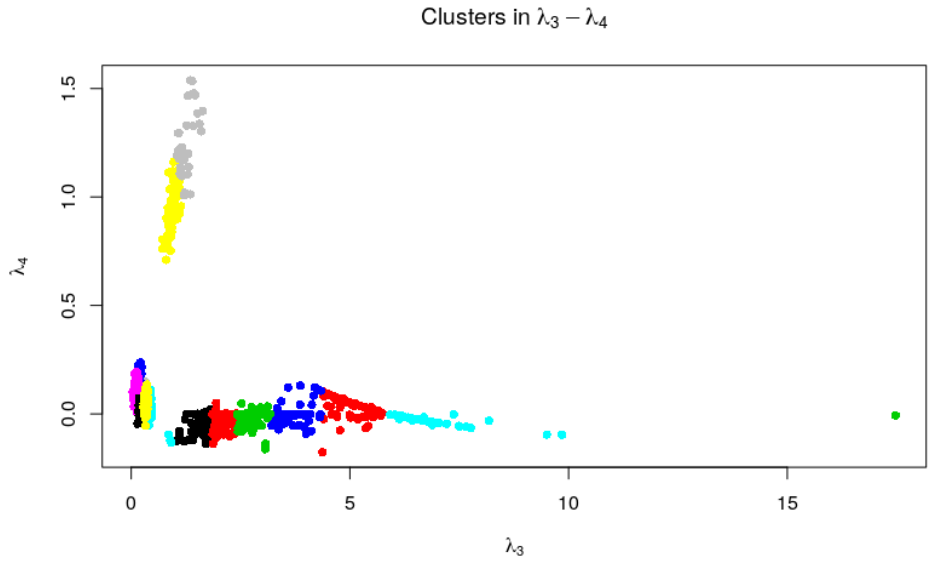

Figure 25 – Distribution of the clusters over the λ_3 and λ_4 space.

5.5 Conclusions

In this chapter, we explore clustering uncertain data based on the similarity between their distributions. The idea is to answer the research question 1 *"how to group the output of the UQ process based on the similarity of the uncertainty?"*. Different to the approaches reported in the literature, we propose the use of a k-means algorithm with euclidean distance over the λ_2, λ_3 and λ_4 space of the *GLD*.

The approach was tested over two synthetic datasets and the results of the test were exactly what we expect.

6 Kriging of the GLD parameters

6.1 Spatio-temporal Interpolation

6.2 Kriging over GLD

6.3 Use Case

6.4 Conclusions

7 Our Approach

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:

$$\mathbf{Q} = (\mathbf{q}(s_1, t_1), \mathbf{q}(s_2, t_2), ..., \mathbf{q}(s_n, t_n)) \tag{7.1}$$

where:

- $(s_1, t_1), (s_2, t_2), ..., (s_n, t_n) \in \mathcal{S} \times \mathcal{T} \subseteq \mathbb{R}^3 \times \mathbb{R}$ 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)

We could have many *QoI*, but for simplicity here we are going to considere just one.

In the presence of a stochastic problem, on each spatio-temporal location (s_i, t_j) we have many realizations of $q(s_i, t_j)$. A structure of a database to store this information can be modeled as:

$$S(s_i, t_j, simId, q(s_i, t_j)) \tag{7.2}$$

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

The first step of our approach consists of trying to find the *GLD* that best fits our simulations on each spatio-temporal location. The algorithms are described in the next section.

7.1 Fitting a GLD to a spatio-temporal dataset

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. In our approach we divide this step in three task:

- 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 the algorithm 2. Before starting the fitting process, we need to group all the simulations that correspond to the same spatio-temporal location (s_i, t_j). As a result we get a new dataset $S^*(s_i, t_j, < q_1, q_2, .., q_n >)$, where $q_i, 1 \leq i \leq n$, represents a vector of all the values of q at point (s_i, t_j).

7.1.1 Fitting process

Now, for each spatio-temporal location $(s_i, t_j) \in \mathcal{S} \times \mathcal{T}$ we use a function of the GLDEX R package described in section **??**, to fit the *GLD* to a vector $< q_1, q_2, .., q_n >$, line 2 of algorithm 2. As a result of this task we get the lambda values of the *GLD* that best fit the dataset at each spatio-temporal location, equation 7.3.

$$S'(s_i, t_j, GLD(\lambda_1, \lambda_2, \lambda_3, \lambda_4)) \tag{7.3}$$

7.1.2 GLD validity check

As we mention in section **??** the *GLD* is not always valid, it depends of 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 **??** the *GLD* is not valid, on the other case is valid.

The validity check is performed in line 3 of the algorithm 2, and as a result we get:

$$S_{validity}(s_i, t_j, valid(s_i, t_j)), \tag{7.4}$$

where:

$$valid(s_i, t_j) = \begin{cases} 1 & \text{if GLD is valid in } (s_i, t_j) \\ 0 & \text{otherwise} \end{cases} \tag{7.5}$$

7.1.3 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 2. The distance D is the maximum distance between both cumulative density functions (CDF), as shown in Figure 26. 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 on each spatio-temporal locations.

$$S_{\mathcal{D}}(s_i, t_j, \mathcal{D}(s_i, t_j)) \tag{7.6}$$

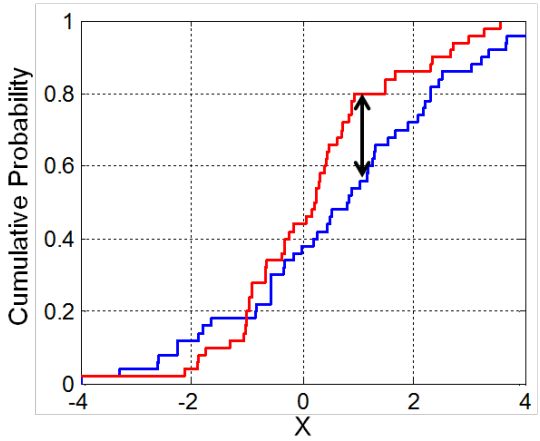

Figure 26 – Illustration of the two-sample Kolmogorov–Smirnov statistic. Red and blue lines each correspond to an empirical distribution function, and the black arrow is the two-sample KS statistic.

$$S_{p_{value}}(s_i, t_j, p_{value}(s_i, t_j)) \tag{7.7}$$

Finally, in line 7 of the algorithm 2 we store the lambda values of those *GLDs* that are valid and return p-values greater than 0.05.

Algorithm 2 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($< \lambda_1, \lambda_2, \lambda_3, \lambda_4 >, s_i, t_j$)

7.2 Clustering the GLD based on its lambda values

In section **??** we discussed the different shapes of the GLD and define the regions of the (λ_3, λ_4) space where the shapes are similar. In Figure **??**, we show how similar values of λ_3 and λ_4 lead to similar shapes. This fact suggests that one can clusterize the *GLD* based on its lambda values. The result of this clusterization are groups of *GLDs* with similar shapes (behaviors).

In addition to λ_3 and λ_4, which represent the right and left tails of the distribution, we have also to consider λ_2, as the latter represents the dispersion of the distribution.

Then, in this step of our workflow, we clusterize the *GLDs* using λ_2, λ_3 and λ_4 values. The final result of this step is:

$$S_{\mathcal{C}}(s_i, t_j, GLD_k, clusterID) \tag{7.8}$$

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

With the *GLD* clusterized, we can use this result to characterize the uncertainty in a particular spatio-temporal region, or to measure numerically the corresponding uncertainty. In subsections 7.3 and 7.4, we describe how those approaches are implemented (see Figure 27).

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

One of the main advantages of assessing the complete probability distribution of the outputs, in place of low order moments (mean and standard deviation), is that we can use the *PDFs* to answer queries. For example, suppose we want to know the mean and standard deviation in a particular spatio-temporal region $(\mathcal{S}_i \times \mathcal{T}_j)$, or we want to observe graphically the distribution of the raw data generated in the simulation process in a spatio-temporal region.

Let us consider the second query. Up to this point, we have discussed the fit of GLDs that characterize the uncertainty at each spatio-temporal locations (s_i, t_j), and the cluster to which the GLD at that particular spatio-temporal location would belong to. If we consider the clusterization of GLD to be of good quality, we can pick the GLD at the centroid of each cluster as a representative of all its members. In this context, in a particular spatio-temporal region, each cluster may be qualified with a weight given by:

$$w_k = \frac{1}{N} \sum_{i=1}^{S} \sum_{j=1}^{T} w(s_i, t_j) \tag{7.9}$$

where:

$$w(s_i, t_j) = \begin{cases} 1 & \text{if } clusterID(s_i, t_j) = k \\ 0 & \text{otherwise} \end{cases} \tag{7.10}$$

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 **??** that $w_k \geq 0$ and $\sum w_k = 1$.

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

$$f(x) = \sum_{k=1}^{K} w_k GLD(\lambda_1, \lambda_2, \lambda_3, \lambda_4) \tag{7.11}$$

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 GLD mixture process is sumarized in algorithm 3.

Algorithm 3 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** $\frac{1}{N} \sum_{c}^{C_{(\mathcal{S}_i \times \mathcal{T}_j)}} w_c * c.getGLD()$

7.4 Information entropy as a measure of the uncertainty in an spatio-temporal region

Now, what happen if we want to measure the uncertainty quantitatively? As we mention in subsubsection 2.2.3.2 the information entropy is useful in this context. The limitation we mention in that section is solved here, because we can use the different clusters we got in section 7.2 as the different outcomes of the system. The equation 2.2 can be rewriten as follow:

$$H(s,t) = -\sum_{c=1}^{C} p_c(s,t) \log p_c(s,t) \tag{7.12}$$

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

Algorithm 4 Information Entropy 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: $p_c(s,t) = \frac{w_c}{N}$
8: $H(s,t) \leftarrow -\sum_{c=1}^{C} p_c(s,t) \log p_c(s,t)$
9: **return** $H(s,t)$

Algorithm 4 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 . Using this probability we compute the information entropy $H(s,t)$, line 8, and finally we return the result in line 9.

7.5 Information entropy and model selection

7.6 UQ Proposed Dataflow

Summarizing, we proposed a workflow to quantify the uncertainty in large-scale spatio-temporal models, figure 27. The workflow is divided into three main steps, the fitting process, the clustering and the queries. We illustrate the use of the workflow with two queries, sections 7.3 and 7.4.

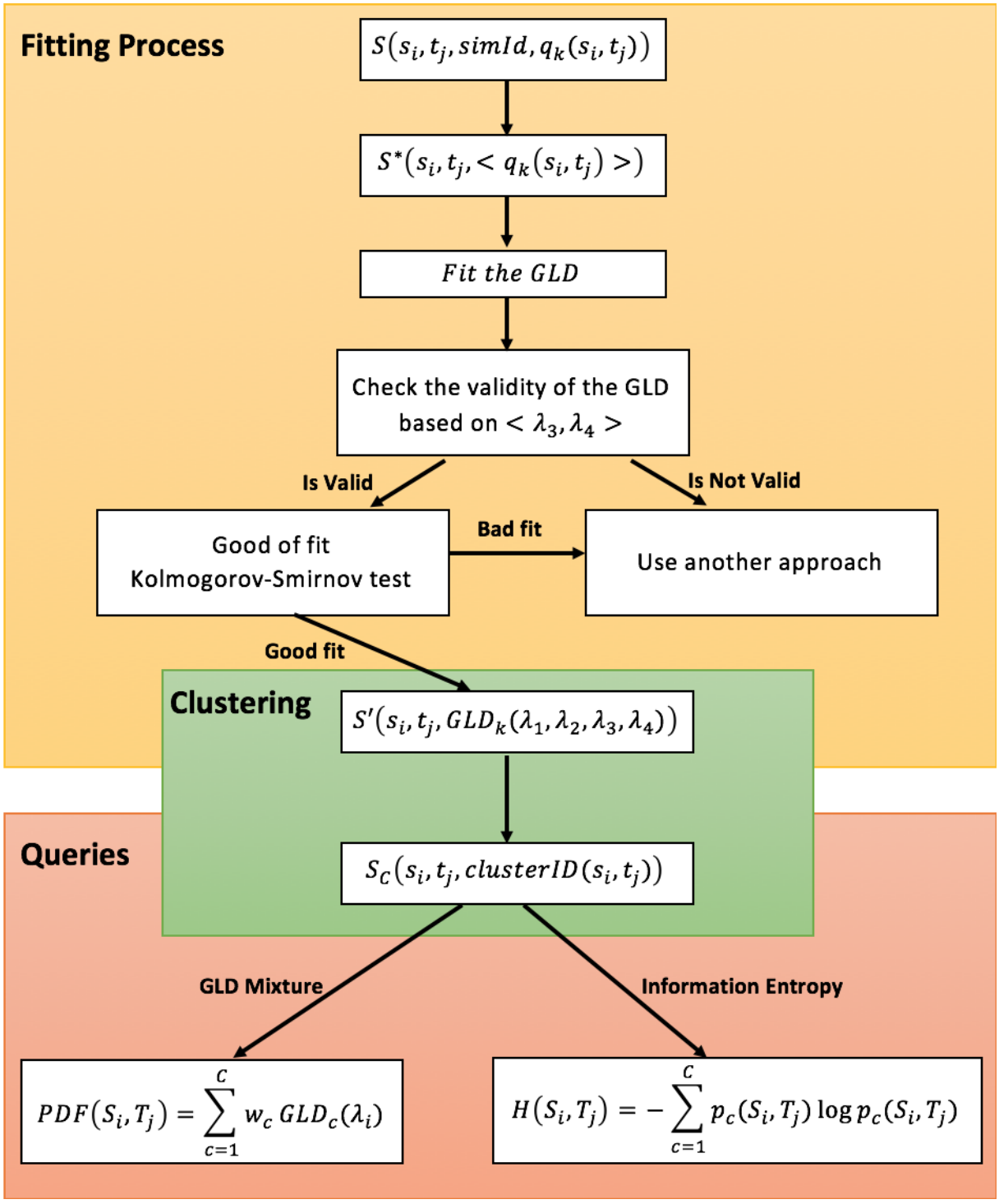

Figure 27 – Proposed workflow. The workflow was divided in three steps, (a) the fitting process, (b) the clustering of the GLDs and, (c) the queries over the results of the clustering process.

7.7 LaSST-UQ R package

7.8 Conclusions

8 Use Cases

In the present chapter we are going to test the UQMS in three different scenarios, spatial only domain, section 8.1 , spatio-temporal domain, section 8.2, and finally a multidisciplinary system, section 8.3.

8.1 Case Study: Wave Propagation Problem

8.1.1 The Dataset

In the HPC4e benchmark, the 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. To generate a single cube with dimensions $250 \times 501 \times 501$ we can use the values provided in the benchmark. For example, to generate a cube in the $v_p(m/s)$ variable we can use the fixed values of Table 6.

The first slice of this cube is shown in Figure 28.

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

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 7 – PDFs and its parameteres used to sampling the v_p, to generate n velocity models.

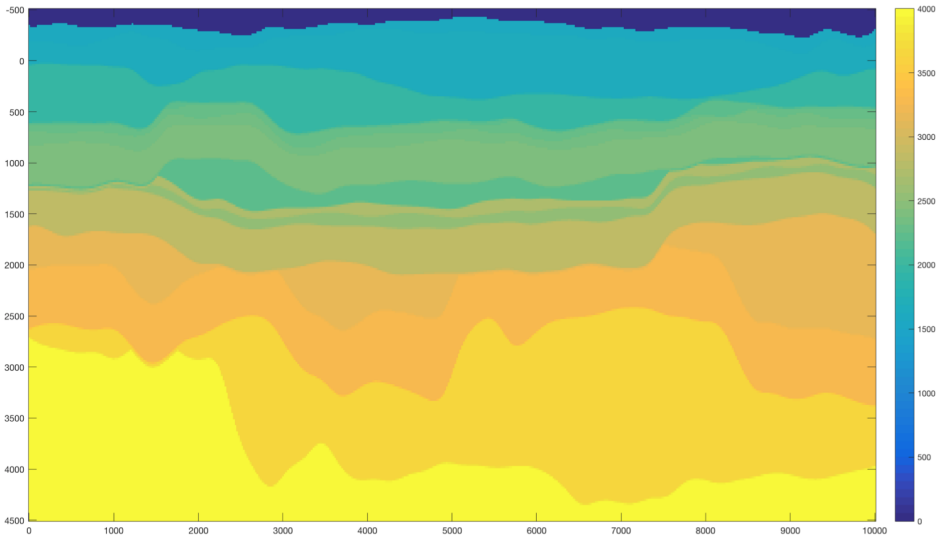

Figure 28 – One slice of the $250 \times 501 \times 501$ cube. In the slice we can distinguish between the different layers.

Now as our purpose is to study the uncertainty in the output as a result of the propagation of the input uncertainty throughout the model, we cannot use this benchmark as it is. We need the input, $v_p(m/s)$ in this case, to be uncertain. In order to achieve so, we compute $v_p(m/s)$ as a random variable with the *PDFs* shown in Table 7.

Then, using a Monte Carlo method we generate a sampling of 1000 realizations of the $v_p(m/s)$ variable, Figure 29; and using a Matlab script provided by the HPC4e

benchmark we simulate 1000 times, one for each realization, and generate 1000 cubes (230 GB) as an output. The resulting cubes are $250 \times 501 \times 501$ multi-dimensional arrays.

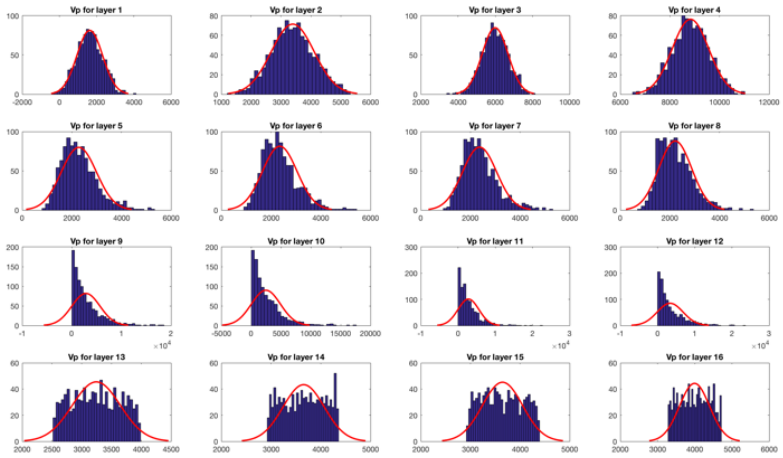

Figure 29 – Histograms of the 1000 samplings generated using Monte Carlo method and the PDFs reported in Table 7.

To simplify the computational process and visualize the results, we select the slice 200 to be used here, then we have 1000 realizations of a slice with size of 250×501. The equation 7.2 can be simplified because we have two dimentions in space and don't have time domain, then our dataset can be represented as $S(x_i, y_j, simId, v_p(x_i, y_j))$. In this new representation (x_i, y_j) are the 2D coordinates and $v_p(x_i, y_j)$ is the velocity value at point (x_i, y_j). *simId* still represents the Id of the simulation and its range here is between 1 and 1000.

Now that we have an experimental dataset we can start to apply our workflow, step by step.

8.1.2 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 7.1.1 we get as a result a new 2D array:

$$S'(x_i, y_j, GLD(\lambda_1, \lambda_2, \lambda_3, \lambda_4)) \tag{8.1}$$

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 8.1.3 and 8.1.4.

8.1.3 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.

8.1.4 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 30, 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 31.

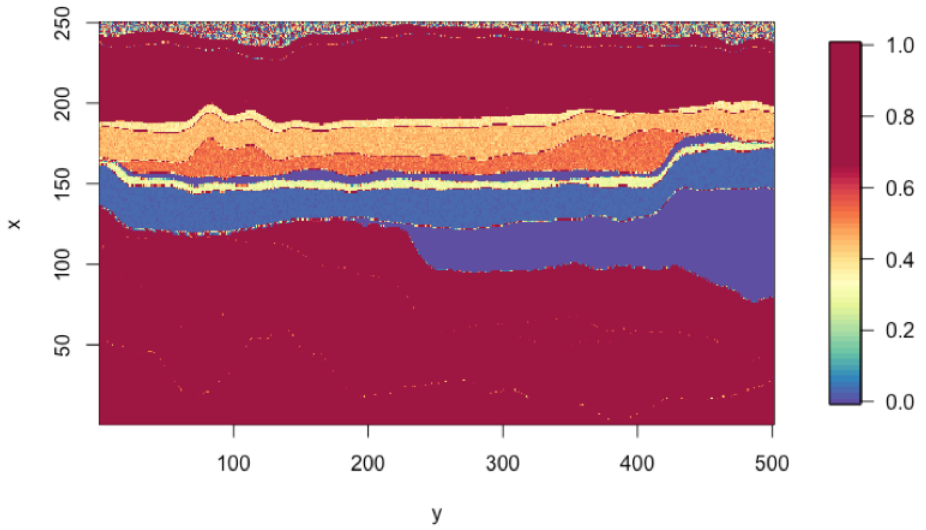

Figure 30 – Goodness of the fit based on the p-value returning by the KS-test. p-value $>$ 0.05 represent a good fit of the GLD to the dataset at (x_i, y_j).

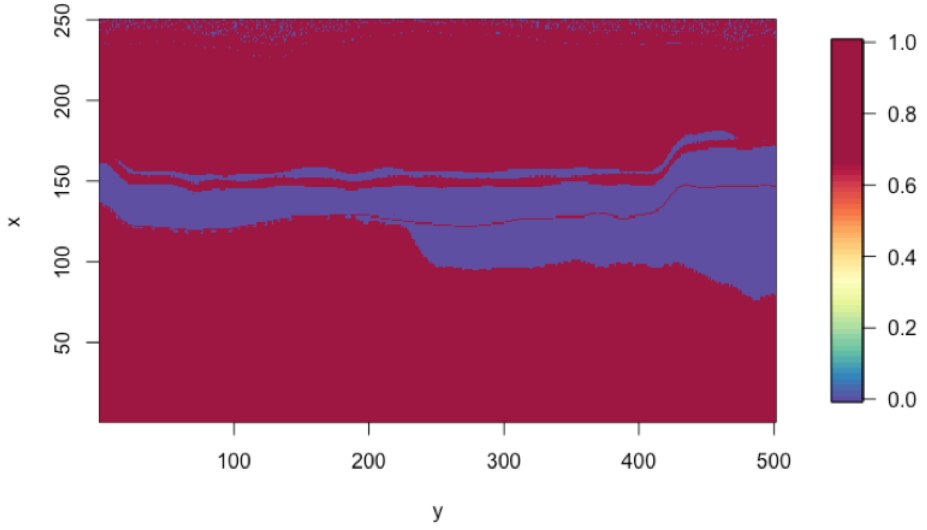

Figure 31 – The red color shows where the p-value was greater than 0.05.

If we consider the distance D, returned by the KS-test, the result is similar, figure 32. We can see a blue region that is common in figures 31 and 32. 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 paper 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.

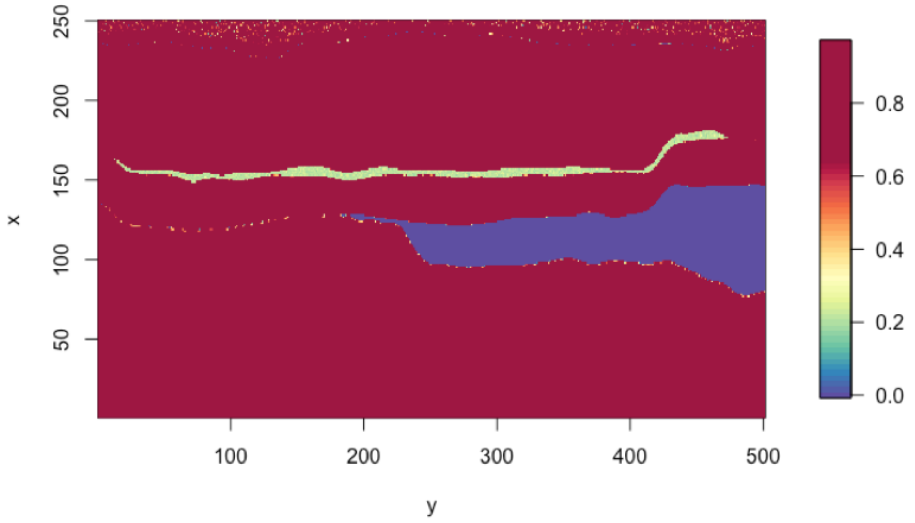

Figure 32 – Kolmogorov-Smirnoff Distance (D). The red regions represent where the GLD fits well.

8.1.5 Clustering

At this point we have our dataset characterized by the schema depicted by Equation 8.1, then using a clustering algorithm, such as k-means, we are going to group the GLDs based on its $(\lambda_2, \lambda_3, \lambda_4)$ values, as those are the values that describe the shape of the distribution at each point of the dataset.

In this paper we use k-means algorithm with $n = 10$, where n is the number of clusters to be made. This is an arbitrary value, we are investigating other algorithms as DBSCAN and what are the ϵ of this algorithm that warranty a good clusterization, but discussing alternative GLDs clustering algorithms is beyond the scope of this paper.

Once the clustering algorithm has been applied, 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 8.2), Figure 33. Note that, in Figure 33, 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 8.1.4.

$$S_{\mathcal{C}}(x_i, y_j, clusterID, GLD_{x_i,y_j}) \tag{8.2}$$

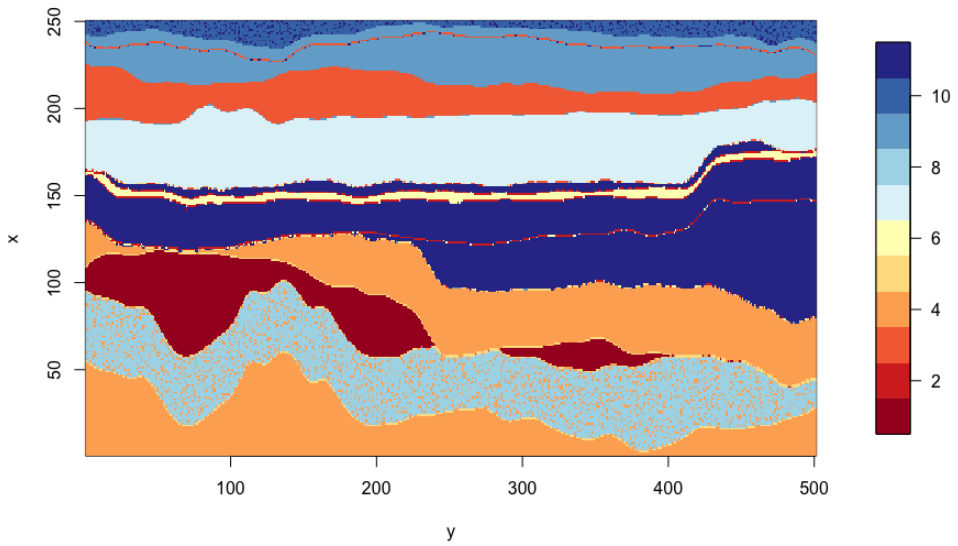

Figure 33 – Result of the clusterization using k-means with $n = 10$.

If we visually compare Figure 28 with Figure 33, we observe a close similarity between the two. It is clear that they can not be equal because we are talking about a slice of a deterministic model, and the result of making clusters on 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 34, where we plot the clusters in (λ_3, λ_4) space. As we mention in section **??**, 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. This is exactly the result we can observe in Figure 34.

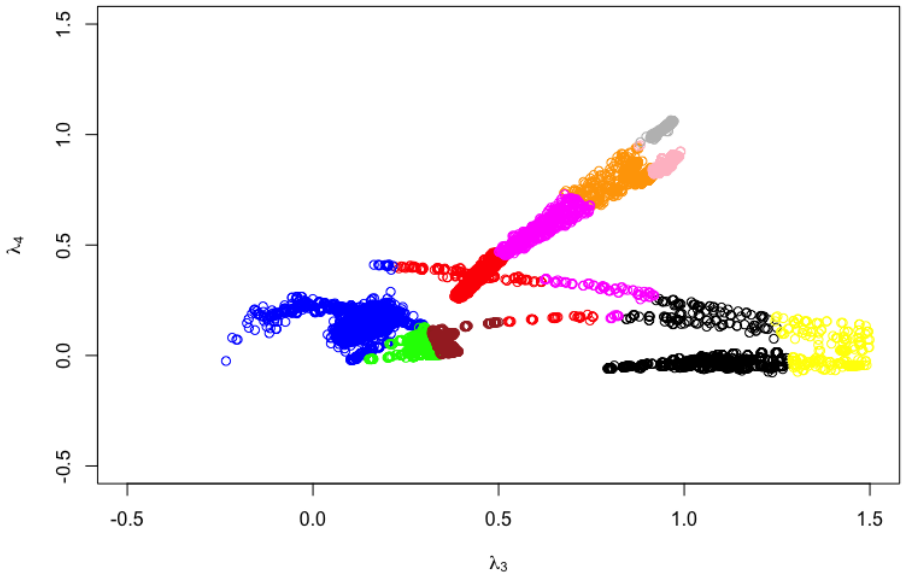

Figure 34 – Distribution of the clusters in the (λ_3, λ_4) space. The points that belongs to a same cluster are one near the others, as was expected.

To further corroborate this fact, in Figures 35 to 44 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 a 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 this cluster, Table 8. The selected member is going to be used to answer the queries in the next sections.

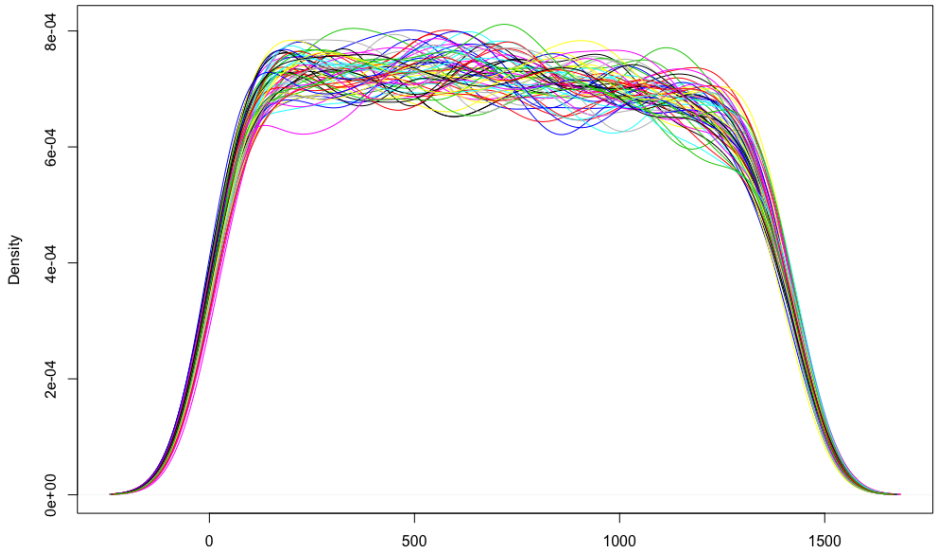

Figure 35 – *PDFs* of 60 members of the cluster 1.

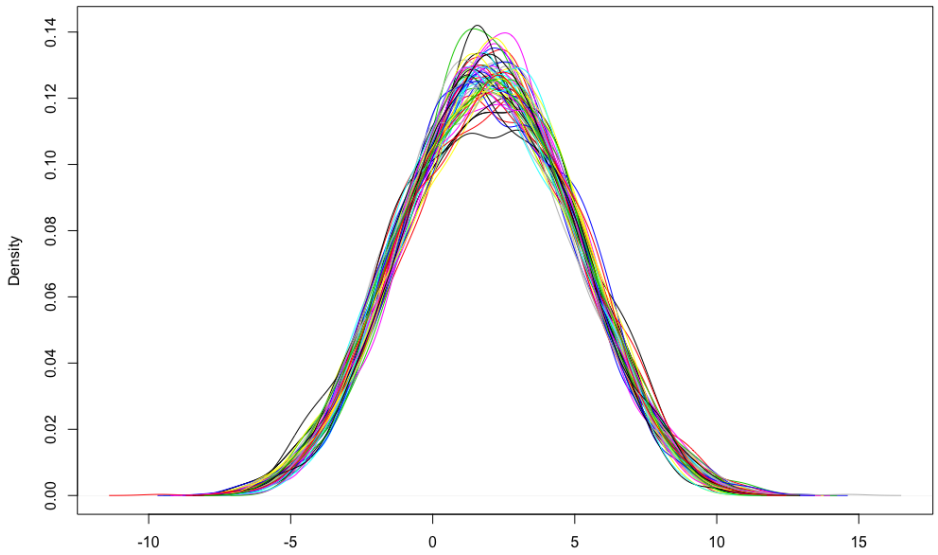

Figure 36 – *PDFs* of 60 members of the cluster 2.

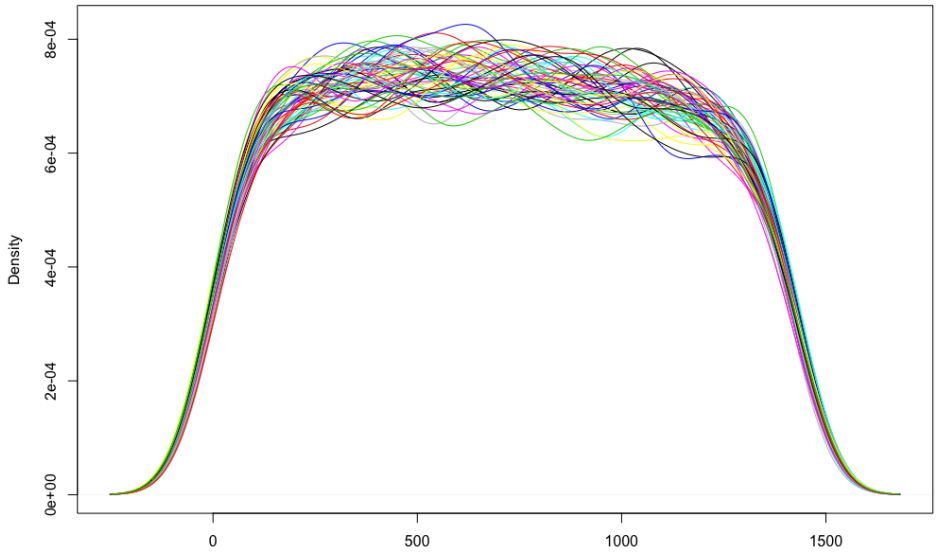

Figure 37 – *PDFs* of 60 members of the cluster 3.

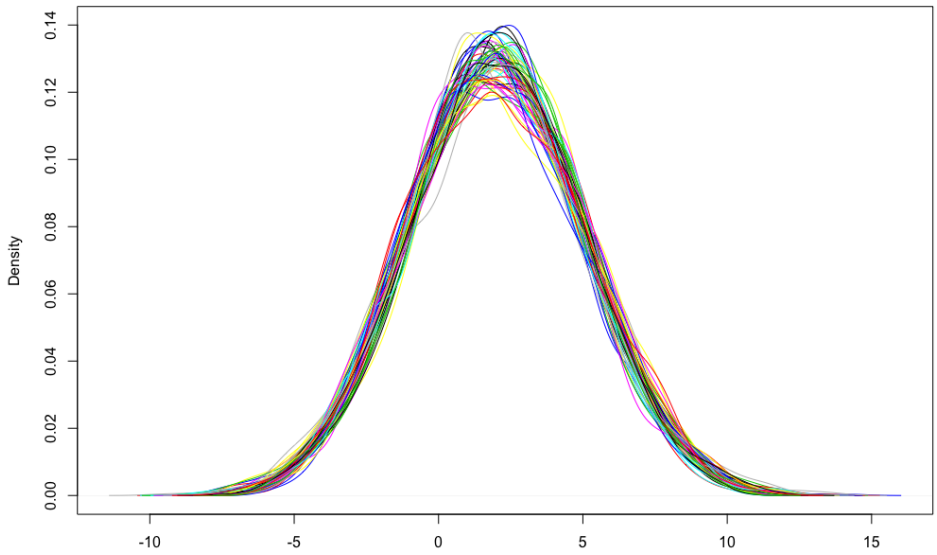

Figure 38 – *PDFs* of 60 members of the cluster 4.

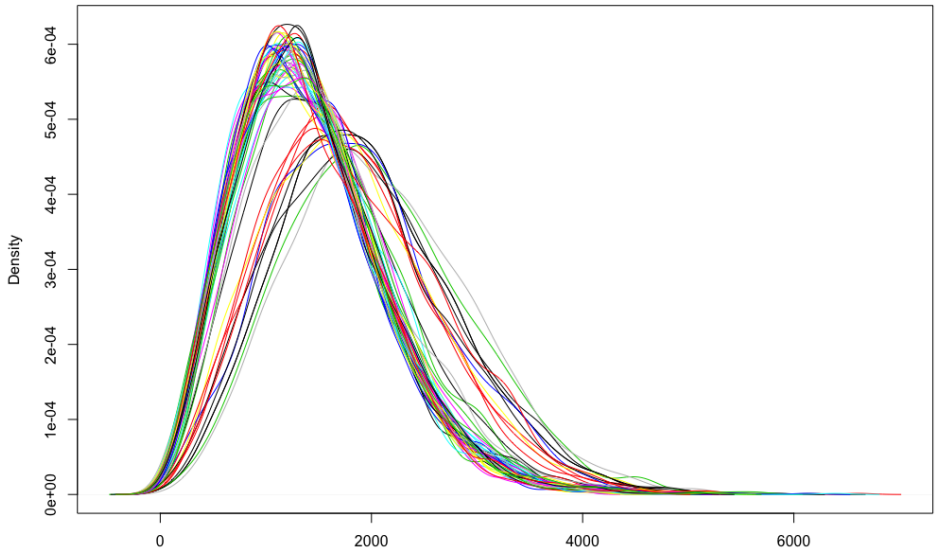

Figure 39 – *PDFs* of 60 members of the cluster 5.

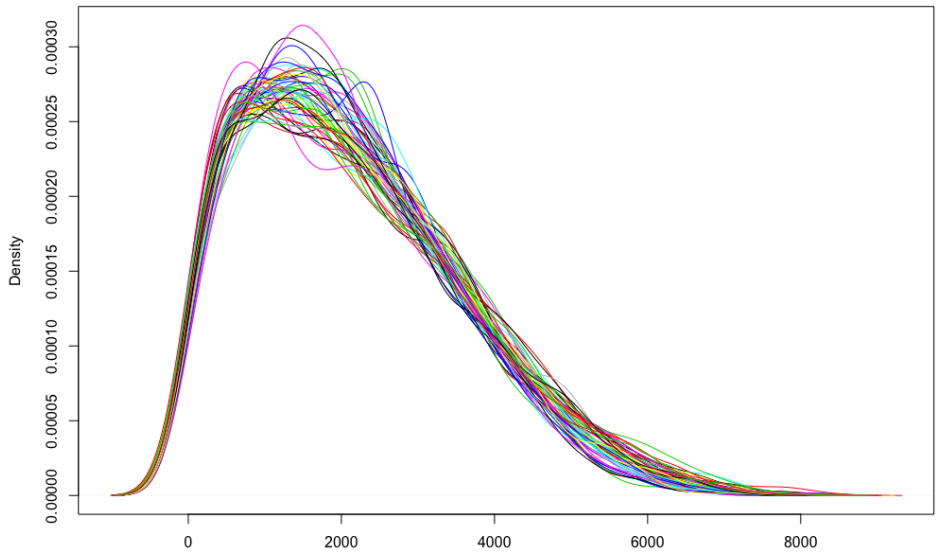

Figure 40 – *PDFs* of 60 members of the cluster 6.

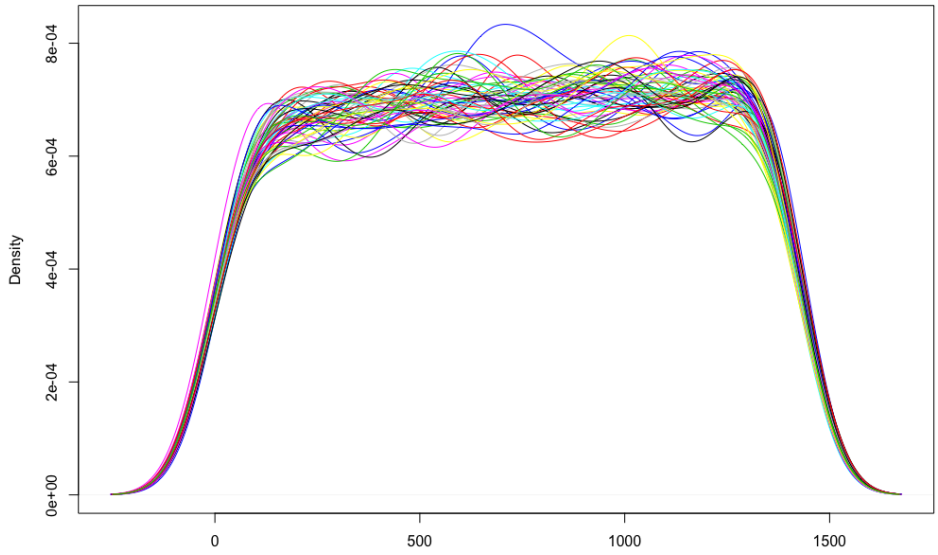

Figure 41 – *PDFs* of 60 members of the cluster 7.

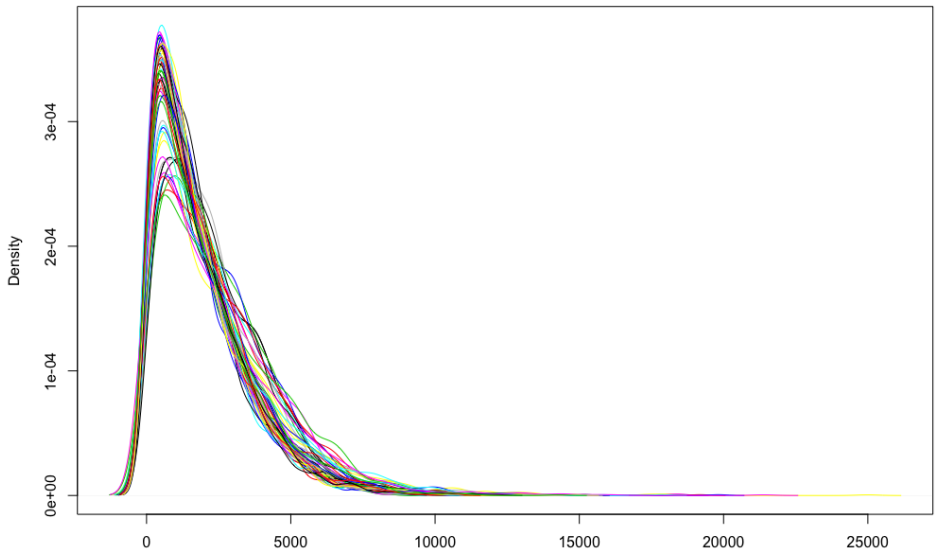

Figure 42 – *PDFs* of 60 members of the cluster 8.

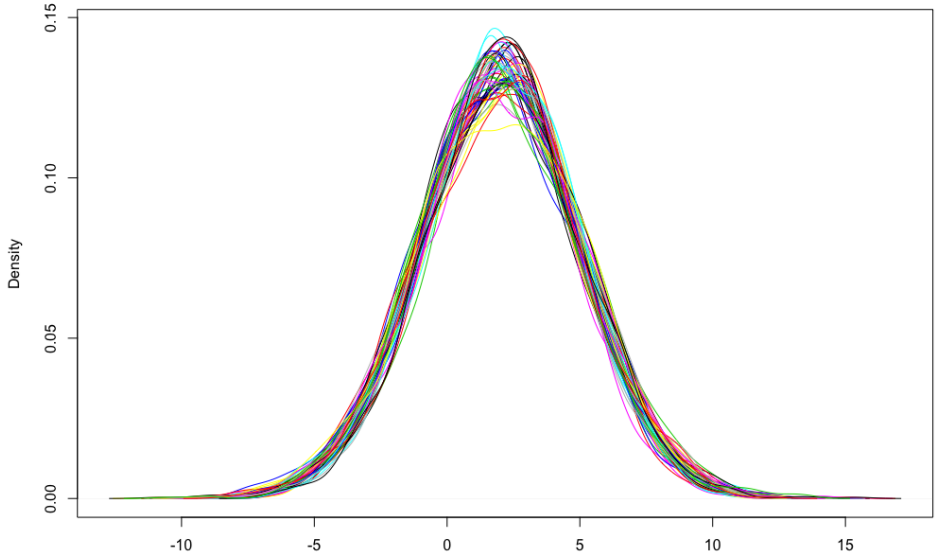

Figure 43 – *PDFs* of 60 members of the cluster 9.

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 8 – Centers of the clusters.

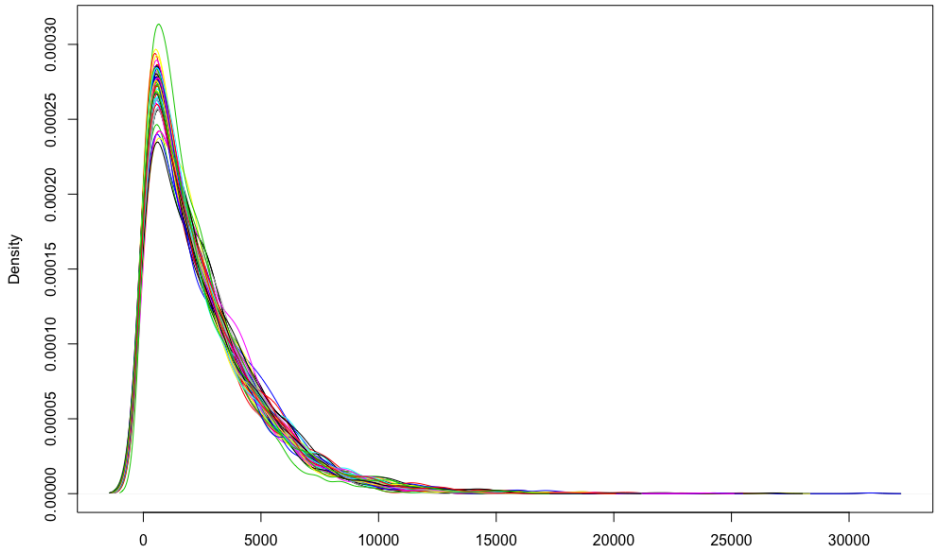

Figure 44 – *PDFs* of 60 members of the cluster 10.

The 125250 points of the slice are distributed through the clusters following the histogram of the figure 45 and Table 9.

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 9 – Distribution of the clusters.

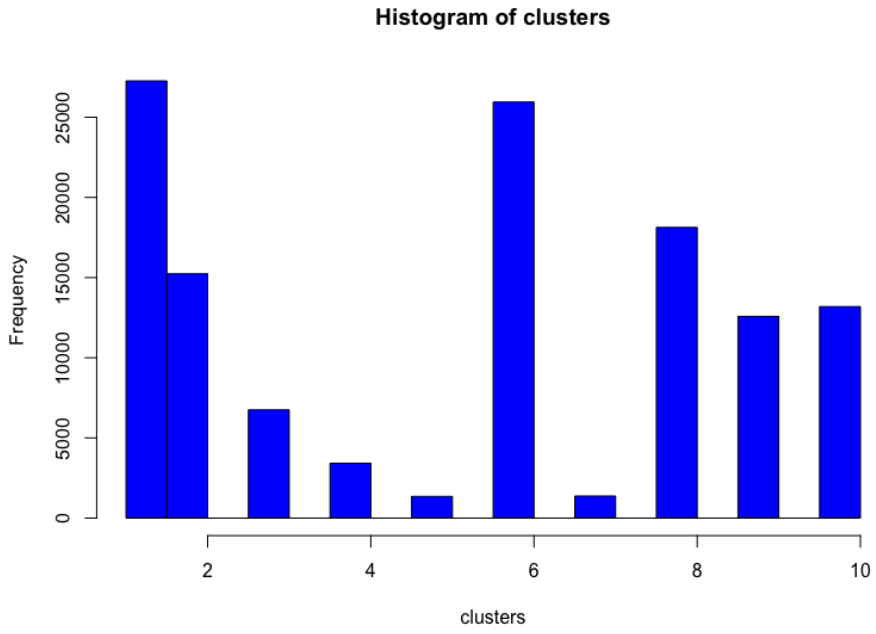

Figure 45 – Distribution of the clusters.

8.1.6 Spatio-temporal queries

At this point, the initial dataset is summarized as depicted by the schema in equation 8.2. 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 46 and the values of $[x_1, x_2], [y_1, y_2]$ that define the regions are shown in Table 10.

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 10 – Analysis Regions.

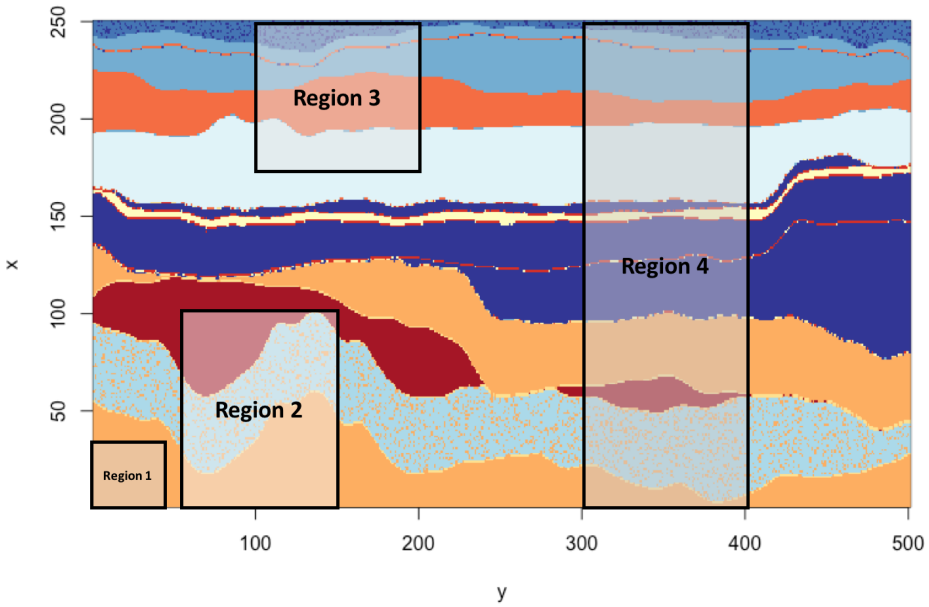

Figure 46 – Analysis Regions.

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 8.1.6.1; and in section 8.1.6.2. We use the Information Entropy to assign a value that measures the uncertainty at each region. In section 8.1.6.1, we expect the GLD mixture to characterize well the raw data; and in 8.1.6.2 we hope that the information entropy is zero in region 1 and increases between regions 2, 3 and 4.

8.1.6.1 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 **??**.

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 11.

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 11 – Distribution of the clusters by regions.

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

Table 12 – p-values by regions.

```
> clRegion1 = clByRegion(210, 250, 0, 40)
> clRegion2 = clByRegion(150, 250, 50, 150)
> clRegion3 = clByRegion(0, 75, 100, 200)
> clRegion4 = clByRegion(0, 250, 300, 400)
```

If we divide the columns of Table 11 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 7.1.3.

Based on the *p-value*, Table 12, we can conclude that in all 4 regions the *mixture of GLDs* is a good fit to the raw data.

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

Table 13 – Information Entropy by regions.

8.1.6.2 Information Entropy

Now we are going to evaluate what happens with the information entropy. Based on the distribution of clusters inside the regions, table 11; 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).

As we expect, Table 13, 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*.

The first one is a geophysical tests for wave propagation problems

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). The models include simple cases that can be used in the development stage of any geophysical imaging practitioner (developer, tester ...) as well as extremely large cases that can only be solved in a reasonable time using ExaFLOPS supercomputers. The models are generated to the required size by means of a Matlab/Octave script and hence can be used by users of any OS or computing platform. The tests can be used to benchmark and compare the capabilities of different and innovative seismic modelling approaches, hence simplifying the task of assessing the algorithmic and computational advantages that they pose.

In our case, we are going to use the "HPC4E Seismic Test Suite" as a case study of the porposed UQMS. As we mention in the introduction of this chapter this model is a spatial only domain problem, because we are going to consider a multidimentional array as an Input and a multidimentional array as an output, but of them time independet.

8.1.7 Mathematical Formulation

8.1.8 Model and Dataset Description

The models have been designed as a set of 16 layers with constant physical properties. The top layer delineates the topography and the other 15 different layer interface surfaces or horizons. In the following, an interface horizon is associated with properties that apply

Table 14 – Layer constant properties and their depth range. "Star" layers are only used in the flat case, in substitution of their non-star equivalents

Layer Id	**Vp (m/s)**	**Vs (m/s)**	**Density (Kg/m3)**	**Max. depth (m)**	**Min. depth (m)**
1	1618.92	500.00	1966.38	-135.55	-476.35
2	1684.08	765.49	1985.88	41.50	-394.90
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
2*					
3*					

to the layer that exists between itself and the immediately next layer horizon. The model covers an area of 10 x 10 x 5 km, with maximum topography at about 500 m and maximum depth at about 4500 m. The layer horizons have been sampled very finely with 1.6667 m spacing so that a highly accurate representation can be honored at high frequencies. For simulation schemes based on unstructured grids, the layer horizons can be used easily to constrain model blocks. For simulation schemes based upon Cartesian grids, a simple script is provided that can generate 3D grids for any desired spatial sampling. Table 14 shows the properties of each of the layers included in the models.

8.1.9 Adding uncertainty into the model

The "HPC4E Seismic Test Suite" does not provide uncertainty sources, because all the input parameters of the model have fixed values. Then, to the purpose of our work we need to add some uncertainties into the inputs. Let's suppose the variable V_p is uncertain. As this variable have 16 different values, one for each layer, we can consider it as a random vector, equation 8.3. We associate to each of the V_{p_i} a Normal distribution with μ_i equal to the value reported in Table 14 and $\sigma = 2$.

$$V_p =< V_{p_i}, \mathcal{N}(\mu_i, \sigma_i) > \tag{8.3}$$

8.2 Case Study: Austin, queso library

8.3 Case Study: Multidisciplinary System (NASA)

8.4 Case Study: Spatio-temporal Nicholson-Bailey model

Este esta en el software uqlab, en la carpeta Doc Manuals

9 Conclusions and Future Works

9.1 Revisiting the Research Questions

9.2 Significance and Limitations

9.3 Open Problems and Future Work

9.4 Final Considerations

Bibliography

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 na página 18.

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 29.

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 na página 18.

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 na página 18.

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 29.

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 na página 18.

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 74.

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 22 and 23.

HELTON, J. C. et al. Representation of analysis results involving aleatory and epistemic uncertainty. *International Journal of General Systems*, Taylor & Francis Group, v. 39, n. 6, p. 605–646, 2010. ISSN 0308-1079. Citado na página 23.

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 24.

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 2 vezes nas páginas 31 and 35.

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 2 vezes nas páginas 18 and 26.

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 2 vezes nas páginas 24 and 25.

KARIAN, Z. A.; DUDEWICZ, E. J. *Handbook of fitting statistical distributions with R*. [S.l.: s.n.], 2011. ISSN 1098-6596. ISBN 9788578110796. Citado 2 vezes nas páginas 19 and 63.

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 22.

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 28 and 29.

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 2 vezes nas páginas 19 and 29.

LIU, J. et al. Parallel Computation of PDFs on Big Spatial Data Using Spark. Disponível em: <https://arxiv.org/pdf/1805.03141.pdf>. Citado na página 19.

LODZIENSIS, A. U. Generalizations of tukey-lambda distributions. 2013. Citado na página 29.

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 na página 29.

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 29.

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 29.

SULLIVAN, T. J. *Introduction to Uncertainty Quantification*. Springer, 2015. ISBN 9783319233949. Disponível em: <http://www.springer.com/series/1214>. Citado na página 26.

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 3 vezes nas páginas 18, 20, and 29.

Appendix

APPENDIX A – uqms R package

A.1 Título da seção

Aqui temos uma seção dentro do Apêndice.

APPENDIX B – Ideas

B.0.1 Variance, Information and Entropy

Variance.

Information and Entropy.

B.0.2 Information Gain, Distances and Divergences

B.1 Sensitivity Analysis

Sensitivity analysis is the systematic study of how model inputs—parameters, initial and boundary conditions—affect key model outputs. Depending on the application, one might use local derivatives or global descriptors such as Sobol's functional decomposition or variance decomposition. Also, the needs of the application may range from simple ranking of the importance of inputs to a response surface model that predicts the output given the input settings. Such sensitivity studies are complicated by a number of factors, including the dimensionality of the input space, the complexity of the computational model, limited forward model runs due to the computational demands of the model, the availability of adjoint solvers or derivative information, stochastic simulation output, and high-dimensional output. Challenges in sensitivity analysis include dealing with these factors while addressing the needs of the application. (**??**)

$$E = mc^2 \tag{B.1}$$

APPENDIX C – Título do apêndice C

Annex

ANNEX A – Título do anexo A

A.1 Título da seção

Aqui temos uma seção dentro do Anexo.

ANNEX B – Título do anexo B

ANNEX C – Título do anexo C