

Imperial College of London  
Department of Aeronautics

---

# UQ and AI: Data Fusion, Inverse Identification, and Multiscale Uncertainty Propagation in Aerospace Components

---

Nick Pepper

October 22, 2021

Supervised by Francesco Montomoli & Sanjiv Sharma

Submitted in part fulfilment of the requirements for the degree of  
Doctor of Philosophy in Aeronautics of Imperial College of London  
and the Diploma of Imperial College London

I hereby certify that all work presented in this thesis is my own and that all other contributions have been appropriately acknowledged and referenced.

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work

# Abstract

A key requirement for engineering designs is that they offer good performance across a range of uncertain conditions while exhibiting an admissibly low probability of failure. In order to design components that offer good performance across a range of uncertain conditions, it is necessary to take account of the effect of the uncertainties associated with a candidate design. Uncertainty Quantification (UQ) methods are statistical methods that may be used to quantify the effect of the uncertainties inherent in a system on its performance. This thesis expands the envelope of UQ methods for the design of aerospace components, supporting the integration of UQ methods in product development by addressing four industrial challenges.

Firstly, a method for propagating uncertainty through computational models in a hierarchy of scales is described that is based on probabilistic equivalence and Non-Intrusive Polynomial Chaos (NIPC). This problem is relevant to the design of aerospace components as the computational models used to evaluate candidate designs are typically multiscale. This method was then extended to develop a formulation for inverse identification, where the probability distributions for the material properties of a coupon are deduced from measurements of its response. We demonstrate how probabilistic equivalence and the Maximum Entropy Principle (MEP) may be used to leverage data from simulations with scarce experimental data- with the intention of making this stage of product design less expensive and time consuming.

The third contribution of this thesis is to develop two novel meta-modelling strategies to promote the wider exploration of the design space during the conceptual design phase. Design Space Exploration (DSE) in this phase is crucial as decisions made at the early, conceptual stages of an aircraft design can restrict the range of alternative designs available at later stages in the design process, despite limited quantitative knowledge of the interaction between requirements being available at this stage. A histogram interpolation algorithm is presented that allows the designer to interactively explore the design space with a model-free formulation, while a meta-model based on Knowledge Based Neural

---

Networks (KBaNNs) is proposed in which the outputs of a high-level, inexpensive computer code are informed by the outputs of a neural network, in this way addressing the criticism of neural networks that they are purely data-driven and operate as black boxes.

The final challenge addressed by this thesis is how to iteratively improve a meta-model by expanding the dataset used to train it. Given the reliance of UQ methods on meta-models this is an important challenge. This thesis proposes an adaptive learning algorithm for Support Vector Machine (SVM) metamodels, which are used to approximate an unknown function. In particular, we apply the adaptive learning algorithm to test cases in reliability analysis.

# Acknowledgements

This thesis collates 4 years of research in the field of Uncertainty Quantification and Artificial Intelligence. Over this period, I have benefited greatly from collaborations with researchers from a number of companies and institutions. I would like to acknowledge and thank the following people for their contributions to this work:

- Francesco Giacomel and Giovanna Cavazzini from the University of Padova and Michele Pinelli and Nicola Casari from the University of Ferrara for the CFD simulations that were used in the scroll compressor test case;
- Phillippe Peters and Linden Harris from Airbus for providing test cases and helpful feedback at our regular meetings throughout my PhD;
- Luca Gerardo-Giorda at the Johannes Kepler University for supervising the work on the invasive species test case;
- Audrey Gaymann at TOffeeAM for collaborating with me on developing the Knowledge Based Neural Network architecture;

- Luis Crespo at NASA Langley for supervising me as we continue to develop the adaptive learning algorithm;
- Finally, special thanks to Francesco Montomoli and Sanjiv Sharma, my PhD supervisors. I greatly appreciate the guidance and support you have both given me throughout my PhD, as well as the opportunities that you have provided for me.

I would also like to thank the EPSRC and Airbus for their financial support that made this research possible and to recognise the work of the staff at the Department of Aeronautics at Imperial College London to make the logistics of the PhD run very smoothly.

On a personal note, I would like to thank the following people for their support throughout my PhD: my friends at Imperial College: Mo, Maria, Krishna, Kyriakos, and Lloyd; my parents Sue and Ed; and my fiancé Eleanor. This thesis was written in the unique circumstances of 2020-21; thanks must also go to my feline office mates during lockdown, Ripley and Benton, for their zoom interruptions, comic relief and general company during these strange times.

# Contents

<b>Acknowledgements</b>	<b>5</b>
<b>Table of Contents</b>	<b>9</b>
<b>List of Publications</b>	<b>10</b>
<b>Nomenclature</b>	<b>12</b>
<b>List of Figures</b>	<b>18</b>
<b>List of Tables</b>	<b>20</b>
<b>1 Introduction</b>	<b>21</b>
1.1 Aims and Objectives	23
1.2 Thesis Outline	25
<b>2 Multi-scale Uncertainty Propagation with arbitrary Polynomial Chaos</b>	<b>27</b>
2.1 Literature discussion	28
2.2 Stochastic model reduction with probabilistic equivalence	33
2.2.1 Fine and coarse-scale probabilistic models . . . . .	33
2.2.2 Problem formulation . . . . .	35
2.2.3 Propagating uncertainty with Non-Intrusive Polynomial Chaos . . .	36
2.2.4 Metrics for the statistical distance . . . . .	44

<b>2.3 Stochastic upscaling with a single common QoI</b>	<b>47</b>
2.3.1 Stochastic upscaling with Gaussian input distributions . . . . .	49
2.3.2 Stochastic upscaling for non-Askey scheme input distributions . . .	49
2.3.3 Choice of univariate orthogonal polynomials . . . . .	50
<b>3 Inverse Identification and Data Fusion: Leveraging Simulations with Experimental Data</b>	<b>53</b>
<b>3.1 Back-calculation of missing input probability distributions</b>	<b>54</b>
3.1.1 Validation with scroll compressor test case . . . . .	59
3.1.2 Carbon fibre coupons test case . . . . .	65
<b>3.2 Data Fusion of NIPC and scarce experimental data</b>	<b>82</b>
3.2.1 Algorithm for data fusion of Polynomial Chaos with experimental results . . . . .	86
3.2.2 p-boxes as a constraint on the optimisation . . . . .	88
3.2.3 Carbon fibre composite test case . . . . .	90
<b>4 Machine learning meta-models for DSE</b>	<b>95</b>
<b>4.1 Histogram interpolation for Design Space Exploration</b>	<b>96</b>
4.1.1 An alg. for non-param., data-driven hist. interp. . . . .	100
4.1.2 Probabilistic Equivalence . . . . .	101
4.1.3 Histogram interpolation test case . . . . .	106
4.1.4 Predictive uncertainty of histogram interpolation . . . . .	115
<b>4.2 Bi-fidelity modelling with Knowledge Based Neural Networks</b>	<b>120</b>
4.2.1 Knowledge Based Neural Networks (KBaNNs) . . . . .	123
4.2.2 KBaNN architecture for a local Navier-Stokes approximation . . . .	126
4.2.3 Test case and results . . . . .	128
<b>5 Adaptive Learning for Reliability Analysis using Support Vector Machines</b>	<b>139</b>



<b>5.1 Meta-modelling strategies for reliability analysis</b>	<b>140</b>
<b>5.2 Support Vector Machines for reliability analysis</b>	<b>144</b>
<b>5.3 Algorithm for Adaptive learning with SVMs</b>	<b>149</b>
5.3.1    Selecting candidates on the LSF . . . . .	150
5.3.2    Finding Sets Containing Points with High Likelihood . . . . .	152
5.3.3    Promoting exploration and quantifying predictive uncertainty . . .	153
<b>5.4 Validation with synthetic test cases</b>	<b>157</b>
5.4.1    Four branch function . . . . .	158
5.4.2    Physical space with discontinuities . . . . .	160
<b>6 Conclusions</b>	<b>164</b>
<b>6.1 Future Developments</b>	<b>166</b>
<b>Bibliography</b>	<b>167</b>
<b>Appendices</b>	<b>197</b>

# List of Publications

## Journal Publications

- **Pepper, N.**, Montomoli, F., and Sharma, S. (2021). "A non-parametric, data-driven histogram interpolation method for design space exploration". Under review in *Journal of Mechanical Design*.
- **Pepper, N.**, Gaymann, A., Sharma, S., and Montomoli, F. (2021). "Local bi-fidelity field approximation with Knowledge Based Neural Networks for Computational Fluid Dynamics". *Scientific Reports*.
- **Pepper, N.**, Montomoli, F., and Sharma, S. (2021). "Data fusion for Uncertainty Quantification with Non-Intrusive Polynomial Chaos". *Computer Methods in Applied Mechanics and Engineering*.
- **Pepper, N.**, Montomoli, F., and Sharma, S. (2021). "Identification of missing input distributions with an inverse multi-modal polynomial chaos approach based on scarce data". *Probabilistic Engineering Mechanics*.
- **Pepper, N.**, Gerardo-Giorda, L., and Montomoli, F. (2019). "Meta-modeling on detailed geography for accurate prediction of invasive alien species dispersal". *Scientific Reports*.

- **Pepper, N.**, Montomoli, F., and Sharma, S. (2019). "Multiscale uncertainty quantification with arbitrary polynomial chaos". *Computer Methods in Applied Mechanics and Engineering*.
- Sakuma, M., **Pepper, N.**, Warnakulasuriya, S., Montomoli, F., Wüchner, R. and Bletzinger, K. (2021). "Multi-Fidelity Uncertainty Quantification of high Reynolds number turbulent flow around a Rectangular 5:1 Cylinder". *Wind Structures: an International Journal*
- Ahlfeld, R., Ciampoli, F., Pietropaoli, M., **Pepper, N.**, and Montomoli, F. (2019). "Data-driven uncertainty quantification for Formula 1: Diffuser, wing tip and front wing variations". *Proceedings of the Institution of Mechanical Engineers Part D Journal of Automobile Engineering*

### Conference Proceedings

- **Pepper, N.**, Crespo, L., and Montomoli, F. (2021). "Adaptive Learning for Reliability Analysis using Support Vector Machines". *ESREL 2021*.
- **Pepper, N.**, Montomoli, F., Giacomel, F., Cavazzini, G., Pinelli, M., Casari, N., and Sharma, S. (2020). "Uncertainty quantification and missing data for turbomachinery with probabilistic equivalence and arbitrary Polynomial Chaos, applied to scroll compressors". *ASME Turbo Expo 2020*.
- Sakuma, M., **Pepper, N.**, Kodakkal, A., Wuchner, R. Bletzinger, K., and Montomoli, F. (2019). "Multi-fidelity uncertainty quantification of the flow around a rectangular 5:1 cylinder". *UNCECOMP 2019*.

# Nomenclature

aPC	arbitrary Polynomial Chaos
CDF	Cumulative Distribution Function
CFD	Computational Fluid Dynamics
DKWM	Dvoretzky–Kiefer–Wolfowitz inequality and the Massart bound
DNS	Direct Numerical Simulation
FEM	Finite Element Model
FEMU	Finite Element Model Updating
GA	Genetic Algorithm
GPM	Gaussian Process model
KBaNN	Knowledge Based Neural Network
KL	Kullback-Leibler (divergence)
KNN	$k$ -nearest neighbors
KS	Kolmogorov-Smirnov (distance)
LSF	Limit State Function

LSV	Limit State Volume
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
MEP	Maximum Entropy Principle
MLP	Multi-Layer Perceptron
mmPCE	multi-modal Polynomial Chaos Expansion
MSE	Mean squared error
NIPC	Non-Intrusive Polynomial Chaos
PCE	Polynomial Chaos Expansion
PDF	Probability Density Function
QoI	Quantity of Interest
RANS	Reynolds Averaged Navier-Stokes
RBDO	Reliability-Based Design Optimisation
RSS	Residual Sum of Squares
SPDE	Stochastic Partial Differential Equation
SQP	Sequential Quadratic Programming
SVM	Support Vector Machine
UQ	Uncertainty Quantification

# List of Figures

1.1	An illustration of the cone of uncertainty: as more decisions are made the range of possible outcomes for a project narrows. . . . .	25
1.2	Schematic illustrating the methods developed in the PhD. . . . .	26
2.1	The stochastic upscaling problem — how to propagate uncertainties from the finest to the coarsest scales? . . . . .	32
2.2	Illustration of the stochastic upscaling algorithm . . . . .	33
2.3	Flowchart of the stochastic upscaling algorithm . . . . .	36
2.4	Schematic illustrating a forwards propagation of uncertainty with NIPC . .	37
2.5	Collocation points and Smolyak grid for a Gaussian PDF and a lognormal histogram . . . . .	43
2.6	CDFs of two arbitrary distributions — the KS distance (arrow in red) is defined as the supremum distance between the two CDFs. . . . .	45
2.7	PDFs and CDFs of four datasets, sampled from Gaussian distributions . .	46
2.8	Wing box, with mesh, created in Abaqus . . . . .	48
2.9	Cross sectional view of the meshed wing box . . . . .	49
2.10	Stochastic upscaling for the case of Gaussian input distributions using a first-order PCE ( $d^{KS} = 0.0163$ ) . . . . .	50
2.11	Stochastic upscaling with non-Askey scheme input distributions using a second-order PCE ( $d^{KS} = 0.0268$ ) . . . . .	51

2.12	Plots comparing the convergence of the statistical distance, as quantified by the Kolmogorov–Smirnov distance, for different orthonormal polynomial bases . . . . .	52
3.1	An illustration of the back-calculation framework . . . . .	58
3.2	An illustration of the midline of the scroll compressor and the fixed and orbiting scrolls . . . . .	60
3.3	A parallel axis plot displaying the results of the CFD simulations . . . . .	61
3.4	A summary of the back-calculation problem for the scroll compressor test case . . . . .	62
3.5	Histograms displaying the synthetic dataset used to validate the framework	62
3.6	CDFs for the simulated and experimental distributions of $\eta$ and $h_{scroll}$ used to validate the framework . . . . .	63
3.7	CDFs and a QQplot for $\eta$ for the case of sparse data . . . . .	64
3.8	Histograms for the synthetic dataset with non-Gaussian input distributions	66
3.9	CDFs and QQ plot for the experimental and simulated distributions of $\eta$ for the case of non-Gaussian input distributions . . . . .	77
3.10	A schematic illustrating how a carbon fibre composite is constructed from layers of unidirectional plies . . . . .	78
3.11	Histograms for the four stochastic material properties in the dataset and the compression and shear loadings responsible for first failure of the laminate	79
3.12	An illustration of a carbon fibre composite . . . . .	79
3.13	Flowchart for the back-calculation algorithm illustrated for the carbon fibre composite test case . . . . .	80
3.14	A comparison of the CDFs for the quantity of interest, $N_{xy}$ . . . . .	80
3.15	A comparison of CDFs for $f_1$ and $f_{12}$ . . . . .	81

3.16	Having determined an initial estimate of the probability distribution for the QoI, scarce experimental measurements of the QoI can be used to update the coefficients of the PCE using a data fusion algorithm based on a MEP	86
3.17	Data fusion algorithm to adjust the coefficients based on an Maximum Entropy Principle, subject to constraints imposed by the available experimental data . . . . .	89
3.18	A plot showing the CDFs of the original estimate of the probability distribution arising from the aPC; the available experimental data and the updated estimate of the probability distribution . . . . .	92
4.1	The performance of a design is typically non-deterministic due to the presence of irreducible uncertainties in the system . . . . .	99
4.2	Schematic offering two views of the cantilever beam model used as a test case. . . . .	109
4.3	We mimic realistic design requirements through constraints on the beam weight and width. . . . .	110
4.4	(a-d) plots of the estimated probability distribution for $D/D_0$ (blue) against the true distribution (red), found through direct MC simulation, for each of the four test points. . . . .	111
4.5	(a-d) Plots comparing the estimated PDF (blue) against the true distribution found by direct MC sampling (red), for each of the test points. This time the interpolated distribution did not belong to a family of distribution in the Askey scheme for which an optimal orthogonal polynomial exists. .	114
4.6	The performance of the algorithm, as quantified by the KS distance, is strongly correlated with the proximity of the test point to the training data.	115



4.7	(a) An example of how $2\sigma$ confidence bands can be used to visualise the predictive uncertainty in a GPM. (b) These bands are difficult to interpret for the PCE coefficients, instead we use the statistical dispersion of $\hat{f}(w \boldsymbol{\lambda}(\mathbf{x}^*), \mathbf{x}^*)$ and $\hat{g}(w \boldsymbol{\lambda}(\mathbf{x}^*), \mathbf{x}^*)$ as a metric for the uncertainty in the estimate. . . . .	116
4.8	(a) Surfaces representing the actual performance of the histogram interpolation algorithm, quantified by the KS distance. (b-d) Surfaces representing the predictive uncertainty using the Hellinger distance, dissimilarity index, and ratio of variances for $d^{fg}$ . . . . .	119
4.9	Architecture of a KBaNN for bi-fidelity modelling . . . . .	122
4.10	Schematic of the $n_g \times n_g$ sample grids used for the local Navier-Stokes approximation . . . . .	127
4.11	A modified KBaNN architecture was used for the local Navier-Stokes approximator . . . . .	127
4.12	Visualisations of the two-dimensional flows used to train the KBaNNs . . .	135
4.13	The performance of the KBaNNs was evaluated for a channel flow at $U = 0.11\text{ms}^{-1}$ . . . . .	136
4.14	Plot of the performance of the KBaNNs as a function of the scarcity of the training data . . . . .	136
4.15	Simulation results for the flow over a NACA 2412 airfoil at $Re = 480$ . . .	137
4.16	The locations of the sample grids used to test the KBaNN . . . . .	138
5.1	A trained Support Vector Machine (SVM) finds a separating hyperplane that maximises the width of the margin between training data that is categorised into two classes . . . . .	146
5.2	An example of how a polynomial transformation may be used to map data to a higher dimensional feature space in which the transformed data becomes linearly separable . . . . .	149

5.3	A polynomial mapping is used to lift a two dimensional physical space to a three dimensional feature space where an orthonormal basis which spans the separating hyperplane can be defined . . . . .	151
5.4	Illustration of the least squares fitting for the empirical function $\hat{g}(s)$ to the training data; and the upper and lower estimates of $\hat{g}$ for two Monte Carlo samples. . . . .	156
5.5	Panels indicating the convergence of the estimated LSF to the four branch LSF during iterations 0, 20, 40, and 80 of AL-SVM, with $W = I$ . . . . .	159
5.6	Plot of the convergence of $P$ to $P_{mc}$ with algorithm iteration for AL-SVM and AK-MCS (left). CPU times (s) for each iteration of the two algorithms (right). . . . .	160
5.7	Plot of the LSF, training data, and estimated training data for the case of discontinuous changes in $g(\mathbf{x})$ (left). Plot displaying the convergence of $P$ to $P_{mc}$ for AL-SVM (right). . . . .	162
6.1	Basque Country test case. . . . .	204
6.2	<b>Panel A.</b> Histograms for the uncertain parameters $\nu$ and $\alpha$ , and the 1D collocation points for the input distributions. <b>Panel B.</b> The sampling grid calculated through application of Smolyak's rule at level 1. . . . .	205
6.3	Uncertainty analysis for the Basque country test case. . . . .	207

# List of Tables

2.1	Common types of parametric distributions and their corresponding univariate orthogonal polynomials in the Askey scheme . . . . .	43
2.2	A comparison of the statistical distances of the distributions A, B, and C from Figure 2.7 from the target distribution for different measures of statistical distance. . . . .	46
3.1	Results of the Sobol sensitivity analysis using the experimental dataset. Given the limited sensitivity of the model to $E_1$ and $G_{12}$ these parameters were treated as constants for the purposes of back-calculating a distributions for $f_1$ and $f_{12}$ . . . . .	73
3.2	The effect of increasing the order of the individual PCEs in the mmPCE on the probabilistic equivalence in the distributions for $N_{xy}$ , quantified by the KS distance . . . . .	75
3.3	A comparison of the computed and experimental probability distributions is not trivial due to the scarcity of the experimental data. Here, the percentage difference in the mean and the p-values from the Wilcoxon rank sum test are presented for each of the uncertain parameters . . . . .	75

4.1	Definitions of the parameters used in the test case. The loads and Young's Modulus of the beam were considered to be stochastic parameters, with training data generated through MC sampling of the distributions detailed in the right column . . . . .	107
4.2	The performance of the algorithm was quantified by comparing the KS distance between the true and estimated distributions, as well as the % differences in the mean and standard deviation. . . . .	112
4.3	The performance of the algorithm tabulated for the second test case, where the interpolated histogram does not have a corresponding optimal Askey polynomial. . . . .	114
4.4	The Euclidean distances between the surfaces representing the uncertainty in the predictions, displayed in Figure 4.8(b-d), and the surfaces representing the actual performance of the algorithm (4.8(a)). . . . .	120
4.5	The residual sum of squares (RSS) between the KBaNN predictions and the fine mesh for each of the sample grids pictured in Figure 4.13 . . . . .	131
4.6	The RSS between the KBaNN predictions and the fine mesh for each of the sample grids pictured in Figure 4.16(a) . . . . .	133
5.1	The number of iterations required by the three implementations of AL-SVM and AK-MCS to achieve a $P$ with percentage error below a given tolerance, as compared to $P_{mc}$ . . . . .	161
5.2	The number of iterations required by AL-SVM to achieve a $P$ with percentage error below a given tolerance, as compared to $P_{mc}$ . AK-MCS did not converge for this case. . . . .	162

# Introduction

A key requirement for engineering designs is that they offer good performance across a range of uncertain conditions while exhibiting an admissibly low probability of failure. A rigorous regime of computational and physical testing is necessary to certify a new product, making it costly to develop multiple designs in parallel and can lead to delays if the requirements change during the design process (see, e.g. [1, 2]). There is industrial and academic interest in developing data-driven meta-models that can estimate the performance and the probability of failure of a candidate design when experimental data is missing or scarce and computational resources may be limited. Such meta-models allow the designer more freedom to explore the design space before committing resources to a single design.

Uncertainty Quantification (UQ) methods are statistical methods that may be used to quantify the effect of the uncertainties inherent in a system on its performance. Broadly, UQ methods may be divided into two groups: the first including methods for forwards uncertainty propagation, in which the effect of uncertain inputs on the Quantity of Interest (QoI) of a computer code are characterised. The uncertain QoI,  $w$ , is typically represented via a probability distribution that can be absolutely continuous and represented by a probability density function (PDF). Methods for reliability analysis, which estimate the probability of a candidate design exceeding a set of requirements imposed

on the system, fit within this category of UQ methods. Conversely, the second family of UQ methods solve inverse problems in which the uncertain system inputs are deduced from measurements of the QoI. An example of such a problem is inverse identification, in which the material properties are deduced from experimental measurements of a QoI.

Regardless of whether the uncertainty propagation is forwards or inverse, most problems in UQ can only be solved through a meta-modelling strategy due to the computational costs associated with each evaluation of an expensive computer code. For instance, consider the problem of estimating the expected value of a QoI for an uncertain system with  $n_u$  uncertain inputs  $\boldsymbol{\xi} \in \mathbb{R}^{n_u}$ . This is calculated through:

$$\langle w \rangle = \int_{\boldsymbol{\xi}} w(\boldsymbol{\xi}) \rho(\boldsymbol{\xi}) d\boldsymbol{\xi}, \quad (1.1)$$

where  $\rho(\boldsymbol{\xi})$  refers to the joint probability density function of the uncertain inputs. In practice this integral is often evaluated numerically, typically through a Monte Carlo sampling strategy:

$$\hat{w} \approx \frac{1}{n_{MC}} \sum_{i=1}^{n_{MC}} w(\hat{\boldsymbol{\xi}}^{(i)}), \quad (1.2)$$

where  $\hat{\boldsymbol{\xi}}$  are  $n_{MC}$  Monte Carlo samples drawn from the joint density. Monte Carlo methods typically require  $10^6 - 10^7$  samples to converge, which can become computationally intractable if the computer code is expensive to evaluate. If this is the case then a meta-model for  $w$  is required. In the context of designing aeronautical components,  $w$ , is typically the output of a behavioural model for structural and fluid mechanics. Examples of meta-modelling strategies in the UQ literature include: polynomial chaos expansions (PCEs) [3], response surface methods [4], artificial neural networks [5], Gaussian Process models (GPMs) [6], and Support Vector Machines (SVMs) [7].

## 1.1 Aims and Objectives

The aim of this PhD studentship was to develop novel methods for UQ that address specific challenges that arise during the design of new aerospace components. We illustrate the design process of a new product with the cone of uncertainty, displayed in Figure 1.2, to indicate where each of these challenges occur in the chronology of product development. Confusingly, in this context ‘uncertainty’ refers to the potential of the project to explore alternative designs, rather than the inherent uncertainty present in the system that might be introduced by manufacturing defects for example. The designs of new engineering products are developed within the design space,  $\mathbb{X}$ , a space of  $n_x$  parameters that influence the performance of the final design. The subspace containing points in the design space that satisfy the requirements imposed on the design is referred to as the feasible region. In the early stage there are numerous candidate designs which lie in the feasible region of the design space and as a consequence there is initially a high degree of uncertainty as there are numerous potential designs. However, over time decisions are made which restrict the feasible region of the design space by narrowing the range of values over which each parameter can take and by reducing the dimensionality of the design space by finalising the values of parameters. These decisions reduce the level of uncertainty until eventually there is convergence to a final design [8, 9, 10].

Some of the challenges that we discuss are general and will be present at all times during product development, while some arise at specific instances. These challenges may be summarised as follows:

### 1. Multiscale uncertainty quantification

How can uncertainty be propagated efficiently between scales in a multiscale model?

This is a problem that is relevant throughout the design process. In particular we consider the stochastic upscaling problem, in which uncertainties at the lowest scales of a hierarchy of models are propagated to coarser scales, where the domain of the model is larger but less granular.

## 2. Design Space Exploration

Decisions made at the early, conceptual stages of product design can restrict the range of alternative designs available at later stages in the design process, despite limited quantitative knowledge of the interaction amongst requirements being available in the conceptual design stage. For this reason there has been interest in methods that enable a wider exploration of the design space at the conceptual design stage. As noted in Figure 1.2 designs are typically evaluated using either historic data (results from experiments/simulations of previous products) or high-level computer code, that can be run quickly, at this stage. Methods for Design Space Exploration (DSE) must balance flexibility in exploring the design space with the associated cost of evaluating potentially expensive computer code at the investigated parameter points in the design space.

## 3. Inverse identification and data fusion

Once a design evolves beyond the conceptual design stage it becomes economic to assess its performance with more granular, expensive computer code. In addition, prototyping and physical tests will begin. Aerospace components must be validated with rigorous physical tests before being certified, requiring thousands of tests of individual carbon fibre composite coupons. Such tests are expensive and time consuming [11]. In addition, the measurement of some properties of composite materials can be very challenging. For these reasons, there has been interest in using computer code to reduce the burden of coupon testing.

We consider two formulations through which data from simulations may be leveraged by experimental data: firstly by considering an inverse problem in which missing data for the material properties of a composite material is deduced from experimental measurements of a QoI. Secondly, by developing a method for data fusion in a forwards uncertainty propagation that allows scarce experimental data to inform



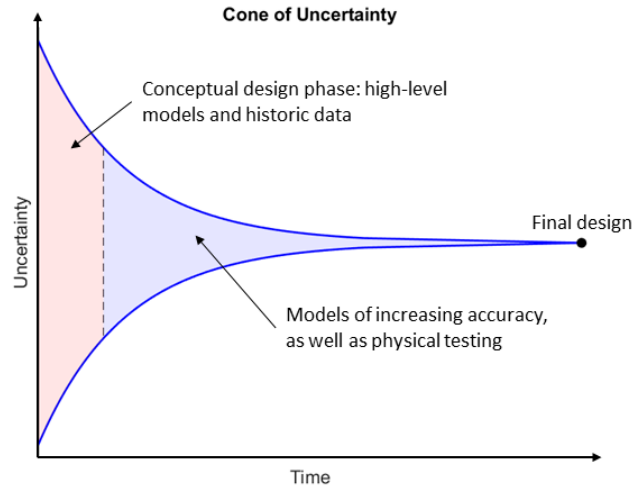


Figure 1.1: An illustration of the cone of uncertainty: as more decisions are made the range of possible outcomes for a project narrows.

the predictions of a computer code, a relatively under-researched problem in the UQ literature.

#### 4. Adaptive learning

In this thesis we develop methods to address the above challenges by employing PCEs, GPMs, SVMs and neural networks as meta-models. A meta-model is developed based on a training set consisting of evaluations of an expensive computer code at certain parameter points. The final challenge considered in this thesis is how such a training set might be expanded to improve the accuracy of a meta-model. Specifically, we develop an adaptive learning method for reliability analysis using SVMs.

## 1.2 Thesis Outline

The remainder of this thesis describes the methods that were developed to address the challenges identified above. While the four challenges are distinct, there are commonalities in the methods that were developed. This thesis has been organised so that these methods and the links between them may be presented in a logical manner; with the

consequence that the chapters are out of chronological order in terms of product development (for instance we consider inverse identification before DSE in the conceptual design phase). The thesis is organised as follows: in **Chapter 2** we present an algorithm for multiscale uncertainty quantification through probabilistic equivalence and Non-Intrusive Polynomial Chaos (NIPC). We expand upon this algorithm to develop a framework for inverse identification problems, which is presented in **Chapter 3**, along with a data fusion algorithm for leveraging PCEs and scarce experimental data. In **Chapter 4** we introduce two machine learning based algorithms for DSE: a model-free formulation that uses probabilistic equivalence and GPMs in a novel, data-driven histogram interpolation algorithm and a Knowledge Based Neural Network (KBaNN) architecture for bi-fidelity modelling, in which a neural network is used to inform the predictions of an inexpensive, high-level computer code. Finally, **Chapter 5** describes an algorithm for adaptive learning of an unknown function with a meta-model, with specific application to reliability analysis with SVMs. Figure 1.2 is a schematic illustrating the methods developed in the PhD and the links between them.

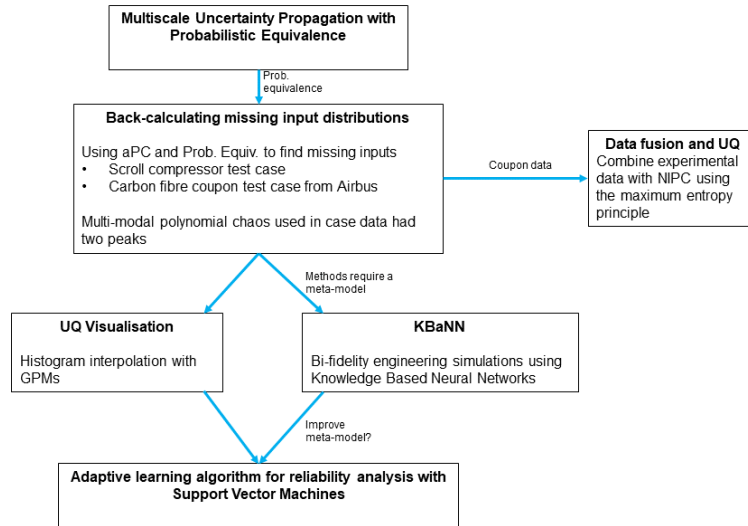


Figure 1.2: Schematic illustrating the methods developed in the PhD.

# Multi-scale Uncertainty Propagation with arbitrary Polynomial Chaos

## Contents

<b>2.1 Literature discussion</b>	<b>28</b>
<b>2.2 Stochastic model reduction with probabilistic equivalence</b>	<b>33</b>
2.2.1 Fine and coarse-scale probabilistic models . . . . .	33
2.2.2 Problem formulation . . . . .	35
2.2.3 Propagating uncertainty with Non-Intrusive Polynomial Chaos . . .	36
2.2.4 Metrics for the statistical distance . . . . .	44
<b>2.3 Stochastic upscaling with a single common QoI</b>	<b>47</b>
2.3.1 Stochastic upscaling with Gaussian input distributions . . . . .	49
2.3.2 Stochastic upscaling for non-Askey scheme input distributions . . .	49
2.3.3 Choice of univariate orthogonal polynomials . . . . .	50

## 2.1 Literature discussion

As has been discussed, computational modelling plays an extensive part in aircraft design throughout product design: Computational Fluid Dynamics (CFD) models are used from the preliminary design phase onwards to predict the aerodynamic performance of aircraft designs; Finite Element Method (FEM) models are used in the aero-elastic analysis of the aircraft structure; heat transfer models are used in the design of propulsion systems and even the economic impact of a new aircraft design can be modelled with market penetration models [12]. Such models may be multiscale in nature. For instance, in the case of multiscale models of composite materials the scales may range from the microscale through to the component level [13]. The multiscale modelling of composite materials is a popular area of research [14, 15, 16, 17, 18]]. Computational fluid dynamics (CFD) models and heat transfer models may also be multiscale [19, 20].

In recent years the potential of Uncertainty Quantification (UQ) to accelerate the process of aircraft design and component certification has come to be recognised [21]. UQ is used to understand how parametric uncertainties affect output quantities of interest by propagating uncertainties through computational models [22]. To conduct UQ on multiscale models it is therefore necessary to develop computationally efficient means of propagating uncertainties through scales. Figure 2.1 shows a schematic of the stochastic upscaling problem: from fine-scale PDF data on coupons we need to forecast the properties of a complete part. The problem can also be seen as the inverse one, in which properties of the materials (as distributions) are needed to achieve a specific aeroplane design. There is no difference in the problem formulation, although in the case of the inverse formulation the solution attained is often not unique.

The aim of the framework presented in this chapter is to use stochastic model reduction, through probabilistic equivalence, to reduce the number of stochastic parameters at each scale as the multi-scale model moves from more fundamental scales to the level of a complete part, or even to the level of the entire aircraft. Reducing the number of

stochastic dimensions makes uncertainty quantification of the entire structure more computationally efficient. Whether the method discussed here is appropriate at every stage of this upscaling process will naturally depend on the nature of the model at each scale. However, there is some precedent in the literature for applying stochastic model upscaling to models of large composite structures. For instance, Sasikumar et al. (2015) argue that when the properties of each individual laminae in a composite plate is modelled as an independent random field the number of stochastic parameters becomes prohibitive. In this case, stochastic model reduction through probabilistic equivalence is achieved in two methods: on the nodal response and on the probabilistic characteristics of the matrix [23]. Gorguluarslan and Choi (2014) demonstrate that stochastic model order reduction can be implemented at the level of a complete part: in this particular case by up-scaling from a fine-scale, mesostructured model of a hydrogen tank to one where it is treated as homogeneous [24]. This test case is analogous to wing structural design in aeronautics, where both finite element and analytical, lumped mass models may be used to model a wing.

In other works in the literature, the problem of stochastic upscaling in multiscale models has commonly been studied from the perspective of modelling groundwater flow through heterogeneous porous media [25, 26, 27, 28, 29]. Achieving accurate predictions from these models is important in contaminant spread, nuclear waste disposal and oil recovery analyses. However, uncertainties exist due to the variation in permeability with length scale and also due to epistemic uncertainties arising from lack of information about the system [25]. Both intrusive and non-intrusive methods have been proposed for propagating uncertainty in stochastic multiscale models, such as the multiscale porous heterogeneous flow problem. Intrusive methods involve changing the underlying model in order to propagate uncertainty through scales and will be discussed first. Non-intrusive methods, which treat the model at each scale as a black box, will then be discussed.

Intrusive approaches to multiscale stochastic upscaling problems typically recast the underlying model equations in the form of stochastic partial differential equations (SPDEs).

Once in this form these equations may be solved through one of a number of established methods such as Monte Carlo simulation [30], spectral stochastic finite element method [31] or the stochastic collocation method. Stochastic collocation is a popular choice because of its fast convergence rates and has been applied frequently to flow problems in heterogeneous media [28]. By representing the random coefficients in the SPDEs through polynomial chaos expansions and the Karhunen–Loève expansion, it is possible to generate a set of sample points through Smolyak sparse grid collocation [25, 27] or through the probabilistic collocation technique [28]. The resulting deterministic equations may be solved through an appropriate strategy such as the multiscale finite element method [29, 32] or the multiscale discontinuous Galerkin method [33]. In multiscale finite elements, basis functions are generated at the coarse grid level which are consistent with the small scale subgrid structures [29]. The use of stochastic collocation coupled with multiscale finite elements is a common approach in the literature [27, 28, 34]. As an alternative to stochastic collocation, Dostert et al. (2008) [26] use Markov Chain Monte Carlo (MCMC) methods in the situation where the prior distribution of the stochastic equation coefficients is known. If the measurement error is assumed to follow a Gaussian distribution then a sampling target distribution may be constructed from which a Markov Chain may be generated.

Another intrusive technique, coarse graining, is used to find a low-fidelity model which may be matched to a complex, high-fidelity model. In future evaluations the low-fidelity model is used in the place of the high-fidelity model; Bilonis and Zabaras (2013) introduce a stochastic optimisation framework to find an effective coarse grained potential that matches the output of a high-fidelity model based on a number of Monte Carlo samples. Such a technique is reminiscent of the stochastic upscaling technique described here, although our technique is non-intrusive and requires fewer high-fidelity model realisations than would be the case when using Monte Carlo sampling, as used in Bilonis and Zabaras [35]. Lastly, a fuzzy stochastic global–local algorithm is proposed in Babuska and Motamed (2016) [36] that upscales uncertainty through a non-stationary fuzzy-stochastic

field, which is evaluated to find a quantity of interest at the coarse-scale. This method is currently limited to the case of upscaling in a one-dimensional fibre composite and is not readily generalisable. In this thesis non-intrusive methods, which do not change the underlying model, are preferred as they are compatible with models already used in industry.

Non-intrusive techniques for multiscale uncertainty propagation attempt to quantify the information loss between scales in a hierarchy of models. Early works sought to integrate tools for uncertainty quantification that were already available in the literature with multiscale problems. For instance, a non-intrusive stochastic solver for multiscale composites is outlined in Fish and Wu (2011) [37]. The Karhunen–Loève expansion is used to reduce the dimensionality of probability space and a stochastic collocation method, in combination with a massively parallel computer architecture, are employed to reduce the computational cost of the uncertainty analysis. More recent works attempt to quantify the information loss between scales and to model the cross scale dependencies within the model. For example a multiscale PCE method was proposed in Mehrez et al. (2018) [38] to model the dependencies of the outputs at a particular level in a hierarchical structure of models on the inputs at finer scales. A generalised hidden Markov model is employed in Wang (2011) [39] for the same purpose. A common non-intrusive stochastic technique is that of probabilistic equivalence, where at each interface between scales the coarse-scale inputs are searched for that will produce a statistically equivalent output to that of the fine-scale model [24, 40, 41]. The resulting optimisation problem is usually solved with a genetic algorithm. Choi et al. (2015) [41] employ a hybrid optimisation procedure with the genetic algorithm and sequential quadratic programming.

Recent works have sought to apply machine learning to the problem of upscaling uncertainty in multiscale problems in both intrusive and non-intrusive methodologies. For instance in Chan and Elsheikh (2018) [42] a neural network is used to generate the coarse-scale basis functions in multiscale finite elements in order to reduce the computational cost of repeated model evaluations. Multi-response Gaussian processes are used in Bostanabad

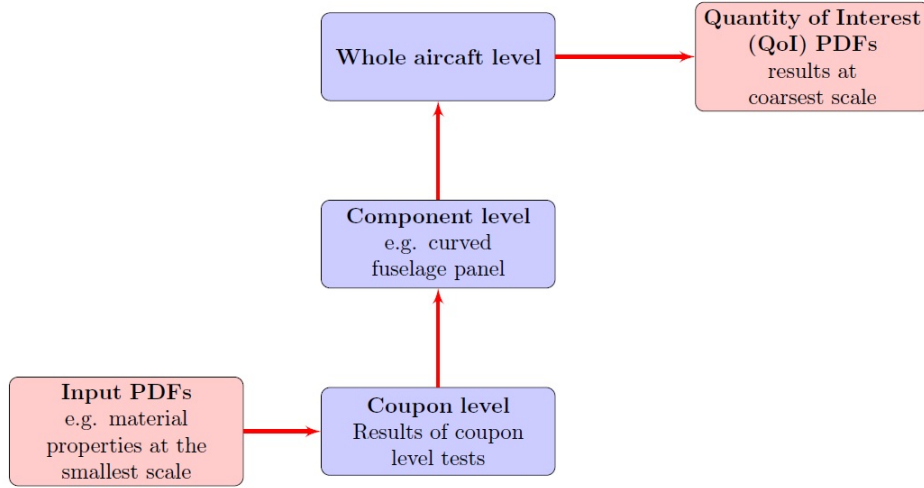


Figure 2.1: The stochastic upscaling problem — how to propagate uncertainties from the finest to the coarsest scales?

et al. (2018) [43] to model the aleatory and epistemic uncertainties arising in multiscale models of woven composites. Trehan and Durlofsky (2018) [44] introduced a Machine Learning framework to model the upscaling error between a high-fidelity model of a reservoir and a low-fidelity model. Simulations of the high-fidelity model are used for a high-dimensional regression to model the error based on user defined features. A random forest is the chosen statistical regression. Scheidt and Caers (2013) [45] similarly addressed the problem of modelling the upscaling error in reservoir models, this time through the use of kernel clustering.

Every method for uncertainty propagation in multiscale models must strike a balance between repeating high-fidelity simulations at the lowest scales in order to understand how uncertainties at the lowest scale impact the coarse-scale quantities of interest with the computational resources available. The method presented in this chapter builds upon the upscaling algorithm presented in Arnst and Ghanem (2008) [40] through the use of a moment base arbitrary Polynomial Chaos (aPC) formulation. The application of aPC to this algorithm, as opposed to using Monte Carlo sampling, greatly increases the computational efficiency by reducing the number of model evaluations needed for the upscaling at both the coarse and fine-scale.



Probabilistic equivalence for upscaling: search for coarse input distributions that give the same probabilistic distribution as the fine model

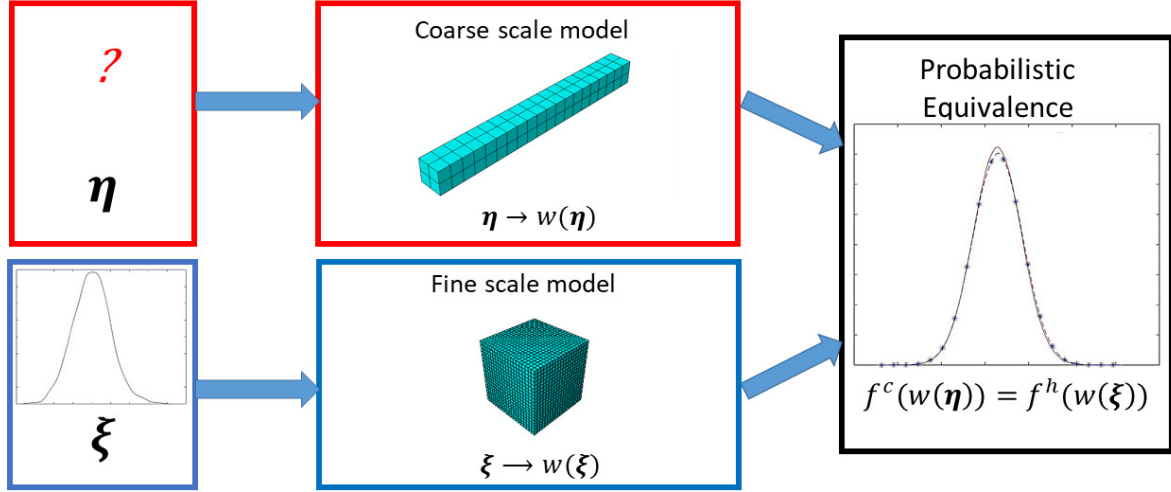


Figure 2.2: Illustration of the stochastic upscaling algorithm. A PCE for the coarse inputs,  $\eta$ , are searched for which, when propagated through the coarse model, give a statistically equal output to that of the fine model when known fine inputs,  $\xi$ , are propagated

## 2.2 Algorithm for stochastic model reduction with probabilistic equivalence

### 2.2.1 Fine and coarse-scale probabilistic models

The uncertainty affecting a computational model may be thought of as belonging to one of two categories: model uncertainty and parametric (or aleatoric) uncertainty. Model uncertainties refer to uncertainties arising from the truncation of the infinite-dimensional real world system that the model has been created to represent [46]. These uncertainties are difficult to quantify but will not be considered here; this chapter will focus on the propagation of parametric uncertainties between scales in multiscale models. Parametric uncertainties refer to incomplete knowledge of constants, boundary conditions or initial conditions in the computational model. It is assumed that such parameters can be identified and the uncertainty represented through a PDF [47].

The upscaling procedure detailed here propagates uncertainty from a fine-scale model to a coarse-scale model at the next highest scale in a hierarchy of models that can be used to represent the whole aircraft structure. In so doing the impact of uncertainties at the most fundamental scales on the complete structure may be quantified. Note that it is also possible to reverse the order of the hierarchy of models for use in inverse problems: the coarse-scale model can be used to explore the design space rapidly and identify interesting features which may be modelled in detail using the fine-scale model.

The fine-scale model occupies the probability space denoted by the triplet  $(\Omega^h, \mathcal{F}^h, \mathcal{P}^h)$  where  $\Omega^h$  denotes the sample space of the fine model,  $\mathcal{F}^h$  the set of events and  $\mathcal{P}^h$  the probabilities assigned to these events. A realisation  $\omega^h$  in the fine model sample space produces a fine random input vector  $\xi = \xi(\omega^h)$ . Evaluating the fine model, when the uncertain parameters take the values of the components of  $\xi$ , yields a fine-scale output  $y^h$ :

$$y^h = w(\xi), \quad (2.1)$$

where  $w$  represents the quantities of interest (QoIs) of the model. Through repeated model realisations at the fine-scale it is possible to generate a joint density for the output statistics of the fine model quantity of interest, denoted  $f^h(w)$ . Similarly, the coarse model occupies the probability space occupied by the triplet  $(\Omega^c, \mathcal{F}^c, \mathcal{P}^c)$  and a realisation  $\omega^c$  produces a coarse random input vector  $\eta = \eta(\omega^c)$ . Evaluating the coarse model with this input vector yields a coarse-scale output vector  $y^c$ , where:

$$y^c = w(\eta). \quad (2.2)$$

The joint density of the coarse-scale output is then denoted by  $f^c(\mathbf{w})$  [41]. The goal of the upscaling procedure is to find the input distributions at the coarse-scale such that the coarse model has a statistically equal output to the fine model.

### 2.2.2 Problem formulation

The problem of propagating uncertainty from one model to another model higher up in the hierarchy of scales is depicted in Figure 2.2. A stochastic upscaling algorithm is used to achieve this. To do so outputs common to both models are identified so that  $\mathbf{y}^h$  and  $\mathbf{y}^c$  may be defined. In order to achieve an abstraction from coupon level to component level, as depicted in Figure 2.1, it is necessary that there will always be some overlap in the model QoIs.

Given that the input distributions at the fine-scale are known, an arbitrary Polynomial Chaos formulation can be used to generate a sampling grid containing  $n_{sp}$  points, corresponding to the set of random input vectors,  $\Xi = [\boldsymbol{\xi}^{(1)}, \dots, \boldsymbol{\xi}^{(n_{sp})}]$ , at which the fine model is evaluated. A meta-model is constructed with a PCE from which the joint density  $f^h(\mathbf{w})$  for the variable(s) common to both the fine and coarse models may be generated with Monte Carlo sampling. In general, we restrict our attention to systems with a single output and from here onward denote the QoI as  $w \in \mathbb{R}^1$ . The stochastic upscaling algorithm then searches for the random input vector,  $\boldsymbol{\eta}$ , which when simulated  $s$  times and propagated with aPC through the coarse-scale model produces an output distribution  $f^c(w)$  that is statistically equivalent to that of the fine-scale model i.e.

$$\forall w : f^c(w(\boldsymbol{\eta})) = f^h(w(\boldsymbol{\xi})). \quad (2.3)$$

Achieving exact equivalence between the two probability distributions is a computationally intractable problem so in practice an input vector  $\boldsymbol{\eta}$  is sought which minimises the statistical distance between the two probability distributions i.e.

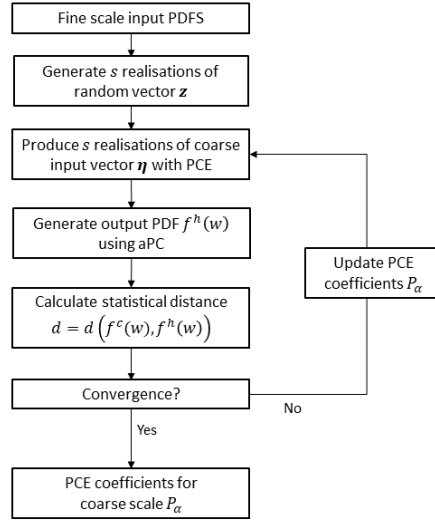


Figure 2.3: Flowchart of the stochastic upscaling algorithm

$$\boldsymbol{\eta} = \underset{\boldsymbol{\eta}}{\operatorname{argmin}} d(f^c(w(\boldsymbol{\eta})), f^h(w(\boldsymbol{\xi}))), \quad (2.4)$$

where  $d$  refers to the statistical distance, a metric that quantifies the difference between the two continuous probability distributions. Possible choices for the statistical distance are discussed below.  $\boldsymbol{\eta}$  may be approximated as a Polynomial Chaos Expansion (PCE). The mathematical background of PCEs is discussed in the next section, together with an explanation of how Non-Intrusive Polynomial Chaos (NIPC) formulations can be used in forwards uncertainty propagation.

### 2.2.3 Propagating uncertainty with Non-Intrusive Polynomial Chaos

Non-Intrusive Polynomial Chaos (NIPC) is a popular technique used widely in Uncertainty Quantification to propagate aleatoric input uncertainties through computational models by providing a spectral representation of uncertainty (see e.g. [48]). A meta-model

for the stochastic response of a model,  $w(\boldsymbol{\xi})$ , is constructed as a function of  $n_u$  independent stochastic input variables  $\boldsymbol{\xi} \in \mathbb{R}^{n_u}$ .  $w(\boldsymbol{\xi})$  is expanded into a basis of orthogonal polynomials to give a spectral representation of uncertainty:

$$w(\boldsymbol{\xi}) = \sum_{k=1}^P \lambda_k \Psi_k(\boldsymbol{\xi}), \quad (2.5)$$

where  $\Psi$  refers to a multivariate orthogonal polynomial, given as the product of a set of univariate, orthogonal polynomials  $\psi$ . The deterministic coefficients of the PCE are denoted  $\boldsymbol{\lambda} \in \Lambda \subseteq \mathbb{R}^P$ . A single realisation of the Polynomial Chaos Expansion (PCE) may be generated by sampling the components of the input random vector  $\boldsymbol{\xi}$  from probability distributions for the  $n_u$  uncertain system inputs, with joint density  $\rho(\boldsymbol{\xi})$ . This process is illustrated in Figure 2.4. For a PCE of order  $p$ , there are  $P = \frac{(n_u+p)!}{n_u!p!}$  linear combination

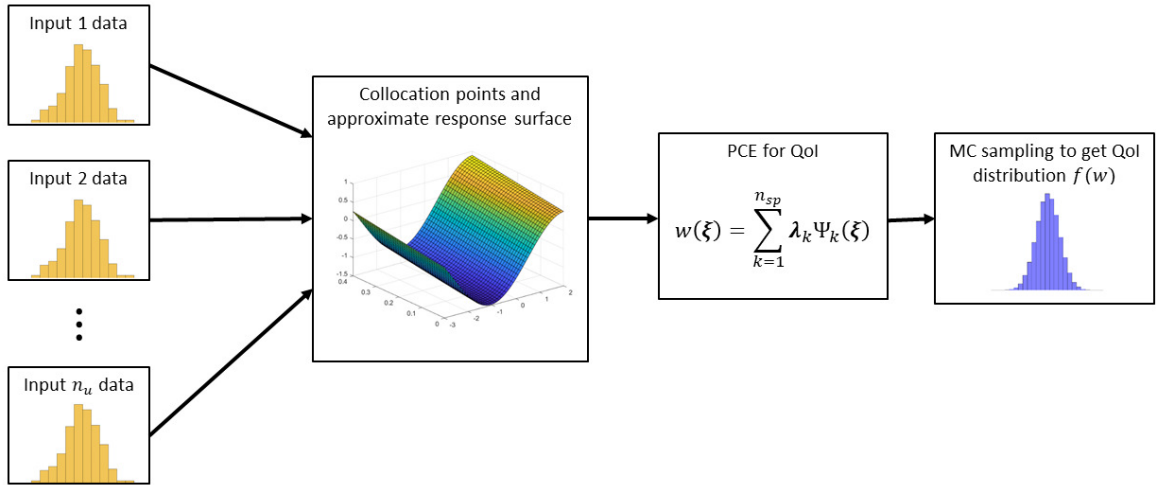


Figure 2.4: Schematic illustrating a forwards propagation of uncertainty with NIPC. Model evaluations at the collocation points are used to derive the coefficients of the PCE, which approximates the response surface of the model.

terms and the multivariate orthogonal polynomial may be calculated as:

$$\Psi_k(\boldsymbol{\xi}) = \prod_{i=1}^{n_u} \psi_{I_{k,i}}(\xi_i), \quad k = 1 \dots P, \quad (2.6)$$

where  $I \in \mathbb{R}^{P \times n_u}$  refers to the index matrix for the expansion. For instance, the case  $n_u = 2$  and  $p = 1$  will have index matrix:

$$I = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad (2.7)$$

with the second row yielding the product  $\Psi_2(\boldsymbol{\xi}) = \psi_1(\boldsymbol{\xi}_1)\psi_0(\boldsymbol{\xi}_2)$ . Note that as a product of orthogonal polynomials, and due to the statistical independence of the inputs,  $\Psi_k$  is itself orthogonal. There are optimal choices for the univariate orthogonal polynomials that correspond to the probability distributions from which the stochastic input variables are drawn; Xiu and Karniadakis (2002) proved that members of the Askey scheme of orthonormal polynomials are optimal for certain types of stochastic processes such as the uniform, normal, and exponential processes [49]. The Askey scheme is discussed in more detail below. Alternatively, the orthogonal polynomials may be derived directly from the input data. Arbitrary Polynomial Chaos (aPC) uses the statistical moments of the input data to calculate the PCE. This is a useful formulation for engineering purposes as it makes it possible to propagate scarce input data without needing to make assumptions as to which family of probability distribution the data belongs [50]. The aPC method used here calculates the orthogonal polynomial for each input by first forming the Hankel matrix of moments:

$$M = \begin{bmatrix} \mu_0 & \mu_1 & \cdots & \mu_p \\ \mu_1 & \mu_2 & & \mu_{p+1} \\ \vdots & & \ddots & \\ \mu_p & \mu_{p+1} & \cdots & \mu_{2p+1} \end{bmatrix} \quad (2.8)$$

where  $\mu_i$  represents the  $i^{\text{th}}$  statistical moment of the input data. Computing the Cholesky decomposition  $M = R^T R$  and calculating the inverse of  $R$  produces an upper triangular matrix, the elements of which form an orthogonal system of polynomials according to the Mysovskih theorem [51]. Using the analytical formulas of Rutishauser (1963) a three term recurrence can be found for these polynomials in terms of the matrix elements  $R_{ij}$ :

$$\psi_{j-1}(\xi) = b_{j-1}\psi_{j-2}(\xi) + a_j\psi_{j-1}(\xi) + b_j\psi_j(\xi), \quad (2.9)$$

where:

$$a_j = \frac{R_{j,j+1}}{R_{j,j}} - \frac{R_{j-1,j}}{R_{j-1,j-1}}, \quad b_j = \frac{R_{j+1,j+1}}{R_{j,j}}, \quad (2.10)$$

and  $R_{0,0} = 1$  and  $R_{0,1} = 0$  [52]. The roots of the polynomial of order  $p$  may then be found by finding the eigenvalues of the symmetric, tri-diagonal Jacobi matrix [53, 54]:

$$J = \begin{bmatrix} a_1 & b_1 & & & \\ b_1 & a_2 & b_2 & & \\ & b_2 & a_3 & b_3 & \\ & & \ddots & \ddots & \ddots \\ & & & b_{p-2} & a_{p-1} & b_{p-1} \\ & & & & b_{p-1} & a_p \end{bmatrix}. \quad (2.11)$$

Having determined the orthogonal polynomials, the coefficients of the PCE are found non-intrusively using evaluations of the computational model. These coefficients may be found through the calculation of the integral

$$\lambda_k = \frac{1}{\langle \Psi_k^2 \rangle} \int_{\xi \in \Omega} w(\xi) \Psi_k(\xi) \rho(\xi) d\xi, \quad (2.12)$$

using either Galerkin projection, collocation or numerical integration [55]. In the above integral  $\rho(\boldsymbol{\xi}) = \prod_{i=1}^{n_u} \rho_i(\boldsymbol{\xi}_i)$  represents the tensor product form of the joint density for the uncertain inputs, while the denominator, the squared norm of the multivariate orthogonal polynomial, is defined as the product of the square norms of the univariate orthogonal polynomials. As noted in Eldred (2009) there are simple, closed form expressions for the univariate inner products if the univariate polynomials belong to the Askey scheme [56, 57]. Alternatively, the coefficients may be found through a linear regression; the computational model is evaluated at each of  $n_{sp}$  sample points, allowing the coefficients to be found through:

$$A\hat{\boldsymbol{\lambda}} = \mathbf{Y}, \quad (2.13)$$

where  $\mathbf{Y} = [w(\boldsymbol{\xi}^{(1)}), w(\boldsymbol{\xi}^{(2)}), \dots, w(\boldsymbol{\xi}^{(n_{sp})})]^T$  is a vector of function evaluations,  $\hat{\boldsymbol{\lambda}} = [\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \dots, \boldsymbol{\lambda}_P]^T$ , i.e. the numerical estimates of the PCE coefficients, and the matrix  $A \in \mathbb{R}^{n_{sp} \times P}$  is defined as:

$$A = \begin{bmatrix} \Psi_1(\boldsymbol{\xi}^{(1)}) & \dots & \Psi_P(\boldsymbol{\xi}^{(1)}) \\ \vdots & \ddots & \vdots \\ \Psi_1(\boldsymbol{\xi}^{(n_{sp})}) & \dots & \Psi_P(\boldsymbol{\xi}^{(n_{sp})}) \end{bmatrix}. \quad (2.14)$$

Note that collocation based approaches exploit the orthonormal properties of the multivariate polynomials in their formulation, which is not the case in linear regression. The cardinality of  $P$  is such that the number of PCE coefficients grows rapidly with  $n_u$ , the so-called ‘curse of dimensionality’. One strategy to mitigate this cost is to use Smolyak’s algorithm to generate a sparse grid on which to sample  $w$  [53]. Other methods to exploit sparsity in high-dimensional spaces include: least angle regression [58], the alternating Least squares scheme [59], compressive sampling [60, 61, 62], and variational inference [63].



In the case of the arbitrary Polynomial Chaos formulation used here the coefficients of the PCE are found by direct numerical approximation. The eigenvalues of the Jacobi matrix in (2.11) give the optimal Gaussian collocation points, associated with each uncertain input, while the first component of each eigenvector can be used to calculate the weight,  $\omega_i$  associated with the  $i^{\text{th}}$  eigenvalue through:

$$\omega_i = v_{i,1}^2, \quad (2.15)$$

where  $v_{i,1}$  refers to the 1<sup>st</sup> element of the  $i^{\text{th}}$  eigenvector of  $J$ . A sequence of one-dimensional Gaussian quadrature rules can be derived from these collocation points and weights:

$$U_{i_j} = \sum_{k=1}^{m_{i_j}} w(\xi_{i_j}^{(k)}) \omega_{i_j}^{(k)}, \quad (2.16)$$

where  $m_{i_j}$   $j \in \{1, \dots, n_u\}$  is the maximum adaptive order for each quadrature rule  $U_{i_j}$  and  $w(\xi_{i_j}^{(k)})$  refers to a model evaluation at the collocation point  $\xi_{i_j}^{(k)}$ . Smolyak's algorithm is then applied to find a sparse quadrature,  $A_s$  containing  $n_{sp}$  points on which the model is evaluated:

$$A_s = \sum_{l+1 \leq |i| \leq l+n_u} (l-1)^{l+n_u-|i|} \binom{n_u}{l+n_u-|i|} \otimes_{k=1}^{n_u} U_{i_k}. \quad (2.17)$$

The parameter  $l$  refers to the level of the Smolyak grid. Increasing the level of the grid increases the accuracy of the PCE by adding more points to the grid, which also increases the computational cost by requiring more model evaluations. Figure 2.5 shows the one-dimensional collocation points associated with a continuous Gaussian distribution and lognormal histogram and the corresponding level 3 Smolyak grid. Having evaluated the

computational model at each of the  $n_{sp}$  sample points in the Smolyak quadrature, the coefficients of the PCE may be numerically approximated through:

$$\lambda_k = \frac{\sum_{i=1}^{n_{sp}} w(\boldsymbol{\xi}^{(i)}) \Psi_k(\boldsymbol{\xi}^{(i)}) \boldsymbol{\theta}_i}{\sum_{i=1}^{n_{sp}} \Psi_k(\boldsymbol{\xi}^{(i)}) \boldsymbol{\theta}_i}, \quad (2.18)$$

where  $\boldsymbol{\xi}^{(i)}$  and  $\boldsymbol{\theta}_i$  represent the sparse grid collocation points and weights. More details on the aPC formulation used may be found in Ahlfeld et al (2016) [53].

Once a PCE is obtained as a meta-model for  $w$ , Monte Carlo simulation can be used to generate a dataset of outputs by drawing  $n_{MC}$  samples from the  $n_u$  input distributions to create a dataset  $W = [w(\hat{\boldsymbol{\xi}}^{(1)}), w(\hat{\boldsymbol{\xi}}^{(2)}), \dots, w(\hat{\boldsymbol{\xi}}^{(n_{MC})})]$ . A PDF for the QoI,  $f(w)$ , may then be estimated using a Kernel estimation method to fit a kernel function to the dataset:

$$f(w) = \frac{1}{n_{MC} \times h} \sum_{i=1}^{n_{MC}} K\left(\frac{w - W_i}{h}\right). \quad (2.19)$$

Possible choices for the kernel function  $K(\cdot)$  are discussed in Scott and Silverman [64, 65]. Gaussian kernel functions are used here, which is a common choice of kernel function [66]. It is widely regarded that the estimation of the bandwidth,  $h$ , is far more crucial than the choice of kernel function [67] and as such a number of methods for performing bandwidth estimation are available (for more details see the review by Jones et al. [68]). We employ a strategy for bandwidth estimation based on cross validation using the Kullback-Leibler divergence [69].

So far this section has described how NIPC may be used in general to propagate input uncertainties through a computational model of a system, with discussion of the main steps of the aPC formulation used (see Appendix A1 for an example of an application of aPC). In the algorithm for multiscale uncertainty quantification presented here Polynomial Chaos is used not only to propagate input uncertainties through a coarse-scale model of the system, but also to represent the uncertain inputs at the coarse-scale. A PCE represents these missing inputs, denoted by  $\boldsymbol{\eta}$ , and is a function of the random

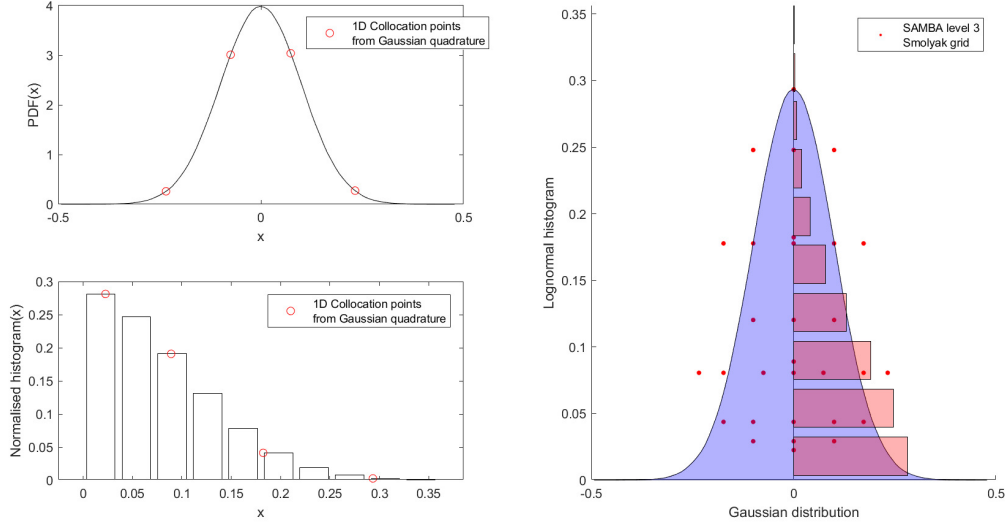


Figure 2.5: One-dimensional collocation points for a Gaussian PDF and a lognormal histogram (left). On the right, a level 3 Smolyak grid produced using aPC

Askey scheme	Interval	Orthogonal	Generating probability distribution
distribution		polynomial	
Uniform	$[a, b]$	Legendre	$U(-1, 1)$
Gaussian	$[-\infty, \infty]$	Hermite	$N(0, 1)$
Gamma	$[0, \infty]$	Laguerre	$\Gamma(\alpha, 1), \alpha > 0$

Table 2.1: Common types of parametric distributions and their corresponding univariate orthogonal polynomials in the Askey scheme

vector,  $\mathbf{z} \in \mathbb{R}^{n_m}$ , where  $n_m$  is the number of missing coarse-scale inputs:

$$\boldsymbol{\eta}(\mathbf{z}) = \sum_{\alpha, |\alpha|=0}^P P_{\alpha} \Psi_{\alpha}(\mathbf{z}), \quad (2.20)$$

where in this instance  $P = \frac{(p+n_m)!}{p!n_m!}$ . The components of  $\mathbf{z}$  are generated from probability distributions that are dependent on the orthonormal basis used. The right hand column of Table 2.1 lists the generating distributions for a selection of probability distributions in the Askey scheme.  $P_{\alpha} \in \mathbb{R}^{n_m \times P}$ , the parameter set of the PCE, is the target of the stochastic upscaling algorithm.

### 2.2.4 Metrics for the statistical distance

For the upscaling procedure to work it is necessary to have a measure of similarity between two probability distributions. As can be seen in Figure 2.3 the evaluation of the statistical distance between the two distributions  $f^h(w)$  and  $f^c(w)$  is necessary in order to determine the convergence of the stochastic upscaling algorithm. The way in which the statistical distance is defined has a significant impact on the speed of convergence and the quality of the matching. Possible choices of the statistical distance include the Kolmogorov-Smirnov (KS) distance and generalised method of moments. Note that neither of these distances (or the Kullback Leibler Divergence discussed in the next chapter) define a metric space.

#### 2.2.4.1 Kolmogorov-Smirnov distance

The Kolmogorov–Smirnov (KS) distance is defined as:

$$d^{KS} = \sup_w |F^c(w) - F^h(w)|, \quad (2.21)$$

where  $F^c(w)$  and  $F^h(w)$  are the Cumulative Distribution Functions (CDFs) of the common variable for the coarse-scale and fine-scale respectively [70]. Note that the CDF is defined as the integral of its PDF i.e. for an arbitrary PDF,  $f^c(w)$ , the CDF,  $F^c(w)$ , is given by:

$$F^c(w) = \int_{-\infty}^w f^c(t) dt. \quad (2.22)$$

Thus, the CDF may be thought of as an alternative representation of the PDF. The KS distance between two arbitrary CDFs is illustrated in Figure 2.6. The ability to attach a normalised significance level to the stochastic upscaling makes this definition of the

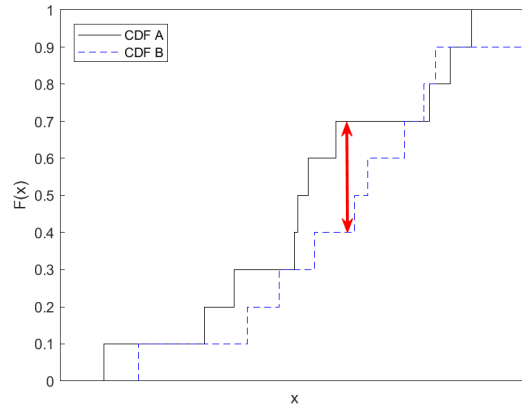


Figure 2.6: CDFs of two arbitrary distributions — the KS distance (arrow in red) is defined as the supremum distance between the two CDFs.

statistical distance appealing. The KS distance also allows matching of higher-order statistical moments, although this may come at the expense of fitting the first two statistical moments. However, the KS distance is only a meaningful distance metric if there is some overlap between the PDFs. This point is illustrated in Figure 2.7, in which four datasets have been created from sampling different Gaussian distributions: while the CDF of B clearly lies further away from the target distribution than that of C, because neither distribution overlaps with the target CDF the KS distance for both is 1. It can therefore be inappropriate to use the KS distance as a measure of statistical distance if there is uncertainty on the upper and lower bounds of the parameter set  $P_\alpha$ . By tabulating the statistical distances between the distributions A, B, and C (Table 2.2) it is clear that the method of moments provides a more appropriate distance measure for when the target and trial distributions are not overlapped. However, the KS distance gives a more meaningful statistic once an overlap between the distributions is established as the statistic can be related to a confidence level; this particular aspect is expanded upon in the next chapter when we discuss a formulation for data fusion between simulations and experimental data.

#### 2.2.4.2 Method of Moments

The statistical distance, quantified by the Method of Moments, is defined in general as:

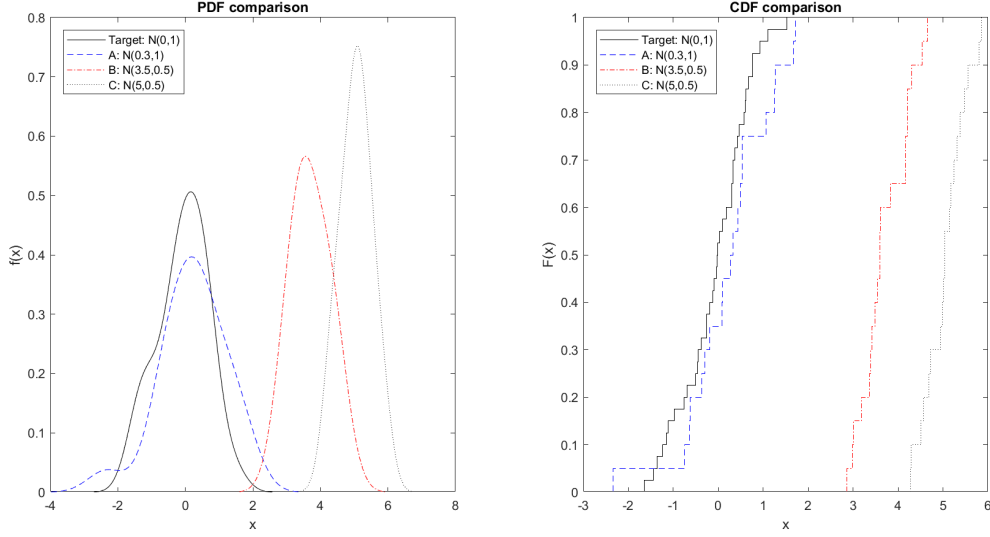


Figure 2.7: PDFs (left) and CDFs (right) of four datasets, sampled from Gaussian distributions. The CDFs of B and C do not overlap with that of the target distribution, hence both have a KS distance of 1.

Metric	$d(\text{Target}, A)$	$d(\text{Target}, B)$	$d(\text{Target}, C)$
KS distance	0.2	1	1
Method of moments ( $\gamma = 1$ )	0.2165	14.34	26.24

Table 2.2: A comparison of the statistical distances of the distributions A, B, and C from Figure 2.7 from the target distribution for different measures of statistical distance.

$$d^{MM}(f^c(\mathbf{w}), f^h(\mathbf{w})) = \|\mathbf{m} - \tilde{\mathbf{m}}\|_2^2 + \gamma \|C - \tilde{C}\|_2^2, \quad (2.23)$$

where  $\gamma$  is a user defined weighting parameter to weight the importance of the first statistical moment against the second [40]. The vectors  $\mathbf{m}$  and  $\tilde{\mathbf{m}}$  contain the mean values of the output quantity of interest, whilst the matrices  $C$  and  $\tilde{C}$  represent the covariance matrices of the fine-scale and coarse-scale quantities of interest respectively.

Defining the statistical distance in this way means that the fitting of higher order statistical moments is sacrificed in favour of fitting the first two moments. Whether this yields satisfactory results will depend on the form of the PDF being matched: for highly skewed distributions it may be more appropriate to use a weighted sum of KS distances, as opposed to the method of moments. On the other hand, matching the first two statistical

moments would be considered acceptable for Gaussian processes or variables. In the case of a single QoI, i.e.  $\mathbf{w} \in \mathbb{R}^1$  this statistical distance is equivalent to the weighted sum of the first two statistical moments.

#### 2.2.4.3 Selection of optimisation strategy

The manner in which the PCE coefficients are updated will depend on the choice of optimisation algorithm. The framework is flexible in the sense that it admits the use of any single or multiple-objective optimisation strategy, however, the problem is non-convex due to the nature of the statistical distance- a real number that quantifies the difference between two continuous functions. For this reason, gradient free optimisation methods are preferred such as the Nelder-Mead method [71], simulated annealing [72], particle swarm optimisation [73] or genetic optimisation methods [74]. If there are multiple QoI then it may be preferable to use a multi-objective optimisation strategy, rather than combining the statistical distances into a single metric. In this work an interior point algorithm from the MATLAB Optimization toolbox was used to perform a constrained local optimisation. The algorithm was started from multiple points in the PCE parameter space to minimise the risk of converging to a local minima. More details of the interior point algorithm may be found in Byrd et al (1999) [75] and Waltz et al (2006) [76].

## 2.3 Stochastic upscaling with a single common QoI

The capacity of the stochastic upscaling algorithm to propagate uncertainty from a fine-scale model to a coarse-scale model of reduced stochastic dimension is demonstrated for the case of a wing box. The wing box is a useful test case in aeronautics and has been used as a benchmark test in a number of publications such as Arnst and Ghanem (2008) and Riccio et al. (2013) [40, 77]. The particular example of propagating uncertainty from a wing box model to a Bernoulli beam to demonstrate multiscale uncertainty propagation was inspired by Arnst and Ghanem (2008) [40]. Only a single QoI, the first natural

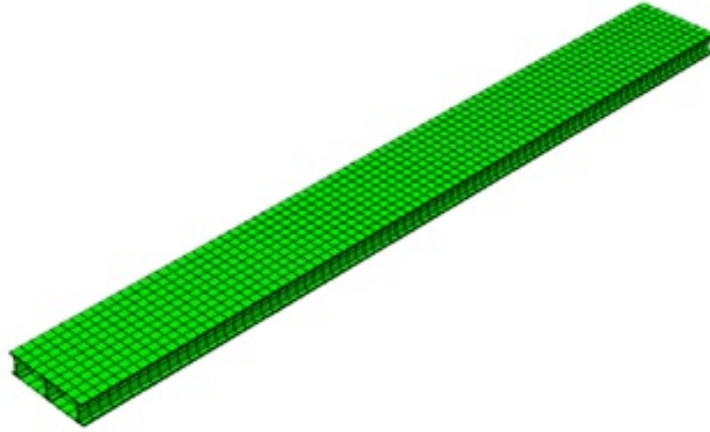


Figure 2.8: Wing box, with mesh, created in Abaqus

frequency, is common to both models. At the more fundamental fine-scale a wing box model was created in Abaqus [78] and a vibration analysis performed to find the first natural frequency of the beam. At the coarse-scale, the wing box was considered to be part of a larger structure where it was instead modelled as a simple cantilever beam using the Bernoulli beam equation. Figure 2.8 shows the wing box that was created in Abaqus with length 8m, width 0.9m, and thickness 0.22m. Figure 2.9 shows a cross section of the wing box: the material properties of each of the 6 strips joining the plates to the ‘I’ beams were characterised as random input variables. Considering the Young’s modulus and Poisson ratio of each strip to be random variables resulted in the fine model being of stochastic dimension 12. In the case of the Bernoulli beam model, the density and Young’s modulus of the entire beam were considered to be random material properties, hence the coarse-scale model was of stochastic dimension 2. Having propagated known fine-scale PDFs using aPC through the wing box model to create a PDF of values for the first natural frequency of the wing box, the stochastic upscaling algorithm was used to find PDFs for the material properties of the Bernoulli beam such that the PDF of its first natural frequency matched that of the wing box model.



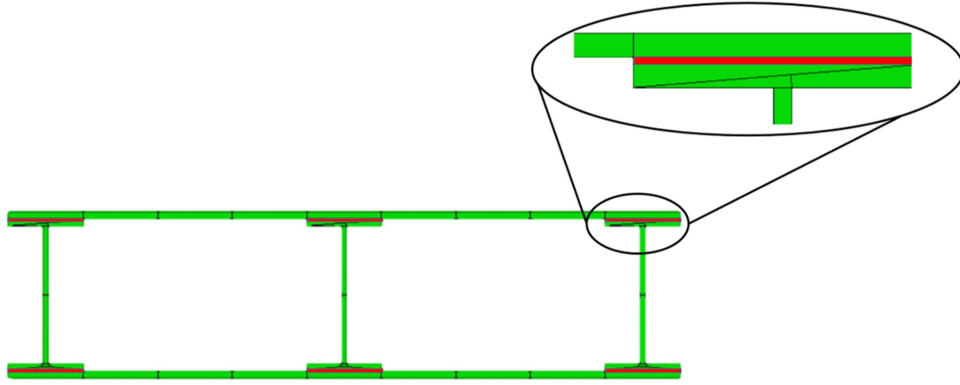


Figure 2.9: Cross sectional view of the meshed wing box. The material properties of the thin strips between the plates and I beams (coloured in red) are considered to be random, hence the model is of stochastic dimension 12

### 2.3.1 Stochastic upscaling with Gaussian input distributions

Figure 2.10 displays the probabilistic equivalence that was established for the QoI, the first natural frequency of vibration of the wing box. In this case the random material properties in the fine-scale model were sampled from two normal distributions:  $N(2.1 \times 10^8, 5 \times 10^7)$  for the Young's modulus and  $N(0.3, 0.01)$  for the Poisson's ratio of each strip. Given that the fine-scale material properties are sampled from normal distributions, Hermite polynomials were chosen as the orthonormal basis in the gPCE representations of the coarse-scale material properties. It was found that it was possible to achieve good probabilistic equivalence between the output PDFs using a first-order Wiener PCE: the KS distance between the two outputs was 0.0163, significantly lower than the target of 0.03, which we used as a threshold KS distance for probabilistic equivalence. This demonstrates that it is possible to achieve stochastic upscaling using a coarse model of reduced stochastic dimension for input PDFs belonging to the Askey scheme.

### 2.3.2 Stochastic upscaling for non-Askey scheme input distributions

The stochastic upscaling algorithm was also demonstrated for non-Askey scheme input distributions. Non-Askey, heavy tailed distributions such as the Cauchy distribution are

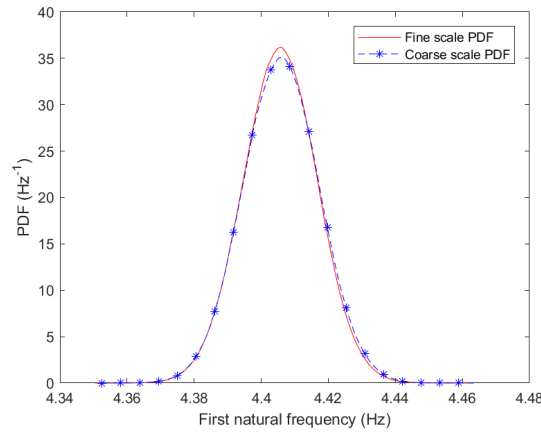


Figure 2.10: Stochastic upscaling for the case of Gaussian input distributions using a first-order PCE ( $d^{KS} = 0.0163$ )

useful in rare events simulation and so important for designing reliable components [79]. Stochastic upscaling of the first natural frequency was repeated for the wing box case study but with non-Askey scheme input distributions. Figure 2.11 shows the result of the stochastic upscaling when the Young's moduli of the 6 strips were selected from a Cauchy distribution with parameters  $C(0.1 \times 10^8, 2.1 \times 10^8)$  and the Poisson's ratios from a stable distribution with parameters  $S(1, 1, 0.008, 0.3)$ . In this case, a second-order Wiener PCE was used in order to achieve satisfactory stochastic upscaling. The KS distance, 0.0268, is below the target of 0.03 for stochastic upscaling. The example case of a wing box modelled in Abaqus serves as a proof of concept for the stochastic upscaling algorithm using aPC. The ability to achieve stochastic upscaling for both Askey and non-Askey scheme input PDFs implies that probabilistic equivalence should be achievable for any reasonable input PDF.

### 2.3.3 Choice of univariate orthogonal polynomials

As has been discussed previously, the choice of which orthonormal polynomials to use as the basis for the PCE can have an impact on the quality of the results. Using Hermite polynomials in the previous section was the natural choice given that the fine-scale inputs to the wing box were normally distributed. However, for cases where the fine-scale inputs

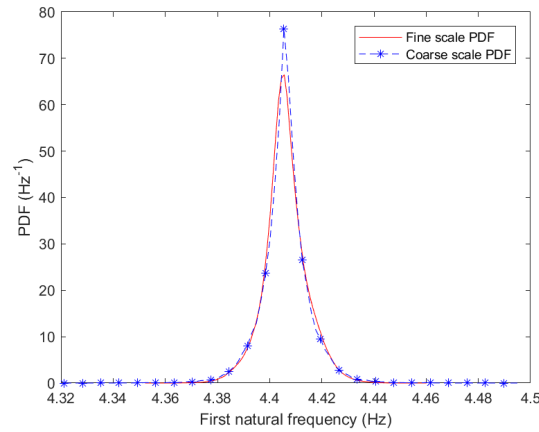


Figure 2.11: Stochastic upscaling with non-Askey scheme input distributions using a second-order PCE ( $d^{KS} = 0.0268$ )

do not follow distributions belonging to the Askey scheme, as is the case in the previous section, the choice of which univariate orthogonal polynomial to use as a basis can become significant and can affect the convergence of the results. This point is demonstrated in Figure 2.12: the same non-Askey scheme inputs to the wing box model are used as in the previous section and the results of the stochastic upscaling between the coarse-scale and fine-scale PDFs are plotted against the gPCE order,  $p$ , for different choices of univariate orthogonal polynomials. The orthonormal bases used are those in Table 2.1. The KS distance is again used to quantify the stochastic upscaling. For the probability distributions in Section 2.3.2 it was found that the choice of Laguerre polynomials led to faster convergence. Such differences in speed of convergence may become significant if computational resources are limited: a judicious choice of univariate orthogonal polynomial can reduce CPU time by limiting the number of coefficients which must be found through optimisation to achieve a satisfactory stochastic upscaling.

## Summary

In summary, a non-intrusive framework for the propagation of uncertainties through multiscale models has been presented here. The method is applicable to field analyses where multi-scale phenomena are of interest, for instance in magnetic and acoustic analyses.

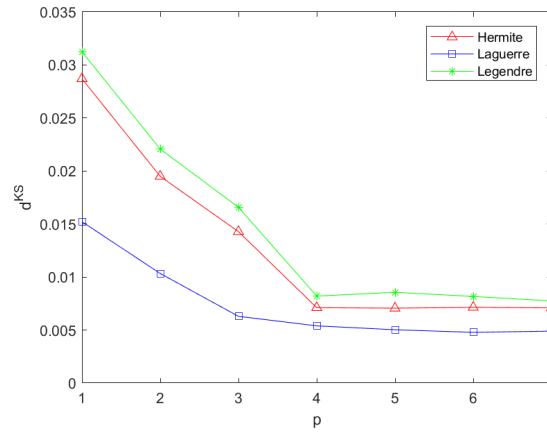


Figure 2.12: Plots comparing the convergence of the statistical distance, as quantified by the Kolmogorov–Smirnov distance, for different orthonormal polynomial bases

The method has been able to achieve satisfactory stochastic upscaling for both Askey and non-Askey scheme input distributions. Achieving a good matching for thick tailed distributions such as the Cauchy distribution is significant as these distributions are used often in accident prediction. As has been discussed above, future works on stochastic model upscaling should focus on propagating uncertainty through more than just two scales in case studies that are more complicated.

Thus far, works on stochastic model upscaling have focused on a two-scale problem: a quantity of interest common to both the fine and coarse-scale models is identified and probabilistic equivalence is achieved for this quantity of interest. In order to achieve the full component roll-up illustrated in Figure 2.1 it is necessary to propagate uncertainty through multiple scales and upscaling through probabilistic equivalence may be used more than once in the course of the roll-up. However, the quantity for which the probabilistic equivalence is found will not be the quantity that is passed up to coarser scales and certainly will not be the quantity of interest of the model at the coarsest scale. A thorough error analysis of a realistic problem involving multiple scales is necessary in order to determine that employing probabilistic equivalence at several stages of the roll-up of scales does not introduce significant model errors at the coarsest scale. An error analysis such as this will be the subject of future research on this topic.

# Inverse Identification and Data Fusion: Leveraging Simulations with Experimental Data

## Contents

---

<b>3.1 Back-calculation of missing input probability distributions</b>	<b>54</b>
3.1.1 Validation with scroll compressor test case . . . . .	59
3.1.2 Carbon fibre coupons test case . . . . .	65
<b>3.2 Data Fusion of NIPC and scarce experimental data</b>	<b>82</b>
3.2.1 Algorithm for data fusion of Polynomial Chaos with experimental results . . . . .	86
3.2.2 p-boxes as a constraint on the optimisation . . . . .	88
3.2.3 Carbon fibre composite test case . . . . .	90

---

In this chapter, two methods are introduced for leveraging the results of computational models with scarce experimental data. To relate these methods to the chronology of product development, depicted in Figure 1.2, these methods are intended for use after

the conceptual phase, when prototyping and limited experimental tests begin. The first of the two methods discussed here adapts the algorithm for multiscale uncertainty propagation, introduced in the previous chapter, for inverse identification problems. Inverse identification problems are a form of inverse problem in which the material properties of a structure are derived from experimental measurements of its response. The second method discussed here is used to fuse the results of forward propagations of uncertainty using NIPC with scarce experimental data.

### 3.1 Back-calculation of missing input probability distributions

Inverse problems occur frequently in science and mathematics, for instance in geophysics [80, 81], computer vision [82], and astronomy [83]. In turbomachinery, a design can be formulated as an inverse problem, where the required blade design is often specified and a blade shape that would accomplish a given performance is sought (see, e.g. [84, 85, 86, 87]). In contrast, the algorithm introduced here uses an inverse framework to leverage the data from the digital twin of a system to enrich and analyze scarce or missing data. As Hadamard established [88], inverse problems are in general ill-posed with non-unique solutions, making them challenging to solve. Inverse problems represent a wide chapter of applied mechanics (see, e.g. [89]), with techniques such as reverse Monte Carlo developed to solve such problems [90]. More recently, there has been an interest in developing inverse methods that are capable of handling missing data. Such problems occur frequently in geophysical exploration problems, where radar and sonar technology is used to deduce subsurface properties based on the boundary of a scattered wave field. For instance, Liu et al (2018) [91] proposes a joint inversion and interpolation problem, where missing sensor data is interpolated and refined as model estimates improve. An Adaptive Eigenspace

Inversion algorithm is proposed in Grote et al (2017) [92] and shown to be more tolerant to missing data than the standard nodal representation of the inverse density.

Inverse identification problems are a subset of inverse problems in which the material properties of a structure are deduced based upon its response. In general, an inverse identification method consists of three components: experimental observations of a set of quantities of interest (QoI); a computational model of the experiment; and an algorithm for identifying values for the missing inputs that, when propagated through the computational model, yield values for the QoI that are sufficiently similar to those obtained through experimental testing. The inverse identification problem is typically cast as an optimisation problem; inputs that minimise the discrepancy between the experimental measurements of the QoI and the outputs of a computational model of the material are sought.

Inverse identification methods are commonly used to deduce the material properties of composite materials. The problem is also relevant to tubomachinery, where many parameters that are important for life expectancy or performance are not measured directly. For instance, the Turbine Entry Temperature is not directly measured in gas turbines and is often inferred from downstream temperature measurements [93]. Previous works in which the material characteristics of composite structures have been found through inverse identification have used a variety of optimisation algorithms such as the Levenberg-Marquardt algorithm [94]; Sequential Quadratic Programming (SQP) [95]; Genetic algorithms [96, 97]; and other evolutionary algorithms [98]. A recent survey of the inverse identification literature found that the popularity of gradient-based optimisation methods has been in decline relative to non-gradient based methods such as genetic algorithms. This is likely due to the inefficiency of gradient-based algorithms in handling the highly nonlinear and typically non-convex nature of the inverse identification problem [99].

Finite Element Model Updating (FEMU) is a popular inverse identification method in the literature, particularly for determining the material characteristics of composite materials (see, e.g. [94, 100, 101]). A Finite Element model of the system being studied

is created, the inputs of which are adjusted until the model outputs match the experimentally measured outputs. Extensions of FEMU such as Regularized Model Updating have also been proposed [102]. Another popular method, the Virtual Fields Method, a non-iterative approach proposed by Grédiac [103] in which a set of equations are built from full field experimental measurements and the Principle of Virtual work, has also been widely used [104, 105, 106]. For a more complete review of the existing literature of identification methods see Avril et al. [107]. A common trait of these works is that they assume that a full set of measurements is available for each coupon. Processing a batch of coupons would thus require the solution of an optimisation problem for each coupon, which would involve an inordinate amount of physical testing. More realistically, coupons in a batch are divided into groups and a single kind of material characterization test is performed on each group. The result of such testing is that the measured material properties are no longer tied to the experimental responses as the tests have been carried out on separate coupons. It is not possible to calculate the missing material properties on a coupon-by-coupon basis, instead the material properties and responses of the composite are treated as realisations of a random process, described with independent probability distributions. The framework introduced here can be used to solve inverse identification problems for incomplete datasets where specimens have not been fully tested.

As has been discussed, the inverse identification algorithm introduced here to back-calculate the probability distributions for missing uncertain inputs has been adapted from the algorithm for multiscale uncertainty propagation based on probabilistic equivalence [40, 108]. While not necessarily considering an application to inverse uncertainty propagations, other works have introduced methods for determining a Polynomial Chaos representation of a random field from a set of experimental observations [109, 110].

The objective of the inverse identification algorithm is to back-calculate probability distributions for  $n_m$  missing uncertain inputs in a system of stochastic dimension  $n_u$ . These  $n_m$  missing probability distributions are generated by producing  $s$  realisations of a PCE that represents the aleatoric uncertainty in the missing inputs. In other words, a random



input vector for the system is defined as:

$$\boldsymbol{\xi} = [\phi_1^m, \phi_2^m, \dots, \phi_{n_m}^m, \xi_{n_m+1}, \dots, \xi_{n_u}], \quad (3.1)$$

where elements 1 to  $n_m$  of the random input vector are a single realisation of a PCE with parameter set  $P_\alpha$ :

$$\phi^m = \sum_{\alpha, |\alpha|=0}^P P_\alpha \Psi_\alpha(\mathbf{z}). \quad (3.2)$$

Elements  $m + 1$  to  $n_u$  of  $\boldsymbol{\xi}$  are found by sampling probability distributions based on the available experimental measurements of the other inputs. Experimental measurements of the system outputs are used to estimate a probability distribution for the QoI,  $f^{exp}(w)$ . Coefficients of the PCE are found such that when the uncertainty in the inputs are propagated through a computational model, the statistical distance,  $d$ , between the subsequent probability distribution for the QoI and the existing probability distribution based on experimental measurements is minimised. This problem may be expressed as an optimisation problem for the set of PCE coefficients,  $P_\alpha$ :

$$P_\alpha = \arg \min_{P_\alpha} d(f^{exp}(w), f(w(\boldsymbol{\xi}))). \quad (3.3)$$

where  $f(w(\boldsymbol{\xi}))$  is an estimate of the QoI probability distribution using aPC. The steps of the back-calculation algorithm may be summarised as follows:

1. Initialise a set of PCE coefficients,  $P_\alpha$ , and generate  $s$  random vectors  $\mathbf{z}_s$
2. Generate a set of  $s$  PCE realisations of  $\phi^m$
3. Propagate the input uncertainties with aPC to generate a probability distribution for the QoI  $f(w(\boldsymbol{\xi}))$
4. Evaluate the statistical distance,  $d$ , between  $f(w(\boldsymbol{\xi}))$  and the experimental probability distribution of the QoI  $f^{exp}(w)$

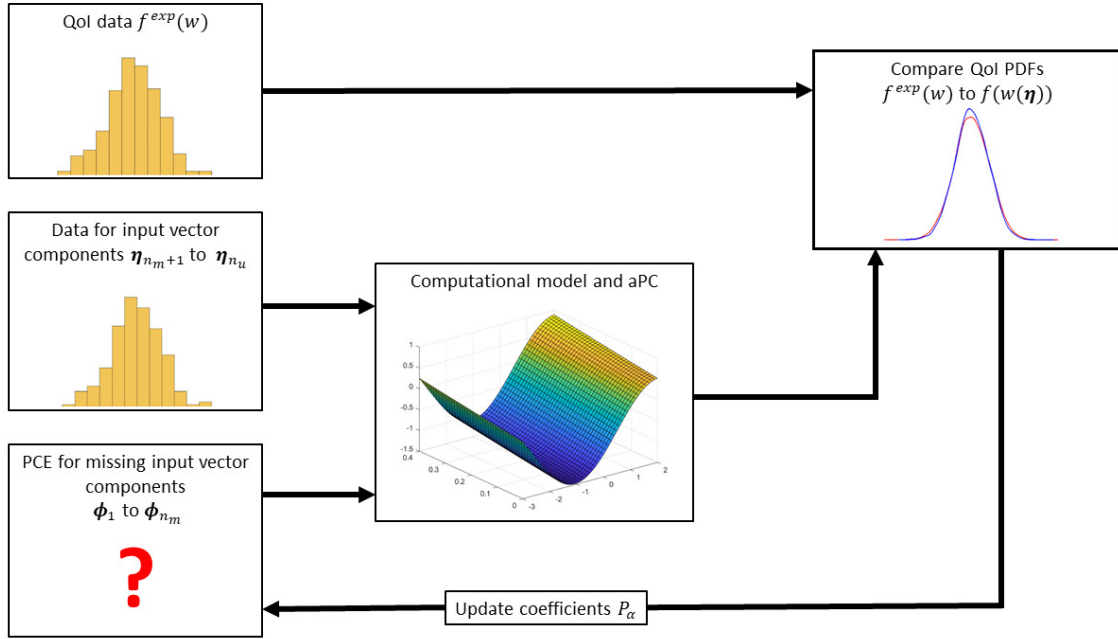


Figure 3.1: An illustration of the back-calculation framework. Arbitrary Polynomial Chaos is used to propagate uncertainty in the input parameters through a computational model. An optimisation algorithm is used to update the coefficients of the PCE such that, when the uncertainties are propagated through the model, the statistical distance between experimental and computational PDFs for the QoI is reduced

5. EXIT if  $d$  has converged, otherwise update coefficients  $P_{\alpha}$  (depending on the optimisation algorithm used) and return to step 2

The optimisation strategy discussed in the previous chapter was reused for this algorithm and the KS distance was again used to calculate the statistical distance between the two probability distributions for the QoI. aPC is used to propagate the uncertainties in the inputs parameters through a computational model of the experiment and generate a polynomial chaos expansion for the QoI. As has been discussed, aPC requires fewer model evaluations than Monte Carlo sampling, so is computationally more efficient. Even in cases where a surrogate model already exists for the system we believe it is still preferable to use aPC to propagate input uncertainties as it can process scarce data efficiently (see [111]). The algorithm is represented in diagrammatic form in Figure 3.1.

The method was applied to two test cases: in the first test case the method was validated by demonstrating that an input distribution could be exactly reconstructed if there is no model error. A synthetic dataset and a surrogate model derived from CFD simulations of a scroll compressor were used to accomplish this. In the second test case, missing probability distributions representing the material properties of a set of carbon fibre coupons were estimated using a dataset of real experimental data.

### 3.1.1 Validation with scroll compressor test case

The back-calculation framework introduced in this paper is generalised and may be applied to any problem in which missing input distributions are back-calculated for small, incomplete datasets. In this section we validate the framework using a scroll compressor as a test case. Scroll compressors are a positive displacement machine that have been used as compressors in refrigeration systems and heat pumps [112, 113]. They are a useful test case for the framework as variations in geometry of the scroll compressors have a significant effect on the efficiency ( $\eta$ ) of the scroll compressor [114], however, the parameters defining the geometry can be difficult to measure experimentally [115].

Recently there has been interest in using Computational Fluid Dynamics (CFD) to analyze the performance of scroll compressors, the performance of which varies according to the shape of the spirals. CFD simulations using a moving mesh, which were tuned to experimental results in the literature, were used to create a response surface for the efficiency of the scroll compressor ( $\eta$ ) as a function of three geometric parameters: the scroll wrap thickness ( $t$ ), the height of the scroll ( $h_{scroll}$ ), and the orbiting radius ( $R_{or}$ ). The midline of the scroll is defined as a circular arc which intersects with a circular involute. The geometries of the fixed and orbiting scrolls are found by displacing a distance of  $R_{or}/2$  above and below the midline, as can be seen in Figure 3.2. In this work the equidistant curve approach from Wang et al (2018) [116] was adopted to define the geometry of the midline.

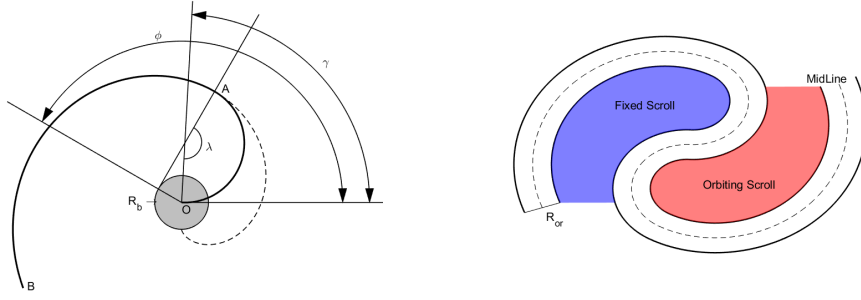


Figure 3.2: An illustration of the midline of the scroll compressor (left) and the fixed and orbiting scrolls (right)

The back-calculation algorithm was validated by finding a distribution for the scroll height based on a synthetic dataset. As is discussed in Arai et al (2018) [115] the scroll height is a difficult and time consuming parameter to measure experimentally, so there is motivation in using a numerical method to estimate a distribution based on experimental measurements of other parameters.

Values of the geometrical parameters of the scroll were taken from Wang et al and the CFD simulations were verified by comparison to the experimental data in this work. Figure 3.3 shows the results of 45 CFD simulations of the scroll compressor. The ten most efficient designs of the 45 simulated are highlighted in red, indicating a clustering of high efficiency designs for compressors with relatively low scroll wrap thicknesses and relatively high orbit radii. A polynomial response surface was fitted to these results, reducing the number of CFD simulations that needed to be performed for the back-calculation.

The aim of the test case is illustrated in Figure 3.4: a PCE is sought for the parameter  $h_{scroll}$ , which is difficult to measure experimentally, using measurements of the efficiency ( $\eta$ , the QoI of the CFD simulations) and measurements of the thickness and orbit radius. Synthetic datasets were used firstly to validate the framework with Gaussian probability distributions and then with multiple missing, non-Askey distributions. In all cases, it was possible to recover a distribution of  $h_{scroll}$  assuming that the computational model is perfectly calibrated to the experiments.

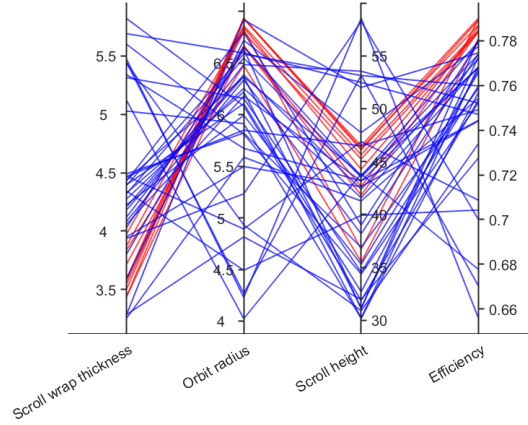


Figure 3.3: A parallel axis plot displaying the results of the CFD simulations. The scroll designs yielding the 10 highest efficiencies are highlighted in red. Lengths are in cm.

#### 3.1.1.1 Validation with Gaussian distributions

A synthetic dataset was used to validate the ability of the framework to recover missing input distributions. The dataset for the inputs of 200 scroll compressors was generated by sampling three Gaussian distributions. Data for  $t$  and  $R_{or}$  was generated using the distribution  $N(5.5, 0.25)$ , while data for  $h_{scroll}$  was generated from  $N(40, 2.5)$ . The response surface based on the results of CFD simulations was used to produce a distribution of the efficiencies of the scroll compressor, in other words the assumption is made that the computational model is perfectly calibrated to experiments for the purposes of validating the framework. Histograms of the dataset are illustrated in Figure 3.5. The framework was validated by back-calculating a distribution for  $h_{scroll}$  based on the distributions of  $\eta$ ,  $R_{or}$ , and  $t$ . A PCE of order  $p = 2$  was found for  $h_{scroll}$ , using Hermite polynomials as the orthonormal basis as this is the optimal choice for Gaussian inputs (see e.g., [108]). As can be seen from Figure 3.6 it is possible to find a PCE distribution which recovers the missing distribution in this instance. The framework can be considered validated in the sense that it can recover the missing distribution exactly when the computational model is perfectly calibrated.

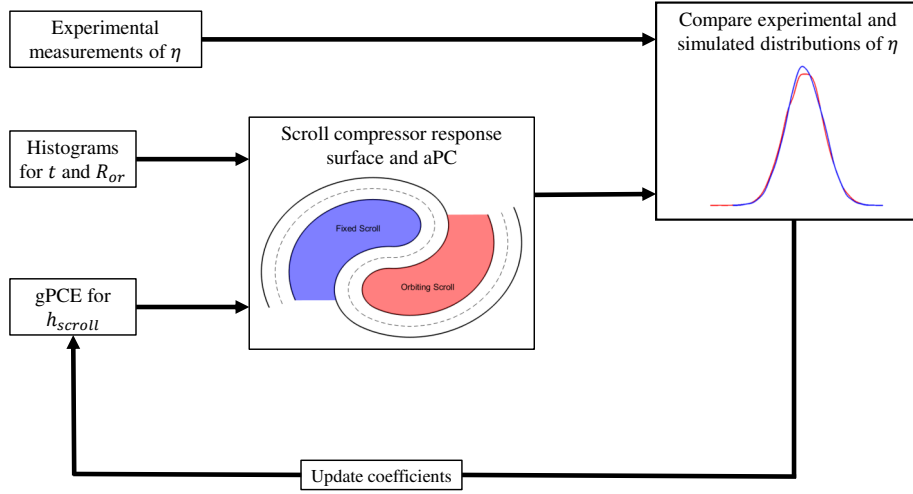


Figure 3.4: A summary of the back-calculation problem for the scroll compressor test case. A PCE is sought to find a distribution for  $h_{scroll}$ , based on a response surface and experimental measurements of  $t$ ,  $R_{or}$ , and  $\eta$ . CFD simulations of the scroll compressor were used to fit the response surface

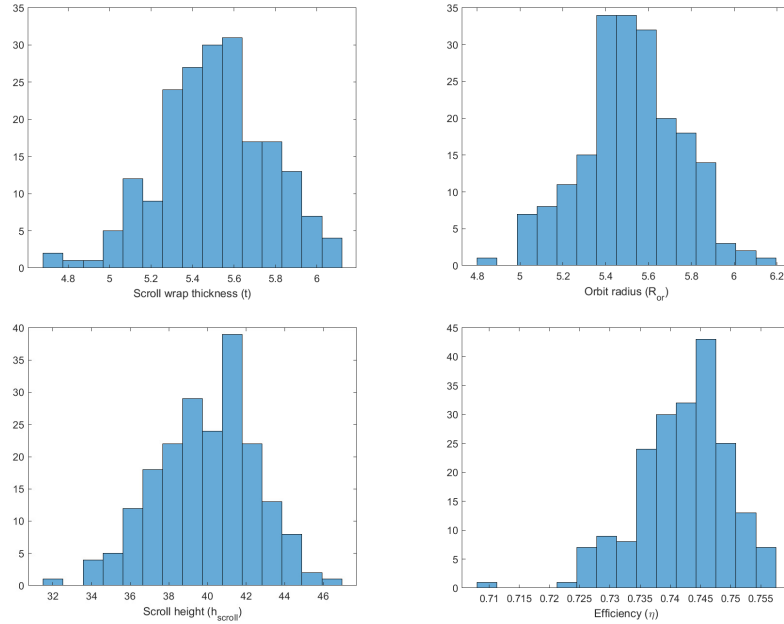


Figure 3.5: Histograms displaying the synthetic dataset used to validate the framework.  $t$ ,  $R_{or}$ , and  $h_{scroll}$  were sampled from the normal distributions  $N(5.5, 0.25)$ ,  $N(5.5, 0.25)$ , and  $N(40, 2.5)$  respectively

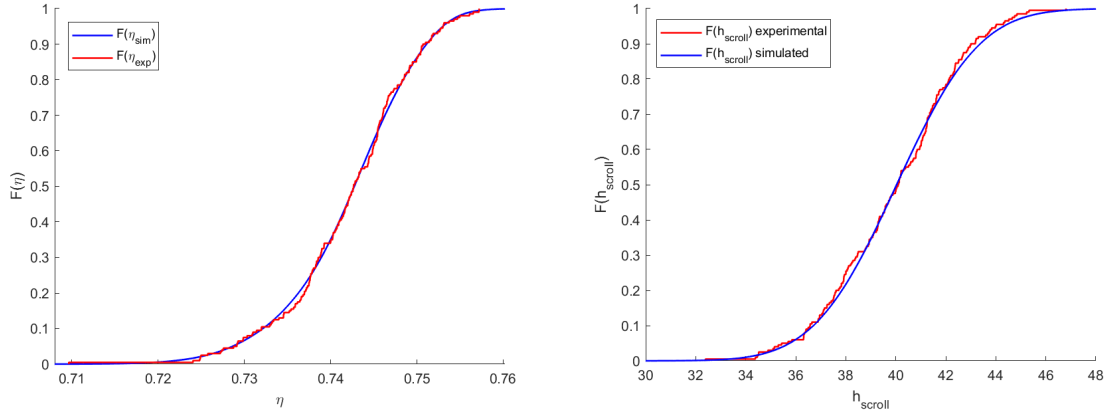


Figure 3.6: CDFs for the simulated and experimental distributions of  $\eta$  (left) and  $h_{\text{scroll}}$  (right) used to validate the framework. It is possible to recover a missing input distribution precisely provided the simulations used in the back-calculation are correctly calibrated

### 3.1.1.2 Back-calculation of missing input distributions with scarce data

Having demonstrated that the back-calculation framework can recover the results of a forward propagation of uncertainty, the capacity of the framework to efficiently handle scarce data was demonstrated. Data was removed at random from the synthetic dataset illustrated in Figure 3.5 such that only 60 measurements in total were available across all of the specimens and that only a single parameter was measured for a given specimen. This is in contrast to the data used for the validation, in which  $t$ ,  $R_{\text{or}}$ , and  $\eta$  were measured for each specimen, resulting in 600 measurements in total. As can be seen from Figure 3.7, it is still possible to find a PCE for  $h_{\text{scroll}}$  such that, when the input uncertainties are propagated using aPC, a reasonable approximation of the distribution for  $\eta$  can be found. Practically, there is no limit to the scarcity of the QoI data for which the statistical distance can be found as the Kolmogorov-Smirnov distance can still be employed to measure the supremum distance between empirical CDFs of the QoI distribution even if the experimental QoI dataset is extremely limited in size, for instance consisting of just a handful of measurements. However, in this case seeking an exact probabilistic equivalence between the QoI distributions may be placing a level of constraint on the PCE that is

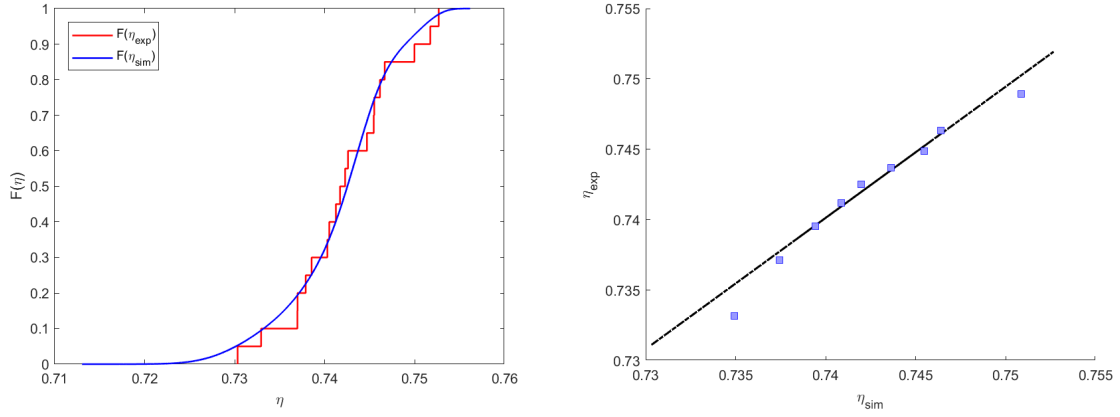


Figure 3.7: CDFs for  $\eta$  for the case of sparse data (left). A QQ plot for every tenth percentile of the experimental and simulated distributions of  $\eta$ . The black line indicates the line of equality, with the majority of percentiles lying on or near this line (right)

not warranted by the size of the dataset. In cases such as these it may be more appropriate to place a non-parametric confidence interval around the experimental CDF (for instance with the Dvoretzky-Kiefer-Wolfowitz inequality and Massart bound [117, 118]) and instead constrain the simulated QoI probability distribution to lie within that bound, rather than attempting to minimise the statistical distance between the two distributions directly.

### 3.1.1.3 Back-calculation of multiple missing, non-Gaussian input distributions

A second synthetic dataset was used to demonstrate the ability of the back-calculation framework to recover multiple missing, non-Gaussian input distributions. The non-Gaussian input distributions used are displayed in Figure 3.8: the scroll thickness was sampled from the uniform distribution  $U(1.3, 5.1)$ , the scroll radius from a Logistic distribution with parameters  $\mu = 5.5$  and  $\sigma = 0.1$ , and the scroll height from the Lognormal distribution with parameters  $\mu = 3.688$  and  $\sigma = 0.0632$ . Monte Carlo sampling of these distributions was used to generate a probability distribution for the scroll efficiency, a histogram of which may be seen in Figure 3.8. The goal of this test was to demonstrate



the generality of the framework by recovering the distributions of  $R_{or}$  and  $h_{scroll}$  based on probability distributions for  $t$  and  $\eta$  only.

The results of the back-calculation are displayed in Figure 3.9. As can be seen from the plots of the CDF of the scroll efficiency in the top left panel of the figure and the QQ plot in the top right, it is possible to find distributions of the missing inputs that, when propagated through the computational model using aPC, yield an almost exact match to the distribution for  $\eta$  displayed in Figure 3.8. CDFs of these input distributions are displayed in the centre left and bottom left panels of Figure 3.9, while kernel density estimates of the PDFs are displayed in the centre right and bottom right panels. As can be seen, the estimates of the input distributions found by Monte Carlo sampling of the PCE are an almost exact match to the distributions in the synthetic dataset. A PCE of order  $p = 2$  was used, which required 12 coefficients to be found by solving (3.3).

In this case Hermite polynomials were used as the basis function of the PCE- given that neither of the missing distributions are in the Askey scheme an optimal choice of orthogonal polynomial does not exist. The choice of polynomial basis does of course have an impact on the probabilistic equivalence, as has been shown in the previous chapter. A judicious choice of polynomial basis can improve the efficiency of the back-calculation as a lower order PCE can be used, which reduces the dimensions of the search space which is explored by the algorithm to find the optimal set of PCE coefficients. In practice the order of the PCE is increased until the statistical distance between the CDFs of the QoI converges.

### 3.1.2 Carbon fibre coupons test case

Having validated the inverse identification algorithm using the scroll compressor as a test case, we now apply the algorithm to a test case in which probability distributions for the material properties of a batch of carbon fibre coupons are back-calculated using a dataset of scarce experimental data. Inverse identification is a problem that frequently arises in the design of new composite materials. The manufacturing process of composite

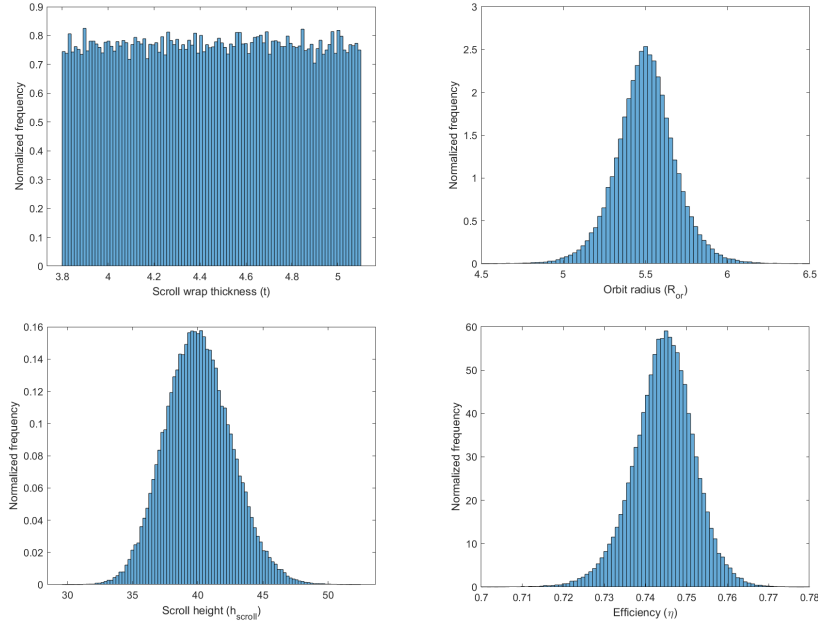


Figure 3.8: Histograms for the synthetic dataset with non-Gaussian input distributions.  $t$  was sampled from the uniform distribution  $U(1.3, 5.1)$ ,  $R_{or}$  from a logistic distribution with parameters  $\mu = 5.5$  and  $\sigma = 0.1$ , and  $h_{scroll}$  is sampled from a lognormal distribution with parameters  $\mu = 3.688$  and  $\sigma = 0.0632$

materials is highly complicated, resulting in significant variance in the material properties when measured in coupon tests [119]. These uncertainties must be quantified in order to produce confidence intervals for the material properties, which in practice requires extensive experimental testing. As an example, the process of certification for a new product in the aeronautical industry requires thousands of tests of individual carbon fibre composite coupons. Extensive testing of coupons can make the design process for a new composite material expensive and time consuming [120]. In addition, the measurement of shear properties of composite materials can be very challenging [121]. For instance, Melin et al. (2000) compared four methods of measuring the inter-laminar shear strength (ILSS) and found significant variation in the stress-strain responses in shear between the different experimental methods, concluding that the ILSS was significantly underestimated by many methods [122]. For these reasons, there has been interest in using computational simulations to reduce the burden of coupon testing, as well as in developing methods that

use computational models of a composite structure in order to deduce material properties from full field observations of the structures' behaviour.

In practice we note that probability distributions estimated from experimental data may well be multi-modal. Instead of modelling such distributions with a very high-order PCE it is instead possible to model multi-modal probability distributions as a mixture of low order PCEs. A consequence of this is that fewer coefficients need to be found numerically (see [123]), reducing the size of the space which must be searched by the optimisation algorithm to find the optimal set of coefficients. For this reason we extend the inverse identification framework by representing the  $n_m$  missing inputs with a multi-modal Polynomial Chaos Expansion (mmPCE). Following the formalism of Nouy (2010), a second random variable,  $z_2$ , is introduced which is independent of  $z_1$ . The probability space occupied by  $\boldsymbol{\eta}$  is now  $(\Omega^m, \mathcal{F}^m, \mathcal{P}^m)$  where  $\Omega^m = \Omega^1 \times \Omega^2$  and  $\mathcal{P}^m = \mathcal{P}^1 \otimes \mathcal{P}^2$  (the tensor product of the set of probabilities  $\mathcal{P}^1$  and  $\mathcal{P}^2$ ). A mixture of  $m$  PCEs can be made, each of which is weighted by an indicator function  $I$  such that the vector of missing inputs  $\boldsymbol{\eta}$  may be written as:

$$\boldsymbol{\eta}(\mathbf{z}) = \sum_{i=1}^m I_{\mathcal{F}_i^m}(z_2) \left[ \sum_{\alpha, \|\alpha\|=0}^P P_{\alpha}^{(i)} \Psi_{\alpha}(\mathbf{z}_1) \right] = \sum_{i=1}^m \sum_{\alpha, \|\alpha\|=0}^P I_{\mathcal{F}_i^m}(z_2) P_{\alpha}^{(i)} \Psi_{\alpha}(\mathbf{z}_1). \quad (3.4)$$

The set of events  $\mathcal{F}_i^m$  is partitioned by  $m$  intervals  $\Omega_i^m = [x_{i-1}, x_i]$ ,  $i = 1, \dots, m$  where  $0 = x_0 < x_1 < \dots < x_m = 1$ .  $z_2$  is a random variable, in this case sampled from the uniform distribution  $U(0, 1)$ . It is possible to define the sample space  $\Omega_i^m = z_2^{-1}(\mathcal{F}_i^m)$  for  $i = 1, \dots, m$ . This characterises the partition with  $x_i = \sum_{j=1}^i \mathcal{P}(\Omega_j^m)$  [123]. The expansion is thus defined by the parameter sets associated with each of the  $m$  PCEs,  $P_{\alpha}^{(i)}$ ,  $i = 1, \dots, m$ , and the partitions  $X = \{x_1, x_2, \dots, x_{m-1}\}$ , which are the targets of the optimisation.

The algorithm was tested using a dataset comprised of laboratory measurements of carbon fibre composites and a computational model based on the equations of classical

laminate theory. A brief explanation of carbon fibre composites and a description of the classical laminate theory is included below, followed by a description of the inverse identification test case and results.

### 3.1.2.1 Classical laminate theory equations and choice of failure criterion

A carbon fibre composite is constructed of layers of unidirectional carbon fibres held together by a resin, referred to as the matrix. A single layer is referred to as a lamina or ply. Note that there are two sets of co-ordinates in a composite,  $(x, y)$  co-ordinates refer to global co-ordinates of the laminate as a whole, while the local co-ordinates  $(1, 2)$  refer to directions parallel with the fibres and transverse to the fibres in each lamina. A rotation is needed to move from global to local co-ordinates, which will depend on the orientation of the fibres in the lamina. For more details on carbon fibre composites see, e.g. [124, 125, 126]. As has been discussed above, significant variance is observed in the material properties of composite materials when measured in coupon tests due to the complexity of the manufacturing process [127]. The stacking of individual, unidirectional plies to form a composite is illustrated in Figure 3.10. The figure also illustrates the directions in which the loads  $N_x$  and  $N_{xy}$  are applied. A rotation of  $\theta$  degrees must be applied to move from the global co-ordinates  $(x, y)$  to the local co-ordinates  $(1, 2)$ .

A deterministic model, based on the equations of classical laminate theory, was used as a computational model of the physical coupon tests. The coupons themselves consisted of 48 laminae in a  $(([90/0_2/45/-45/0]_4)_s)_s$  stacking sequence. The applied stresses  $N_x$  and  $N_{xy}$  were incremented in ratio until a failure criterion was met at any of the laminae:

1. Initialise  $N_x$  and  $N_{xy}$
2. Apply loads  $N_x$  and  $N_{xy}$  ( $N_x = 3.33N_{xy}$ )
3. Find the mid surface strains  $\epsilon_x, \epsilon_y$  and  $\gamma_{xy}$
4. Apply rotations to find the local strains in each ply  $(\epsilon_1, \epsilon_2, \gamma_{12})$

5. Evaluate the Yamada-Sun failure criterion for each ply (described below)
6. If the failure criterion is met at any of the plies then the laminate is considered to have met the first failure stress, otherwise increment the applied load and return to Step 2

The relationship between the mid surface strains and the local stresses in a given ply is represented using the reduced stiffness matrix  $Q$ . Through the plane stress assumption the full  $6 \times 6$  can be reduced to a  $3 \times 3$  matrix, the components of which are determined by the material properties of the laminate:

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \gamma_{12} \end{bmatrix}, \quad (3.5)$$

where  $\sigma_1$  is refers to tension/compression in the principal fibre direction,  $\sigma_2$  is the tension/compression in the transverse direction and  $\tau_{12}$  the shear stress. The elements of the reduced stiffness matrix are dependent on the material properties of the ply, determined by the following relations:

$$Q_{11} = \frac{E_1}{1 - \nu_{12}\nu_{21}}, \quad Q_{12} = \frac{\nu_{12}E_2}{1 - \nu_{12}\nu_{21}} = \frac{\nu_{21}E_1}{1 - \nu_{12}\nu_{21}}, \quad Q_{66} = G_{12}, \quad (3.6)$$

where  $E_1$  represents the Young's modulus in the fibre direction,  $E_2$  the Young's modulus in the transverse direction,  $G_{12}$  the shear modulus, and  $\nu_{12}$  Poisson's ratio. A linear strain distribution is assumed through the thickness of the laminate:

$$\begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \epsilon_x^0 \\ \epsilon_y^0 \\ \gamma_{xy}^0 \end{bmatrix} + z \begin{bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{bmatrix}, \quad (3.7)$$

where  $\epsilon_x^0$ ,  $\epsilon_y^0$ , and  $\gamma_{xy}^0$  represent the mid-surface strains and  $\kappa_x$ ,  $\kappa_y$ , and  $\kappa_{xy}$  the curvature in the laminate. The applied loads  $(N_x, N_y, N_{xy})$  and moments  $(M_x, M_y, M_{xy})$  are related to the mid-surface strains and curvatures through:

$$\begin{bmatrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{bmatrix} = \begin{bmatrix} [A] & [B] \\ [B] & [D] \end{bmatrix} \begin{bmatrix} \epsilon_x^0 \\ \epsilon_y^0 \\ \gamma_{xy}^0 \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{bmatrix}, \quad (3.8)$$

where the elements of the  $ABD$  matrices depend on a summation across all  $n$  plies in the laminate:

$$A_{ij} = \sum_{k=1}^n (\bar{Q}_{ij})_k (z_k - z_{k-1}), \quad B_{ij} = \sum_{k=1}^n \frac{1}{2} (\bar{Q}_{ij})_k (z_k^2 - z_{k-1}^2), \quad D_{ij} = \sum_{k=1}^n \frac{1}{3} (\bar{Q}_{ij})_k (z_k^3 - z_{k-1}^3), \quad (3.9)$$

where  $\bar{Q}_{ij}$  represents the  $ij^{\text{th}}$  element of the reduced stiffness matrix of the  $k^{\text{th}}$  ply when it has been rotated to lie in the  $(x,y)$  co-ordinate system. It is possible to greatly simplify equation (3.8) through symmetry considerations, for instance  $[B] = [D] = 0$  and  $A_{16} = A_{26} = 0$  due to the symmetrical layup of the laminate. In the test case only horizontal and shear loads are applied, meaning there is no curvature due to the applied moments. This allows a further simplification of equation (3.8):

$$\begin{bmatrix} N_x \\ N_y \\ N_{xy} \end{bmatrix} = \begin{bmatrix} N_x \\ 0 \\ N_{xy} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & 0 \\ A_{12} & A_{22} & 0 \\ 0 & 0 & A_{66} \end{bmatrix} \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{bmatrix} \quad (3.10)$$

The Yamada Sun failure criterion was used to determine whether failure had occurred in an individual lamina [128]. The equations were used to relate the applied load to the mid surface strains and in turn the local stresses in each ply. The laminate was considered to have reached first failure when the condition:

$$\left(\frac{\sigma_1}{f_1}\right)^2 + \left(\frac{\tau_{12}}{f_{12}}\right)^2 = 1, \quad (3.11)$$

was met in any ply. In (3.11)  $f_1$  and  $f_{12}$  represent the longitudinal and transverse strengths of the composite respectively. Note that each of these linear transformations is dependent on the material properties of the laminate.

### 3.1.2.2 Inverse identification test case

Experimental measurements were made of the longitudinal and transverse strengths of the composite ( $f_1$  and  $f_{12}$ ); the shear modulus ( $G_{12}$ ); the Young's modulus in the fibre direction ( $E_1$ ) and the compression and shear stress required to cause the first failure of the laminate ( $N_x$  and  $N_{xy}$ ). Histograms of the available experimental data are displayed in Figure 3.11; note that the data for  $N_x$  and  $N_{xy}$  is particularly scarce, consisting of only 6 measurements. The difficulty in accurately determining the shear strength of composite materials is well known (see, e.g. Odegard and Kumosa (2000) [121]) so there is motivation in replacing difficult experimental tests with a computational model that can capture the physics of experiment. By incorporating the model in the framework described here it is possible to back calculate PDFs for quantities that may be difficult or costly to measure, such as the shear properties, without having to directly measure them.

Arbitrary Polynomial Chaos was again used to propagate the input uncertainties through the classical laminate model. As the compression and shear stresses are increased in ratio, a single PCE can be used to generate both probability distributions. An advantage of using NIPC methods, such as aPC, to propagate uncertainty is that it is possible to use the coefficients of the PCE to conduct a sensitivity study. Sudret (2008) demonstrated that once a Polynomial Chaos representation of a model is available, the Sobol sensitivity indices (defined below) may be calculated analytically at almost no extra cost by post-processing the PCE coefficients [129]. For a system with  $N_u$  stochastic inputs there are  $2N_u - 1$  Sobol indices in total, with each Sobol index,  $S_{i_1, \dots, i_s}$  weighting the contribution of an input to the total variance,  $D$ :

$$S_{i_1, \dots, i_s} = \frac{D_{i_1, \dots, i_s}}{D}, \quad (3.12)$$

where  $D_{i_1, \dots, i_s}$  represents the partial variance and  $D$  is defined as the sum of the partial variances:

$$D = \text{var}(w(\boldsymbol{\xi})) = \sum_{i=1}^{N_u} D_i + \sum_{1 \leq i \leq j \leq N_u} D_{ij} + D_{1,2,\dots,N_u}, \quad (3.13)$$

$$D_{i_1, \dots, i_s} = \int_{\Omega^s} w_{i_1, \dots, i_s}^2(\xi_{i_1}, \dots, \xi_{i_s}) d\xi_{i_1} \dots d\xi_{i_s}, \quad 1 \leq i_1 < \dots < i_s \leq N_u$$

where the integral is performed over a  $s$ -dimensional space of input parameters. By exploiting the orthogonality of the multivariate PCE basis, it can be shown that each Sobol index may be approximated by the ratio of two sums over the PCE coefficients [130]:

$$S_k \approx \frac{\sum_{k \in I_k} \lambda_k^2 \langle \psi_k, \psi_k \rangle}{\sum_{i=1}^P \lambda_k^2 \langle \psi_k, \psi_k \rangle} \quad (3.14)$$



Material property	Sensitivity (%)
$E_1$	0.35
$f_1$	93.12
$G_{12}$	0.04
$f_{12}$	6.49

Table 3.1: Results of the Sobol sensitivity analysis using the experimental dataset. Given the limited sensitivity of the model to  $E_1$  and  $G_{12}$  these parameters were treated as constants for the purposes of back-calculating a distributions for  $f_1$  and  $f_{12}$

where the counting variable  $k$  sums up all rows of the multi-index matrix for which an index  $i_j$  is not zero and  $\boldsymbol{\lambda}_k$  are the PCE coefficients found using the aPC formulation. A Sobol sensitivity study was performed using the first-order Sobol indices, which quantify the independent contribution of each input parameter to the total variance. Higher-order Sobol indices account for the influence of combinations of parameters but are typically not calculated. Table 3.1 displays the results of the Sobol sensitivity study, which used input distributions from the first iteration of the back-calculation algorithm as inputs. The sensitivity of the failure stresses to each of the four material properties can be quantified as a percentage. As can be seen from the table, the classical laminate theory model has limited sensitivity to  $f_1$  and  $f_{12}$  compared to  $E_1$  and  $G_{12}$ . For this reason,  $E_1$  and  $G_{12}$  were subsequently treated as constants, allowing the stochastic dimension of the model to be reduced to 2. Reducing the maximum adaptive order to 2 made the back-calculation more computationally efficient as there were fewer points in the Smolyak sampling grid at each iteration (see (2.17)). In addition, the dimensionality of the search space for the mmPCE parameters is reduced.

Having established through the Sobol indices that the model had little sensitivity to  $E_1$  and  $G_{12}$  the algorithm was used to back-calculate distributions for  $f_1$  and  $f_{12}$ . The situation is illustrated graphically in Figures 3.12 and 3.13. Figure 3.12 illustrates how a carbon fibre laminate may be formed from a stack of unidirectional laminae. It also tabulates which parameters have experimental data associated with them, while Figure 3.13 illustrates the framework for the specific test case presented here. The QoI of the

experiment are the stresses  $N_x$  and  $N_{xy}$ , with the four stochastic material properties as the input parameters to the computational model.

The Kolmogorov-Smirnov distance was used as the metric of statistical distance between the experimental and computational distributions of  $N_x$  and  $N_{xy}$ . The use of the Kolmogorov-Smirnov distance makes the CDF a natural choice for visualising the distributions. In Figure 3.14, the CDFs of the experimental and computational distributions of  $N_{xy}$  are plotted, together with a QQ plot for every 10th quantile. Note that the compression stress and shear stress are in ratio ( $N_x = 3.33N_{xy}$ ) it is only necessary to plot  $N_{xy}$  for the purposes of comparison. The majority of the quantiles lie on the identity line, indicating that the distributions are close to being equivalent with one another. While the statistical distance between the two CDFs has been minimised through an optimisation procedure, the scarcity of the experimental data for  $N_{xy}$  is such that there are a range of solutions which would have a comparable statistical distance which may be deemed satisfactory for the purposes of probabilistic equivalence. If necessary, a bound could be placed around the experimental CDFs, indicating the range of distributions that would yield what would be considered an acceptable matching. An mmPCE consisting of a mixture of 4 PCEs of order  $p = 3$  was used. First-order PCEs were added to the mixture until adding additional PCEs no longer reduced the KS distance, at which point the orders of the PCEs in the mixture were increased. Table 3.2 displays the effect of increasing the order of the PCEs on the KS distance.

Finally, the back-calculated distributions for  $f_1$  and  $f_{12}$  are compared with the available experimental data in Figure 3.15. Given the scarce nature of the experimental data, quantifying the similarity between the calculated and experimental probability distributions is not trivial. One option is to represent the probability distributions graphically: in Figure 3.14 the CDFs of the computed and experimental distributions are plotted on the same axes for comparison, in addition to a set of QQ plots which allow us to compare the quantiles of the probability distributions directly. From these plots we can see that the distributions generated using the mmPCE are slightly less conservative than the ex-

PCE order, $p$	$d^{KS}$
1	0.0934
2	0.0921
3	0.0912
4	0.0915

Table 3.2: The effect of increasing the order of the individual PCEs in the mmPCE on the probabilistic equivalence in the distributions for  $N_{xy}$ , quantified by the KS distance

Material property	$\Delta\mu$ (%)	Wilcoxon rank sum test ( $p$ )
$N_{xy}$	0.39	0.9737
$f_1$	2.72	0.0507
$f_{12}$	0.56	0.6832

Table 3.3: A comparison of the computed and experimental probability distributions is not trivial due to the scarcity of the experimental data. Here, the percentage difference in the mean and the  $p$ -values from the Wilcoxon rank sum test are presented for each of the uncertain parameters

perimental results, but comparable nonetheless. The 10<sup>th</sup>-80<sup>th</sup> quantiles lie on or near the identity line, with the mmPCE yielding a more conservative value for the 90<sup>th</sup> quantile in both cases. Table 3.3 presents the results of a quantitative approach to assessing the similarity between the probability distributions. As can be seen from the second column of Table 3.3, the means of the distributions are relatively similar. The third column contains the  $p$ -values associated with the Wilcoxon rank sum test, which tests the null hypothesis that the computed and experimental probability distributions come from continuous probability distributions with equal medians. The null hypothesis cannot be rejected at the 5% significance level for every parameter, although the  $p$ -value for  $f_1$  is significantly lower than the other two. Nevertheless, these results are encouraging given the scarcity of the experimental data, containing only 6 measurements of  $N_x$  and  $N_{xy}$ , and a relatively simple computational model was employed to establish the probabilistic equivalence. The test case serves as a proof of concept, illustrating that the framework has the potential to yield reasonable estimates of missing probability distributions, even when the size of the dataset is very limited.

In this section we have introduced an algorithm for solving inverse identification problems based on probabilistic equivalence. The algorithm has been validated using two test cases that reflect real-world engineering problems. In the case of the test case involving carbon fibre coupons we demonstrate how mmPCEs may be used to efficiently handle multi-modal data. Having discussed how probability distributions may be inferred from scarce experimental distributions in an inverse formulation, we now examine the forwards propagation of uncertainty and introduce an algorithm that enables the estimate of a probability distribution based upon NIPC to be informed by scarce experimental measurements of a QoI.

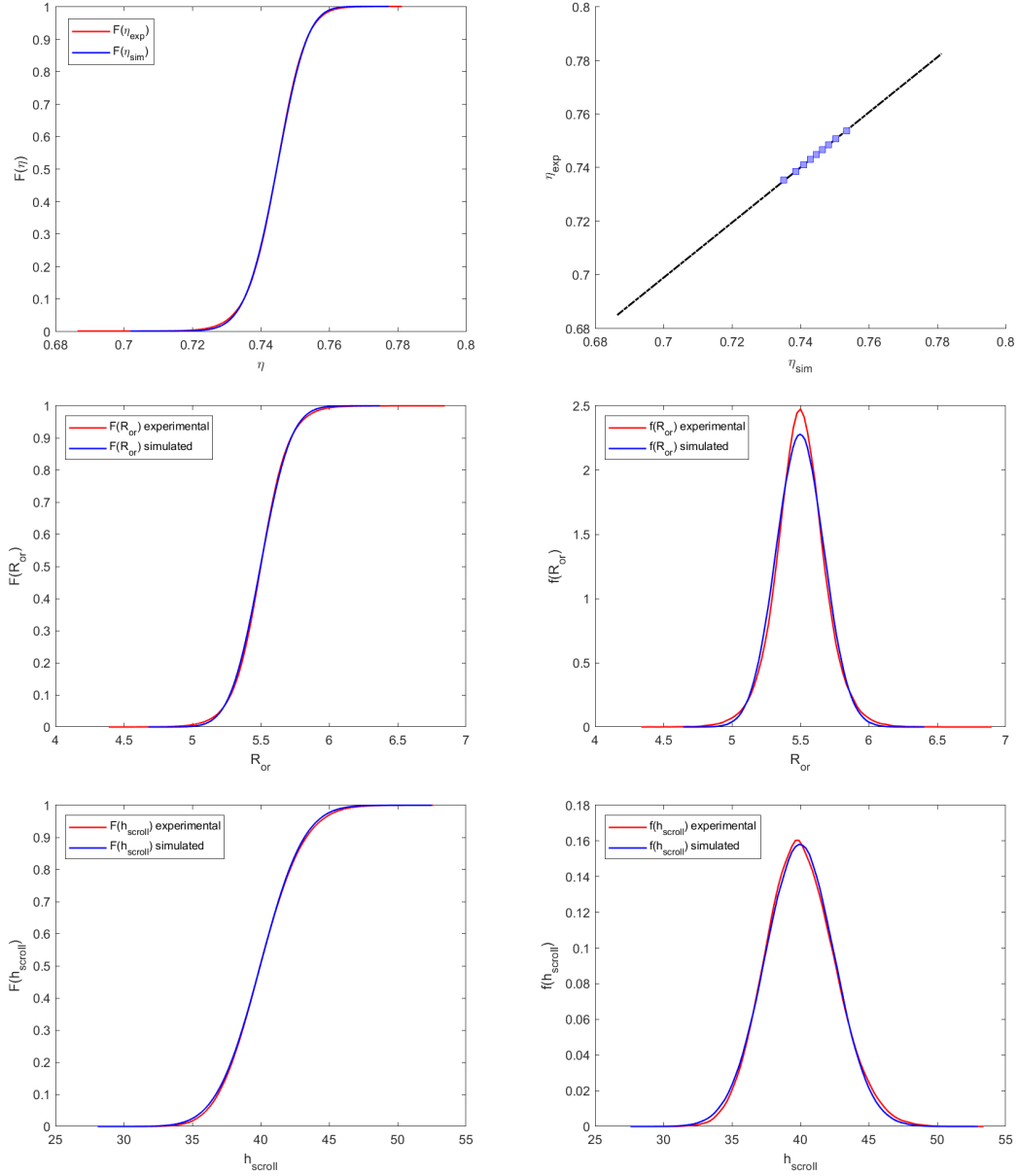


Figure 3.9: CDFs for the experimental and simulated distributions of  $\eta$  for the case of non-Gaussian input distributions (top left). The QQ plot of the same distributions indicates the almost exact equivalence between the two (top right). Similarly there is an almost exact equivalence between the distributions for the inputs  $R_{or}$  and  $h_{scroll}$ . This can be seen in the CDFs (centre left and bottom left) and in the kernel density estimates of the PDFs (centre right and bottom right)

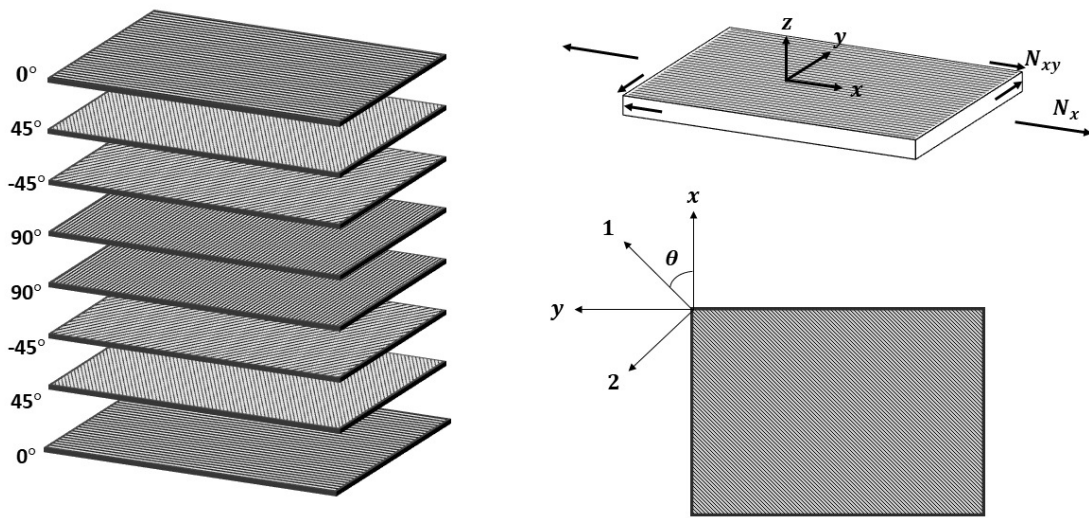


Figure 3.10: A schematic illustrating how a carbon fibre composite is constructed from layers of unidirectional plies (left). The directions of the global co-ordinates are indicated, together with the directions of the compression load  $N_x$  and shear load  $N_{xy}$  (top right). The angle  $\theta$  defines the rotation necessary to move from global co-ordinates  $(x, y)$  to the local co-ordinates of the ply  $(1, 2)$  (bottom right)

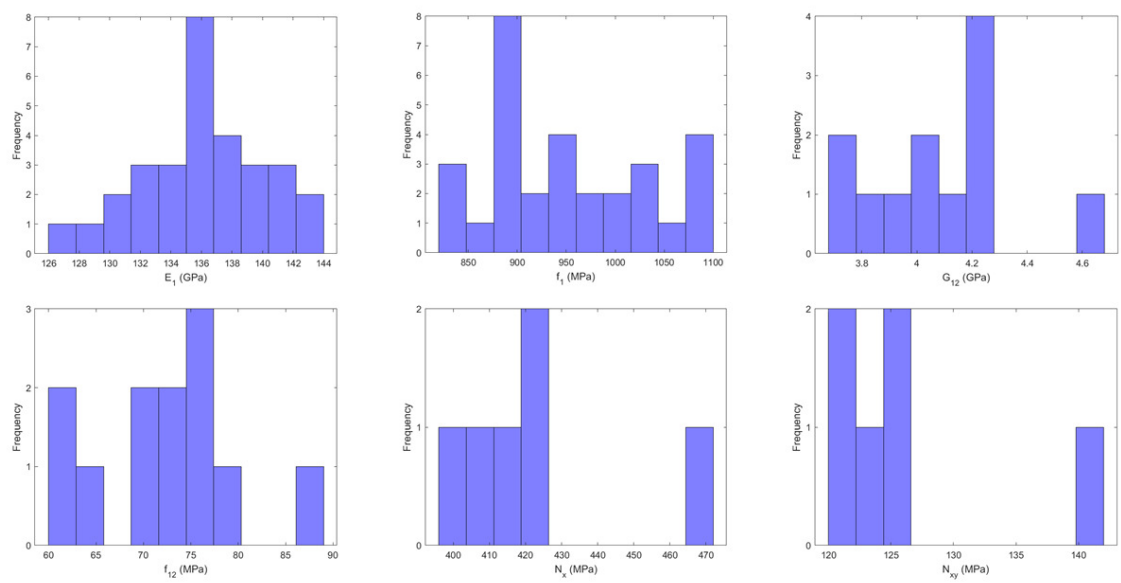


Figure 3.11: Histograms for the four stochastic material properties in the dataset ( $E_1, f_1, G_{12}, f_{12}$ ) and the compression and shear loadings responsible for first failure of the laminate( $N_x$  and  $N_{xy}$ ). Note that the data for  $N_x$  and  $N_{xy}$  is particularly scarce

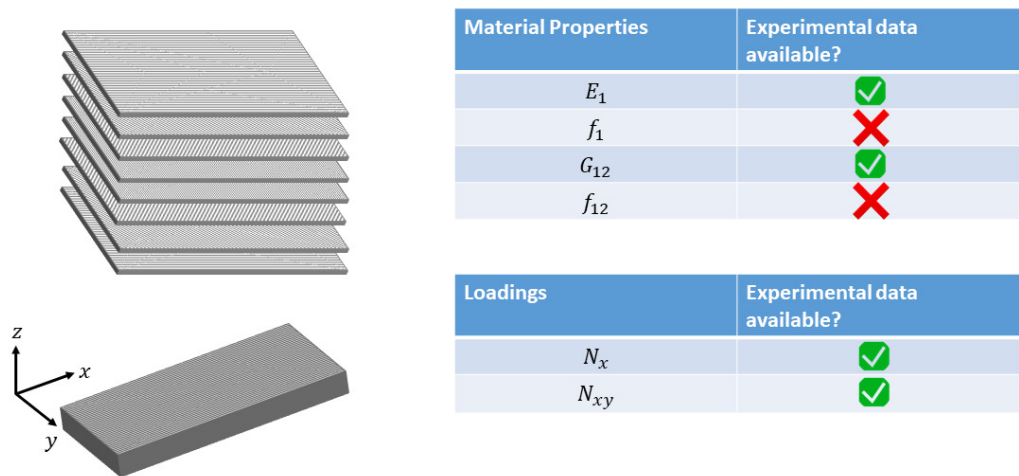


Figure 3.12: An illustration of a carbon fibre composite, which is comprised of a stack of unidirectional plies (left), and a summary of the experimental dataset for the test case (right). It was assumed that experimental data was not present for the strengths  $f_1$  and  $f_{12}$ . Distributions for these quantities were back-calculated and the results were compared to the experimental data

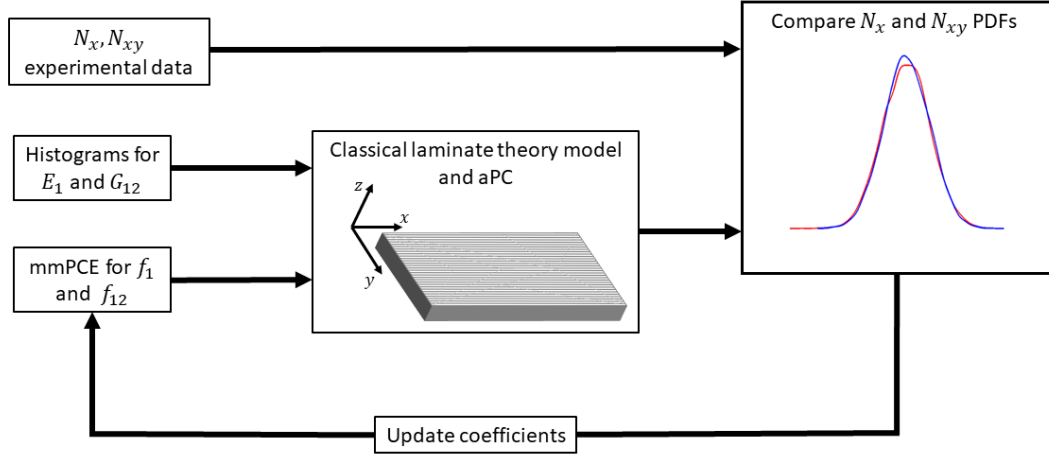


Figure 3.13: Flowchart for the back-calculation algorithm illustrated for the carbon fibre composite test case

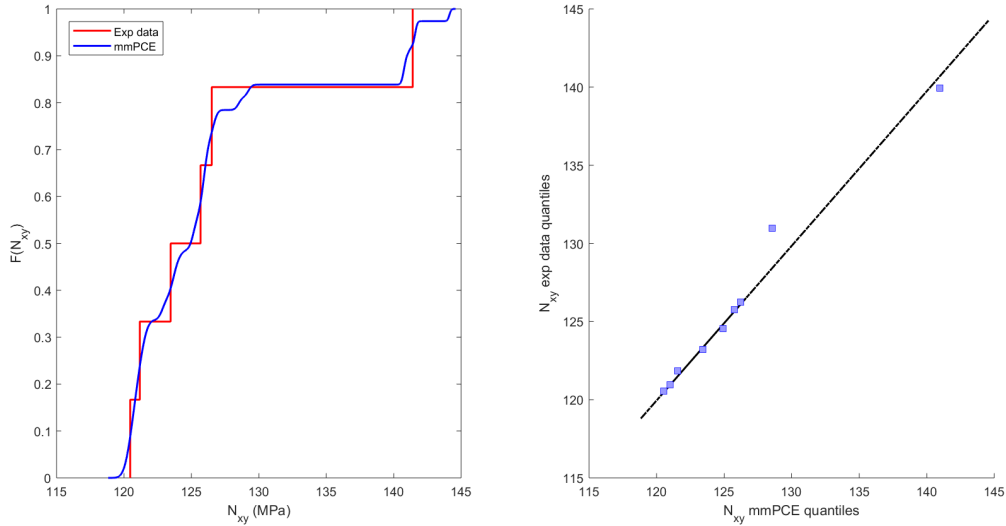


Figure 3.14: A comparison of the CDFs for the quantity of interest,  $N_{xy}$  (left). The optimum Kolmogorov Smirnov distance was found to be  $d^{ks} = 0.0912$ . A QQ plot for every 10<sup>th</sup> quantile (right). The majority of quantiles lie on the identity line, implying that the two distributions are statistically very similar



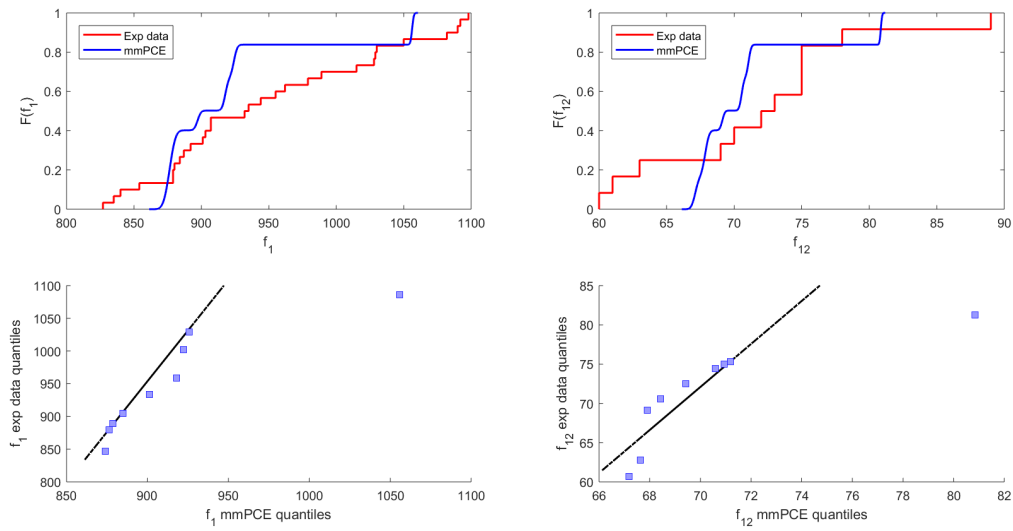


Figure 3.15: A comparison of CDFs for  $f_1$  (top left) and  $f_{12}$  (top right). While comparable, in general the back-calculated distributions are slightly less conservative than the experimental results. QQ plots for  $f_1$  (bottom left) and  $f_{12}$  (bottom right) for every 10<sup>th</sup> quantile indicate that the distributions are broadly similar, but that the right tails of the back-calculated distributions are more conservative (i.e. predict a lower strength for the composite)

## 3.2 Data Fusion of NIPC and scarce experimental data

As has been discussed, engineers must frequently produce designs that can yield good performance in a wide variety of operating conditions and for this reason, UQ techniques are used to evaluate the performance of candidate designs. To save both time and expense the performance of these candidate designs are typically tested using complex computational models. Non-intrusive Polynomial Chaos (NIPC) is a popular technique for propagating the effect of aleatoric uncertainties through complex computational models. As has been discussed previously, NIPC has been shown to yield results that are consistent with the more established Monte Carlo (MC) sampling techniques but at a much reduced computational cost, with NIPC requiring many times fewer model evaluations than traditional MC methods (see, e.g. [131]).

In recent years there has been great interest within the field of UQ in using multi-fidelity techniques to carry out uncertainty propagations. The accuracy of uncertainty propagations are fundamentally limited by the accuracy of the computational model that is used, however more accurate models are usually very computationally expensive. For instance, in fluid dynamics a Direct Numerical Simulation (DNS) can give a highly accurate resolution of a flow field but usually takes several days to run, making the repeated model evaluations necessary for UQ impractical. On the other hand, Reynolds-Averaged Navier-Stokes (RANS) simulations are less accurate due to assumptions made in the Reynolds stress closure but also far less computationally expensive and more suitable to repeated evaluations. This has led to the development of machine learning [132] and Bayesian [133] approaches which use RANS to leverage the results of DNS simulations. A bi-fidelity approach for combining RANS models of different fidelities was introduced by Doostan et al. (2016), in which the low-fidelity model is used to learn a reduced basis, which is used together with an interpolation rule and a limited number of high fidelity evaluations to approximate the high fidelity solution [134]. A multi-fidelity extension to NIPC was

introduced in Eldred et al. (2017) [56]. Methods such as Multi-level Monte Carlo [135, 136, 137], co-kriging [138] and Gaussian Process models (GPMs) [139, 140, 141] are used widely in the UQ community to leverage the results of a small number of high-fidelity simulations with many low fidelity model evaluations. A more comprehensive review of multi-fidelity methods in Uncertainty Quantification may be found in the review paper of Peherstorfer et al. (2018) [142].

However, while there has been great interest in developing methods that allow models of different fidelities to be combined, there has been relatively little attention paid to combining the results of uncertainty propagations with experimental data. While there are established Bayesian methods for updating a probability distribution as more data is collected (see e.g., [143]), this is an under-studied problem for frequentist methods. In this section we introduce a framework by which the results of an uncertainty propagation using a computational model may be adjusted based on the evidence of a limited set of experimental measurements of a Quantity of Interest (QoI). An example of where this problem is relevant is in the design process of composite materials, where thousands of coupon tests are required in order to authorise the use of a new composite material. The material properties of composite materials can vary significantly due to defects introduced in the manufacturing process [127], thus materials are typically characterised by statistical quantities such as the b-basis [144]. A current topic of research is how computational simulations of composite materials can be used to leverage the results of experimental coupon tests.

It should be noted that while relatively under researched in the Engineering sciences, the problem of combining evidence from different sources has been treated in other disciplines. For instance, Dempster-Shafer theory has been developed to combine evidence from multiple sources to reach a subjective judgement [145, 146]. It has been applied in software failure detection [147] and in Machine learning, where the accuracy of a decision is improved by combining the evidence of multiple classifiers [148]. However, significant criticism of Dempster-Shafer theory has been made in the literature, for instance in Pearl

(1990) [149]. In econometrics and weather forecasting density forecasts from different experts are frequently combined directly [150, 151]. The earliest and most popular attempts at forecast combination has been the ‘Linear Opinion Pool’, in which the resultant forecast is simply the weighted sum of the individual forecasts [152]. For more complete reviews of the probability distribution combining literature see, e.g. [150, 153, 154, 155]. A drawback of the Linear Opinion Pool, shown by Hora (2004) [156], is that if each predictive distribution is calibrated then any non-trivial linear combination will itself be un-calibrated [157, 158]. To this end, methods of non-linear combination have been developed in recent years. Gneiting and Ranjan (2010) introduce the Beta calibrated linear pools, in which the beta transform of the traditional linear pool is used to find the resultant distribution [159]. Bassetti et al. (2018) placed the Beta calibrated linear pool in a Bayesian framework [160] while Casarin et al. (2016) compared the performance of linear, harmonic and logarithmic opinion pools in a Bayesian beta mixture model but found the schemes to be substantially equivalent [161]. Kapetanios et al. (2015) introduced Generalised density forecast combinations, in which the weight functions of the Linear Opinion Pool are allowed to vary spatially [162]. One of these schemes could well be an appropriate way to directly combine experimental and computational probability distributions of a QoI, but this would assume that there was firstly a sufficient number of experimental results to estimate a probability distribution with sufficient confidence and secondly that some limitation in the design of the experiment meant that it too was an abstract representation of the system being designed, in the same way that a computational model is an abstract or idealised version of a system. It is assumed that the experimental data is too scarce to meaningfully estimate a probability distribution when taken in isolation, but that the experiment measures the QoI to a higher degree of accuracy than the computational model. Additionally, we assume that there is insufficient related evidence, for instance comparisons of experimental and simulation results of previous test cases, to formulate belief functions that could be combined according to Dempster-Shafer Theory.

A number of works in the literature that have addressed the problem of updating a probability distribution based on additional sources of data have done so using the Maximum Entropy Principle (MEP). First developed by Jaynes (1982) for estimating a canonical ensemble distribution constrained by particle number and total energy [163, 164], the MEP states that the least biased next estimate of a probability distribution is the distribution that maximises the relative entropy between the old and new estimate of the distribution, while at the same time conforming to a constraint imposed by the experimental data. Typically, the maximum entropy approach involves finding a set of Lagrange multipliers to reweight the initial estimate of the distribution and the imposed constraint is usually the experimental mean. This is the approach adopted in Cesari et al. (2018) to combine Molecular Dynamics simulations with experimental results [165] and also in Lou and Cukier (2018) [166]. The Maximum entropy approach was expanded in another paper by Cesari et al. (2016) to allow experimental errors to be included in the formulation [167]. Such an approach has a number of limitations. Firstly, the Lagrange multiplier formulation makes it very difficult to change the modality of the new estimate. Secondly, the use of moment based constraints is inappropriate for a limited set of experimental results as the statistical moments are unlikely to have converged, thus the approach taken in these works would force the new estimate to converge to a set of statistical moments which are unlikely to reflect the ‘true’ distribution. The limitations of moment based constraints are also discussed in Baggenstoss, where the joint distribution of a set of scalar measurements is used as a constraint, as opposed to the average values of each measurement [168]. Here, a p-box based constraint is used to update the results of an uncertainty propagation using NIPC. In this way scarce experimental data can be used to inform the estimate of the QoI’s probability distribution, without placing constraints which are harsher than the weight of the experimental evidence implies.

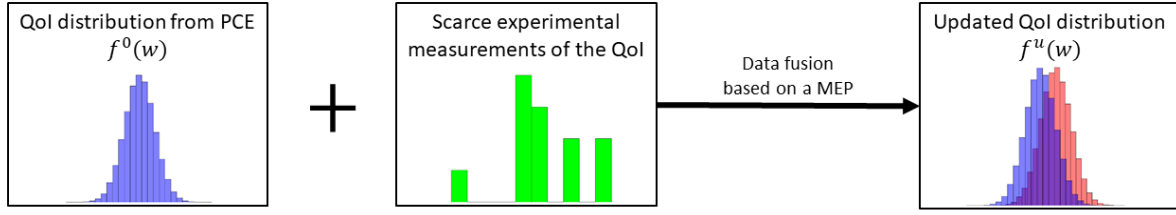


Figure 3.16: Having determined an initial estimate of the probability distribution for the QoI, scarce experimental measurements of the QoI can be used to update the coefficients of the PCE using a data fusion algorithm based on a MEP. The PCE can then be resampled to generate an updated probability distribution

### 3.2.1 Algorithm for data fusion of Polynomial Chaos with experimental results

The algorithm proposed here allows an initial estimate of a probability distribution to be updated through the fusion of an uncertainty propagation with NIPC and the evidence of a scarce experimental dataset. The initial estimate of the QoI probability distribution is arrived at by using an NIPC method to derive a PCE to approximate the response surface of a computational model,  $w(\boldsymbol{\xi})$ :

$$w(\boldsymbol{\xi}) = \sum_{k=1}^P \boldsymbol{\lambda}_k^0 \Psi_k(\boldsymbol{\xi}), \quad (3.15)$$

where  $\boldsymbol{\lambda}^0 \in \mathbb{R}^P$  refers to the initial estimate of the PCE coefficients. The algorithm introduced here is general and can be applied to a PCE arising from either a forwards or inverse propagation of uncertainty, although only the forwards case is considered here. Having estimated a PCE for the QoI, Monte Carlo evaluations of this PCE are then used to generate an initial estimate of the distribution. The algorithm updates the coefficients of the PCE such that the new estimate of the distribution maximises the cross-entropy between the initial estimate and the new estimate, while at the same time satisfying a constraint on the new estimate imposed by the available experimental data. This is illustrated schematically in Figure 3.16.

The Kullback-Leibler (KL) divergence is used to quantify the statistical distance between the updated PDF  $f^u(x)$  and the initial estimate of the PDF  $f^0(w)$ . For a continuous set of normalised probability distributions, the Kullback-Leibler divergence is defined as [69]:

$$d^{KL}(f^u||f^0) = \int_{-\infty}^{\infty} f^u(w) \log \frac{f^u(w)}{f^0(w)} dw. \quad (3.16)$$

The Kullback-Leibler (KL) divergence is defined as the negative of the cross entropy, hence minimising the KL divergence between two probability distributions is the equivalent of maximising the cross-entropy. Minimising the KL divergence between the updated distribution and the initial estimate produces a distribution which introduces the minimum amount of new information with respect to the prior knowledge [166, 167]. The steps of the data fusion algorithm are illustrated in Figure 4.2. The initial estimate of the PCE coefficients in equation (3.15),  $\lambda_k^0$ , are then replaced with the updated set of coefficients,  $\lambda_k^u$ , which are the solution of an optimisation problem, with a constraint imposed by the experimental data. In this work a p-box placed around the experimental data is used to constrain the CDF of the updated distribution  $F^u(w)$  [169]. It should be noted that the formulation is flexible and can admit a range of user chosen constraints. For instance, moment based constraints could be used such as the experimental mean or the experimental standard deviation. However, given the scarcity of the experimental data the statistical moments of the experimental data are unlikely to have converged to the true distribution, making constraints based on the statistical moments of the experimental data unsuitable. p-boxes themselves are discussed further below. The optimisation problem to find the updated coefficients is formulated as:

$$\lambda_k^u = \underset{\lambda_k^u}{\operatorname{argmin}} d^{KL}(f^u(w)||f^0(w)) \quad \text{subject to } \underline{F}(w) \leq F^u(w) \leq \bar{F}(w), \quad (3.17)$$

where  $\underline{F}(w)$  and  $\bar{F}(w)$  refer to the lower and upper limits of the p-box respectively. In general, a six sigma criterion is applied in reliability problems (see, e.g. [170]). Montomoli and Massini (2013) demonstrated that for aerospace accidents this criterion is not accurate enough and tends to underestimate the likelihood of accidents [171]. Given the scarcity of the experimental data, accurate estimation of the tails of the distribution is not possible using existing methods in the literature (see, e.g. [172, 173]). We focus on a more general case, where constraints are imposed that are likely to only affect the first two or three statistical moments in order to make the experimental data consistent with the probability distribution estimated from the PCE.

As with the algorithms based on probabilistic equivalence discussed above, the manner in which the PCE coefficients are updated in the algorithm will depend on the optimisation strategy used. The use of an inequality constraint such as a p-box, in addition to the lack of an analytical relationship between the PCE coefficients and  $f^u(w)$ , precludes the use of Lagrange multipliers, which have been used in other works which use an MEP to update a probability distribution. As with the algorithms based on probabilistic equivalence, an interior point algorithm was used, initialised at a point in PCE space within the feasible region. It was found that the variation of  $d^{KL}$  in the  $P$ -dimensional PCE parameter space was sufficiently smooth for the algorithm to avoid becoming trapped in a local minima [174, 175].

### 3.2.2 p-boxes as a constraint on the optimisation

A p-box is defined by two curves  $\bar{F}(w)$  and  $\underline{F}(w)$  which act as upper and lower bounds to the updated CDF  $F^u(w)$  respectively. The bounds are defined such that the true but unknown CDF lies between  $\underline{F}(w)$  and  $\bar{F}(w)$ , i.e.:  $\underline{F}(w) \leq F^{true}(w) \leq \bar{F}(w) \forall w$ . p-boxes may be non-parametric or parametric. Parametric p-boxes arise from distributions which are known to conform to a particular probability distribution (normal, exponential, Weibull, uniform, etc.) but with parameters which are specified imprecisely as intervals. Non-parametric p-boxes are distribution-free and make no assumptions about the form



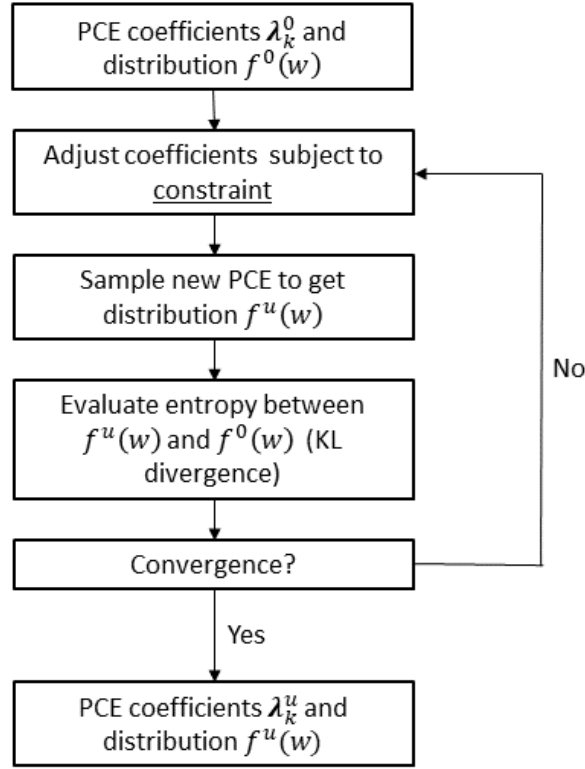


Figure 3.17: Data fusion algorithm to adjust the coefficients based on an Maximum Entropy Principle, subject to constraints imposed by the available experimental data

of a distribution and so are a more appropriate choice as a bound when there is insufficient data to specify a family of distributions to which the experimental data for the QoI belongs. In this work a non-parametric technique based upon the Dvoretzky-Kiefer-Wolfowitz inequality and the Massart bound (referred to as the DKWM bound) is used to place a bound around the experimental data for the QoI [176]. Bounds based on the DKWM bound place a confidence interval around the Kolmogorov Smirnov statistic between the unknown, true distribution  $F^{true}$  and the distribution for the experimental data  $F^{exp}$  [117, 118]:

$$\mathbb{P}[\sup_w |F^{exp}(w) - F^{true}(w)| \leq 2e^{-2n\epsilon^2}] = \alpha. \quad (3.18)$$

With a confidence level  $\alpha$  the true CDF does not differ from the experimental CDF by more than the error,  $\epsilon$ . Rearranging the RHS of this equation yields an expression for  $\epsilon$  given a desired confidence  $\alpha$ :

$$\epsilon = \sqrt{\frac{-\log\left(\frac{\alpha}{2}\right)}{2n}}, \quad (3.19)$$

where  $n$  is the number of experimental data points available. Note that the inverse dependence on  $n$  means that the bound becomes tighter if more experimental points are available. By translating the experimental CDF vertically by  $\pm\epsilon$ , a p-box may be created, i.e.:

$$F^{exp}(w) - \epsilon \leq F^{true}(w) \leq F^{exp}(w) + \epsilon. \quad (3.20)$$

The CDF of the updated distribution,  $F^u(w)$  is constrained such that it lies within the p-box. As the p-box is created through a vertical translation of  $F^{exp}$  it is defined only over the support of  $F^{exp}$  and so does not constrain the tails of  $F^u(w)$  directly. As has been discussed, this is considered acceptable as the intention is to place a constraint that affects the first two or three statistical moments of the updated probability distribution. The tails of a distribution are typically determined by higher order statistical moments.

### 3.2.3 Carbon fibre composite test case

The algorithm for the fusion of simulations and scarce experimental measurements of a QoI was tested using the dataset of laboratory measurements of carbon fibre composites and the computational model based on the equations of classical laminate theory used in the previous test case. In this instance, aPC is used to construct a surrogate for  $N_x$  and  $N_{xy}$  based on the scarce experimental data for  $E_1$ ,  $G_{12}$ ,  $f_1$ , and  $f_{12}$ . The test case is

relevant as, as has been discussed above, significant variance is observed in the material properties of composite materials when measured in coupon tests due to the complexity of the manufacturing process [127]. Certifying a new composite material can require thousands of coupon tests, which is both expensive and time consuming, providing motivation to replace physical tests with simulations wherever possible. Previous works have attempted to reduce the computational cost of these simulations using non-intrusive polynomial chaos approaches (see e.g. [177, 178, 179]). The data fusion algorithm presented here can be used to enrich these PCE surrogates with scarce experimental measurements of the QoI.

Having produced a surrogate using aPC, an estimate for the shear load probability distribution,  $f^0(N_{xy})$ , was generated from  $10^6$  Monte Carlo realisations of the PCE. The data fusion algorithm was then used to adjust the coefficients of the PCE based on the evidence of the available experimental measurements of the failure stresses. Given the relative simplicity of the computational model used, we expect the model error to result in a discrepancy between the estimate of the failure stress distribution and the ‘true’ but unknown distribution, which we assume can only be found with many experimental tests. The aim of the data fusion algorithm is to improve the initial estimate provided by NIPC, based on the evidence of a small number of experiments.

The results of the uncertainty propagation and subsequent data fusion are displayed in Figure 3.18. The blue line in the left panel represents the CDF of the initial estimate of the probability distribution from propagating the uncertainties in the material properties with aPC,  $F^0(N_{xy})$ . The green line indicates the CDF of the 6 available experimental results,  $F^{exp}(N_{xy})$ . The DKWM bound was used to place a p-box around  $F^{exp}(N_{xy})$  with a 95% confidence level. An interior point algorithm was used to find a set of updated PCE coefficients which minimised the KL divergence between the initial and updated estimate of the probability distribution of  $N_{xy}$ , while at the same time ensuring that the updated probability distribution lies within the bounds set by the p-box. The algorithm was initialised at a point in the PCE parameter space that satisfied the bounds and was

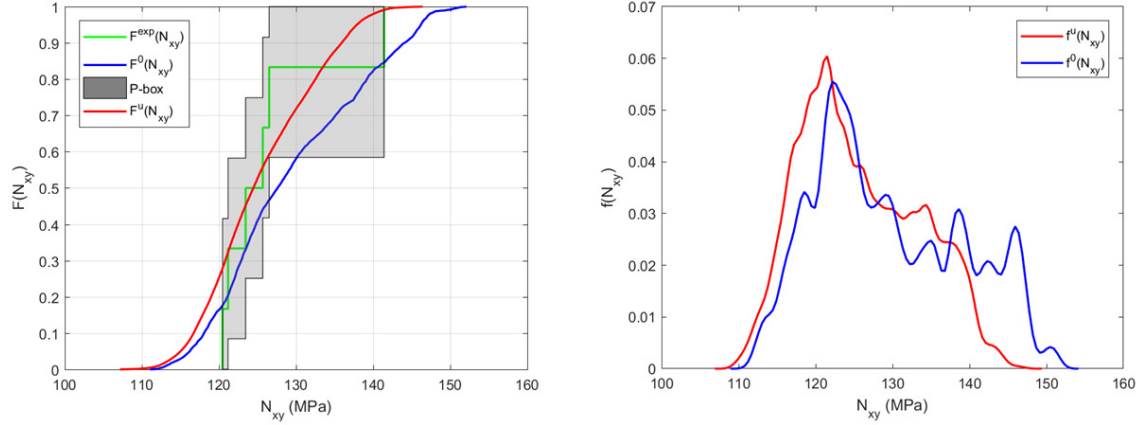


Figure 3.18: A plot showing the CDFs of the original estimate of the probability distribution arising from the aPC (blue); the available experimental data (green) and the updated estimate of the probability distribution (red). The p-box, generated by the DKWM bound, is shaded in grey (left subplot). Kernel density estimates of the original and updated estimates- note how the maximum entropy estimator has smoothed the peaks in the original estimate arising from the scarce input data (right subplot)

found to converge quickly, requiring only 74 function evaluations. The CDF of the updated distribution is represented in red in the left panel of Figure 3.18. The right panel of Figure 3.18 illustrates the kernel density estimates of the initial estimate of the probability distribution  $f^0(N_{xy})$  and the updated probability distribution  $f^u(N_{xy})$ . Kernel density estimates were made for  $f^0(N_{xy})$  and  $f^u(N_{xy})$  based on  $10^6$  Monte Carlo realisations of the PCE. As can be seen the updated distribution is weighted more heavily to the left, giving a more conservative estimate of the failure strength distribution. The potential of the data fusion algorithm to adjust the modality of the updated distribution is demonstrated, with some peaks in  $f^0(N_{xy})$  becoming smoothed out in the updated estimate as a result of the MEP. These peaks likely arise due to the effect of the scarce input data on the uncertainty propagation using aPC.

In recent years much attention has been paid to developing methods of combining models of different accuracies in order to propagate uncertainty, but relatively little attention has been paid to improving uncertainty propagations by considering evidence from both

computational models and experiments. The growing accuracy of computational models offers industries the possibility of reducing the amount of physical testing needed in the design phase of a product by supplementing physical tests with simulations, making this research problem increasingly relevant. Data fusion allows for evidence from multiple sources to be considered in order to estimate a probability distribution. The framework introduced here is intended for cases where a probability distribution must be estimated based on the evidence of simulations, which are relatively plentiful but are considered less reliable than experimental measurements as a result of the epistemic uncertainties present in the models (model error). The experimental data, while more reliable, is expensive or time consuming to obtain and as a consequence there are not enough points to fit a probability distribution to the data without making significant assumptions.

## Summary

In this chapter we have presented two methods through which simulations might be used to leverage scarce experimental data in an uncertainty propagation. The first method, for inverse identification, was validated with synthetic data and the response surface of a scroll compressor, before it was applied to real experimental data from coupon tests of carbon fibre composites. The second method provided a framework for data fusion between NIPC and experimental data in a forwards propagation of uncertainty. The Maximum Entropy Principle is a key aspect of the framework. This assumption, together with the data driven NIPC formulation used to propagate uncertainties, means that the method is non-parametric as no assumptions are made as to the family to which the underlying distribution belongs. This makes the framework well suited to handling scarce data. As can be seen from Figure 3.18, the Maximum Entropy Principle tends to have a smoothing effect in practice and can eliminate spurious peaks which arise in the original estimate of the probability distribution due to the scarcity of the input data. The use of a p-box as a constraint, as opposed to moment based constraints, is another feature of the

framework that allows it to handle scarce data efficiently. Scarce experimental data can be used to inform the updated estimate of the probability distribution, without placing a set of onerous moment based constraints on the estimate which are unlikely to reflect the moments of the true distribution.

# Machine learning meta-models for DSE

## Contents

---

<b>4.1 Histogram interpolation for Design Space Exploration</b>	<b>96</b>
4.1.1 An alg. for non-param., data-driven hist. interp. . . . .	100
4.1.2 Probabilistic Equivalence . . . . .	101
4.1.3 Histogram interpolation test case . . . . .	106
4.1.4 Predictive uncertainty of histogram interpolation . . . . .	115
<b>4.2 Bi-fidelity modelling with Knowledge Based Neural Networks</b>	<b>120</b>
4.2.1 Knowledge Based Neural Networks (KBaNNs) . . . . .	123
4.2.2 KBaNN architecture for a local Navier-Stokes approximation . . . .	126
4.2.3 Test case and results . . . . .	128

---

In the conceptual design phase of a new product, where the cone of uncertainty is widest, high-level computational models tend to be used to evaluate candidate designs; with more granular models being employed once desired areas of the design space are identified and more resources can be budgeted for each candidate design. In this chapter we present two machine learning based methods to assist design space exploration. Firstly, we propose an interpolation method for model-free histogram interpolation, based

on the principle of probabilistic equivalence discussed in previous chapters. Secondly, we present a data fusion approach based on Knowledge Based Neural Networks (KBaNNs). A KBaNN acts as a meta-model for a function that is expensive to evaluate by training a neural network with a limited set of expensive model evaluations.

## 4.1 Histogram interpolation for Design Space Exploration

In this section we present a novel algorithm to aid designers in the exploration of the design space of a new engineering product during the conceptual design phase. As has been discussed, the performance of any design will be non-deterministic due to the presence of irreducible uncertainties in the system, such as imperfections introduced during the manufacturing process. The uncertain performance of a design can be represented with a probability distribution for a quantity of interest. The algorithm presented here develops a meta-model that allows the designer to estimate the probability distribution for a candidate design without requiring additional experiments or simulations.

The designs for new engineering products are developed within the design space,  $\mathbb{X}$ , a  $n_x$ -dimensional space of parameters that influence the performance of the final design. A candidate design may be uniquely specified as a parameter point in the design space, denoted  $\mathbf{x} \in \mathbb{X} \subseteq \mathbb{R}^{n_x}$ . The subspace containing points in the design space that satisfy a set of requirements imposed on the design is referred to as the feasible region. Mavris et al (2000) discussed how decisions made at the early, conceptual stages of an aircraft design can restrict the range of alternative designs available at later stages in the design process, despite limited quantitative knowledge of the interaction between requirements being available in the conceptual design stage [180]. Consequently, methods have been developed to promote a wider exploration of the design space in the conceptual design stage, with approaches based on evolutionary algorithms [181], the morphological matrix



[182], and architecture design space graphs [183] proposed. Set based design strategies develop multiple designs concurrently, deferring decisions that limit the scope of the design space for as long as possible, allowing more time for the designers to accumulate knowledge about the system trade-offs [184, 185, 186]. Methods for design space exploration (DSE) must balance flexibility in exploring the design space with the associated cost of evaluating potentially expensive computer code at the investigated parameter points in the design space [187, 188]. For this reason there has been interest in developing interpolation schemes that predict the performance of a candidate design based on historic data. Examples include the interpolation method for Computer Aided Design (CAD) models in Schulz et al (2017) [189] and the approach to DSE for architectural design in Ipek et al (2008), in which neural networks were used as meta-models linking the design parameters to experimental data [190].

In this section we present an algorithm for model-free DSE in the early stages of product development by informing the decisions of designers using estimations based on historic data. In this context historic data refers to either the results of simulations evaluated according to a design of experiments (DoE) or the results of experiments and simulations collected during the development of previous products. While the values of some parameters may be selected by the designers, the performance of any design will be non-deterministic due to the irreducible uncertainties present in the system, an example being natural variability in material properties. The uncertain performance of a design can be represented using a continuous function such as a PDF or CDF for a given Quantity of Interest (QoI),  $w$ . Each candidate design is associated with a probability distribution representing its performance. We propose an algorithm that will allow designers to estimate the performance of a candidate design from a set of training data. Figure 4.1 illustrates this problem for a  $n_x = 2$  design space with parameters  $x_1$  and  $x_2$ . The blue dots indicate the parameter points in the training dataset. An uncertainty analysis, using either experiments or the evaluations of expensive computer code, has been conducted at each of these points, hence a probability distribution for the QoI is associated with

each parameter point. The objective of the algorithm presented here is to estimate the probability distribution at a query parameter point in the design space, indicated by the red cross, without requiring further experiments or model evaluations.

Histogram interpolation has been used in the past to accomplish this objective: Read (1999) used a linear histogram interpolation method to find the reconstructed invariant mass distribution of a Higgs boson. It was found that the interpolated probability distribution was consistent with the results of a Monte Carlo simulation, but much less computationally expensive to obtain. A limitation of the method is that it is defined for probability distributions that have a linear dependence on a particular parameter; in the chapter, interpolation forms are defined for the normal and exponential distributions [191]. A multi-dimensional, non-linear moment morphing technique was later introduced in Baak et al (2015), again with a high energy physics application, to account for distributions with a non-linear dependence on their parameters and to account for dependency between them [192]. The method presented here is intended to provide a more flexible, data-driven framework for histogram interpolation which is non-parametric (free from assumptions about the forms of distributions).

The algorithm described here employs probabilistic equivalence to project continuous probability distributions to a finite-dimensional space, corresponding to the coefficients of a polynomial chaos expansion (PCE). These coefficients are allowed to vary spatially within the design space and an interpolation function is developed for each coefficient by fitting to the available training data. In this respect the algorithm is similar to deep learning methods such as autoencoders that reduce the dimensionality of the input space and train a neural network within the reduced space [193].

Gaussian Process models (GPMs) are used as interpolation functions for the PCE coefficients. While the algorithm presented here is novel, there is precedent for combining Gaussian processes and polynomial chaos within a meta-model in the literature. Schöbi et al (2015) developed Polyomial Chaos-based kriging, a meta-model that combines the global interpolation properties of PCEs with a local correction provided by a Gaussian

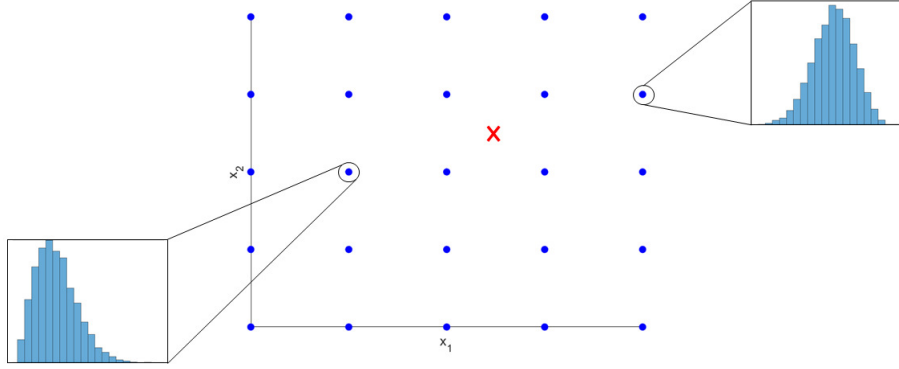


Figure 4.1: The performance of a design is typically non-deterministic due to the presence of irreducible uncertainties in the system. Here an  $n_x = 2$  design space is illustrated. Blue dots indicate parameter points for which QoI data is available. The goal of the algorithm is to estimate the probability distribution at a new parameter point (red cross).

process [194]. Polynomial Chaos-based kriging was later employed in an adaptive sampling algorithm which was used to iteratively improve the accuracy of a meta-model for the limit state function (LSF). As has been discussed, the feasible region of design space refers to the subset of points for which a design satisfies a set of performance requirements cast as inequality constraints. The LSF separates this region from the subspace containing parameter points that violate the constraints [195]. Approximating a LSF is a key aspect of reliability analysis, where it is used to calculate the probability of a system violating the constraints, however, this is an issue we do not address in this chapter (although we note that it would be possible to extend our scheme to do this). The formulation for multi-scale uncertainty propagation in Thimmisetty et al (2018) employs Gaussian Processes and PCEs in a similar manner to our algorithm, where the coefficients of the PCE are themselves Gaussian random variables. This allowed the meta-model to reflect both the global uncertainty structure through the GPMs, while also accounting for the local uncertainty structure through the polynomial chaos [196].

### 4.1.1 An algorithm for non-parametric, data-driven histogram interpolation

In this section the steps of a novel algorithm for non-parametric, data-driven histogram interpolation are outlined. A data-driven formulation is chosen so that the method might be applicable to a wider variety of datasets than existing methods, which assume that the histograms belong to a particular family of probability distribution. The goal of the algorithm is to estimate the probability distribution,  $\hat{f}(w|\mathbf{x}^*)$ , for the performance of a candidate design at the parameter point  $\mathbf{x}^* \in \mathbb{X} \subseteq \mathbb{R}^{n_x}$  in the design space. Historic data, which as discussed above may include either the results of experiments or computer simulations, is used as training data to develop a set of interpolation functions. The training dataset includes the probability distributions  $f(w|\mathbf{x}^{(i)})$  for a Quantity of Interest,  $w$ , and conditional on their locations in the design space  $\mathbf{x}^{(i)}$   $i = 1 \dots n$ . The mathematical aspects of the algorithm are discussed in greater detail below. Note that in this work we restrict our attention to the interpolation of univariate probability distributions for the QoI, i.e.  $w \in \mathbb{R}^1$ , although there is scope to extend our formulation to the multivariate case.

#### 4.1.1.1 Polynomial Chaos Expansions

In the algorithm presented here polynomial chaos is not applied to propagate uncertainty but rather as a means for projecting continuous probability distributions into a finite-dimensional space for the purposes of interpolation. To keep the algorithm general we work only with the probability distribution of the QoI and assume that there is no knowledge of the stochastic process by which this distribution is generated. A PCE is constructed that is a function of a single random variable,  $\xi$ , that is sampled from a generating probability distribution which is dependent on the orthonormal basis used. The right hand column of Table 2.1 lists the generating distributions for a selection of probability distributions in the Askey scheme. As has been discussed above, the stochastic

response of a candidate design is dependent on its position in the design space as well as on the uncertain inputs  $\boldsymbol{\xi}$ . For this reason, the PCE parameter set is no longer treated as a set of constants, but rather a set of functions dependent on  $\boldsymbol{x}$ :

$$w(\boldsymbol{x}, \xi) = \sum_{k=1}^P \boldsymbol{\lambda}_k(\boldsymbol{x}) \Psi_k(\xi), \quad (4.1)$$

where  $\xi \in \mathbb{R}^1$ . At a given parameter point,  $\boldsymbol{x}$ , a probability distribution for  $w$  may be estimated through Monte Carlo sampling of  $\xi$ :  $n_{MC}$  samples are generated from the probability distribution associated with the chosen polynomial basis- referred to as the generating probability distributions in Table 2.1. This yields the dataset  $W = [w(\boldsymbol{x}, \xi^{(1)}), w(\boldsymbol{x}, \xi^{(2)}), \dots, w(\boldsymbol{x}, \xi^{(n_{MC})})]$  from which the CDF  $\hat{F}(w|\boldsymbol{\lambda}, \boldsymbol{x})$  may be calculated directly. The PDF  $\hat{f}(x|\boldsymbol{\lambda}, \boldsymbol{x})$  can be estimated from  $W$  through kernel density estimation (see, e.g. [197, 198]).

#### 4.1.2 Probabilistic Equivalence

The algorithm described here represents the uncertainty in the performance of a product across the design space with a PCE, the coefficients of which are estimated with a set of  $P$  interpolation functions. In order to find a meta-model for these coefficients it is necessary to generate a set of training data by projecting PDFs or CDFs into a finite,  $P$ -dimensional, PCE coefficient space,  $\Lambda$ . For each probability distribution in the training set a probabilistic equivalence is sought:

$$\forall w : f(w|\boldsymbol{x}^{(i)}) = \hat{f}(w|\boldsymbol{\lambda}^{(i)}, \boldsymbol{x}^{(i)}), \quad i = 1 \dots n. \quad (4.2)$$

In this instance we define the probabilistic equivalence in terms of the PDFs rather than the CDFs:  $f(w|\mathbf{x}^{(i)})$  refers to the PDF at the  $i^{\text{th}}$  parameter point in the training data and  $\hat{f}(w|\boldsymbol{\lambda}^{(i)}, \mathbf{x}^{(i)})$  represents the PDF estimated from the  $i^{\text{th}}$  set of PCE coefficients.

As in previous chapters, achieving an exact probabilistic equivalence between the two sets of PDFs is intractable. Instead, sets of coefficients are sought that minimise the statistical distance between each pair of probability distributions and (4.2) may be reformulated as  $n$  optimisation problems:

$$\boldsymbol{\lambda}^{(i)} = \underset{\boldsymbol{\lambda}^{(i)}}{\operatorname{argmin}} \quad d(f(w|\mathbf{x}^{(i)}), \hat{f}(w|\boldsymbol{\lambda}^{(i)}, \mathbf{x}^{(i)})), \quad i = 1 \dots n, \quad (4.3)$$

where  $d$  refers to a measure of statistical distance between each pair of probability distributions. The steps of the sub-algorithm used to estimate the PCE coefficients for the  $i^{\text{th}}$  training parameter point are as follows:

Input: QoI PDF  $f(w|\mathbf{x}^{(i)})$  at  $i^{\text{th}}$  parameter point  $\mathbf{x}^{(i)}$  and initialised PCE coefficients  $\boldsymbol{\lambda}^{(i)}$

Output: PCE coefficients  $\boldsymbol{\lambda}^{(i)}$

1. MC sampling of the PCE to generate  $n_{MC}$  QoI samples,  $W$
2. Kernel estimate of  $\hat{f}(w|\boldsymbol{\lambda}^{(i)}, \mathbf{x}^{(i)})$  from  $W$
3. Evaluate  $d(f(w|\mathbf{x}^{(i)}), \hat{f}(w|\boldsymbol{\lambda}^{(i)}, \mathbf{x}^{(i)}))$
4. Check for convergence in  $d$ , otherwise update  $\boldsymbol{\lambda}^{(i)}$  and return to step 1

Note that there is an equivalent sub-algorithm for CDFs instead of PDFs if a statistical distance based on CDFs is used. The manner in which the PCE coefficients are updated will depend on the choice of optimisation algorithm. In general this is likely to be a non-convex optimisation and for this reason non-gradient based optimisation strategies are preferred such as genetic algorithms.

The histogram interpolation algorithm described here uses probabilistic equivalence to find a PCE representation for each probability distribution in the training dataset and develops interpolation functions for the PCE coefficients. There is a certain level of correlation between the PCE parameters which is difficult to quantify. For this reason, the QR decomposition was used to decorrelate the training data, transforming the PCE coefficients in the training data  $\boldsymbol{\lambda}^{(i)}, i = 1 \dots n$  to a  $P$ -dimensional space in which the correlations are removed, denoted as  $\tilde{\Lambda}$  (see, e.g. [199]). A set of  $P$  interpolation functions are trained on each component of the transformed data,  $\boldsymbol{\lambda}^{(i)} \rightarrow \tilde{\boldsymbol{\lambda}}^{(i)}, i = 1 \dots n$ . An inverse transformation is used to map the estimates of the meta-models back to  $\Lambda$  space.

It should be noted that there will be some inevitable information loss at the probabilistic equivalence stage that is difficult to quantify. Given that any model-free interpolation is only ever intended as a preliminary estimate and that efficient probabilistic equivalence has been demonstrated for several non-Askey scheme distributions, this information loss is considered to be negligible. In other words, the errors associated with the interpolation of the PCE coefficients are assumed to be significantly greater than the errors in obtaining the coefficients used as training data.

#### 4.1.2.1 Gaussian process modelling

Having obtained a set of (transformed) PCE coefficients for each parameter point in the training set using probabilistic equivalence and a QR decomposition, an interpolation function,  $\Phi_k$ , is developed for each transformed coefficient,  $\tilde{\boldsymbol{\lambda}}_k$ :

$$\tilde{\boldsymbol{\lambda}}_k(\mathbf{x}) = \Phi_k(\mathbf{x}) + \epsilon_k, \quad k = 1 \dots P, \quad (4.4)$$

where  $\epsilon_k \sim \mathcal{N}(0, \sigma_{\epsilon_k})$  is an independent, identically distributed Gaussian noise with variance  $\sigma_{\epsilon_k}^2$ . In this work, Gaussian Process models are used as interpolation functions.

A widely used supervised learning algorithm, GPMs have previously been used in uncertainty quantification [200, 201] and reliability based design [202, 203]. A Gaussian Process prior is assumed over the regression functions, i.e.

$$\begin{bmatrix} Y_k \\ \tilde{\mathbf{y}}_k(\mathbf{x}^*) \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} \mu(X) \\ \mu(\mathbf{x}^*) \end{bmatrix}, \begin{bmatrix} K(X, X) & K(\mathbf{x}^*, X)^T \\ K(\mathbf{x}^*, X) & K(\mathbf{x}^*, \mathbf{x}^*) \end{bmatrix} \right), \quad (4.5)$$

where  $\mu$  is a function representing the mean of the process. The matrix,  $K$ , reflects the covariances of the training data, which is organised into a matrix of inputs  $X = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}]^\top \in \mathbb{R}^{n \times n_x}$  and outputs  $Y_k = [\tilde{\mathbf{y}}_k^{(1)}, \tilde{\mathbf{y}}_k^{(2)}, \dots, \tilde{\mathbf{y}}_k^{(n)}]^\top \in \mathbb{R}^n$  [204, 205]. Elements of the covariance matrix are given by:

$$K_{ij}(X, X) = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) + \sigma_{\epsilon_k}^2 \delta_{ij}, \quad (4.6)$$

where  $k(\cdot)$  represents the covariance function. Common choices for the covariance function, or kernel, include: the squared exponential, Matern and rational quadratic kernels. In this work the squared exponential kernel is used:

$$k(\mathbf{g}, \mathbf{h}) = \sigma^2 \exp \left( - \frac{(\mathbf{g} - \mathbf{h})^T \Lambda^{-1} (\mathbf{g} - \mathbf{h})}{2} \right), \quad (4.7)$$

where  $\Lambda = \text{diag}(l_1^2, l_2^2, \dots, l_{n_x}^2)$  and  $\mathbf{g}, \mathbf{h} \in \mathbb{R}^{n_x}$ . The correlation lengths  $l_i$ ,  $i = 1 \dots n_x$  and amplitude  $\sigma$  are hyperparameters which must be determined numerically for each of the  $P$  GPMs, together with  $\sigma_{\epsilon_k}$ . In this work a squared exponential covariance function was used as this is well suited for smoothly varying data, which is typical of the response surfaces of engineering products. A consequence of assuming a Gaussian Process prior is that the posterior predictive density is also Gaussian [204]:



$$\tilde{\lambda}_k | X, Y_k, \mathbf{x}^* \sim \mathcal{N}(\hat{\mu}_k, \hat{\sigma}_k), \quad (4.8)$$

where

$$\hat{\mu}_k = \mu_k(\mathbf{x}^*) + K(\mathbf{x}^*, X)^\top K(X, X)^{-1}(Y_k - \mu_k(\mathbf{x}^*)), \quad (4.9)$$

$$\hat{\sigma}_k^2 = K(\mathbf{x}^*, \mathbf{x}^*) - K(\mathbf{x}^*, X)^\top K(X, X)^{-1}K(X, \mathbf{x}^*). \quad (4.10)$$

Typically the mean function,  $\mu_k(\cdot)$ , is set to 0 and emphasis is instead placed on choosing an optimal covariance function. Having decided upon a covariance function for the GP and a form for the mean function, the set of unknown hyperparameters  $\Theta_k = [\sigma, l_1, l_2, \dots, l_{n_x}, \sigma_{\epsilon_k}]^\top$  must be computed to obtain the posterior distribution of the GPM. It is not possible to place a prior on these parameters and solve analytically, instead the hyperparameters must be estimated by assuming a relatively flat prior distribution and by maximising the log marginal likelihood:

$$\begin{aligned} \mathcal{L}(\Theta_k) &= \log[\mathbb{P}(Y_k | X, \Theta_k)] \\ &= -\frac{1}{2} \left( Y_k^\top (K(X, X) + \sigma_{\epsilon_k}^2 I_n)^{-1} Y_k + \log(|K(X, X) + \sigma_{\epsilon_k}^2 I_n|) + n \log(2\pi) \right). \end{aligned} \quad (4.11)$$

There are a variety of gradient based approaches to accomplish this estimation, assuming that the gradient of the log likelihood is known [206]. However, as discussed in Petelin et al (2011), deterministic optimisation approaches can be very sensitive to the initial choice of hyperparameter value. Given that the marginal likelihood function usually features many local maxima, evolutionary algorithms may be more appropriate for this non-convex optimisation [207]. The set of hyperparameters that minimise the negative of the log likelihood are found through a genetic algorithm. The steps of the algorithm may be summarised as follows:

Input: Training data  $f(w|\mathbf{x}^{(i)})$ ,  $\mathbf{x}^{(i)}$ ,  $i = 1 \dots n$  and query parameter point  $\mathbf{x}^*$

Output: Estimated PDF  $\hat{f}(w|\boldsymbol{\lambda}(\mathbf{x}^*), \mathbf{x}^*)$

1. Probabilistic equivalence sub-algorithm to find the set of PCE coefficients  $\boldsymbol{\lambda}^{(i)}$ ,  $i = 1 \dots n$
2. Transform the coefficients to remove correlations  $\boldsymbol{\lambda}^{(i)} \rightarrow \tilde{\boldsymbol{\lambda}}^{(i)}$
3. Train a GPM for each  $\tilde{\boldsymbol{\lambda}}_k$ ,  $k = 1 \dots P$
4. Estimate transformed PCE parameters  $\tilde{\boldsymbol{\lambda}}(\mathbf{x}^*)$  with GPMs
5. Reverse transformation  $\tilde{\boldsymbol{\lambda}}(\mathbf{x}^*) \rightarrow \boldsymbol{\lambda}(\mathbf{x}^*)$
6. Resample the PCE with Monte Carlo sampling
7. Kernel density estimate of  $\hat{f}(w|\boldsymbol{\lambda}(\mathbf{x}^*), \mathbf{x}^*)$

### 4.1.3 Histogram interpolation test case

The histogram interpolation algorithm for DSE was validated using a benchmark test case for methods in reliability based design [208, 209]. A function for the displacement of a cantilever beam was used to generate synthetic data; the test function expresses the displacement of the cantilever beam as a function of its dimensions, material properties, and applied load:

$$D = \frac{4x_l^3}{Ex_w x_t} \sqrt{\left(\frac{N_y}{x_t}\right)^2 + \left(\frac{N_x}{x_w^2}\right)^2}, \quad (4.12)$$

where the parameters in this equation are defined in Table 4.1 and the geometry of the test case is illustrated in Figure 4.2. The designer is assumed to have free choice in selecting the dimensions of the beam, corresponding to a  $n_x = 3$  design space with parameter points  $\mathbf{x} = [x_l, x_w, x_t]^\top$ . The loadings and Young's Modulus of the beam are uncertain

Parameter	Definition	Probability distribution
$x_l$	Beam length	—
$x_w$	Beam width	—
$x_t$	Beam thickness	—
$D$	Beam displacement	—
$D_0$	Displacement tolerance	—
$N_x$	Horizontal load	$N(500, 100)$
$N_y$	Vertical load	$N(1000, 100)$
$E$	Young's modulus	$N(2.9\text{E}7, 1.45\text{E}6)$

Table 4.1: Definitions of the parameters used in the test case. The loads and Young's Modulus of the beam were considered to be stochastic parameters, with training data generated through MC sampling of the distributions detailed in the right column

parameters, used to generate a probability distribution for the QoI,  $D$ . This quantity represents the total magnitude of displacement, with components in the  $x$  and  $y$  directions, taken at the tip of the beam.

As has been discussed, we do not consider the problem of estimating the LSF: in this test case we have attempted to mimic realistic design requirements through constraints on the weight of the beam and an upper and lower bound on the beam width. The length of the beam itself was treated as a discrete variable with three possible values. Including a discrete variable within the test case was significant as continuous-discrete design spaces are common in engineering applications. Not all variables can take arbitrary values, for instance a component may be manufactured in standard sizes. Alternatively, some parameters such as the number of stringers reinforcing a panel or the number of bolts in a connection may only take integer values [210]. The design space for the cantilever beam test case is illustrated in Figure 4.3. The 20 black circles in plots (a)-(c) represent the locations of the parameter points in the training dataset, while the red crosses indicate the locations of query parameter points for which we wish to estimate the probability distribution of  $D$ , normalised by the displacement tolerance,  $D_0$ . Synthetic training data was generated through Monte Carlo sampling of (4.12) at these locations in the design space.

We note that there is a rich literature in methods for the design of experiments (DoE) based on space filling [211] and adaptive sampling approaches [212]. Adaptive methods attempt to improve the accuracy of a meta-model by balancing the exploration of the design space against exploitation, placing additional sampling points in regions of interest and will be discussed further in the next chapter [213]. However, determining a sampling strategy for generating the training data is not the focus of this chapter: the histogram interpolation algorithm is intended for use with historic data, which may not have been collected according to an organised scheme, and so instead we randomly disperse the training parameter points. This has the interesting effect of creating regions of the design space where the training data is clustered tightly, as well as other regions where the training data is sparse or non-existent.

The response surface of a beam with a length of 100in is illustrated in Figure 4.3(d) (with the stochastic parameters set to their mean values in Table 4.1). This response surface is fairly typical for engineering products: a smooth, continuous surface with regions of relatively large gradients, indicating a rapid deterioration in performance once some limit is reached. The cantilever beam test function is a useful test case in this respect as it exhibits typical behaviour of an engineering product but is computationally cheap to evaluate. Two synthetic datasets were generated to test the histogram interpolation algorithm: the first from Monte Carlo samples taken from the Gaussian distributions described in Table 4.1, the second from a mixture of uniform and Gaussian distributions, meaning that the probability distribution for  $D/D_0$  was no longer a member of the Askey family of probability distributions. Interpolation using this dataset was expected to be more challenging as there was no longer an optimal choice of polynomial chaos with which to establish a probabilistic equivalence with the training data.

We demonstrate the benefits of a non-parametric, data-driven approach by comparing the estimations of the algorithm presented here against a parametric method that assumes that the data is Gaussian and constructs a polynomial interpolation function for the mean and standard deviation using the training data. We found that a polynomial interpolation

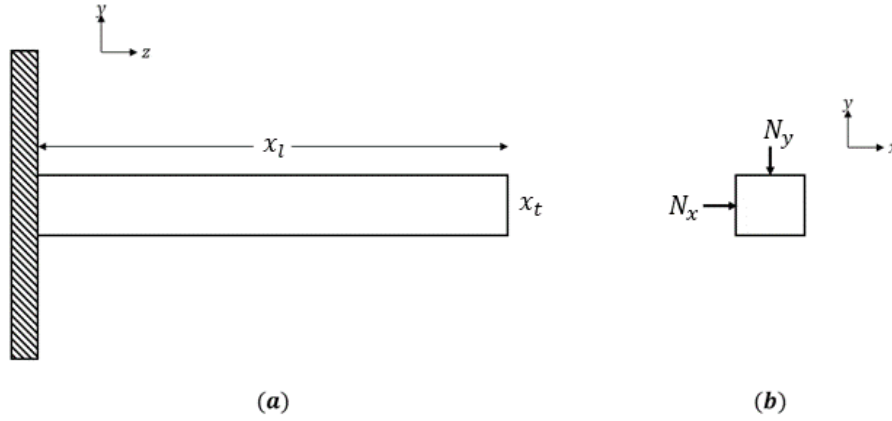


Figure 4.2: Schematic offering two views of the cantilever beam model used as a test case. The dimensions of the beam formed a three variable design space, while the loadings and material properties of the beam were stochastic.

function that was fitted to the entire training dataset performed poorly. Instead, we fitted a separate polynomial to each of the interval values of  $x_l$  to provide a more challenging test of the histogram interpolation algorithm. The maximum polynomial order before the regression became underdetermined was chosen, i.e. order  $p = 2$  for  $x_l = 80 \& 100\text{in}$  and order  $p = 1$  for  $x_l = 120\text{in}$ . In the following, this parametric method is referred to as the ‘polynomial method’.

#### 4.1.3.1 Histogram interpolation with Gaussian-like distributions

Monte Carlo sampling of (4.12) was used to generate a probability distribution for  $D/D_0$  at each of the 20 training points in the design space using  $10^6$  samples drawn from the probability distributions in Table 4.1. The locations of these training points are illustrated in Figure 4.3(a)-(c). The Monte Carlo sampling generated probability distributions for  $D/D_0$  that were Gaussian-like.

The histogram interpolation algorithm was employed to estimate the probability distribution for  $D/D_0$  at four test points in the design space. These represent candidate designs for which the designer wishes to estimate the uncertain QoI. The locations of these test points are also illustrated in Figure 4.3(a)-(c). A probabilistic equivalence was

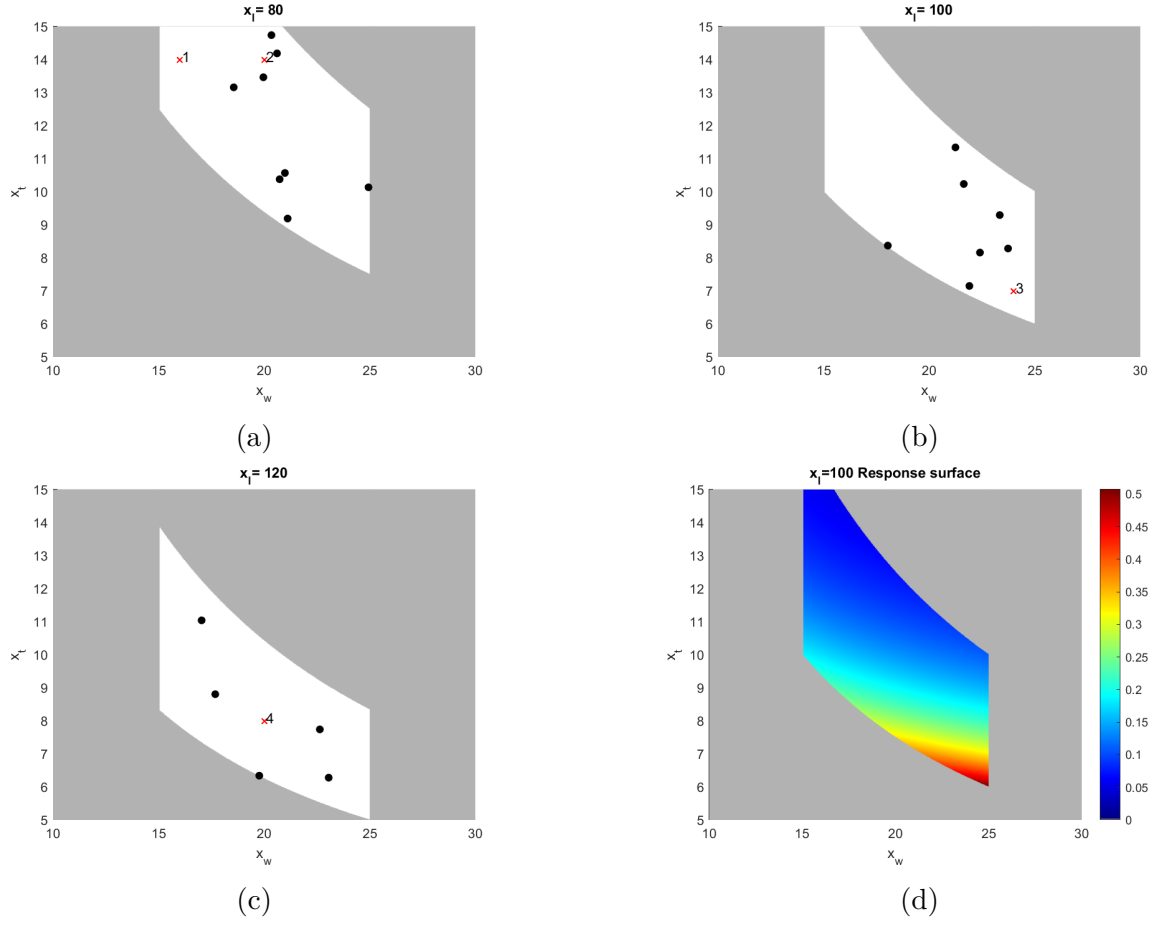


Figure 4.3: (a-c) We mimic realistic design requirements through constraints on the beam weight and width. The black dots indicate parameter points in the training dataset, the red crosses indicate the 4 query parameter points used to test the algorithm. (d) a surface plot of the response surface of the beam, using mean values of the stochastic parameters.

established between the training data and a PCE with a Hermite polynomial basis, as these polynomials are optimal for Gaussian distributions. A convergence study on the order of the PCE was performed, in which the order was incremented until it no longer reduced the statistical distance between the PCE and the training data. For this case the order was set to  $p = 3$ . The results of the histogram interpolation are plotted in Figure 4.4. The estimated PDF, plotted in blue, is compared to the ‘true’ PDF for  $D/D_0$ , plotted in red. The ‘true’ PDF was computed from Monte Carlo sampling of (4.12). As can be seen from Figure 4.4, the histogram interpolation algorithm provides almost an exact

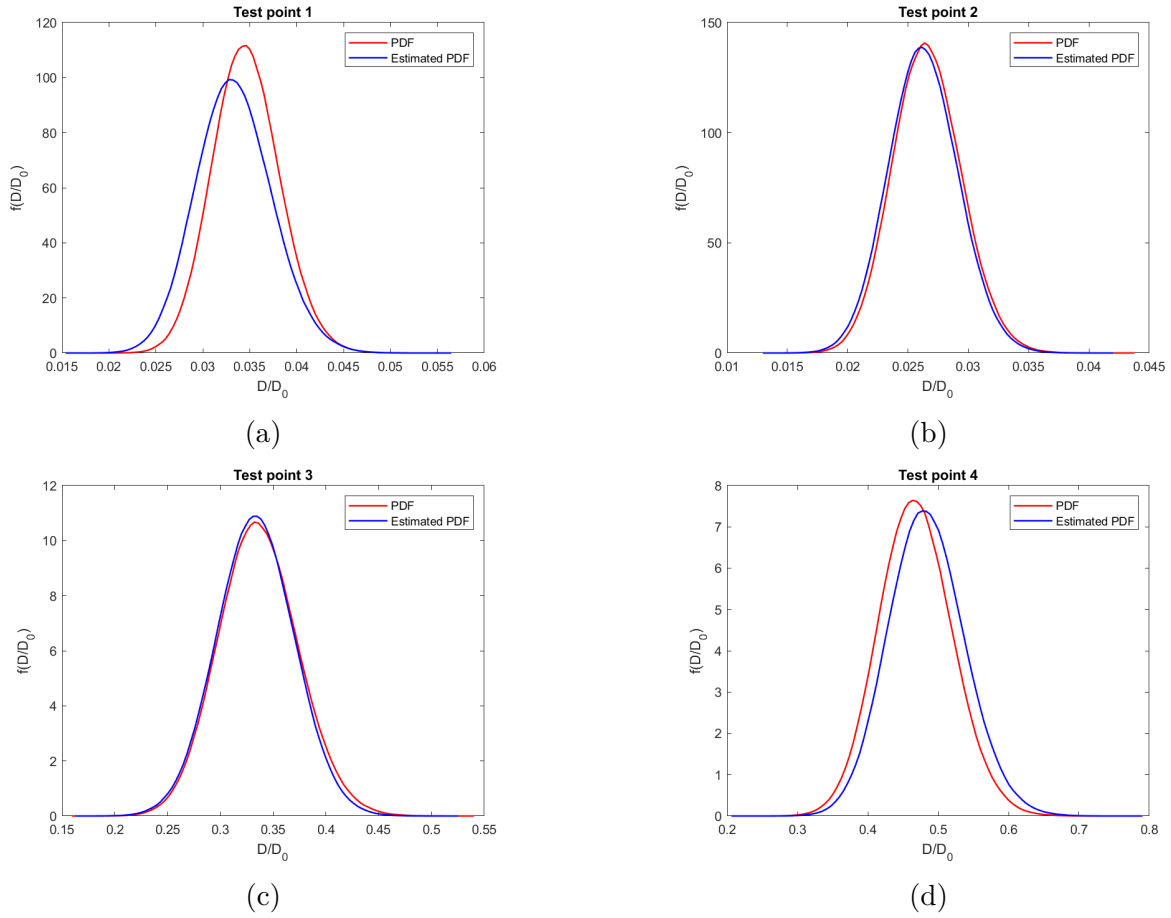


Figure 4.4: (a-d) plots of the estimated probability distribution for  $D/D_0$  (blue) against the true distribution (red), found through direct MC simulation, for each of the four test points.

estimate at points 2 and 3 (Figure 4.4(b)-(c)). The interpolation is less accurate at points 1 and 4, which are more isolated from the training data.

Table 4.2 quantifies the performance of the interpolation through the Kolmogorov-Smirnov distance between the ‘true’ and estimated probability distributions and the percentage error in the mean and standard deviation. In terms of the percentage error in the mean, all four estimates are relatively accurate. The trends in the KS distance and standard deviation confirm our observation that the proximity of the test data was correlated with the performance of the algorithm. With the exception of test point 3, the algorithm significantly outperforms the polynomial method.

Method	Test point	$d^{KS}$	$\Delta\mu$ (%)	$\Delta\sigma$ (%)
Histogram Interpolation	1	0.1441	3.6237	12.2038
	2	0.0417	1.1719	0.9035
	3	0.0276	0.8808	2.8061
	4	0.1155	3.3071	3.5007
Polynomial Method	1	0.2343	6.1308	7.3489
	2	0.0495	1.0539	1.2269
	3	0.2782	7.8364	7.7359
	4	0.6639	23.864	23.8472

Table 4.2: The performance of the algorithm was quantified by comparing the KS distance between the true and estimated distributions, as well as the % differences in the mean and standard deviation.

#### 4.1.3.2 Histogram interpolation with non-Askey scheme distributions

Having demonstrated that the histogram interpolation algorithm was capable of providing reasonably accurate estimations for Gaussian distributions, using a PCE with a basis that was optimal for the family of probability distribution being interpolated, the algorithm was then tested by interpolating a type of distribution with no corresponding optimal Askey polynomial. In this case, probability distributions for  $D/D_0$  were generated at the training parameter points by sampling  $N_x$  and  $N_y$  from the uniform distributions  $U(100, 400)$  and  $U(900, 1200)$  respectively. The Anderson-Darling test was used to reject the null hypothesis that the distributions for  $D/D_0$  belonged to a family of probability distribution within the Askey scheme at the 5% significance level (see Table 2.1), confirming that no optimal orthogonal polynomial existed for this distribution. Note that if the input distributions were known to the designer a PCE with optimal polynomials could be defined, in which the PCE was a function of the uncertain inputs  $\xi$ , along the lines of Eq. (2.5). However, in this case we assume that only the data for  $D/D_0$  is available and so the PCE is defined using the dummy variable  $z$  as in Eq. (4.1).

Hermite polynomials were again used as the multivariate polynomial basis for the PCE. The choice of non-optimal basis polynomial has been shown to effect the quality of the probabilistic equivalence, some polynomial bases might require a higher order PCE expan-



sion to achieve the same level of probabilistic equivalence as others [108]. A convergence study was performed, with the degree of the PCE used set at  $p = 5$ .

Figure 4.8 illustrates the estimated PDF against the ‘true’ PDF found by Monte Carlo sampling of (4.12). As can be seen from this figure and Table 4.3, the performance of the algorithm for all the points is generally lower than for the case of the Gaussian inputs, although the estimations for points 2 and 3 could still be considered to be sufficiently accurate estimations for the purposes of an initial estimate. The results again demonstrate the dependence of the algorithms performance on the proximity of the training data, as can be seen in Figure 4.6 where the performance of the algorithm at each test point, as quantified by the KS distance, is plotted against the  $k$  nearest neighbor (KNN) distance of the test point from the training dataset (with  $k = 3$  and the columns of  $X$  re-scaled to lie in the range  $[-1,1]$ ).

Again, we note from Table 4.3 that the Histogram Interpolation algorithm compared favourably to the polynomial method. However, it is clear from both datasets that the estimate of the algorithm alone is insufficient as it does not convey the level of predictive uncertainty. Finding a suitable statistic for this, that correlates with the performance of the histogram interpolation algorithm as quantified by the KS distance, is the subject of the next section.

Method	Test point	$d^{KS}$	$\Delta\mu$ (%)	$\Delta\sigma$ (%)
Histogram Interpolation	1	0.2607	7.2485	18.2026
	2	0.0223	0.4678	8.4021
	3	0.0688	1.8413	3.2259
	4	0.1460	4.2228	7.4868
Polynomial Method	1	0.3001	8.0457	13.0399
	2	0.064	0.9248	0.6394
	3	0.3179	7.9270	7.4079
	4	0.7220	24.0514	24.0653

Table 4.3: The performance of the algorithm tabulated for the second test case, where the interpolated histogram does not have a corresponding optimal Askey polynomial.

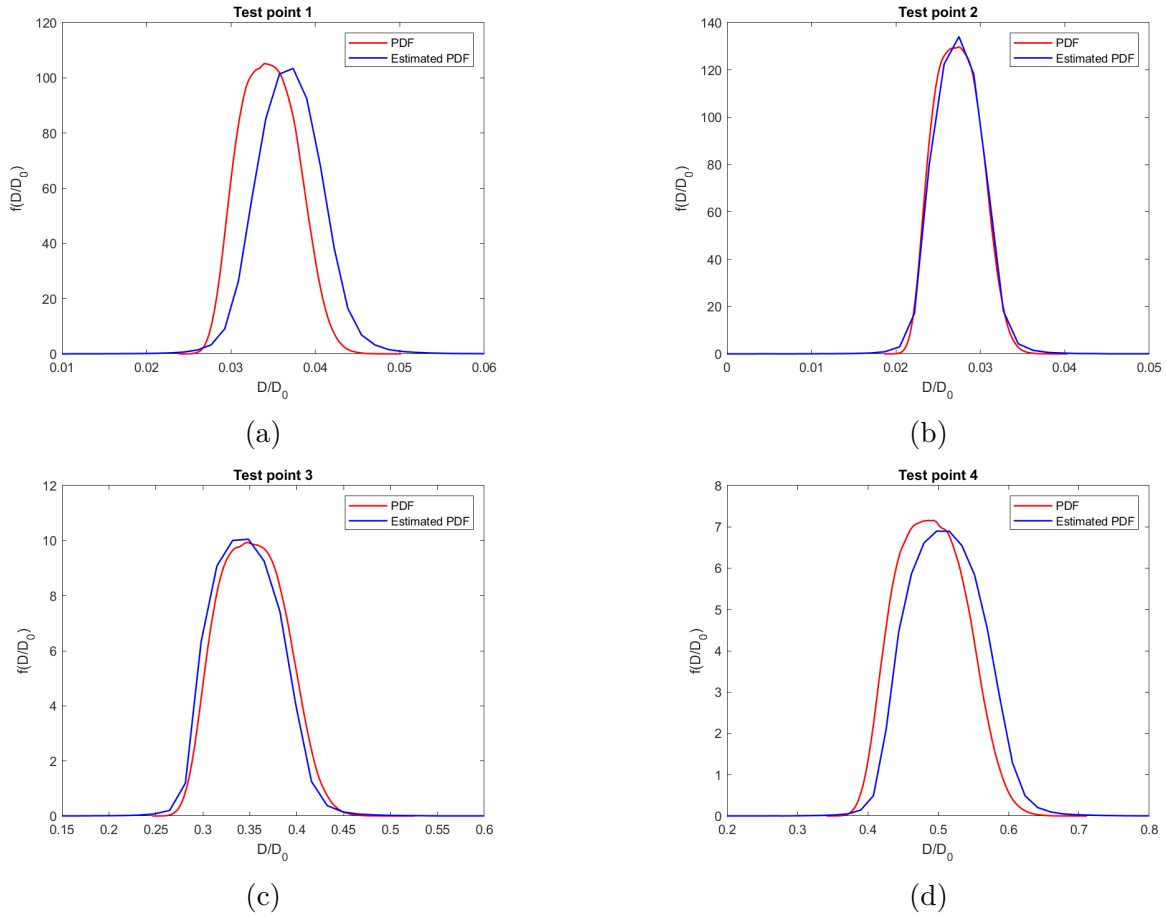


Figure 4.5: (a-d) Plots comparing the estimated PDF (blue) against the true distribution found by direct MC sampling (red), for each of the test points. This time the interpolated distribution did not belong to a family of distribution in the Askey scheme for which an optimal orthogonal polynomial exists.

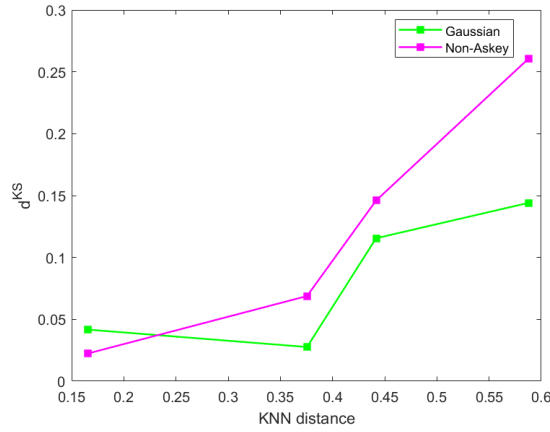


Figure 4.6: The performance of the algorithm, as quantified by the KS distance, is strongly correlated with the proximity of the test point to the training data. Here we plot  $d^{KS}$  against the nearest neighbor distance of the test point from the parameter points in the training dataset.

#### 4.1.4 Predictive uncertainty of histogram interpolation

The test cases presented above have demonstrated that the histogram interpolation algorithm is capable of producing a relatively accurate estimate of a probability distribution in a continuous-discrete design space. Naturally, the accuracy of the interpolation is strongly dependent on the proximity of the test points to the training data, as shown in Figure 4.6. There is motivation in defining a statistic to accompany the estimated probability distribution,  $\hat{f}(w|\boldsymbol{\lambda}(\mathbf{x}^*), \mathbf{x}^*)$ , that indicates the level of predictive uncertainty.

While such an approach may be novel in the context of histogram interpolation, communicating the level of uncertainty in a meta-model is a common requirement. An advantage of GPMs is that the predictive uncertainty may be estimated by leveraging the variance of the Gaussian Process,  $\sigma^2$ . Plots such as Figure 4.7(a) are common in the literature, where a 95% confidence interval may be approximated by  $\mu \pm 2\sigma$ .

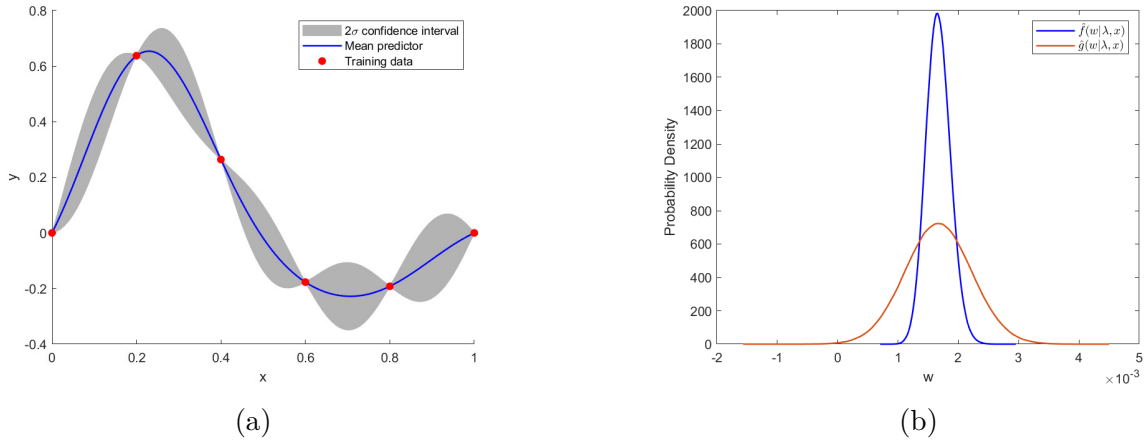


Figure 4.7: (a) An example of how  $2\sigma$  confidence bands can be used to visualise the predictive uncertainty in a GPM. (b) These bands are difficult to interpret for the PCE coefficients, instead we use the statistical dispersion of  $\hat{f}(w|\lambda(\mathbf{x}^*), \mathbf{x}^*)$  and  $\hat{g}(w|\lambda(\mathbf{x}^*), \mathbf{x}^*)$  as a metric for the uncertainty in the estimate.

It is straightforward to visualise the predictive uncertainty when only a single output parameter is estimated directly, however, in the algorithm presented here  $P$  transformed PCE coefficients are estimated simultaneously. The abstract nature of these coefficients makes communicating the level of uncertainty in the estimate challenging as the transformed PCE parameters do not correspond to any physical or statistical quantity. A further complication is that the sensitivity of the PCE to each coefficient varies spatially in the design space. We propose a method that condenses the predictive uncertainties of the  $P$  GPMs into a single statistic.

The approach taken here is to estimate two PDFs through Monte Carlo (MC) simulation of the PCE coefficients and fitting a kernel by maximum likelihood estimation, as discussed above. The first PDF,  $\hat{f}(w|\lambda(\mathbf{x}^*), \mathbf{x}^*)$ , is generated from a set of PCE coefficients predicted by  $\hat{\mu}_k$  and random sampling of the generating distribution for  $\xi$  (Figure 4.7(b)). A second PDF,  $\hat{g}(w|\lambda(\mathbf{x}^*), \mathbf{x}^*)$ , is estimated by sampling each  $\tilde{\lambda}_k$  from its posterior distribution, in addition to the generating distribution for  $\xi$ . An estimate of the predictive uncertainty is made by evaluating the statistical distance between  $\hat{f}(w|\lambda(\mathbf{x}^*), \mathbf{x}^*)$  and  $\hat{g}(w|\lambda(\mathbf{x}^*), \mathbf{x}^*)$ , using a metric that emphasises the differences in the statistical dispersion between the two probability distributions. There are a number of normalised measure of

the statistical distance than be used for this, such as the Hellinger distance [214] or the dissimilarity index [215]. Engineering designers are interested in measuring the similarity of one distribution against another. As such we use the terms metric and measures in the informal sense; It is clear that these measures of similarity do not conform fully to the properties of a Metric Space. The dissimilarity index is defined as the integral of the absolute difference between the distributions:

$$d^{fg} = \frac{1}{2} \int_{-\infty}^{\infty} |\hat{f}(w|\boldsymbol{\lambda}(\mathbf{x}^*), \mathbf{x}^*) - \hat{g}(w|\boldsymbol{\lambda}(\mathbf{x}^*), \mathbf{x}^*)| dw, \quad (4.13)$$

where we denote the statistical distance as  $d^{fg}$  in this context. Regardless of whether the Hellinger distance or dissimilarity index is used, as the confidence of the meta-model estimate increases the kriging variances will shrink and consequently  $\hat{g}(w|\boldsymbol{\lambda}(\mathbf{x}^*), \mathbf{x}^*) \rightarrow \hat{f}(w|\boldsymbol{\lambda}(\mathbf{x}^*), \mathbf{x}^*)$  and  $d^{fg} \rightarrow 0$ .

This method can be used to visualise the uncertainty in the estimates of the histogram interpolation algorithm within the design space. In Figure 4.8(a) we use the meta-model developed in the previous section to estimate the probability distribution for  $D/D_0$  on a mesh of parameter points that fills the design space. The KS distance between the estimated distribution and the ‘true’ distribution found by direct Monte Carlo simulation of (4.12) is used to quantify the performance of the algorithm on these test points. Figure 4.8(a) displays the spatial variation in the performance of the algorithm, as quantified by the KS distance. The procedure outlined above is used to estimate the level of uncertainty in the interpolation at each of these test points. The intention is to create a surface for the spatial variation of the predictive uncertainty that is consistent with the surface representing the algorithm’s performance. The choice of metric for the statistical distance will influence the appearance of this surface. In Figure 4.8(b)-(c) the surface representing the predictive uncertainty is plotted, using the Hellinger distance and the dissimilarity index respectively. For the purposes of comparison we include a metric for the statistical

distance that is not dependent on integration of the two distributions, which can have a filtering effect, but rather on the ratio of variances (RoV) of the two distributions:

$$d^{fg} = 1 - \frac{\sigma_f^2}{\sigma_g^2}. \quad (4.14)$$

Assuming that  $\sigma_g \geq \sigma_f$ , then this distance will also be normalised. The surfaces generated with the RoV used as  $d^{fg}$  are plotted in Figure 4.8(d).

As can be seen, the surfaces for the predictive uncertainty in Figure 4.8(b)-(c) appear to be consistent with the surface representing the performance, with the surfaces based on the dissimilarity index appearing to be the most consistent with Figure 4.8(a). This observation is confirmed by Table 4.4, which tabulates the Euclidean distance between the surfaces representing the predictive error and the surface representing the performance of the algorithm. In this case we find that a normalised metric based on the second statistical moment is too conservative. Integration of the tails of the distributions appears to give a more accurate metric for the predictive error of the histogram interpolation. For the sake of clarity, statistical distances greater than 0.4 have been plotted as the same color in Figure 4.8.

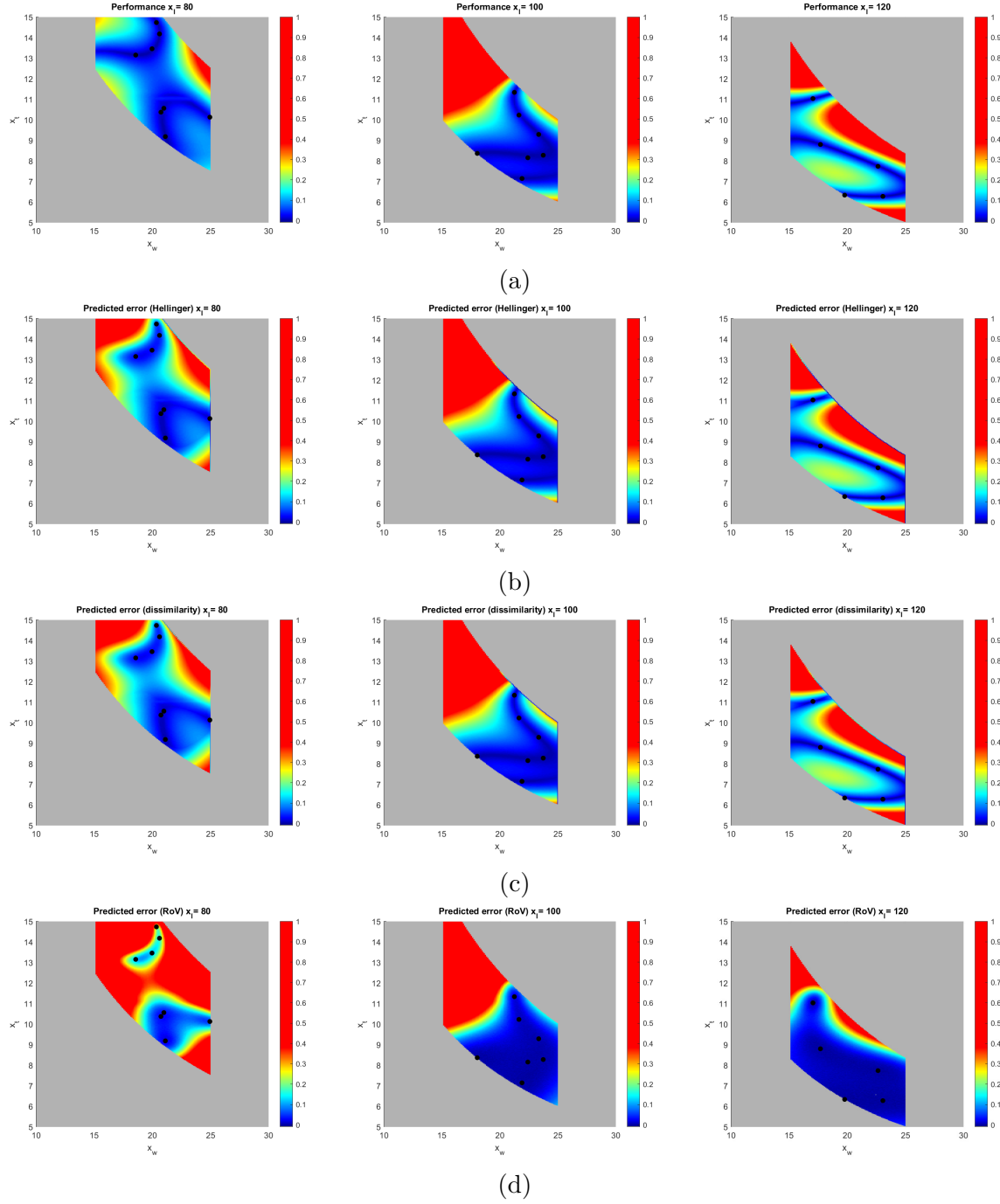


Figure 4.8: (a) Surfaces representing the actual performance of the histogram interpolation algorithm, quantified by the KS distance. (b-d) Surfaces representing the predictive uncertainty using the Hellinger distance, dissimilarity index, and ratio of variances for  $d^{fg}$ .

In this half of the chapter we have introduced an algorithm for DSE that processes raw data and performs a first estimate of the uncertain performance of a new design without

L	Dissimilarity index	Hellinger distance	Ratio of variances
80	20.25	11.75	63.84
100	24.70	42.95	13.81
120	39.02	45.34	27.66
Total	83.97	100.04	105.31

Table 4.4: The Euclidean distances between the surfaces representing the uncertainty in the predictions, displayed in Figure 4.8(b-d), and the surfaces representing the actual performance of the algorithm (4.8(a)).

requiring further evaluations of computer code or experiments. In the next section we discuss a novel method for DSE which is intended to give a more accurate evaluation of a candidate design in the conceptual design phase by combining a neural network with the outputs of an inexpensive, high-level computer code.

## 4.2 Bi-fidelity modelling with Knowledge Based Neural Networks

As has been discussed in previous chapters, an attractive strategy for mitigating the computational cost of UQ algorithms is to supplement the results of the most accurate models of a system, referred to as high-fidelity models, with models that are less computationally expensive. These low-fidelity models may have simplified physics, a coarser meshing or less detailed geometries and as a consequence are not as accurate. A popular area of research has been in developing multi-fidelity methods that can leverage relatively scarce high-fidelity data with low-fidelity data that is less accurate but much cheaper to obtain.

A particularly popular example of a multi-fidelity method is co-kriging [216, 217]. An auto-regressive model is constructed to combine datasets of multiple fidelities, with the outputs of the model treated as a realisation of a Gaussian random variable. A Markov property is assumed: the coarse models cannot add additional information at locations where high-fidelity data is available. Co-kriging has been used widely among many disciplines, however, there are a number of drawbacks that make its application to industrial



problems challenging. Training the kernel involves repeated inversions of the correlation matrix. This matrix scales with the size of the training dataset, which can become intractable for large datasets [218]. Secondly, the kernel is sensitive to the choice of kernel function, the choice of which is not trivial in high dimensional problems [219]. Conversely, neural methods have proved efficient for high dimensional meta-modelling [220, 221]. However, while neural networks can act as powerful approximators, they are purely data-driven and function as black box models. Predictions made by neural networks are often criticised due to this lack of transparency [222]. For these reasons there is interest in developing deep neural networks which incorporate physical knowledge of a system [223].

Knowledge based neural networks use prior knowledge of a system to inform their predictions. In the original formulation for a hybrid learning system proposed in Towell and Shavlik (1994) the prior knowledge is encoded in the network in the form of symbolic rules which determine the neural network structure and the initialisation of its weights. It was demonstrated that the performance of a classifier could be improved by incorporating these ‘domain theories’, especially if the data used to train the classifier was limited [224, 225]. An alternative formulation for KBaNNs was later introduced in Wang and Zhang (1997) in which prior knowledge is embedded in a neural network in the form of a ‘knowledge layer’ consisting of empirical functions [226]. Later works replaced the empirical functions with a low-fidelity model, producing a neural network for bi-fidelity modelling [227, 228]. The general architecture for a KBaNN capable of bi-fidelity modelling is illustrated in Figure 4.9.

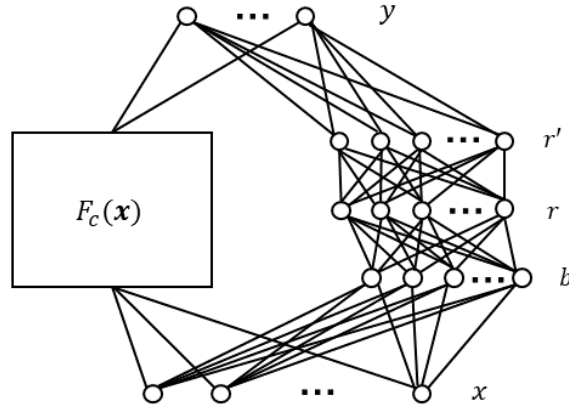


Figure 4.9: Architecture of a KBaNN for bi-fidelity modelling. A correction is made to the outputs ( $\mathbf{y}$ ) of a coarse, low-fidelity model  $F_c(\mathbf{x})$  by a network consisting of boundary neurons ( $b$ ), region neurons ( $r$ ), and normalised region neurons ( $r'$ ) for an input  $\mathbf{x}$ .

As was referenced in the previous chapters' discussion of data fusion, the high computational cost associated with individual simulations make CFD simulations a popular application in the multi-fidelity modelling literature. There are two approaches to defining a low-fidelity CFD simulation: the first approach is to alter the physics of the model so that it is cheaper to evaluate. For instance, Direct Numerical Simulation (DNS) is a high fidelity technique for modelling turbulent flows. The Navier-Stokes equations are solved at every length scale, allowing the turbulence to be completely resolved and hence providing complete knowledge of the flow. The drawback to the method is that it is very computationally expensive, with each simulation typically taking a number of days to run[229]. A number of works in the literature leverage a small set of DNS results with Reynolds-Averaged Navier-Stokes (RANS) simulations, which are less accurate due to simplifications in the turbulence closure but also cheaper to evaluate [132, 230, 231]. An alternative approach is to use the same model with meshes of varying coarseness in order to create low-fidelity surrogates. Examples of such an approach have been used in the literature for the design of a transonic compression rotor in Shapar et al (2011)[232] and in Shah et al (2015)[233] for airfoil design. This approach is similar to multi-scale mod-

elling, where a relatively coarse model used to model the entire domain can be informed by more accurate models of a sub domain (see e.g. [12, 13]).

In this chapter we introduce a novel KBaNN architecture and demonstrate the power of the method by developing a local Navier-Stokes approximator that corrects the results of low-fidelity CFD simulations using a coarse mesh. The approximator is a KBaNN that has been trained using a dataset of high-fidelity and low-fidelity CFD data. A crucial distinction between this work and the multi-fidelity modelling literature is that the multi-fidelity approximator is tested on data harvested from a flow around a geometry that is not the same as the geometry that was used to train it. In the test case presented here, the approximator is trained on a dataset comprising simulations of a two-dimensional channel flow and flow over a converging channel before being tested on a flow across a NACA 2412 airfoil. By learning the local corrections that must be made to the velocity field due to the coarseness of the mesh for a simple geometry, the system can be generalised to flows around more complex geometries that share similar physics. As computational based design becomes an increasingly important part of designing new products, databases of simulation results will begin to accumulate. This chapter presents a framework by which these databases may continue to add value, by training a multi-fidelity system that can be used to inform future designs.

### 4.2.1 Knowledge Based Neural Networks (KBaNNs)

Knowledge Based Neural Networks are a bi-fidelity machine learning architecture that allow the outputs of a coarse scale model,  $F_c(\mathbf{x})$ , to inform the predictions of a neural network. Having been trained using a dataset comprising outputs of a high-fidelity model,  $F_e(\mathbf{x})$ , the KBaNN corrects the outputs of the coarse model to emulate the output of  $F_e(\mathbf{x})$ . The generalised KBaNN architecture is illustrated in Figure 4.9. The architecture is based on the KBaNN proposed in Wang et al (1997) but adapted for bi-fidelity modelling. The formulism has also been modified to produce an additive rather than multiplicative correction to  $F_c(\mathbf{x})$ . It also now incorporates  $L2$  regularisation. The neural

architecture consists of five layers: input layer  $\mathbf{x}$ , boundary layer  $\mathbf{b}$ , region layer  $\mathbf{r}$ , normalised region layer  $\mathbf{r}'$ , and output layer  $\mathbf{y}$ . Neurons in the boundary layer are related to the KBaNN inputs through the parameters  $V = [\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(n_b)}]$  associated with each of the  $n_b$  boundary neurons:

$$\mathbf{b}_i = \mathbf{v}^{(i)T} \mathbf{x}, \quad i = 1, \dots, n_b. \quad (4.15)$$

This is a generalised formulation for the boundary neurons in which linear boundaries are assumed. It is noted in Wang et al (1997) that there is the potential to include problem specific boundary functions if these are known. Neurons in the region layer are evaluated in a similar fashion to the hidden layers of a Multi-Layer Perceptron (MLP), with weighted connections between the neurons in the region layer and boundary layer:

$$\mathbf{r}_i = \prod_{j=1}^{n_b} \sigma(\alpha_{ij} \mathbf{b}_j + \theta_{ij}), \quad i = 1, \dots, n_r, \quad (4.16)$$

where  $\sigma(\cdot)$  refers to a sigmoid activation function and  $n_r$  the number of region neurons. The parameters  $\alpha_{ij}$  and  $\theta_{ij}$  refer to the weight and bias in the connection between the  $i^{\text{th}}$  region neuron and  $j^{\text{th}}$  boundary neuron. Rational function based neurons [234] are used in the normalized region layer to normalize the region layer outputs:

$$\mathbf{r}'_i = \frac{\mathbf{r}_i}{\sum_{j=1}^{n_r} \mathbf{r}_j}, \quad i = 1, \dots, n_r. \quad (4.17)$$

Finally, an additive correction with second order neurons [235] is applied to the outputs of the coarse model in the output layer:

$$\mathbf{y}_j = \beta_j F_{cj}(\mathbf{x}) + \sum_{i=1}^{n_r} \rho_{ij} \mathbf{r}'_i + \beta_{0j}, \quad j = 1, \dots, n_y, \quad (4.18)$$

where  $F_{cj}$  refers to the  $j^{\text{th}}$  output of the coarse model. The parameter  $\rho_{ij}$  weights the contribution of the  $i^{\text{th}}$  region to the  $j^{\text{th}}$  output neuron. The KBaNN is trained using a

dataset that includes  $n$  evaluations of the fine model and the corresponding coarse model evaluations:  $[\mathbf{x}^{(i)}, F_e(\mathbf{x}^{(i)}), F_c(\mathbf{x}^{(i)})]$ ,  $i = 1, \dots, n$ . The error in the KBaNN prediction for the  $i^{\text{th}}$  element of the training set is given by:

$$\epsilon = \frac{1}{2} \sum_{j=1}^{n_y} (\mathbf{y}_j^{(i)} - F_{e_j}(\mathbf{x}^{(i)}))^2 + \delta \mathcal{C}(\Phi), \quad (4.19)$$

i.e. the sum of the mean square error (MSE) and a regularisation function  $\mathcal{C}(\cdot)$  that is dependent on the complexity of the KBaNN, with:

$$\mathcal{C}(\Phi) = \sum_{i=1}^{n_b} \|\mathbf{v}^{(i)}\|_2^2 + \sum_{i=1}^{n_r} \sum_{j=1}^{n_b} \alpha_{ij}^2 + \sum_{i=1}^{n_r} \sum_{j=1}^{n_b} \theta_{ij}^2 + \sum_{i=1}^{n_r} \sum_{j=1}^{n_y} \rho_{ij}^2 + \sum_{j=1}^{n_y} \beta_{0j}^2 + \sum_{j=1}^{n_y} (1 - |\beta_j|)^2, \quad (4.20)$$

where  $\Phi$  is the set of all KBaNN hyperparameters and the parameter  $\delta$ , the regularisation constant, weights the contribution of the model complexity term to the error. Penalising model complexity in this way is intended to produce a more parsimonious model. Whilst regularisation techniques in MLP's typically act to force the network's prediction to zero, in the case of the KBaNN the regularisation is instead used to punish predictions that deviate from the outputs of the coarse model. The KBaNN is trained through back-propagation using gradient descent optimisation. In this chapter the adagrad algorithm for gradient based optimisation is used [236]. Employing an adaptive learning rate, the magnitude of the updates made to the KBaNN parameters each iteration is tailored to the frequency with which the features in the training set occur and is well suited for sparse datasets [236]. More details on the error back-propagation, including the derivatives for the prediction error and Adagrad may be found in Appendix A2. An open source code for implementing the KBaNN is available in MATLAB [237] and Python [238].

### 4.2.2 KBaNN architecture for a local Navier-Stokes approximation

In this chapter a KBaNN is used to perform a local correction to the results of a low-fidelity CFD simulation of a two-dimensional laminar flow. The KBaNN is trained using data taken from CFD simulations of a flow at two levels of meshing: a fine mesh which has fully converged to a solution (i.e. the mesh quality no longer affects the solution) and a coarser, more inaccurate mesh which is less computationally expensive to run. Having learnt the discrepancies between the high and low-fidelity mesh the KBaNN can then be employed to make corrections to CFD simulations of flows with similar physics.

Local velocity data is extracted from each of the simulations by superimposing  $n_g \times n_g$  sample grids on the flow and evaluating the velocity point at each data point. This is illustrated in Figure 4.10, where a grid of data points is superimposed over a square mesh. For a two-dimensional flow with no separation there are two configurations these sample grids may take depending on whether a wall is present. A KBaNN is then trained for each of these to situations i.e. a KBaNN trained for sample grids in the free-stream and a separate KBaNN trained for sample grids in the boundary layer, with a row of points inside a wall. The inputs of the KBaNN are the horizontal and vertical velocities along the ‘inlet’ side of the sample grid and the outputs the horizontal and vertical velocities of the entire sample grid. The output dimensions therefore scale quadratically with  $n_g$ , which demonstrates why it is advantageous to use a neural method for the bi-fidelity modelling as the number of dimensions can be significant even for a relatively small sample grid. Note that the spatial location of the grid is not included in the training data. This allows the KBaNN to be used on sample grids in the same flow in different locations or on sample grids in flows with similar physics but different geometries.

In order to prevent the grid size,  $L$ , from impacting the results the velocity data is scaled by a factor  $\kappa = \frac{L}{L^*}$ , where  $L^*$  is the characteristic length scale of the flow. The KBaNN is used to predict the normalised velocities  $u^* = \kappa u$  and  $v^* = \kappa v$ .

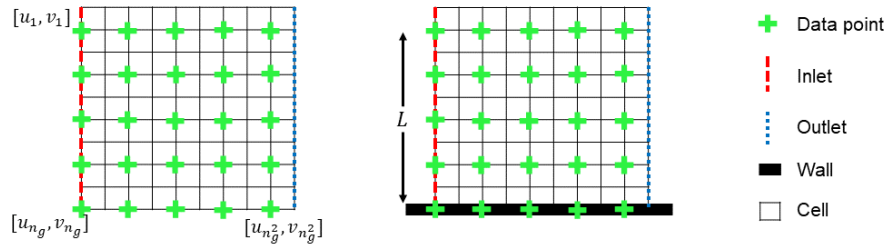


Figure 4.10: Schematic of the  $n_g \times n_g$  sample grids used for the local Navier-Stokes approximation. There are two configurations for the grids: in the freestream or near a wall with a boundary layer present. Velocity data from the inlet of the grids were used as an input to the KBaNN, which estimated the velocities for the entire sample grid.

The architecture of the KBaNN used for local Navier-Stokes approximation is illustrated in Figure 4.11. Note that this is a slight modification of the general KBaNN architecture illustrated in Figure 4.9. The network is effectively split in half, with one portion of the network learning the corrections to the horizontal velocity and the other portion the vertical velocity correction. Splitting the network in this way was found to be more efficient as fewer neurons in total were required to fit the training data. This is a common technique that has been used in modular neural networks (see, e.g. [239]).

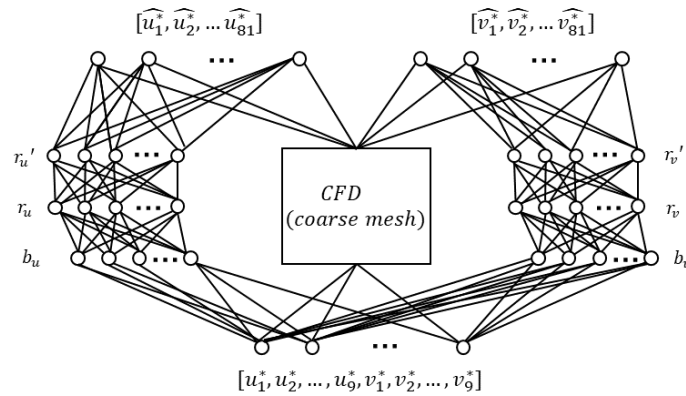


Figure 4.11: A modified KBaNN architecture was used for the local Navier-Stokes approximator. It was found that training a separate set of boundary and region neurons for each velocity component was more efficient (fewer neurons were required overall) than the general KBaNN architecture illustrated in Figure 4.9.

### 4.2.3 Test case and results

The potential of the KBaNN architecture described above to act as a local Navier-Stokes approximator was demonstrated through two test cases. As described above, the system consisted of two KBaNNs: one of which was trained on data taken from the freestream and the other trained on grids located in the boundary layer. The KBaNNs were trained using data taken from seven simulations of a simple two-dimensional channel flow. In six of the simulations the channel walls were parallel, while in the seventh simulation one of the walls was inclined, creating a converging channel. The system was first validated using a channel flow simulation at a different Reynolds number before being applied to the more challenging case of a flow around a NACA 2412 airfoil. The intention was to train the system with a small dataset harvested from a relatively simple geometry and then to use it to make local Navier-Stokes estimations of a flow around a more complex geometry that shared similar physics.

For the test case of a channel flow at a different Reynolds number it was found that the system greatly improved the accuracy of the coarse mesh simulation. In the case of the airfoil the system led to an overall improvement in accuracy, despite the more complex geometry of the flow around the airfoil compared to the training data.

#### 4.2.3.1 Training and channel validation

Bi-fidelity CFD simulations of a laminar flow through a two-dimensional channel were used to train two KBaNNs so that the system could learn the discrepancies between the fine and coarse meshes. Figure 4.12(a) shows the two-dimensional channel flow and the locations of the sample grids which were used to train the KBaNNs. One KBaNN was trained using the blue sample grids located in the free stream, while the other was trained on the data from the red sample grids lying in the boundary layer attached to the channel wall. In Figure 4.12(b) a sample grid is superimposed on plots of the coarse and fine meshes. As can be seen the spatial resolution of the fine mesh is significantly shorter than



the length of the sample grid. It was chosen to be short enough to no longer affected the results of the simulation. The spatial resolution of the coarse mesh is around 8 times larger than that of the fine mesh. No-slip boundary conditions are applied to the boundaries at  $y = 0$  and  $y = 6$ , an outflow at the boundary at  $x = 10$ , and finally a uniform inflow was applied between  $x = 0$  and  $x = 1$ . Simulations of the channel were run at varying values of the inflow velocity,  $U$ , for each mesh in order to generate a set of training data.

The training dataset was generated from the velocity data of the square,  $9 \times 9$  sample grids illustrated in Figure 4.12(a). As has been discussed above, the velocities were scaled to account for effects introduced by changing the size of the grid. Data was collected from grids of size  $L = 0.1, 0.5$ , and  $0.75$  so that these effects could be learned by the system.

A common problem in training neural networks is over-fitting the network to the training data. A strategy to prevent over-fitting is to hold aside some training data and use this as a validation dataset. Data was held aside from the channel configuration illustrated in Figure 4.12(a) and also from a single simulation of the two-dimensional flow through a converging channel. This flow is illustrated in Figure 4.12(c). The spatial resolutions of the fine and coarse mesh were the same as with the other simulations, as can be seen in Figure 4.12(d). The bottom wall was inclined to give a 5% slope and the sample grids placed in different locations to the configuration in Figure 4.12(a). By including data from a flow around a different geometry in the validation set the KBaNNs were encouraged not to over-learn the discrepancies between the fine and coarse meshes. The following simulations were used for the training and validation datasets:

**Training data:** Channel flow simulations at  $Re = [300, 480, 600, 720, 900]$

**Validation data:** Channel flow simulation at  $Re = 660$  and converging channel flow at  $Re = 480$

with data extracted from the sample grids locations displayed in Figures 4.12(a) and 4.12(c) at three length scales. The KBaNN architecture illustrated in Figure 4.11 was used. The sub networks were relatively small, with between 8-10 neurons in each layer and an equal number of neurons in the boundary and region layers.  $9 \times 9$  sample grids required each sub network to have 18 neurons in the input layer and 81 neurons in the output layer. Error back-propagation was used to train the KBaNNs, more details of which may be found in Appendix A. A gradient based optimisation algorithm was used to find values of the KBaNN parameter set  $\Phi$  that minimised the mean squared error (MSE) of the validation dataset.

Figure 4.13 displays the predictions of the KBaNNs for two sample grids in the channel simulation at  $Re = 660$  that was used in the validation dataset. These sample grids are labelled in Figure 4.13(a). In 4.13(b-c) the fine mesh solution is plotted against the estimations of the KBaNNs, with the coarse mesh solution included for comparison. As can be seen there is very good agreement between the fine mesh solution and the predictions of the KBaNNs. This agreement is reflected in Table 1, which tabulates the Residual Sum of Squares (RSS) differences between the coarse mesh and fine mesh solutions and between the KBaNN estimations and the fine mesh solution. Figure 4.14 illustrates the performance of the system as a function of the number of grids included in the training data. The RSS error between the fine and coarse mesh solutions are included as a comparison, represented by the horizontal lines in the figure. Increasing the scarcity of the training data reduced the performance of the system, however, it was found that even the KBaNNs trained with a minimal amount of data were more accurate than the coarse mesh solution in isolation.

This test demonstrates that the system can give accurate predictions for a flow that shares the same geometry as the flow used to train it. As can be seen in Figure 4.13 the additive corrections made by the relatively small KBaNN effectively act as a high-order transformation of the coarse mesh solution.

Sample grid	Velocity component	RSS between coarse and fine mesh	RSS between KBaNN and fine mesh
1	$u^*$	7.2058e-4	8.3983e-6
	$v^*$	6.6582e-5	1.0856e-6
2	$u^*$	4.0879e-4	9.2442e-7
	$v^*$	2.5176e-4	1.2501e-8

Table 4.5: The residual sum of squares (RSS) between the KBaNN predictions and the fine mesh for each of the sample grids pictured in Figure 4.13. The corrections made by the KBaNNs greatly improve the accuracy of the solution

#### 4.2.3.2 Airfoil test case

Having demonstrated that the KBaNNs were capable of giving highly accurate estimations of the local velocity field for data harvested from a geometry identical to that used to train it, the system was tested on a test dataset harvested from a more complex, unfamiliar geometry. A laminar flow around a NACA 2412 airfoil at  $Re = 480$  was chosen as the test geometry. Estimations are made challenging in the boundary layer by the curvature of the airfoil and in the freestream by the displacement caused by the airfoil.

The flow is visualised in Figure 4.15(a), with the locations of the sample grids superimposed. A sample grid was placed in the boundary layer on the top surface of the airfoil. Figure 4.15(b) displays this grid, superimposed on the two meshes. One challenge for the system is that the grid sits on a section that is locally flat in the coarse mesh, while the box in the coarse mesh curves to match the contours of the airfoil. The predictive power of the system is dependent on the data used to train it. Consequently, the sample grid in the boundary layer is situated upstream of the separation point, as the system has no experience of a separated boundary layer. Similarly, the sample grid placed in the freestream is located above the airfoil wake. The intention of the test case is to demonstrate that a system trained using data taken from simple flows can be used to make estimations in more complex flows, provided that the flow physics are sufficiently similar. Making predictions in separated boundary layers and wakes would require a more expansive training dataset, including flows with these features.

Figure 4.16 displays the predictions of the KBaNNs for the sample grids around the NACA 2412 airfoil. The results of the fine mesh and coarse mesh CFD simulations are plotted for comparison. The RSS between the KBaNN predictions and the fine mesh results are tabulated in Table 2, along with the RSS between the results of the coarse and fine meshes for these grids. As with the channel test case, the freestream KBaNN is able to significantly improve the accuracy of the coarse mesh. The boundary layer KBaNN also leads to an overall increase in accuracy, although the system has difficulty estimating the vertical velocity of this grid due to the curvature. Nevertheless, this test shows that it is possible to use the KBaNNs on flows around different geometries to those in the training data provided the physics of the flow is sufficiently similar.

There are several differences between the sample grids used to test the KBaNNs and the grids used in the training and validation sets which make predictions challenging. Firstly, the sample grids are smaller spatially. The KBaNNs accept the velocity components of the sample points along the leading edge of the sample grid as inputs; the KBaNNs are not trained to recognise the effect of altering the size of the sample grid explicitly, instead the effects of the altered grid size are implicit in the velocity field of the coarse mesh. Secondly, while the airfoil surface may be locally flat, there is curvature to the boundary layer of the airfoil which is unlike the boundary layer KBaNN has been trained on. As can be seen in Figure 4.15(b) the bottom edge of this sample grid curves to match the shape of the airfoil, rather than being arranged in regular, evenly spaced rows and columns as in the training and validation sets. For the case of the fine mesh this effect is not particularly significant as the airfoil is locally flat, but aligning the sample grid to the top surface of the airfoil when it is meshed more coarsely requires a rotation of the grid. The vertical component of the velocity will be negative in the rotated grid. The flows used to train the boundary layer KBaNN all have positive vertical velocities, so the KBaNN is tested on a dataset that partly lies outside the range of the training data.

As can be seen from the table, the predictions of the KBaNN are a significant improves the accuracy of the coarse mesh velocity field. The improvement is particularly significant

Sample grid	Velocity component	RSS between coarse and fine mesh	RSS between KBaNN and fine mesh
1	$u^*$	1.618e-6	3.1175e-7
	$v^*$	4.1606e-7	1.713e-6
2	$u^*$	5.2534e-5	2.0391e-6
	$v^*$	1.9255e-4	9.094e-6

Table 4.6: The RSS between the KBaNN predictions and the fine mesh for each of the sample grids pictured in Figure 4.16(a). For comparison the RSS between the coarse and fine mesh is included. As can be seen, the KBaNN improves significantly on the fine mesh prediction.

for the sample grid in the freestream, labelled sample grid 2 in Figure 4.16(a). As can be seen from Figure 4.16(c) the freestream velocity field of both the fine and coarse mesh is approximately flat. The additive correction provided by the KBaNN acts to translate the coarse mesh velocity field towards the fine mesh. The boundary layer KBaNN also improves the prediction of the velocity field, as can be seen in Figure 4.16(b), although at the cost of slightly underfitting the velocity field at the wall. However, given that the velocities in this sample grid lie partially outside of the training data this is an encouraging result. The test case is intended as a proof of concept, demonstrating that it is possible for a KBaNN to apply a local approximation to velocity fields that it has not encountered before by learning from simpler flows. An area of future development is to train a KBaNN with a more diverse set of training data, to allow it to be applied to a wider variety of flows.

## Summary

This chapter has introduced two meta-modelling approaches that can be used for DSE, both of which leverage machine learning techniques. The histogram interpolation algorithm defines a data-driven approach to DSE, in which estimations are made without requiring the training data to belong to a particular family of probability distribution and without requiring further evaluations of a computer code. In a departure from other methods, we proposed a statistic to quantify the level of uncertainty associated with the

estimated probability distribution. This statistic was found to closely correlate with the actual performance of the algorithm for a test case in which synthetic data in a continuous-discrete design space was interpolated. The second meta-modelling approach we described here employed Knowledge-Based Neural Networks (KBaNNs), a data-driven method in which a neural network is used to inform the output of a coarse computer code. Allowing the KBaNN to be informed by the low-fidelity model of the system allows the KBaNN to capture some of the physics of the system and in so doing helps to address the criticism of neural networks that they are black boxes which are completely data-driven. The test case, involving corrections to a two-dimensional velocity field from a RANS simulation, illustrates how KBaNNs may be used in the future to estimate the performance of new designs through a meta-model that is informed from the results of simulations that share similar physics.

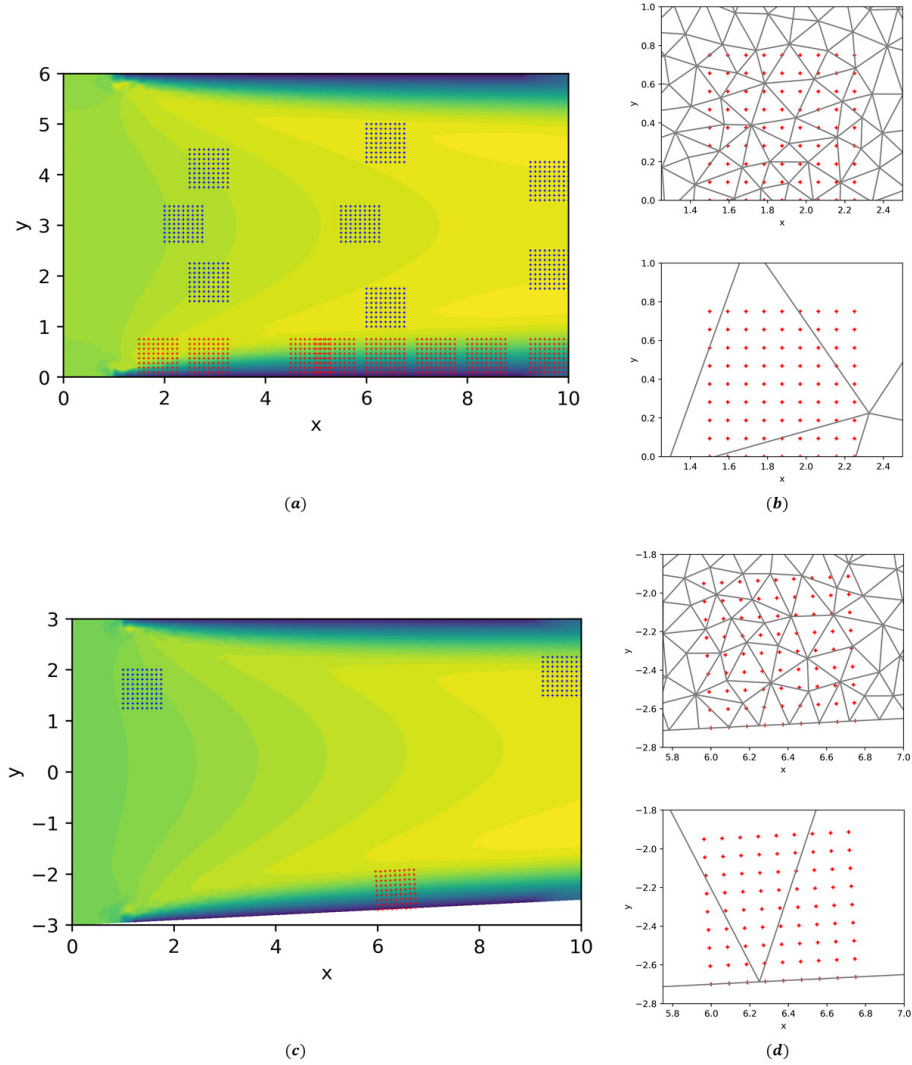


Figure 4.12: (a) A visualisation of the two-dimensional channel flow used to train the KBaNNs. The locations of the sample grids used to harvest velocity data are superimposed. Blue grids correspond to sample grids in the freestream, red to grids that bound the wall and capture the boundary layer. (b) Close ups of the mesh near the wall of the channel, with a sample grid superimposed. (c) A simulation of the flow over a slightly converging channel was included in the validation data set to prevent overfitting (d) the spatial resolution of the two meshes was kept the same.

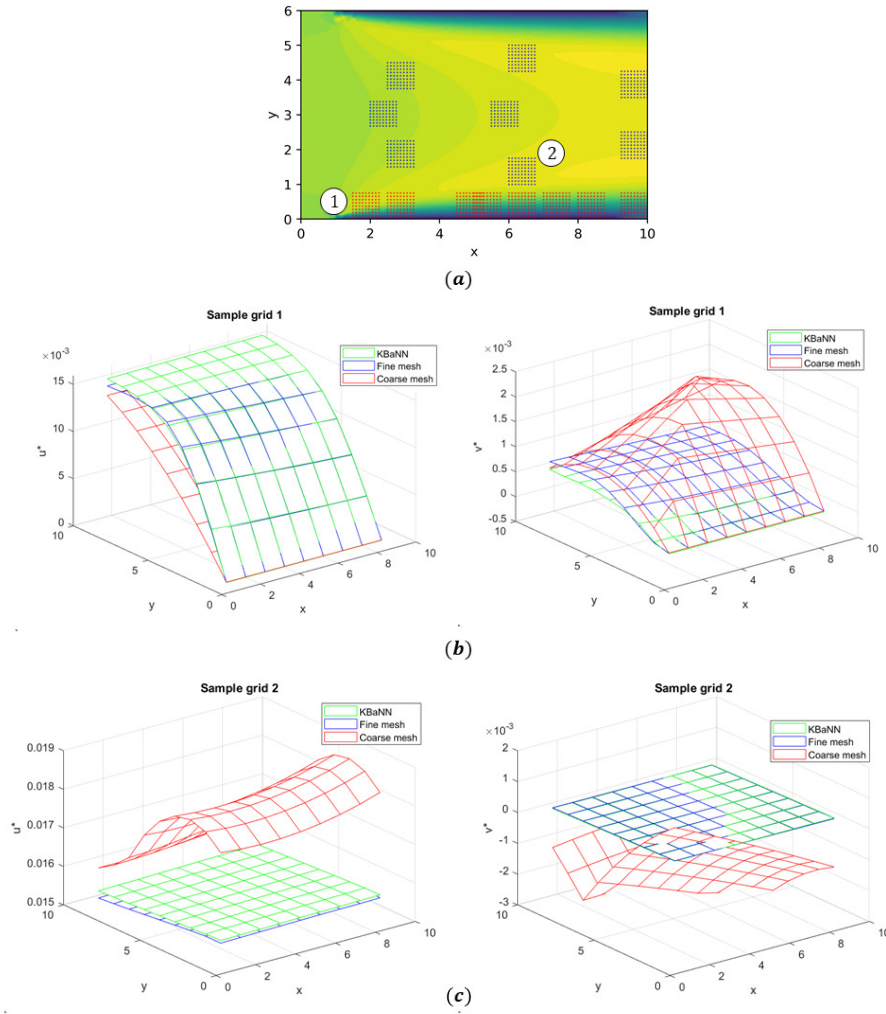


Figure 4.13: The performance of the KBaNNs was evaluated for a channel flow at  $U = 0.11\text{ms}^{-1}$ . (a) shows the locations of the two sample grids plotted. (b-c) the KBaNN predictions significantly improve the coarse mesh velocity field, while continuing to respect the no-slip boundary condition at the wall.

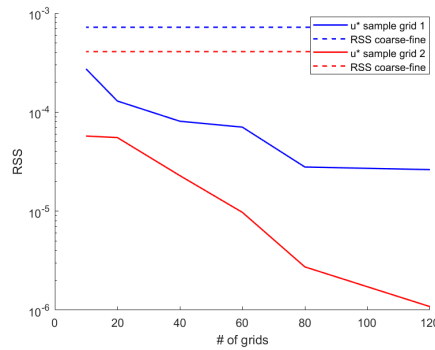


Figure 4.14: Plot of the performance of the KBaNNs as a function of the scarcity of the training data. Note that in all cases the RSS error of the KBaNNs was less than that of the coarse mesh.



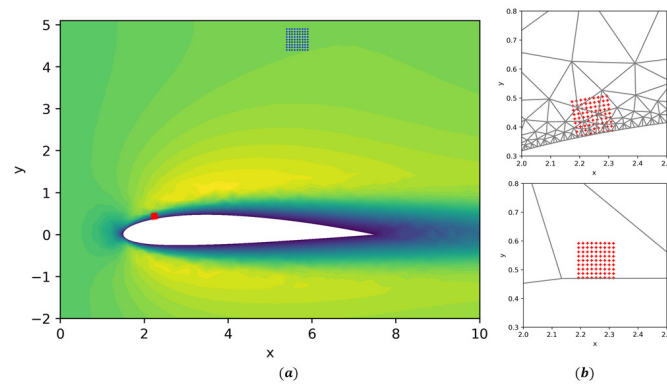


Figure 4.15: (a) Simulation results for the flow over a NACA 2412 airfoil at  $Re = 480$ . Two sample grids were placed in the freestream, out of the wake of the airfoil, and one on the top surface of the airfoil. (b) A closer view of the sample grid on the top surface of the airfoil.

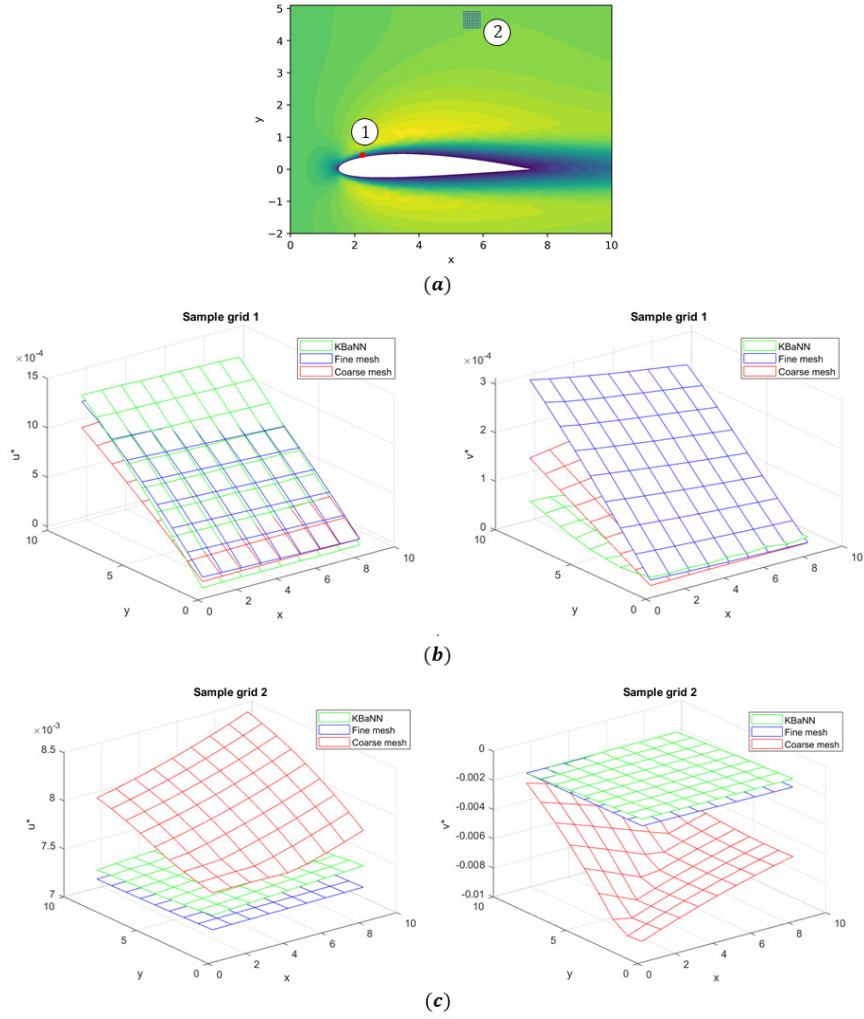


Figure 4.16: (a) the locations of the sample grids used to test the KBaNN. (b-c) A comparison of the predicted velocities in the sample grids from the KBaNN versus the fine mesh results. For comparison the velocities of the coarse mesh (which the KBaNN estimation is in part based on) are included.

# Adaptive Learning for Reliability Analysis

## using Support Vector Machines

### Contents

---

<b>5.1 Meta-modelling strategies for reliability analysis</b>	<b>140</b>
<b>5.2 Support Vector Machines for reliability analysis</b>	<b>144</b>
<b>5.3 Algorithm for Adaptive learning with SVMs</b>	<b>149</b>
5.3.1 Selecting candidates on the LSF . . . . .	150
5.3.2 Finding Sets Containing Points with High Likelihood . . . . .	152
5.3.3 Promoting exploration and quantifying predictive uncertainty . . .	153
<b>5.4 Validation with synthetic test cases</b>	<b>157</b>
5.4.1 Four branch function . . . . .	158
5.4.2 Physical space with discontinuities . . . . .	160
<b>6.1 Future Developments</b>	<b>166</b>

---

Having presented two strategies for DSE, employing machine learning meta-models, in the previous chapter we now discuss the final aspect of this PhD studentship, adaptive

learning. We introduce a method to iteratively improve a meta-model through an adaptive learning algorithm that selects informative parameter points to add to the training dataset according to an optimality criterion. Specifically, we focus on the case of a Support Vector Machine (SVM) meta-model that is used to learn an unknown function that separates two regions of a domain. In the context of reliability analysis these two regions represent the failure domain, where a set of constraints or requirements are violated, and a safe domain where they are satisfied. The Limit State Function (LSF) separates these two regions. Evaluating the constraints for a given parameter point requires the evaluation of a computational model that may well be expensive. For this reason we wish to construct a meta-model that can estimate the LSF as accurately as possible, using only a limited amount of training data. Applications in material science, structural dynamics, aviation-safety and control systems can benefit from this methodology.

This chapter presents an adaptive strategy for estimating an LSF, using Support Vector Machine (SVM) as a meta-model. An SVM with a polynomial kernel provides a semi-algebraic approximation of the LSF. We describe an optimisation process that is used to select informative parameter points to add to training data at each iteration to improve the accuracy of this approximation. A formulation is introduced for bounding the predictions of the meta-model; in this way we seek to incorporate this aspect of Gaussian Process Models (GPMs) within an SVM meta-model. Finally, we apply our algorithm to two benchmark test cases, demonstrating performance that is comparable with, if not superior, to a standard technique for reliability analysis that employs GPMs.

## 5.1 Meta-modelling strategies for reliability analysis

A key requirement for engineering designs is that they offer good performance across a range of uncertain conditions while exhibiting an admissibly low probability of failure. Denote the following:  $\mathbf{x}$  as a number of uncertain parameters with joint density  $f_x(\mathbf{x})$  with  $\mathbf{x} \in X \subseteq \mathbb{R}^{n_x}$ ;  $\mathbf{y} = M(\mathbf{x})$  as the system response with  $\mathbf{y} \in Y \subseteq \mathbb{R}^{n_y}$ ; and

$g(\mathbf{x}) = M(\mathbf{x}) - \mathbf{y}_0 \leq 0$  with  $g : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_g}$  as a set of requirements imposed upon the system. The physical space  $X$  (distinct from the design space  $\mathbb{X}$ ) is divided into a safe domain,  $\{\mathbf{x} : g(\mathbf{x}) \leq 0\}$ , and a failure domain  $\{\mathbf{x} : g(\mathbf{x}) > 0\}$ , with the two separated by the limit state function (LSF)  $\{\mathbf{x} : g(\mathbf{x}) = 0\}$ . The main goal of reliability analysis is to evaluate the probability of failure,  $\mathbb{P}$ :

$$\mathbb{P}[g(\mathbf{x}) > 0] = \int_{g(\mathbf{x}) > 0} f_x(\mathbf{x}) d\mathbf{x}. \quad (5.1)$$

Evaluating the failure probability is often difficult since it entails performing a multi-dimensional integral over a complex integration domain, thereby requiring an approximation. In many cases  $M$  is cheap to evaluate so the probability of failure can be readily estimated through Monte Carlo sampling. A Monte Carlo sampling approximation to the probability of failure is:

$$P = \frac{1}{n_{mc}} \sum_{i=1}^{n_{mc}} \mathcal{I}(g(\hat{\mathbf{x}}^{(i)}) > 0) \quad (5.2)$$

where  $\mathcal{I}$  is an indicator function (1 if  $g(\mathbf{x}) > 0$ , 0 otherwise),  $P$  the estimated probability of failure, and  $n_{mc}$  represents the number of Monte Carlo samples [240].  $\hat{\mathbf{x}}^{(i)}$  denotes the  $i^{\text{th}}$  Monte Carlo sample, drawn from  $f_x$ . The resulting approximation is subjected to sampling error because  $n_{mc}$  is finite. However, when  $M$  is expensive to evaluate and  $n_{mc}$  is large, the cost of (5.2) becomes inadmissibly high. This has motivated the development of methods for reliability analysis which replace  $M$  with a meta-model based on a training set of expensive model evaluations. The meta-model, which is cheap to evaluate, is then used to estimate  $P$  through Monte Carlo sampling. This model, which is trained using a limited number of function evaluations of  $g(\mathbf{x})$ , is given by  $\hat{g}(\mathbf{x})$ . Therefore, a Monte Carlo approximation to the failure probability based on  $\hat{g}(\mathbf{x})$  will suffer from both sampling error and response error. Gaussian Processes [241, 242], Artificial Neural Networks

[243], Polynomial Chaos [79], and response surface methods [244] have been commonly used in the literature as surrogates to obtain  $\hat{g}(\mathbf{x})$ .

A surrogate model for reliability analysis will perform well as long as the LSFs  $g(\mathbf{x}) = 0$  and  $\hat{g}(\mathbf{x})$  are sufficiently close. Large offsets in the response away from these functions will be immaterial. This consideration enables casting the surrogate modeling problem as the problem of creating a 2-classifier. These two categories correspond to points falling in either the safe domain or the failure domain. This chapter proposes a strategy for adaptively training a reliability-based classifier according to a limited number of evaluations of  $g(\mathbf{x})$  while accounting for uncertainty in the learned function.

Introduced by Vapnik in the field of statistical learning theory, Support Vector Machines (SVMs) [245, 246, 247] have been commonly used in reliability analysis [248, 249, 250]. This approach has several advantages: firstly, training a meta-model as a classifier, rather than directly learning  $M(\mathbf{x})$  or  $g(\mathbf{x})$ , allows discontinuous or binary responses to be efficiently handled. Secondly, multiple constraints or failure modes, i.e.  $n_g \geq 1$ , may be reduced to a single decision function [251]. A particular advantage of SVMs is the efficiency with which the training data is handled as only the support vectors, a subset of the training data, contribute to the model predictions.

Naturally, the use of a meta-model will sacrifice some accuracy in favour of reducing the computational cost regardless of which strategy is used. For this reason there has been much interest in developing adaptive methods for reliability analysis, which seek to improve the accuracy of a meta-model iteratively, by adding additional model evaluations to the dataset [252]. Several adaptive learning approaches have been proposed for adding to the training set of a SVM. A common aspect of these approaches is to encourage exploration of the physical space,  $X$ , by adding points to the training data that are separate from the existing labelled samples. For instance, Basudhar and Missoum (2008,2010) introduced an adaptive sampling method for constructing an estimate of the LSF using SVMs based on optimisation. Informative samples are found by maximising an objective function that depends on the nearest neighbor distance of the candidate from the train-

ing data in  $X$  [253, 254]. Pan and Dias (2017) proposed a pool based sampling scheme, in which the most informative parameter point is chosen from the Monte Carlo samples [255]. In this context informativeness quantifies the degree of exploration associated with a given sample. In the algorithm of Pan and Dias informativeness is calculated with a scoring function that depends in part on the nearest neighbour distance between the parameter point and the training data. Alibrandi et al (2015) reduced the complexity of the problem by confining the search for candidate points to a limited number of sampling directions, as opposed to searching the entirety of the physical space [256].

There are several challenges inherent to adaptive learning with SVMs that the algorithm presented here aims to address. Firstly, it is well recognised in the literature that the most informative candidates to add to the training set lie on the estimated LSF [257]. However, the non-linear nature of the LSF can make it challenging to efficiently restrict the search to such a subspace. One solution is to search for new candidates in the vicinity of the support vectors, however, this can lead to a divergent search which may not converge to the true LSF, referred to as ‘SVM locking’ in Basudhar and Missoum [254]. Secondly, while SVMs may generally scale more efficiently to higher dimensional physical spaces and larger training sets than Gaussian processes, an estimate from a Gaussian Process is accompanied by the predicted variance of the random process that gives an indication of the uncertainty associated with that estimate. An analogous measure does not exist for SVMs. The SVM solution to the classification problem yields not only an approximation to the true LSF but also a volume where this function might be contained. This subspace will be called the Limit State Volume (LSV). As more data is collected, the LSV shrinks towards the LSF approximation.

The algorithm proposed here sequentially expands the training data set by identifying optimal data points using a previously obtained SVM representation. This process accounts for the uncertainty caused by learning the LSV from limited data. In particular, we introduce a non-negative, polynomial function that at any value of  $\mathbf{x}$  takes on a value proportional to the distance from such a point to the closest member of the training data

set, which we name the uncertainty function. The points being added to the training set are optimal in the sense that they maximize the likelihood of the data and the uncertainty function within the LSV. Furthermore, the best and worst-case LSFs within the LSV are identified and used to compute a range of predicted failure probabilities. As more data is gathered, the width of this interval approaches zero. As such, the role of the uncertainty function is analogous to the predicted variance of the Gaussian Process Model (GPM) (see e.g. [241, 242]). In this way we seek to combine the flexibility and efficiency of SVMs when applied to reliability analysis, with the desirable attributes of Gaussian Processes when applied to adaptive learning.

The remainder of this chapter is structured as follows: in the next section we discuss the theoretical background of SVMs; the proposed algorithm for adaptive learning for reliability analysis using SVMs is then described; and finally the algorithm is implemented for two benchmark test cases in which the performance of the algorithm is compared with an established adaptive learning method for reliability analysis that employs GPMs.

## 5.2 Support Vector Machines for reliability analysis

Approaches that employ Support Vector Machines (SVMs) to estimate the LSF formulate the problem as a two class classification problem, as opposed to developing an explicit surrogate model for  $g(\mathbf{x})$ . Membership of the two classes depends on whether  $g(\mathbf{x}) > 0$ . We first consider a case in which the failure and safe domains are linearly separable. Using a set of  $n$  labelled training points  $(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(n)}, y^{(n)})$ , where the class labels  $y^{(i)} \in \{-1, 1\}$  indicate whether the inequality constraints are satisfied (-1) or violated (1), an SVM aims to approximate the LSF by finding the hyperplane with maximal margin that separates the two classes:

$$\mathbf{w}^\top \mathbf{x} - b = 0, \quad (5.3)$$



where  $\mathbf{w} \in \Re^{n_x}$  is the normal to the hyperplane and the bias,  $b$ , is a scalar. This is illustrated in Figure 5.1 for a  $n_x = 2$  physical space. Hereafter, we will refer to the training data as the sequence  $D = \{\mathbf{x}^{(i)}, y^{(i)}\}$ , where  $i = 1, \dots, n$ . In order to determine  $\mathbf{w}$ , the following constraints are imposed:

$$\begin{aligned} \mathbf{w}^\top \mathbf{x} - b &\geq 1 \text{ if } y^{(i)} = 1 \\ \mathbf{w}^\top \mathbf{x} - b &\leq -1 \text{ if } y^{(i)} = -1, \end{aligned} \tag{5.4}$$

These constraints are equivalent to  $y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} - b) \geq 1$  for  $i = 1, \dots, n$  and have the effect of defining a margin of width  $\frac{2}{\|\mathbf{w}\|}$  within which no training data falls. The separating hyperplane attaining the largest margin is given by the quadratic optimisation program:

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \text{ such that } y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} - b) \geq 1 \text{ for } i = 1, \dots, n \tag{5.5}$$

This optimisation problem can be reformulated with the Lagrangian dual as:

$$\mathbb{L} = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i (y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} - b) - y^{(i)}) \tag{5.6}$$

where  $\alpha_i$  are a set of Lagrange multipliers. The derivatives of the Lagrangian,  $\mathbb{L}$ , are set to 0 to yield the final form of this optimisation problem [247]:

$$\begin{aligned} \max_{\alpha} \mathbb{L} &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y^{(i)} y_j \mathbf{x}^{(i)\top} \mathbf{x}^{(j)} \\ \text{such that } \alpha_i &\geq 0 \text{ and } \sum_i \alpha_i y^{(i)} = 0. \end{aligned} \tag{5.7}$$

Using the Lagrangian dual we see that the only non-zero Lagrange multipliers correspond to the subset of points in the training data that lie on the boundary of the LSV. These data points are referred to as the support vectors. These Lagrange multipliers refer to those samples for which  $|\mathbf{w}^\top \mathbf{x} - b| = 1$  and can be found by using quadratic

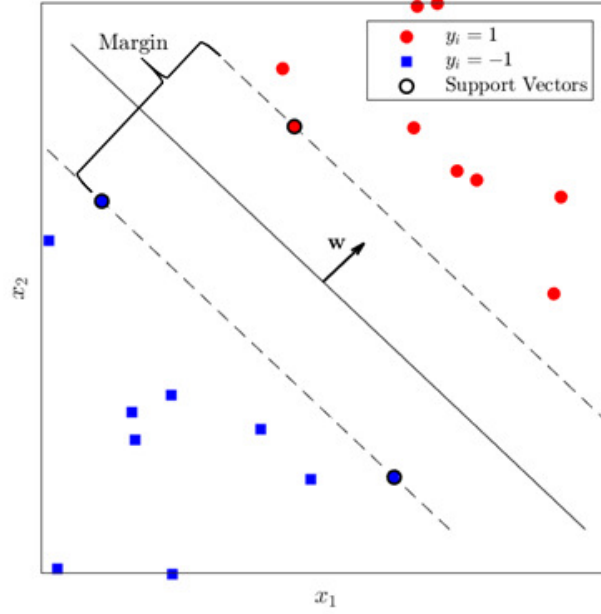


Figure 5.1: A trained Support Vector Machine (SVM) finds a separating hyperplane that maximises the width of the margin between training data that is categorised into two classes. Samples that lie on the margin are referred to as the Support Vectors.

programming to solve the above optimisation problem, which has been demonstrated to be convex [258]. The classification of an unlabelled sample,  $\mathbf{x}$ , is determined by the sign of its distance from the separating hyperplane,  $s(\mathbf{x})$ :

$$s(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} - b \text{ with } \mathbf{w} = \sum_{i=1}^{n_{sv}} \alpha_i \bar{y}^{(i)} \bar{\mathbf{x}}^{(i)} \quad (5.8)$$

where  $n_{sv}$  refers to the number of support vectors and  $\bar{\mathbf{x}}^{(i)}$  the  $i^{\text{th}}$  support vector with class label  $\bar{y}^{(i)}$ . The separating hyperplane, given by  $s(\mathbf{x}) = 0$ , is the approximation to the LSF; with the sign of  $s(\mathbf{x})$  used to determine class membership. The LSV is given by  $\{\mathbf{x} : -1 \leq s(\mathbf{x}) \leq 1\}$ . The decision function may be evaluated inexpensively, allowing the probability of failure to be estimated using  $n_{mc}$  Monte Carlo samples drawn from the joint distribution  $f_{\mathbf{x}}(\mathbf{x})$ :

$$P = \frac{1}{n_{mc}} \sum_{i=1}^{n_{mc}} \mathcal{I}(s(\hat{\mathbf{x}}^{(i)}) > 0) \quad (5.9)$$

where  $\hat{\mathbf{x}}^{(i)}$  is a Monte Carlo parameter point drawn from  $f_x$  and  $\mathcal{I}$  is the indicator function defined earlier.

The formulation described above can be used to find the maximally separating hyperplane in physical space provided that the training points are linearly separable. However, this might not be the case in many practical applications. In such cases, a transformation,  $\Phi : X \rightarrow Z$ , is used to map the training data from physical space,  $X$ , to a higher dimensional feature space,  $Z \in \mathbb{R}^{n_z}$ , where the data is linearly separable and a separating hyperplane can be found:

$$\mathbf{w}^\top \Phi(\mathbf{x}) - b = 0 \text{ with } \mathbf{w} = \sum_{i=1}^{n_{sv}} \alpha_i \bar{y}^{(i)} \Phi(\bar{\mathbf{x}}^{(i)}) \quad (5.10)$$

where  $\mathbf{w} \in \mathbb{R}^{n_z}$ . The form of the Lagrangian dual is such that in theory it is not necessary to specify  $\Phi$  explicitly in order to find the Lagrange multipliers; only the kernel of the transformation,  $\mathcal{K}(\mathbf{p}, \mathbf{q}) = \Phi(\mathbf{p})\Phi(\mathbf{q})$ ,  $\mathbf{p}, \mathbf{q} \in X$  need be defined. This is referred to as the ‘kernel trick’ and allows transformations to infinite dimensional feature spaces through the use of Gaussian kernels. However, in this work we restrict our attention to polynomial feature maps, which are of finite dimension, to not only work on a finite dimensional feature space but also to use polynomial optimisation strategies. This mapping is parametrized by the polynomial degree of the transformation and the number of monomials that are kept. Note that the separating plane might only exist when the degree exceeds a threshold value. This value can be found by using a bisection algorithm on the natural numbers. In particular we use polynomial mappings with kernels:

$$\mathcal{K}(\mathbf{p}, \mathbf{q}) = (1 - \mathbf{p}^\top \mathbf{q})^d, \quad (5.11)$$

where the order,  $d$ , of the kernel is user defined. For instance, a polynomial kernel of degree  $d = 2$  and  $n_x = 2$  corresponds to the feature map [259]:

$$\mathbf{z} = \Phi(\mathbf{x} = \{x_1, x_2\}) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \end{bmatrix}. \quad (5.12)$$

The inverse feature map,  $\Phi^{-1}$  is found using the monomial terms of degree 1, in this case:

$$\mathbf{x} = \Phi^{-1}(\mathbf{z}) = \frac{1}{\sqrt{2}} \begin{bmatrix} z_4 \\ z_5 \end{bmatrix}. \quad (5.13)$$

Figure 5.2 demonstrates how a polynomial transformation may be used to lift training data to a higher dimensional feature space where it becomes linearly separable. In this case, a separating plane in physical space does not exist. However, the polynomial transformation  $\Phi(x) = \begin{bmatrix} x^2 & x \end{bmatrix}^\top$  may be used to lift the data to a two dimensional space where the data is linearly separable and a maximally separating hyperplane may be found. Note that the entire physical space is mapped into a polynomial manifold in feature space. Every point at the intersection between this manifold and the separating hyperplane  $s(\mathbf{z}) = 0$  will map to the LSF,  $g(\mathbf{x}) = 0$ . In Figure 5.2 there are two such points, with the vertical construction lines indicating their positions in physical space. However, in higher dimensional feature spaces there will be infinitely many such points, with the estimate of the LSF corresponding to the locus of points on the manifold which intersect the separating hyperplane. In the next section we describe an adaptive learning algorithm which at every iteration selects the most informative point from among these to label and add to the training data.

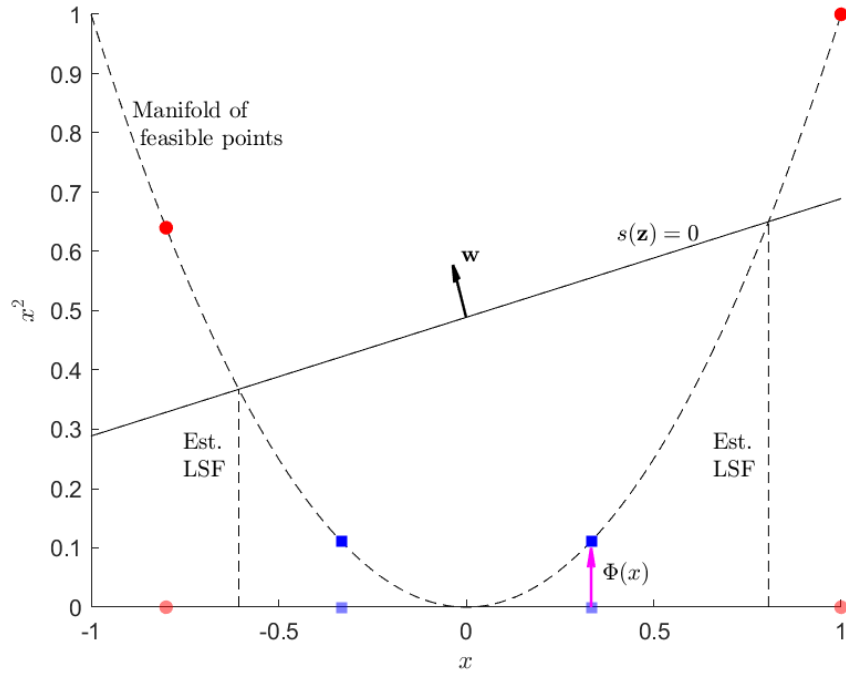


Figure 5.2: An example of how a polynomial transformation may be used to map data to a higher dimensional feature space in which the transformed data becomes linearly separable. The transformation,  $\Phi$ , creates a manifold of feasible points. Points on the estimated LSF correspond to the intersections of the manifold with the separating hyperplane.

## 5.3 Algorithm for Adaptive learning with Support Vector Machines

Any adaptive learning algorithm must strike a compromise between refining the meta-model in regions of  $X$  that already contain samples and placing new samples in unsampled regions of  $X$ . These two search features are referred to as exploitation and exploration respectively [260]. In the context of adaptive learning for reliability analysis, exploitation may be characterised as identifying a high-likelihood point where the uncertainty function takes on comparatively small values. In contrast, exploration may be thought as identifying a high-likelihood point where the uncertainty function takes on comparatively large values. The likelihood of a parameter point  $\mathbf{x}$  is quantified with the likelihood function,  $\mathcal{L}(\mathbf{x})$ . In this work we propose an algorithm for identifying a parameter point driven by

an optimality criterion that balances exploration and exploitation. The response of the expensive code is then evaluated at such a point, and the corresponding outcome is used to expand the training dataset which will be used in the following iteration. There are three desirable properties for a candidate training point to possess:

1. Lying on the estimated LSF
2. Attaining relatively high likelihood
3. Attaining comparatively large uncertainty function value

The range of failure probabilities described above can be computed at any iteration. The algorithm will proceed until the spread of this interval is below a given threshold. In the following sections we discuss how the proposed algorithm selects candidates with these desirable features.

### 5.3.1 Selecting candidates on the LSF

In general, the most informative candidates lie on the estimated LSF. If an SVM meta-model is employed, then these points will be those that satisfy  $\{\mathbf{x} : s(\mathbf{x}) = 0\}$ . For numerical reasons this equality is typically relaxed to the inequality constraint  $\{\mathbf{x} : |s(\mathbf{x})| \leq \epsilon\}$ , where  $\epsilon$  is a tolerance on the distance of the candidate from the separating hyperplane in feature space (see, e.g. [254]). Identifying points on the LSF by exploring its physical space representation might be difficult. For this reason we introduce a formulation for exploring the LSF in feature space. Recall from equation (5.10) that the SVM fits a separating hyperplane between the labelled data in feature space with normal  $\mathbf{w}$ . Through Gram-Schmidt orthogonalisation it is possible to develop an orthonormal basis that spans this separating hyperplane:

$$\mathbf{z} = \sum_{i=1}^{n_z-1} \gamma_i \mathbf{e}^{(i)} + \mathbf{e}^{(0)} \quad (5.14)$$

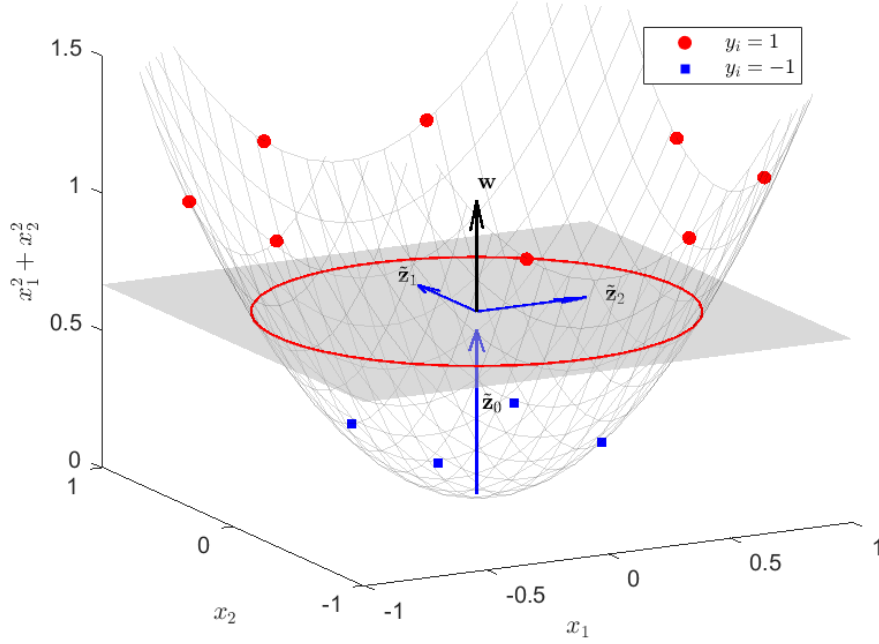


Figure 5.3: A polynomial mapping is used to lift a two dimensional physical space to a three dimensional feature space where an orthonormal basis which spans the separating hyperplane can be defined. The red circle indicates the locus of the intersections between the separating hyperplane and the polynomial manifold, i.e. the estimated LSF.

where  $\tilde{\mathbf{z}} \in \mathbb{R}^{n_z \times n_z - 1}$  represents the orthonormal basis vectors,  $\boldsymbol{\gamma} \in \Gamma \subseteq \mathbb{R}^{n_z - 1}$  the associated weights, and  $\mathbf{e}^{(0)} \in \mathbb{R}^{n_z}$  a vector of scalars. By definition, any choice of the coefficients  $\gamma$  in this basis will produce a point that lies on the separating hyperplane in feature space. However, the region of interest is the intersection between this hyperplane and the mapping of physical space onto feature space, i.e.,  $\mathbf{z} = \Phi(\Phi^{-1}(\mathbf{z}))$  where  $\mathbf{z}$  is in (5.14). This is illustrated in Figure 5.3, in which a two-dimensional physical space is mapped to a three-dimensional feature space with a polynomial feature map. The red circle indicates those points with weights  $\gamma_1$  and  $\gamma_2$  such that  $\mathbf{z}$  lies on both the separating hyperplane and the polynomial manifold.

### 5.3.2 Finding Sets Containing Points with High Likelihood

As has been discussed, exploitation in the context of reliability analysis may be thought of as sampling the LSF in regions with relatively high likelihood where the uncertainty of the meta-model is low. The likelihood is given by  $\mathcal{L}(\mathbf{x})$ . The joint density is an example of such a likelihood function. The high likelihood region near the LSV can be prescribed in terms of super-level sets of the joint density. For instance, in the case of Gaussian joint densities these level sets are ellipsoids, for which a closed-form expression is readily available. Alternatively, if a closed form expression for this contour is unavailable then we propose constraining the search to lie within an the region satisfying the inequality  $\hat{\mathcal{L}}(\mathbf{x}) \geq \underline{\mathcal{L}}$ , where  $\underline{\mathcal{L}}$  represents a minimum admissible likelihood value. Note that this region might be multiply connected and non-representable in closed form. This tolerance is calculated as a function of the maximum value that the likelihood function takes on the estimated LSF, which we denote  $\hat{\mathcal{L}}$ . This quantity is found through the solution of the optimisation problem:

$$\begin{aligned} \max_{\boldsymbol{\gamma}, \boldsymbol{\zeta}} \mathcal{L}(\Phi^{-1}(\mathbf{z})) \quad (5.15) \\ \text{such that } \|\mathbf{z} - \Phi(\boldsymbol{\zeta})\|^2 \leq \epsilon, \mathbf{z} = \sum_{i=1}^{n_z-1} \gamma_i \mathbf{e}^{(i)} + \mathbf{e}^{(0)}, \end{aligned}$$

where  $\boldsymbol{\gamma} \in \mathbb{R}^{n_z-1}$  and  $\boldsymbol{\zeta} \in \mathbb{R}^{n_x}$ . The dummy vector  $\boldsymbol{\zeta} \in X \subseteq \mathbb{R}^{n_x}$ , is used in a manner analogous to the Lagrange multipliers in (7). Hence, the solution to (5.15) yields a point that is in both the physical space and the LSF,  $\boldsymbol{\zeta}^*$ , for which the likelihood is maximal. Note that by assuming  $\epsilon > 0$  we relax what should nominally be an equality constraint. With  $\hat{\mathcal{L}}$  determined through optimisation, a simple formulation to guarantee exploration of regions with relatively high likelihood is to impose the inequality constraint  $\mathcal{L}(\mathbf{x}) \geq \underline{\mathcal{L}} = \kappa \hat{\mathcal{L}}$ , where  $\kappa < 1$  is a positive constant specified by the user.



### 5.3.3 Promoting exploration and quantifying predictive uncertainty

Having introduced a formulation to explore the separating hyperplane in feature space, with a set of constraints that ensure exploration of regions in  $X$  where the likelihood function exceeds a threshold, we introduce a function for quantifying the separation of a candidate parameter point from the training set in  $X$ . We term this function the uncertainty function,  $\mathcal{U}(\mathbf{x}; D_x)$ , where  $\mathbf{x} \in \mathbb{R}^{n_x}$  represents the candidate parameter point in  $X$  and  $D_x = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}\} \in \mathbb{R}^{n_x \times n}$  the sub-sequence of the training data corresponding to the locations of the training points in  $X$ . In addition to quantifying the informativeness of the candidate point, the uncertainty function may also be used to quantify the approximation error in the failure probability estimate resulting from only using a limited number of training points. In so doing, the uncertainty function is analogous to the variance in a GPM and is intended to marry this feature of GPMs with the efficiency with which SVMs scale with  $n_x$  and  $n$ . Any non-negative function which at any fixed value of  $\mathbf{x}$  takes on values proportional to the distance from that  $\mathbf{x}$  to the closest element of  $D_x$  can be used as an uncertainty function. This property can be written as:

$$\mathcal{U}(\mathbf{x}; D_x) = \min_i \mathcal{D}(\mathbf{x}; \mathbf{x}^{(i)}), \quad (5.16)$$

where  $\mathcal{D}$  is any norm. In this chapter we use the Mahalanobis distance given by:

$$\mathcal{D}(\mathbf{p}; \mathbf{q}) = (\mathbf{p} - \mathbf{q})^\top W (\mathbf{p} - \mathbf{q}), \quad (5.17)$$

where  $W \in \mathbb{R}^{n_x \times n_x}$  is a positive definite matrix of weights. In the case of  $W = I$  this distance is equivalent to the squared Euclidean distance. Candidates in  $X$  that maximise the uncertainty function are considered to be the most informative and are identified through an optimization process:

$$\begin{aligned}
& \max_{\gamma, \zeta} \mathcal{U}(\Phi^{-1}(\mathbf{z}); D_x) \text{ such that} \\
& \|\mathbf{z} - \Phi(\zeta)\|^2 \leq \epsilon, \quad \mathcal{L}(\Phi^{-1}(\mathbf{z})) \geq \underline{\mathcal{L}}, \\
& \mathbf{z} = \sum_{i=1}^{n_z-1} \gamma_i \mathbf{e}^{(i)} + \mathbf{e}^{(0)}.
\end{aligned} \tag{5.18}$$

Hence, we seek a parameter point that maximizes the uncertainty function while attaining a sufficiently large likelihood. In general, the optimization problem described here is likely to be non-convex and for this reason a Sequential Quadratic Programming (SQP) algorithm is run from several initial conditions in order to select the best point.

In practice it may not be feasible to run an adaptive algorithm to convergence due to the computational expense in evaluating  $g(\mathbf{x})$  at each iteration. For this reason, it is convenient to quantify the uncertainty in the empirical estimate of the failure probability at each iteration, which could be expressed as the bounded interval  $[P_l, P_u]$ . The formulation of a SVM does not provide a means to explicitly calculate this interval; we propose a method that uses the available knowledge of the requirements  $g$ , together with the uncertainty function. This data corresponds to the evaluation of the requirements for the parameter points in  $D_x$ , which we denote:  $G = \{g(\mathbf{x}^{(i)})\}$ ,  $i = 1, \dots, n$ .

Firstly, we use an empirical function to approximate  $g$  as a function of the distance from the separating hyperplane, which we denote  $\hat{g}(s)$ .  $\hat{g}(s)$  is an increasing function that passes through the origin. We represent  $\hat{g}$  as the weighted sum of a basis of functions that share these properties, for instance in this work we choose:

$$\hat{g}(s) = \begin{cases} \boldsymbol{\beta}^\top f(s), & s \geq 0 \\ \boldsymbol{\delta}^\top f(s), & s < 0 \end{cases} \tag{5.19}$$

where  $f(s) = [s, s^3, \text{erf}(s)]^\top$  and  $\boldsymbol{\beta}, \boldsymbol{\delta} \in \mathbb{R}^3$  contain non-negative weighing coefficients. This format guarantees that  $\hat{g}$  and  $s$  have the same sign. Alternative forms can be used instead. These coefficients can be found through a least squares fit to the training data,

using  $G$  and the evaluation of  $s(\mathbf{x})$  for the parameter points in  $D_x$ . We combine the empirical approximation for  $\hat{g}(s)$  with the uncertainty function to estimate the interval  $[P_l, P_u]$  through:

$$\begin{aligned} P_u &= \frac{1}{n_{mc}} \sum_{i=1}^{n_{mc}} \mathcal{I}(\hat{g}_u(s(\hat{\mathbf{x}}^{(i)})) > 0), \\ P_l &= \frac{1}{n_{mc}} \sum_{i=1}^{n_{mc}} \mathcal{I}(\hat{g}_l(s(\hat{\mathbf{x}}^{(i)})) > 0), \end{aligned} \quad (5.20)$$

where  $\mathcal{I}$  refers to the indicator function defined above.  $\hat{g}_u(s(\hat{\mathbf{x}}^{(i)}))$  and  $\hat{g}_l(s(\hat{\mathbf{x}}^{(i)}))$ , which represent a pessimistic and optimistic estimate of  $\hat{g}$  for the  $i^{\text{th}}$  Monte Carlo sample respectively, are determined as:

$$\begin{aligned} \hat{g}_u(s(\hat{\mathbf{x}}^{(i)})) &= \hat{g}(s(\hat{\mathbf{x}}^{(i)})) + \rho_u \mathcal{U}(\hat{\mathbf{x}}^{(i)}; D_x), \\ \hat{g}_l(s(\hat{\mathbf{x}}^{(i)})) &= \hat{g}(s(\hat{\mathbf{x}}^{(i)})) - \rho_l \mathcal{U}(\hat{\mathbf{x}}^{(i)}; D_x), \end{aligned} \quad (5.21)$$

where the coefficients  $\rho_u$  and  $\rho_l$  are chosen by the user to make  $\mathcal{U}$  and  $\hat{g}$  comparable. We suggest defining these coefficients as a percentage of the values that  $\hat{g}$  takes at the boundaries of the LSV (we choose 90% here). The set  $\{s : g(s) > 0\}$  corresponds to the failure domain, therefore a positive shift will result in a more pessimistic estimate of  $P$  as some Monte Carlo samples for which  $\hat{g}(s) \leq 0$  will be shifted to the upper half plane. Conversely, subtracting the second term corresponds to a more optimistic of  $P$  (recall that  $\mathcal{U}$  is a non-negative function). We illustrate this in Figure 5.4 for two Monte Carlo samples, where  $\hat{g}(s)$  is estimated from a training set containing 10 parameter points. The formulation described in (5.20) and (5.21) has several desirable qualities:

1. The interval is based on  $\mathcal{U}$  and is a function of  $D_x$ , instead of the support vectors.
2.  $g(s)$  is smooth of discontinuities since no training data falls in the LSV.

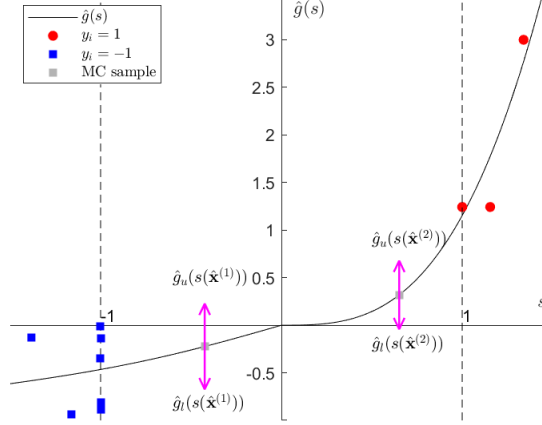


Figure 5.4: Illustration of the least squares fitting for the empirical function  $\hat{g}(s)$  to the training data; and the upper and lower estimates of  $\hat{g}$  for two Monte Carlo samples.

3. The algorithm is guaranteed to decrease the width of the interval  $[P_l, P_u]$  as  $\mathcal{U}$  and the LSV tend to 0 as more points are added.

Having discussed in detail how the formulation of the adaptive sampling algorithm is intended to promote the selection of an informative parameter point to add to the training data at each iteration, we present the corresponding algorithm next:

**Algorithm 1 (Adaptive learning with SVMs for reliability analysis):**

Inputs: Training dataset  $D = \{(\mathbf{x}^{(i)}, y^{(i)})\}$ ,  $G = \{g(\mathbf{x}^{(i)})\}$ ,  $i = 1, \dots, n$  and MC samples  $\{\hat{\mathbf{x}}^{(i)}\}$ ,  $i = 1, \dots, n_{mc}$

Outputs: Trained SVM meta-model,  $P$ ,  $P_u$  &  $P_l$

1. Train a SVM meta-model using the training dataset
2. Generate the orthonormal basis  $\mathbf{e}$  & constant vector  $\mathbf{e}^{(0)}$  in  $Z$
3. Determine  $\hat{\mathcal{L}}$  through (5.15), calculate  $\underline{\mathcal{L}}$
4. Estimate  $P$ ,  $P_u$ , &  $P_l$  through (5.9), (5.20) & (5.21)
5. If  $|P_u - P_l| < \epsilon$  stop. Otherwise, continue.
6. Determine next parameter point  $\zeta^*$  through (5.18)

7. Evaluate  $g(\zeta^*)$
8. Update  $D$  &  $G$ . Go to step 1.

Recall that if closed form expressions for contours of equal likelihood exist, for example ellipsoids in the case of Gaussian joint densities, then these can be used as a constraint on the likelihood, making step 3 unnecessary. The convergence of  $P_u$ , which represents a worst case estimation of the failure probability given the available data, can be used as a stopping criteria for the algorithm. We suggest a stopping criteria based on the interval value of  $P_u$  and  $P_l$ , i.e.,  $|P_u - P_l| < \epsilon$ . In the next section we apply the above algorithm to two benchmark examples and compare its performance against an adaptive method employing Gaussian Process models.

## 5.4 Validation with synthetic test cases

In this section we apply the algorithm for adaptive learning with SVMs presented above, which we refer to here as AL-SVM for convenience, to two synthetic test cases. For now we restrict our attention to applications involving low-dimensional physical spaces: both test cases presented here have  $n_x = 2$  physical spaces. The first test case, the four branch function, is a common benchmark problem in the reliability analysis literature, while the second case features a simple feasible region that is bordered by discontinuities. In both instances we compare the performance of AL-SVM to AK-MCS, an adaptive learning algorithm that utilises GPMs. In AK-MCS a GPM is trained to approximate  $g(\mathbf{x})$  directly using the training data. At each iteration the Monte Carlo parameter point for which a scoring function is maximal is added to the training data. The scoring function is dependent on the estimated constraint and the kriging variance. More details on AK-MCS may be found in Echard et al [241].

### 5.4.1 Four branch function

The first test case involves a  $n_x = 2$  physical space constrained by two polynomial and two linear requirements:

$$g(x_1, x_2) = \max \begin{cases} 3 + 0.1(x_1 - x_2)^2 - \frac{1}{\sqrt{2}}(x_1 + x_2) \\ 3 + 0.1(x_1 - x_2)^2 + \frac{1}{\sqrt{2}}(x_1 + x_2) \\ x_1 - x_2 + \frac{6}{\sqrt{2}} \\ x_2 - x_1 + \frac{6}{\sqrt{2}} \end{cases} \quad (5.22)$$

where  $x_1$  and  $x_2$  represent the two uncertain inputs. Referred to as the four branch function, this is a common benchmark test case in reliability analysis [255, 261, 262]. Monte Carlo samples are drawn from a multivariate Gaussian joint density  $N(\boldsymbol{\mu}, \Sigma)$ , where:

$$\boldsymbol{\mu} = \begin{bmatrix} 0 & 0 \end{bmatrix} \text{ and } \Sigma = \begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix}, \quad (5.23)$$

in which  $\boldsymbol{\mu}$  refers to the means and  $\Sigma$  the covariances of the joint density. The four branch function is a useful test case as the geometry of the requirements in physical space can be easily visualised and are cheap to evaluate, allowing us to study the convergence of the two adaptive sampling methods to the failure probability estimated by Monte Carlo sampling, referred to as  $P_{mc}$ .

The top left panel in Figure 5.5 illustrates the true LSF of the four branch function, the location of the Monte Carlo samples and the initial parameter points in the training dataset that was fed to both adaptive learning algorithms. The remaining panels indicate the training data at subsequent iterations of AL-SVM (with  $W = I$ ), with the estimated LSF indicated in red at each iteration. The convergence of this estimate is clear, however we see an interesting effect occur at iteration 80 as the SVM tries to fit the data near the four points in the four branch function where the requirements intersect. In both test cases we used a SVM with a polynomial kernel of degree 4, which remains fixed

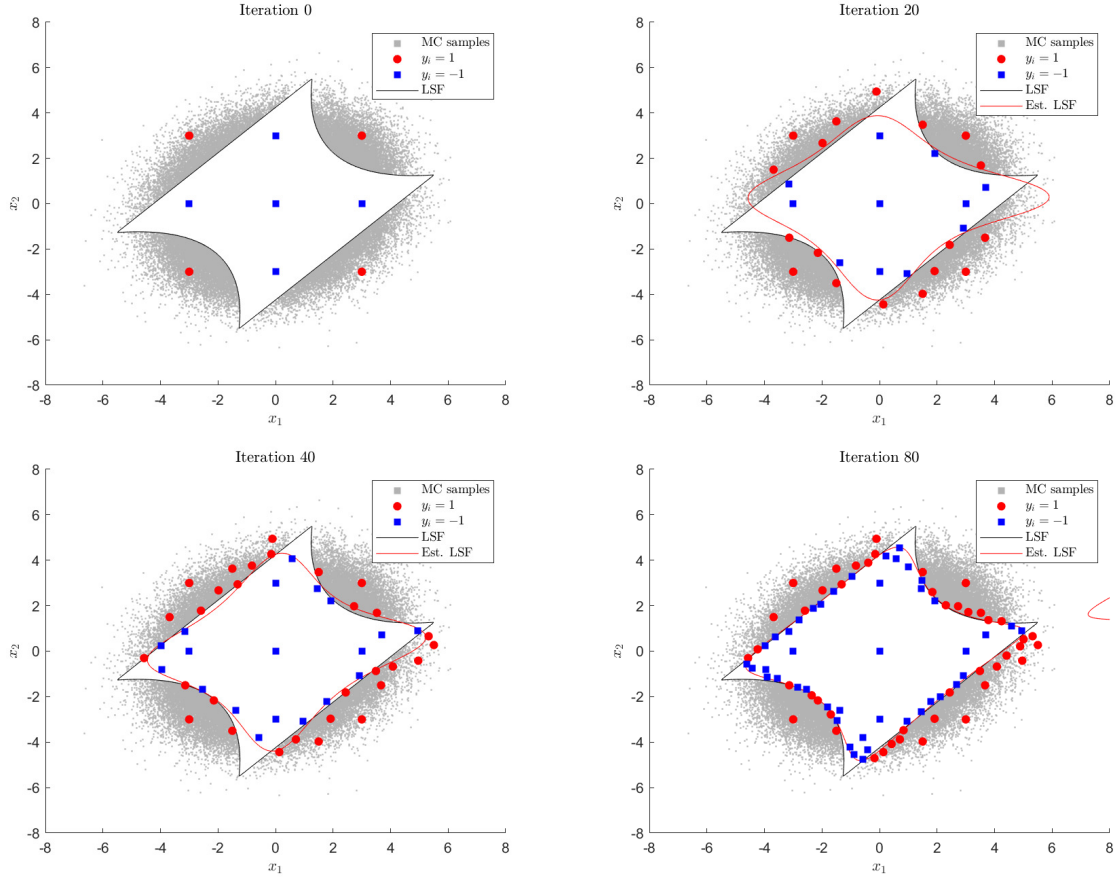


Figure 5.5: Panels indicating the convergence of the estimated LSF to the four branch LSF during iterations 0, 20, 40, and 80 of AL-SVM, with  $W = I$ .

throughout the learning. We also use a ‘hard margin’ SVM, which does not allow for misclassification of the training data. This can lead to a situation where the SVM does not have sufficient degrees of freedom to adequately fit the data. In this case there is a low density of Monte Carlo samples near these four points and consequently this numerical effect has little impact on the estimated failure probability.

The convergence of  $P$  for both AL-SVM and AK-MCS are plotted in the left panel of Figure 5.6. The panel also indicates the interval  $[P_l, P_u]$  at each iteration. We note from this plot that the convergence of our implementation of AK-MCS is similar to the implementation of AK-MCS presented in Pan and Dias [255]. The number of iterations for the percentage deviation between  $P$  and  $P_{mc}$  to decrease to below a given tolerance are presented in Table 5.1. AL-SVM was implemented three times, with varying choices of

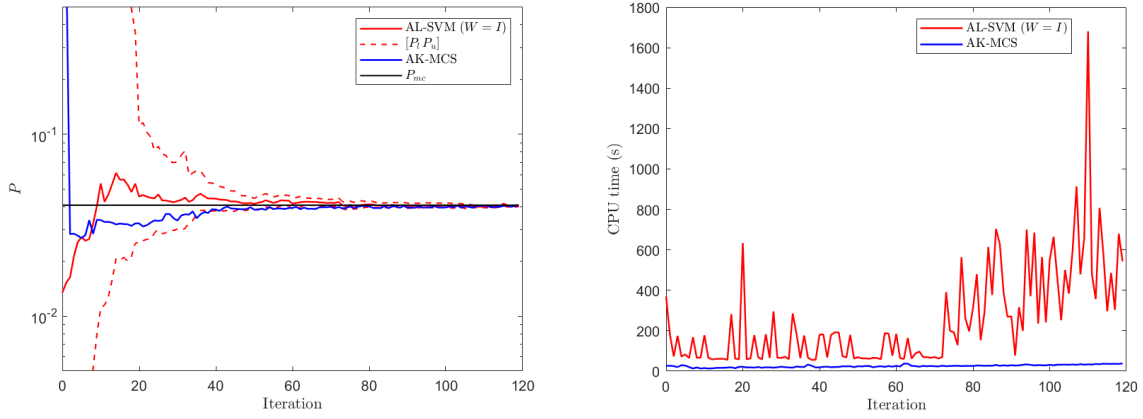


Figure 5.6: Plot of the convergence of  $P$  to  $P_{mc}$  with algorithm iteration for AL-SVM and AK-MCS (left). CPU times (s) for each iteration of the two algorithms (right).

$W$ . It was found that the choice  $W = I$ , corresponding to the squared Euclidean distance, yielded the fastest convergence of the three. In any case, it appears that the convergence of AL-SVM is at least comparable to, if not faster than, AK-MCS with respect to the number of requirement evaluations needed. This speed of convergence comes at a cost, however, as can be seen from the right panel of Figure 5.6. This panel compares the CPU time for AL-SVM and AK-MCS, which is clearly much greater for AL-SVM, especially in later iterations. This cost arises from the solution of the optimisation described in (5.18). As has been discussed, this optimisation is likely to be non-convex and for that reason a local solver is run from 12 separate starting points on each iteration. The computational cost of AL-SVM in its present formulation may be prohibitive for certain applications, where the timescale of each evaluation of the requirements may be on the order of seconds as opposed to minutes or hours.

### 5.4.2 Physical space with discontinuities

In the second test case we consider a simple feasible region corresponding to a square in a  $n_x = 2$  physical space with two constraints:



Method	Iterations until $\epsilon \leq \epsilon_{tol}$		
	10%	5%	2%
AL-SVM ( $W = I$ )	38	60	73
AL-SVM ( $W = \Sigma$ )	62	65	88
AL-SVM ( $W = \Sigma^{-1}$ )	57	80	97
AK-MCS	38	55	116

Table 5.1: The number of iterations required by the three implementations of AL-SVM and AK-MCS to achieve a  $P$  with percentage error below a given tolerance, as compared to  $P_{mc}$ .

$$g(x_1, x_2) = \begin{cases} 10 & \text{if } x_1 > 0.8 \vee x_2 > 0.8 \\ -2 & \text{otherwise} \end{cases} \quad (5.24)$$

where  $x_1$  and  $x_2$  again represent the uncertain inputs. Monte Carlo samples for these inputs are generated from two independent uniform distributions  $U(0, 1)$ . This LSF is plotted in the left panel of Figure 5.7, together with the estimated LSF from AL-SVM and the locations of the parameter points added to the training set by the algorithm. The locations of the 5 parameter points in the initial training set are ringed in black. While the geometry of the feasible region is straightforward, it can be considered a challenging case for adaptive methods that employ meta-models to learn  $g(\mathbf{x})$  directly as  $g(\mathbf{x})$  changes discontinuously across the LSF. Such discontinuities can occur frequently in real-world problems, for example in buckling behaviour [7, 263].

The convergence of AL-SVM is plotted in the right panel of Figure 5.7 and the iterations required to reduce the percentage error below given tolerances are displayed in Table 5.2. For this test case the weight matrix  $W = I$  was chosen. The AL-SVM approach converges to a value of  $P$  with a deviation from  $P_{mc}$  of less than 2% after 17 iterations. There will be an offset to this estimated  $P$ , however, as an SVM with a polynomial kernel will not have sufficient degrees of freedom to exactly replicate the corner of the feasible region, formed by the intersection of the requirements. On the other hand, a naive implementation of AK-MCS fails to converge as the variances shrink near the discontinuities.

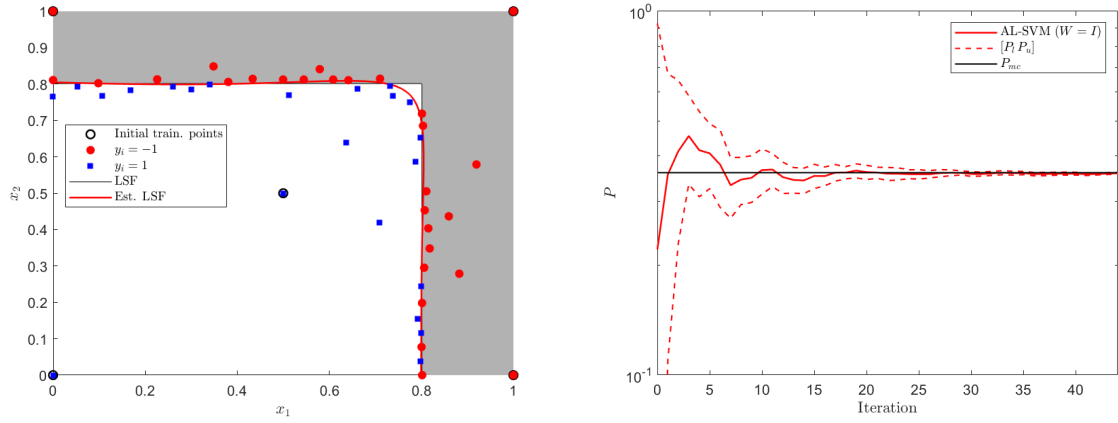


Figure 5.7: Plot of the LSF, training data, and estimated training data for the case of discontinuous changes in  $g(\mathbf{x})$  (left). Plot displaying the convergence of  $P$  to  $P_{mc}$  for AL-SVM (right).

This test case demonstrates an inherent advantage of classifier based approaches to reliability analysis: that they can be used to approximate constraints that vary discontinuously without modifying the algorithm. While there are established methods for modelling discontinuities with Gaussian Processes, for instance through piecewise Gaussian processes [264], this can prove challenging in the context of adaptive learning, where the presence of discontinuities may not be known a priori. Methods for detecting discontinuities in high dimensional spaces, for example through polynomial annihilation, require many function evaluations that might not be available in the initial iterations of an adaptive learning algorithm [265].

Method	Iterations until $\epsilon \leq \epsilon_{tol}$		
	10%	5%	2%
Adaptive learning with SVM	6	8	17
AK-MCS	N/A	N/A	N/A

Table 5.2: The number of iterations required by AL-SVM to achieve a  $P$  with percentage error below a given tolerance, as compared to  $P_{mc}$ . AK-MCS did not converge for this case.

## Summary

A novel algorithm has been presented for adaptive learning of an LSF, using SVMs to construct a semi-algebraic approximation to the LSF. The method identifies informative parameter points through an optimisation process in which the uncertainty function, a polynomial function that depends on the proximity of a candidate parameter point to the training data, is maximised. By exploiting the properties of polynomial kernel functions, the algorithm executes a constrained search in feature space, ensuring that candidate parameter points have an admissibly high likelihood and lie on the estimated LSF. The predictive uncertainty of the SVM meta-model is expressed using an analytical expression for the constraints as a function of distance from the separating hyperplane and the uncertainty function. We applied the algorithm to two benchmark cases in  $n_x = 2$  physical spaces and demonstrated performance that was consistent with, if not superior to, the performance of a method utilising GPMs.

While these results are encouraging there are several aspects in which the work presented here may be extended. Firstly, an application of the method to a higher-dimensional physical spaces is crucial and may be instructive to the future development of the method. Secondly, we do not fully exploit the polynomial formulation of the algorithm here. Implementing the method using polynomial optimisation methods will improve the performance of the algorithm. The first test case demonstrated that while the algorithm required fewer model evaluations than the Gaussian Process method to achieve the same accuracy, the computational cost per iteration of the algorithm was significantly higher. Finally, the algorithm may benefit from a subroutine for tuning the polynomial degree of the SVM, perhaps based on a bisection algorithm. Such a subroutine might lead to quicker convergence and an analysis of the variation of the degree with iteration number might provide a means of assessing the quality of the convergence.

## Conclusions

This thesis has presented a range of novel Uncertainty Quantification (UQ) methods that may be integrated into the design process in order to assess the uncertain performance of a candidate design. The intention of these methods is to assist designers in designing reliable products that offer good performance across a range of uncertain conditions by informing decision making, particularly in the conceptual design phase. There is industrial and academic interest in developing data-driven meta-models that can estimate the performance and the probability of failure of a candidate design when experimental data is missing or scarce and computational resources may be limited. Such meta-models allow the designer more freedom to explore the design space before committing resources to a single design. The work in this thesis expands the envelope of methods for Uncertainty Quantification for the design of aerospace components and addresses four challenges:

1. In **Chapter 2** a stochastic model updating algorithm for multiscale uncertainty quantification was introduced. Uncertainty was up-scaled from one scale to the next in a hierarchy of models through a coupling between scales, which applied the principle of probabilistic equivalence and a meta-model based on polynomial chaos. The algorithm was applied to a test case involving a wing box, where an equivalence was sought between an analytical model and a more granular, finite element model.

This test case is relevant to aeronautics, where components can be modelled with both finite element models and simpler, lumped mass models.

2. A relatively under-researched problem in the UQ literature is how to leverage the results of simulations with scarce experimental data. Two novel methods for this are proposed in **Chapter 3**, based on the principles of probabilistic equivalence and the Maximum Entropy Principle (MEP). The methods were demonstrated using a dataset from experimental tests on carbon fibre composite coupons.
3. In **Chapter 4** two methods for Design Space Exploration (DSE) are proposed, in which meta-models are developed from a training set of either experimental data or the results of evaluations of expensive computer code. In this way, the meta-models can be used by a designer to explore the design space of a new aerospace component, with the meta-models used to minimise the computational cost associated with evaluating each candidate design.
4. All of the methods proposed in this thesis develop data-driven meta-models from a training dataset that has been collected according to a Design of Experiments (DoE), with the accuracy of the meta-model determined by the size and informativeness of this dataset. Given the expense associated with each parameter point in the training dataset, a challenge that arises is how to improve a meta-model iteratively so that an accurate meta-model may be trained with a minimum of data. In the final chapter of this thesis, **Chapter 5**, an adaptive learning algorithm is introduced that was developed in partnership with NASA Langley, with the specific application of reliability analysis with a Support Vector Machine (SVM) meta-model.

## 6.1 Future Developments

There are a number of directions in which the research presented in this thesis might be expanded upon. There are three areas in which the research could be improved that would aid its dissemination/adoption among industry and academia:

1. Probabilistic Equivalence is a powerful principle, however, in the author's opinion its potential has not been fully realised. This thesis has demonstrated how probabilistic equivalence might be applied to problems beyond stochastic model upscaling, in this case to histogram interpolation and inverse identification. However, a thorough error analysis is needed for the method to reach maturity.
2. The Knowledge Based Neural Network (KBaNN) architecture introduced here is a powerful tool for bi-fidelity simulation. There are a number of ways in which the formulation could be extended, for instance by including a metric to describe the predictive uncertainty of the meta-model, analogous to the kriging variance in Gaussian Processes. Such a metric isn't explicitly defined in the formulation of a neural network, motivating the development of methods such as dropout and Bayesian neural networks, that can quantify the level of uncertainty associated with a prediction from the meta-model [266]. For the KBaNN, such a metric could then be used in an algorithm to adaptively improve the accuracy of the KBaNN meta-model, in a similar manner to the uncertainty function in the adaptive learning algorithm in **Chapter 5**.

With regard to the local Navier-Stokes approximator introduced in **Chapter 4**, this work may be expanded, diversifying the training data available to the KBaNNs to allow it to make estimations for more complex flows, for instance flows in three dimensions or including wakes and separation. Topology Optimisations (TOs) can involve repeated calls to a Computational Fluid Dynamics (CFD) solver, so there is motivation in developing a meta-model for the CFD solver that can reduce the

computational resources required by the TO. Further to this, the economic impact of KBaNNs more generally can be generated with test cases that involving high-dimensional spaces.

3. The results presented in **Chapter 5** are encouraging. One of the next steps for this research is to demonstrate the applicability of the algorithm to problems in high-dimensional spaces, which reflect the design spaces of industrial aerospace components. In addition, the polynomial formulation of the algorithm is not fully exploited currently. This will entail applying techniques for polynomial optimisation, which will increase the speed at which the algorithm can be run.

# Bibliography

- [1] P.J. Clarkson, C. Simons, and C. Eckert. “Predicting Change Propagation in Complex Design”. *J. Mech. Des.*, **126** (2004), pp. 788–797.
- [2] J. Rios, R. Roy, and A. Lopez. “Design requirements change and cost impact analysis in airplane structures”. *International Journal of Production Economics*, **109** (2007), pp. 65–80.
- [3] M. Eldred and J. Burkardt. “Comparison of Non-Intrusive Polynomial Chaos and Stochastic Collocation Methods for Uncertainty Quantification”. *AIAA* (2009).
- [4] V.J. Romero, L.P. Swiler, and A.A. Giunta. “Construction of response surfaces based on progressive-lattice-sampling experimental designs”. *Struct. Saf.*, **26** (2004), pp. 201–219.
- [5] H.M. Gomes and A.M. Awruch. “Comparison of response surface and neural network with other methods for structural reliability analysis”. *Struct. Saf.*, **26** (2004), pp. 49–67.
- [6] D. Crevillén-García et al. “Gaussian process modelling for uncertainty quantification in convectively-enhanced dissolution processes in porous media”. *Advances in Water Resources*, **99** (2017), pp. 1–14.
- [7] A. Basudhar, S. Missoum, and A. Harrison Sanchez. “Limit state function identification using Support Vector Machines for discontinuous responses and



- disjoint failure domains”. *Probabilistic Engineering Mechanics*, **23** (2008), pp. 1–11.
- [8] T. Little. “Schedule estimation and uncertainty surrounding the cone of uncertainty”. *IEEE Software*, **23** (2006), pp. 48–54.
- [9] S. McConnell. *Rapid development : taming wild software schedules*. Redmond Wash : Microsoft Press, 1996.
- [10] L. Williams et al. “Scrum + Engineering Practices: Experiences of Three Microsoft Teams”. *2011 International Symposium on Empirical Software Engineering and Measurement* (2011), pp. 463–471.
- [11] D.E. Guest. *Adams Simulation Saves Millions by Replacing Physical Testing in Aircraft Certification - Industrial Technology Centre*. 2016. URL: <https://www.itc.mb.ca/simulation/adams-simulation-saves-millions-by-replacing-physical-testing-in-aircraft-certification/>. (visited on 01/19/2021).
- [12] P. Jenny, S.H. Lee, and H.A. Tchelepi. “Multi-scale finite-volume method for elliptic problems in subsurface flow simulation”. *J. Comput. Phys.*, **187** (2003), pp. 47–67.
- [13] T.Y. Hou and X. Wu. “A multiscale finite element method for elliptic problems in composite materials and porous media”. *J. Comput. Phys.*, **134** (1997), pp. 169–189.
- [14] T. Yamanaka et al. “Multiscale finite element analysis of mode I delamination growth in a fabric composite”. *Compos. Struct.*, **133** (2015), pp. 157–165.
- [15] T. Niezgoda and A. Derew onko. “Multiscale composite FEM modeling”. *Procedia Eng.*, **1** (2009), pp. 209–212.
- [16] B.R. Reddy and K. Ramji. “Modeling and simulation of nano and multiscale composites”. *Int. J. Hybrid Inf. Technol.*, **9** (2016), pp. 133–144.

- 
- [17] H.Y. Chou et al. “Stochastic factors controlling the failure of carbon/epoxy composites”. *J. Mater. Sci.*, **51** (2015), pp. 311–333.
- [18] G. Couegnat, E. Martin, and J. Lamon. “Multiscale Modelling of the Mechanical Behaviour of Woven Composite Materials” (2010).
- [19] S. Chen, M. Wang, and Z. Xia. “Multiscale fluid mechanics and modeling”. *Procedia IUTAM* 10, **10** (2013), pp. 100–114.
- [20] Q. Xu et al. “Multiscale modeling and simulation of directional solidification process of turbine blade casting with MCA method”. *Metallurgical and Materials Transactions B*, **45** (2014), pp. 555–561.
- [21] B. Pipes. *Accelerating the Certification Process for Aerospace Composites*. 2014. URL: <https://www.compositesworld.com/columns/accelerating-the-certification-process-for-aerospace-composites> (visited on 01/19/2021).
- [22] A.R. Ervilha, J.M.C. Pereira, and J.C.F. Pereira. “On the parametric uncertainty quantification of the Rothermel’s rate of spread model”. *Appl. Math. Model.*, **41** (2017), pp. 37–53.
- [23] P. Sasikumar, R. Suresh, and S. Gupta. “Stochastic model order reduction in uncertainty quantification of composite structures”. *Compos. Struct.*, **128** ().
- [24] R.M. Gorguluarslan, S.-K. Choi, and G.W. Woodruff. “A simulation - based upscaling technique for multiscale modeling of engineering systems under uncertainty”. *J. Multiscale Comput. Eng.*, **12** (2014), pp. 549–566.
- [25] B. Ganapathysubramanian and N. Zabaras. “A stochastic multiscale framework for modeling flow through random heterogeneous porous media”. *J. Comput. Phys.*, **228** (2009), pp. 591–618.
- [26] P. Dostert, Y. Efendiev, and T.Y. Hou. “Multiscale finite element methods for stochastic porous media flow equations and application to uncertainty

- quantification”. *Comput. Methods Appl. Mech. Engrg.*, **197** (2008), pp. 3445–3455.
- [27] X. He, L. Jiang, and J.D. Moulton. “A stochastic dimension reduction multiscale finite element method for groundwater flow problems in heterogeneous random porous media”. *J. Hydrol.*, **478** (2013), pp. 77–88.
- [28] L. Shi et al. “A multiscale probabilistic collocation method for subsurface flow in heterogeneous media”. *Water Resour. Res.*, **46** (2010), pp. 1–18.
- [29] N.C. Nguyen. “A multiscale reduced-basis method for parametrized elliptic partial differential equations with multiple scales”. *J. Comput. Phys.*, **227** (2008), pp. 9807–9822.
- [30] F. Anker et al. “SDE based regression for linear random PDEs”. *SIAM J. Sci. Comput.*, **39** (2017), A1168–A1200.
- [31] K. Sepahv and S. Marburg. “Spectral stochastic finite element method in vibroacoustic analysis of fiber-reinforced composites”. *Procedia Eng.*, **199** (2017), pp. 1134–1139.
- [32] T.Y. Hou and X.H. Wu. “A multiscale finite element method for elliptic problems in composite materials and porous media”. *J. Comput. Phys.*, **134** (1997), pp. 169–189.
- [33] B.O. Heimsund J. Aarnes. *Multiscale Discontinuous Galerkin Methods for Elliptic Problems with Multiple Scales*. Vol. 44. In: Engquist B., Runborg O., Lötstedt P. (eds) *Multiscale Methods in Science, Engineering. Lecture Notes in Computational Science, and Engineering*, Springer, 2005.
- [34] X.F. Xu. “A multiscale stochastic finite element method on elliptic problems involving uncertainties”. *Comput. Methods Appl. Mech. Engrg.*, **196** (2007), pp. 2723–2736.

- 
- [35] I. Bilonis and N. Zabaras. “A stochastic optimization approach to coarse-graining using a relative-entropy framework”. *J. Chem. Phys.*, **138** (2013).
- [36] I. Babuška and M. Motamed. “A fuzzy-stochastic multiscale model for fiber composites: A one-dimensional study”. *Comput. Methods Appl. Mech. Engrg.*, **302** (2016), pp. 109–130.
- [37] J. Fish and W. Wu. “A nonintrusive stochastic multiscale solver”. *Int. J. Numer. Methods Eng.*, **88** (2011), pp. 862–879.
- [38] L. Mehrez et al. “A PCE-based multiscale framework for the characterization of uncertainties in complex systems”. *Comput. Mech.*, **61** (2018), pp. 219–236.
- [39] Y. Wang. “Multiscale uncertainty quantification based on a generalized hidden Markov model”. *J. Mech. Des.*, **133** (2011).
- [40] M. Arnst and R. Ghanem. “Probabilistic equivalence and stochastic model reduction in multi-scale analysis”. *Computer Methods in Applied Mechanics and Engineering*, **197** (2008), pp. 3584–3592.
- [41] S.K. Choi et al. “Simulation-based uncertainty quantification for additively manufactured cellular structures”. *J. Electron. Mater.*, **44** (2015), pp. 4035–4041.
- [42] S. Chan and A.H. Elsheikh. “A Machine Learning Approach for Efficient Uncertainty Quantification using Multiscale Methods”. *Journal of Computational Physics*, **354** (2018), pp. 493–511.
- [43] R. Bostanabad et al. “Uncertainty quantification in multiscale simulation of woven fiber composites”. *Comput. Methods Appl. Mech. Engrg.*, **338** (2018), pp. 506–532.
- [44] S. Trehan and L.J. Durlofsky. “Machine-learning-based modeling of coarse-scale error, with application to uncertainty quantification”. *Comput. Geosci.*, **22** (2018), pp. 1–21.

- 
- [45] C. Scheidt and J. Caers. “Uncertainty quantification in reservoir performance using distances and kernel methods—application to a West Africa deepwater turbidite reservoir”. *SPE J.*, **14** (2013), pp. 680–692.
- [46] L. Uusitalo et al. “An overview of methods to evaluate uncertainty of deterministic models in decision support”. *Environ. Model. Softw.*, **63** (2015), pp. 24–31.
- [47] M.A. Tatang et al. “An efficient method for parametric uncertainty analysis of numerical geophysical models”. *Journal of geophysical research*, **102** (1997), pp. 21925–21932.
- [48] M. Eldred and J. Burkardt. “Comparison of Non-Intrusive Polynomial Chaos and Stochastic Collocation Methods for Uncertainty Quantification”. *AIAA* (2009).
- [49] D. Xiu and G.E. Karniadakis. “The Wiener Askey polynomial chaos for stochastic differential equations”. *SIAM journal on scientific computing*, **24** (2002), pp. 619–644.
- [50] S. Oladyshkin and W. Nowak. “Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion”. *Reliability Engineering & System Safety*, **106** (2012), pp. 179–190.
- [51] I.P. Mysovskikh. “On the construction of cubature formulas with fewest nodes”. *Dokl. Akad. Nauk SSSR*, **178** (1968), pp. 1252–1254.
- [52] H. Rutishauser. “On a modification of the QD-algorithm with Graeffe-type convergence”. *Proc. IFIP Congr.*, **62** (1963), pp. 93–96.
- [53] R. Ahlfeld, B. Belkouchi, and F. Montomoli. “SAMBA: Sparse approximation of moment-based arbitrary polynomial chaos”. *J. Comput. Phys.*, **320** (2016), pp. 1–16.

- 
- [54] R. Ahlfeld et al. “Data-driven uncertainty quantification for Formula 1: Diffuser, wing tip and front wing variations”. *Proc IMechE Part D: J Automobile Engineering*, **233** (2019), pp. 1495–1506.
- [55] A. Kaintura, T. Dhaene, and D. Spina. “Review of Polynomial Chaos-Based Methods for Uncertainty Quantification in Modern Integrated Circuits”. *Electronics*, **7** (2018), pp. 1–21.
- [56] M. S. Eldred et al. *Multifidelity Uncertainty Quantification Using Spectral Stochastic Discrepancy Models*. In *Handbook of Uncertainty Quantification*. Springer, 2017.
- [57] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, 1965.
- [58] G. Blatman and B. Sudret. “Adaptive sparse polynomial chaos expansion based on least angle regression”. *J. Comp. Phys.*, **230** (2011), pp. 2345–2367.
- [59] A. Doostan and G. Iaccarino. “A least-squares approximation of partial differential equations with high-dimensional random inputs”. *J. Comp. Phys.*, **228** (2009), pp. 4332–4345.
- [60] J. Peng, J. Hampton, and A. Doostan. “A weighted l1-minimization approach for sparse polynomial chaos expansions”. *J. Comp. Phys.*, **267** (2014), pp. 92–111.
- [61] X. Yang and G.E. Karniadakis. “Reweighted l1 minimization method for stochastic elliptic differential equations”. *Journal of Computational Physics*, **248** (2013), pp. 87–108.
- [62] K. Sargsyan et al. “Dimensionality reduction for complex models via Bayesian compressive sensing”. *International Journal for Uncertainty Quantification*, **4** (2014).
- [63] P. Tsilifis et al. “Sparse Polynomial Chaos expansions using variational relevance vector machines”. *J. Comp. Phys.*, **416** (2020).

- 
- [64] D. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley, 1992.
- [65] B. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.
- [66] J. Hwang, S. Lay, and A. Lippman. “Nonparametric Multivariate Density Estimation: A Comparative Study”. *IEEE Trans. Signal Process.*, **42** (1994), pp. 2795–2810.
- [67] A. Bowman. “An alternative method of cross-validation for the smoothing of density estimates”. *Biometrics*, **71** (1984), pp. 353–360.
- [68] M. Jones, J. Marron, and J. Sheather. “A Brief Survey of Bandwidth Selection for Density Estimation”. *J. Am. Stat. Assoc.*, **91** (1996), pp. 401–407.
- [69] S. Kullback and R. A. Leibler. “On information and sufficiency”. *Ann. Math. Stat.*, **22** (1951), pp. 79–86.
- [70] G. Corder and D. Foreman. *Nonparametric Statistics: A Step-by-Step Approach*. Wiley, 2014.
- [71] J. A. Nelder and R. Mead. “A Simplex Method for Function Minimization”. *Comput. J.*, **7** (1965), pp. 308–313.
- [72] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. “Optimization by simulated annealing”. *Science*, **220** (1983), pp. 671–680.
- [73] J. Kennedy and R. Eberhart. “Particle Swarm Optimization”. *Proceedings of IEEE International Conference on Neural Networks* (1995), pp. 1942–1948.
- [74] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., 1989.
- [75] R. H. Byrd, M. E. Hribar, and J. Nocedal. “An Interior Point Algorithm for Large-Scale Nonlinear Programming”. *SIAM J. Optim.*, **9** (1999), pp. 877–900.

- 
- [76] R.A. Waltz et al. “An interior algorithm for nonlinear optimization that combines line search and trust region steps”. *Math. Program.*, **107** (2006), pp. 391–408.
- [77] A. Riccio et al. “Numerical simulations of inter-laminar damage evolution in a composite wing box”. *Appl. Compos. Mater.*, **21** (2013), pp. 467–481.
- [78] M. Smith. *ABAQUS/Standard User’s Manual, Version 6.9*. Dassault Systèmes Simulia Corp, 2009.
- [79] R. Ahlfeld et al. “Uncertainty quantification for fat-tailed probability distributions in aircraft engine simulations”. *J. Propuls. Power*, **33** (2017), pp. 881–890.
- [80] M. Zhdanov. *Inverse Theory and Applications in Geophysics*. Elsevier, 2015.
- [81] H. Ben Ameer, G. Chavent, and J. Jaffré. “Refinement and coarsening indicators for adaptive parametrization: application to the estimation of hydraulic transmissivities”. *Inv. Pr.*, **18** (2002), pp. 775–794.
- [82] A. Jones and C. Taylor. “Solving Inverse Problems in Computer Vision by Scale Space Reconstruction”. *IAPR Workshop on Machine Vision Applications* (1994), pp. 401–404.
- [83] L. Lucy. “Astronomical Inverse Problems”. *Reviews in Modern Astronomy*, **7** (1994), pp. 31–50.
- [84] M. Ahmadi and W. S. Ghaly. “Aerodynamic inverse design of turbomachinery cascades using a finite volume method on unstructured meshes”. *Inverse Problems in Engineering*, **6** (1998), pp. 281–298.
- [85] B. Roidl and W. Ghaly. “Redesign of a low speed turbine stage using a new viscous inverse design method”. *J. Turbomach.*, **133** (2011), pp. 011009–011009.
- [86] M. Nili-Ahmadabadi et al. “A Novel 2D Incompressible Viscous Inverse Design Method for Internal Flows Using Flexible String Algorithm”. *J. Fluids Eng.*, **132** (2010), 031401 (10 pages).



- 
- [87] N. P. Kruyt and R. W. Westra. “On the inverse problem of blade design for centrifugal pumps and fans”. *Inverse Problems*, **30** (2014), p. 065003.
- [88] J. Hadamard. *Lectures on Coghy’s problem in Linear Partial Differential Equations*. Yale University Press, 1923.
- [89] P.C. Sabatier. “Past and future of inverse problems”. *Journal of Mathematical Physics*, **41** (2000).
- [90] R. McGreevy and L. Pusztai. “Reverse Monte Carlo Simulation: A New Technique for the Determination of Disordered Structures”. *Mol. Simul.*, **1** (1988), pp. 359–367.
- [91] M. Liu et al. “Simultaneous-shot inversion for PDE-constrained optimization problems with missing data”. *Inverse Problems*, **35** (2018), pp. 1–20.
- [92] M. Grote, M. Kray, and U. Nahum. “Adaptive eigenspace method for inverse scattering problems in the frequency domain”. *Inverse Problems*, **33** (2017), pp. 1–22.
- [93] A. Von Moll et al. “A Review of Exhaust Gas Temperature Sensing Techniques for Modern Turbine Engine Controls”. *50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference* (2014).
- [94] J. Molimard et al. “Identification of Orthotropic Plate Stiffness Using Open Hole Tensile Test”. *Exp. Mech.*, **45** (2005), pp. 404–411.
- [95] M. Anghileri et al. “An inverse approach to identify the constitutive model parameters for crashworthiness modelling of composite structures”. *Compos. Struct.*, **68** (2005), pp. 65–74.
- [96] Y. L. Kang, X. H. Lin, and Q. H. Qin. “Inverse / genetic method and its application in identification of mechanical parameters of interface in composite”. *Compos. Struct.*, **66** (2004), pp. 449–458.

- 
- [97] W. Ogierman. “Inverse Identification of Elastic Properties of Constituents of Discontinuously Reinforced Composites”. *Materials (Basel)*, **11** (2018), p. 232.
- [98] T. Vo-Duy et al. “A two-step approach for damage detection in laminated composite structures using modal strain energy method and an improved differential evolution algorithm”. *Compos. Struct.*, **147** (2016), pp. 42–53.
- [99] E. Ficarella, L. Lamberti, and S. O. Degertekin. “Mechanical Identification of Materials and Structures with Optical Methods and Metaheuristic Optimization”. *Materials (Basel)*, **12** (2019), p. 2133.
- [100] R. Viala, V. Placet, and S. Cogan. “Identification of the anisotropic elastic and damping properties of complex shape composite parts using an inverse method based on finite element model updating and 3D velocity fields measurements ( FEMU-3DVF ): Application to bio-based composite violins”. *Compos. Part A*, **106** (2018), pp. 91–103.
- [101] M. Springmann and M. Kuna. “Identification of material parameters of the Rousselier model by non-linear optimization”. *Comp. Mat. Sci.*, **26** (2003), pp. 202–209.
- [102] B. Rahmani et al. “A new approach to inverse identification of mechanical properties of composite materials : Regularized model updating”. *Compos. Struct.*, **105** (2013), pp. 116–125.
- [103] M. Grédiac. “Principe des travaux virtuels et identification”. *Comptes Rendus l’Académie des Sci.* (1989), pp. 1–5.
- [104] M. Grédiac, E. Toussaint, and F. Pierron. “Special virtual fields for the direct determination of material parameters with the virtual fields method. 2–Application to in-plane properties”. *Int. J. Solids Struct.*, **39** (2002), pp. 2707–2730.

- 
- [105] B. Rahmani, I. Villemure, and M. Levesque. “Regularized virtual fields method for mechanical properties identification of composite materials”. *Comput. Methods Appl. Mech. Engrg.*, **278** (2014), pp. 543–566.
- [106] F. Pierron et al. “Identification of the Orthotropic Elastic Stiffnesses of Composites with the Virtual Fields Method: Sensitivity Study and Experimental Validation”. *Strain*, **43** (2007), pp. 250–259.
- [107] S. Avril et al. “Overview of Identification Methods of Mechanical Parameters Based on Full-field Measurements”. *Exp. Mech.*, **48** (2008), pp. 381–402.
- [108] N. Pepper, F. Montomoli, and S. Sharma. “Multiscale uncertainty quantification with arbitrary polynomial chaos”. *Computer Methods in Applied Mechanics and Engineering*, **357** (2019), pp. 1–20.
- [109] S. Das, R. Ghanem, and J. C. Spall. “Asymptotic Sampling Distribution for Polynomial Chaos Representation of Data: A Maximum Entropy and Fisher information approach”. *Proc. 45th IEEE Conf. Decis. Control* (2006), pp. 4139–4144.
- [110] S. Das, R. Ghanem, and S. Finette. “Polynomial chaos representation of spatio-temporal random fields from experimental measurements”. *J. Comput. Phys.*, **228** (2009), pp. 8726–8751.
- [111] R. Ahlfeld. “A Data-Driven Uncertainty Quantification Method for Scarce Data and Rare Events”. *PhD thesis* (2017).
- [112] M. Fukuta et al. “Performance of scroll expander for co2 refrigeration cycle”. *International Compressor Engineering Conference* (2006).
- [113] X. Wang, Y. Hwang, and R. Radermacher. “Two-stage heat pump system with vapor-injected scroll compressor using r140a as a refrigerant”. *International Journal of Refrigeration*, **32** (2009), pp. 1442–1451.

- 
- [114] Y.R. Lee and W.F. Wu. “On the profile design of a scroll compressor”. *International Journal of refrigeration*, **18** (1995), pp. 308–317.
- [115] Y. Arai et al. “A High Speed and Compact System for Profile Measurement of Scroll Compressors”. *International Journal of Precision Engineering and Manufacturing*, **10** (2009), pp. 27–32.
- [116] J. Wang et al. “Design methodology and geometric modeling of complete meshing profiles for scroll compressors”. *International Journal of Refrigeration*, **91** (2018), pp. 199–210.
- [117] A. Dvoretzky, J. Kiefer, and J. Wolfowitz. “Asymptotic Minimax Character of the Sample Distribution Function and of the Classical Multinomial Estimator”. *Ann. Math. Stat.*, **27** (1956), pp. 642–669.
- [118] P. Massart. “The Tight Constant in the Dvoretzky-Kiefer-Wolfowitz Inequality”. *Ann. Probab.*, **18** (1990), pp. 1269–1283.
- [119] L. Bek et al. “Calculation of B-Basis Values from Composite Material Strength Parameters Obtained from Measurements of Non-Identical Batches”. *EAN 2017 - 55th Conference on Experimental Stress Analysis 2017* (2017), pp. 477–485.
- [120] O. Martin et al. “Virtual calculation of the B-value allowables of notched composite laminates”. *Compos. Struct.*, **212** (2018), pp. 11–21.
- [121] G. Odegard and M. Kumosa. “Determination of shear strength of unidirectional composite materials with the Iosipescu and 10 degree off-axis shear tests”. *Compos. Sci. Technol.*, **60** (2000), pp. 2917–2943.
- [122] L. Melin et al. “Evaluation of Four Composite Shear Test Methods by Digital Speckle Strain Mapping and Fractographic Analysis”. *J. Compos. Technol. Res.*, **22** (2000), pp. 161–172.

- 
- [123] A. Nouy. “Identification of multi-modal random variables through mixtures of polynomial chaos expansions”. *Comptes Rendus Mécanique*, **338** (2010), pp. 689–703.
- [124] D. Hull. *An Introduction to Composite Materials*. Cambridge University Press, 1981.
- [125] R. M. Jones. *Mechanics of Composite Materials*. Taylor & Francis, 1999.
- [126] I. M. Daniel and O. Ishai. *Engineering Mechanics of Composite Materials*. Oxford University Press, 2005.
- [127] K. Potter. “Understanding the origin of defects and variability in composites manufacture”. *Proceedings of the 17th international conference on composite materials* (2009).
- [128] S. E. Yamada and C. T. Sun. “Analysis of Laminate Strength and its Distribution”. *J. Compos. Mater*, **12** (1987), pp. 275–284.
- [129] B. Sudret. “Global sensitivity analysis using polynomial chaos expansions”. *Reliab. Eng. Syst. Saf.*, **93** (2008), pp. 964–979.
- [130] T. Crestaux, O. Le Maitre, and J. M. Martinez. “Polynomial chaos expansion for sensitivity analysis”. *Reliab. Eng. Syst. Saf.*, **94** (2009), pp. 1161–1172.
- [131] S. Hosder, R. W. Walters, and R. Perez. “A Non-Intrusive Polynomial Chaos Method For Uncertainty Propagation in CFD Simulations”. *44th AIAA Aerospace Sciences Meeting and Exhibit* (2006).
- [132] J.L. Wu, H. Xiao, and E.G. Paterson. “Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework”. *Phys. Rev. Fluids*, **3** (2018).
- [133] H. Xiao et al. “Quantifying and reducing model-form uncertainties in Reynolds-averaged Navier-Stokes simulations: A data-driven, physics-informed bayesian approach”. *J. Comput. Phys.*, **324** (2016), pp. 115–136.

- 
- [134] A. Doostan, G. Geraci, and G. Iaccarino. “A Bi-fidelity for Uncertainty Quantification of Heat Transfer in a Rectangular Ribbed Channel”. *Proceedings of ASME Turbo Expo 2016: Turbomachinery Technical Conference and Exposition* (2016), pp. 1–9.
- [135] M. Giles. “Multilevel Monte Carlo methods”. *Acta Numer.*, **24** (2015), pp. 259–328.
- [136] J. Charrier, R. Scheichl, and A. L. Teckentrup. “Finite Element Error Analysis of Elliptic PDEs with Random Coefficients and Its Application to Multilevel Monte Carlo Methods”. *SIAM Journal on Numerical Analysis*, **51** (2013), pp. 322–352.
- [137] M. Pisaroni, F. Nobile, and P. Leyland. “A Continuation Multi Level Monte Carlo (C-MLMC) method for uncertainty quantification in compressible inviscid aerodynamics”. *Comput. Methods Appl. Mech. Eng.*, **326** (2017), pp. 20–50.
- [138] M. C. Kennedy and A. O’Hagan. “Bayesian calibration of computer models”. *J. Roy. Stat. Soc. Ser. B*, **63** (2001), pp. 425–464.
- [139] D. Crevillén-García et al. “Gaussian process modelling for uncertainty quantification in convectively-enhanced dissolution processes in porous media”. *Adv. Water Resour.*, **99** (2017), pp. 1–14.
- [140] P. T. Roy et al. “Comparison of polynomial chaos and Gaussian process surrogates for uncertainty quantification and correlation estimation of spatially distributed open-channel steady flows”. *Stoch. Env. Res Risk Assess*, **32** (2018), pp. 1723–1741.
- [141] I. Bilonis and N. Zabarar. *Handbook of Uncertainty Quantification*. Springer, 2015.
- [142] B. Peherstorfer, K. Willcox, and M. Gunzburger. “Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization”. *SIAM Rev.*, **60** (2018), pp. 550–591.

- 
- [143] P. Diaconis and D. Freedman. “On the Consistency of Bayes Estimates”. *Ann. Statist.*, **14** (1986), pp. 1–26.
- [144] O. Vallmajó et al. “Virtual calculation of the B-value allowables of notched composite laminates”. *Compos. Struct.*, **212** (2018), pp. 11–21.
- [145] A.P. Dempster. “Upper and lower probabilities induced by a multivalued mapping”. *The Annals of Mathematical Statistics*, **38** (1967), pp. 325–339.
- [146] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
- [147] A. Paksoy and M. Göktürk. “Information fusion with dempster-shafer evidence theory for software defect prediction”. *Procedia Computer Science*, **3** (2011), pp. 600–605.
- [148] B. Quost, M.H. Masson, and T. Denceux. “Classifier fusion in the Dempster–Shafer framework using optimized t-norm based combination rules”. *International Journal of Approximate Reasoning*, **52** (2011), pp. 353–374.
- [149] J. Pearl. “Reasoning with belief functions: An analysis of compatibility”. *International Journal of Approximate Reasoning*, **4** (1990), pp. 363–389.
- [150] S. Hall and J. Mitchell. “Combining density forecasts”. *Int. J. Forecast.*, **23** (2007), pp. 1–13.
- [151] A. Garratt et al. “Forecast Uncertainties in Macroeconomic Modeling: An Application to the U.K. Economy”. *J. Am. Stat. Assoc.*, **98** (2003), pp. 829–838.
- [152] M. Stone. “The Opinion Pool”. *Ann. Math. Stat.*, **32** (1961), pp. 1339–1342.
- [153] C. Genest and J. Zidek. “Combining probability distributions: A critique and an annotated bibliography”. *Stat. Sci.*, **1** (1986), pp. 114–135.
- [154] R. Clemen and R. Winkler. “Combining Probability Distributions From Experts in Risk Analysis”. *Risk Anal.*, **19** (1999), pp. 187–203.
- [155] T. Wallsten et al. “Evaluating and Combining Subjective Probability Estimates”. *J. Behav. Decis. Mak.*, **10** (1997), pp. 243–268.

- 
- [156] S. Hora. “Probability judgements for continuous quantities: Linear combinations and calibration”. *Manage. Sci.*, **50** (2004), pp. 597–604.
- [157] R. Ranjan and T. Gneiting. “Combining probability forecasts”. *J. R. Stat. Soc.*, **72** (2010), pp. 71–91.
- [158] A. Allard, A. Comunian, and P. Renard. “Probability aggregation methods in geoscience”. *Math. Geosci.*, **44** (2012), pp. 545–581.
- [159] T. Gneiting and R. Ranjan. “Combining predictive distributions”. *Electron. J. Stat.*, **7** (2013), pp. 1747–1782.
- [160] F. Bassetti, R. Casarin, and F. Ravazzolo. “Bayesian Nonparametric Calibration and Combination of Predictive Distributions”. *J. Am. Stat. Assoc.*, **113** (2018), pp. 675–685.
- [161] R. Casarin, G. Mantoan, and F. Ravazzolo. “Bayesian Calibration of Generalized Pools of Predictive Distributions”. *Econometrics*, **4** (2016).
- [162] G. Kapetanios et al. “Generalised density forecast combinations”. *J. Econom.*, **188** (2015), pp. 150–165.
- [163] E. Jaynes. “Information Theory and Statistical Mechanics”. *Phys. Rev.*, **106** (1957), pp. 620–630.
- [164] E. Jaynes. “On the Rationale of Maximum-Entropy Methods”. *Proc. IEEE*, **70** (1982), pp. 939–952.
- [165] A. Cesari, S. Reißer, and G. Bussi. “Using the Maximum Entropy Principle to Combine Simulations and Solution Experiments”. *Computation*, **6** (2018), pp. 1–25.
- [166] H. Lou and R. Cukier. “Reweight ensemble probabilities with experimental histogram data constraints using a maximum entropy principle”. *J. Chem. Phys.*, **149** (2018).



- 
- [167] A. Cesari, A. Gil-Ley, and G. Bussi. “Combining Simulations and Solution Experiments as a Paradigm for RNA Force Field Refinement”. *J. Chem. Theory Comput.*, **12** (2016), pp. 6192–6200.
- [168] P. M. Baggenstoss. “Beyond Moments: Extending the Maximum Entropy Principle to Feature Distribution Constraints”. *Entropy*, **20** (2018).
- [169] S. Ferson et al. “Constructing probability boxes and Dempster-Shafer structures”. *Sandia National Laboratories* (2003).
- [170] F. Montomoli et al. “Interaction of wheelspace coolant and main flow in a new aeroderivative low pressure turbine”. *Journal of turbomachinery*, **132** (2010), pp. 1–7.
- [171] F. Montomoli and M. Massini. “Gas turbines and uncertainty quantification: Impact of PDF tails on UQ predictions, the Black Swan”. *Proceedings of ASME Turbo Expo 2013: Power for Land, Sea, and Air* (2013).
- [172] J.R.M. Hosking and J.R. Wallis. “Parameter and Quantile Estimation for the Generalized Pareto Distribution”. *Technometrics*, **23** (1987), pp. 339–349.
- [173] R.L. Smith. “Estimating the tails of probability distributions”. *Ann. Statist.*, **15** (1987), pp. 1174–1207.
- [174] R.H. Byrd, M.E. Hribar, and J. Nocedal. “An Interior Point Algorithm for Large-Scale Nonlinear Programming”. *SIAM Journal on Optimization*, **9** (1999), pp. 877–900.
- [175] R. A. Waltz et al. “An interior algorithm for nonlinear optimization that combines line search and trust region steps”. *Mathematical Programming*, **107** (2006), pp. 391–408.
- [176] A. Gaymann et al. “Random Variable Estimation and Model Calibration in the Presence of Epistemic and Aleatory Uncertainties”. *WCX 18 SAE World Congr. Exp.* (2018), pp. 1–14.

- 
- [177] Z.G. Ghauch et al. “Integrated stochastic analysis of fiber composites manufacturing using adapted polynomial chaos expansions”. *Composites Part A: Applied Science and Manufacturing*, **118** ().
- [178] P. Sasikumar et al. “A data driven polynomial chaos based approach for stochastic analysis of CFRP laminated composite plates”. *Composite Structures*, **125** (2015), pp. 212–227.
- [179] F. Zhang et al. “Efficient stochastic simulation approach for RTM process with random fibrous permeability”. *Composites Science and Technology*, **71** (2011), pp. 1478–1485.
- [180] D. Mavris and D. DeLaurentis. “Methodology for Examining the Simultaneous Impact of Requirements, Vehicle Characteristics, and Technologies on Military Aircraft Design”. *ICAS* (2000).
- [181] C.P. Frank et al. “Evolutionary multi-objective multi-architecture design space exploration methodology”. *Optim Eng*, **19** (2018), pp. 359–381.
- [182] J. Ölvander, B. Lundéna, and H. Gavel. “A computerized optimization framework for the morphological matrix applied to aircraft conceptual design”. *Computer-Aided Design*, **41** (2009), pp. 187–196.
- [183] J.H. Bussemaker, P.D. Ciampa, and B. Nagel. “System Architecture Design Space Exploration: An Approach to Modeling and Optimization”. *AIAA Aviation 2020 Forum* (2020).
- [184] D.J. Singer, N. Doerry, and M.E. Buckley. “What Is Set-Based Design?” *Naval Engineers Journal*, **121** (2009), pp. 31–43.
- [185] M. Guenov et al. “Aircadia -an interactive tool for the composition and exploration of aircraft computational studies at early design stage”. *Proceedings of the 29th Congress of the International Council of the Aeronautical Sciences* (2014).

- 
- [186] A. Georgiades et al. “ADOPT: An augmented set-based design framework with optimisation”. *Design Science*, **5** (2019), pp. 1–40.
- [187] A. D. Pimentel. “Exploring Exploration: A Tutorial Introduction to Embedded Systems Design Space Exploration”. *IEEE Design & Test*, **34** (2017), pp. 77–90.
- [188] M. Gries. “Methods for evaluating and covering the design space during early design development”. *Integration*, **38** (2004), pp. 131–183.
- [189] A. Schulz et al. “Interactive Design Space Exploration and Optimization for CAD Models”. *ACM Transactions on Graphics*, **36** (2017).
- [190] E. Ipek et al. “Efficient Architectural Design Space Exploration via Predictive Modeling”. *ACM. Trans. Architec. Code Optim.*, **4** (2008).
- [191] A.L. Read. “Linear interpolation of histograms”. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, **425** (1999), pp. 357–360.
- [192] M. Baak et al. “Interpolation between multi-dimensional histograms using a new non-linear moment morphing method”. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, **771** (2015), pp. 39–48.
- [193] P. Baldi. “Autoencoders, Unsupervised Learning, and Deep Architectures”. *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, **27** (2012), pp. 37–49.
- [194] R. Schöbi, B. Sudret, and J. Wiart. “Polynomial-Chaos-Based-Kriging”. *International Journal for Uncertainty Quantification*, **5** (2015), pp. 171–193.
- [195] R. Schöbi, B. Sudret, and S. Marelli. “Rare Event Estimation Using Polynomial-Chaos Kriging”. *ASCEASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, **3** (2016).

- 
- [196] C. Thimmisetty et al. “Multiscale stochastic representations using Polynomial Chaos Expansions with Gaussian Process Coefficients”. *Data-Enabled Discov. Appl.*, **2** (2018).
- [197] A. Bowman. “An alternative method of cross-validation for the smoothing of density estimates”. *Biometrics*, **71** (1984), pp. 353–360.
- [198] M. Jones, J. Marron, and J. Sheather. “A Brief Survey of Bandwidth Selection for Density Estimation”. *J. Am. Stat. Assoc.*, **91** (1996), pp. 401–407.
- [199] K. Z. Mao. “Orthogonal forward selection and backward elimination algorithms for feature subset selection”. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, **34** (2004), pp. 629–634.
- [200] R.P. Dwight and Z.H. Han. “Efficient Uncertainty Quantification using Gradient-Enhanced Kriging”. *11th AIAA Non-Deterministic Approaches Conference* (2009).
- [201] B.A. Lockwood and M. Anitescu. “Gradient-Enhanced Universal Kriging for Uncertainty Propagation,” *Nuclear Science and Engineering*, **170** (2012), pp. 168–195.
- [202] B. Echard et al. “A combined Importance Sampling and Kriging reliability method for small failure probabilities with time-demanding numerical models”. *Reliability Engineering & System Safety*, **111** (2013), pp. 232–240.
- [203] V. Dubourg, B. Sudret, and J.M. Bourinet. “Reliability-based design optimization using kriging surrogates and subset simulation”. *Struct Multidisc Optim*, **44** (2011), pp. 673–690.
- [204] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
- [205] T.J. Santner, B.J. Williams, and W.I. Notz. *The design and analysis of computer experiments*. Springer, 2003.

- 
- [206] C.J. Moore et al. “Fast methods for training gaussian processes on large datasets”. *Royal Society Open Science*, **3** (2016).
- [207] D. Petelin, B. Filipič, and J. Kocijan. “Optimization of Gaussian Process Models with Evolutionary Algorithms”. *Adaptive and Natural Computing Algorithms*. Ed. by A. Dobnikar, U. Lotrič, and B. Šter. Vol. **6593**. Lecture Notes in Computer Science. Springer, 2011.
- [208] M. Eldred et al. “Investigation of reliability method formulations in DAKOTA/UQ”. *Structure and Infrastructure Engineering*, **3** (2007), pp. 199–213.
- [209] Y.T. Wu et al. “Safety-factor based approach for probability-based design optimization”. In *Proc. 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, **196** (2001), pp. 199–342.
- [210] M.W. Huang and J.S. Arora. “Optimal Design with Discrete Variables: Some Numerical Examples”. *International Journal for Numerical Methods in Engineering*, **40** (1997), pp. 165–188.
- [211] L. Pronzato and W.G. Müller. “Design of computer experiments: space filling and beyond”. *Stat Comput*, **22** (2012), pp. 681–701.
- [212] H. Liu, Y.S. Ong, and J. Cai. “A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design”. *Struct Multidisc Optim*, **57** (2018), pp. 393–416.
- [213] D. Deschrijver et al. “Adaptive Sampling Algorithm for Macromodeling of Parameterized S-Parameter Responses”. *IEEE Transactions on Microwave Theory and Techniques*, **59** (2011), pp. 39–45.
- [214] A. Bhattacharyya. “On a measure of divergence between two statistical populations defined by their probability distributions”. *Bulletin of the Calcutta Mathematical Society*, **35** (1943), pp. 99–109.

- 
- [215] H.F. Inman and E.L. Bradley. “The overlapping coefficient as a measure of agreement between probability distributions and point estimation of the overlap of two normal densities”. *Communications in Statistics - Theory and Methods*, **18** (1989), pp. 3851–3874.
- [216] M.C. Kennedy and A. O’Hagan. “Predicting the output from a complex computer code when fast approximations are available”. *Biometrika*, **87** (2000), pp. 1–13.
- [217] A.I.J. Forrester, A. Sobester, and A. Keane. “Multi-fidelity optimization via surrogate modelling”. *Proc. Roy. Soc. A*, **463** (2007), pp. 3251–3269.
- [218] J. Hensman, N. Fusi, and N.D. Lawrence. “Gaussian processes for big data”. *Proceedings of the twenty-ninth conference on uncertainty in artificial intelligence* (2015).
- [219] N. Durrande, D. Ginsbourger, and O. Roustant. “Additive covariance kernels for high-dimensional gaussian process modeling”. *Ann Fac Sci Toulouse Math*, **21** (2012), pp. 481–499.
- [220] D. Liu and Y. Wang. “Multi-Fidelity Physics-Constrained Neural Network and Its Application in Materials Modeling”. *Journal of Mechanical Design*, **141** (2019), pp. 1–13.
- [221] E. Weinan, J. Han, and A. Jentzen. “Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations”. *Commun. Math. Stat.*, **5** (2017), pp. 349–380.
- [222] Z. Zhang et al. “Opening the black box of neural networks: methods for interpreting neural network models in clinical applications”. *Ann Transl Med.*, **6** (2018).
- [223] M. Xuhui and G.E. Karniadakis. “A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems”. *Journal of Computational Physics*, **401** (2020), pp. 1–15.

- 
- [224] G.G. Towell and J.W. Shavlik. “Knowledge-based artificial neural networks”. *Artificial Intelligence*, **70** (1994), pp. 119–165.
- [225] G.G. Towell and J.W. Shavlik. “Extracting refined rules from knowledge-based neural networks”. *Machine Learning*, **13** (1993), pp. 71–101.
- [226] F. Wang and Q. Zhang. “Knowledge-based neural networks for microwave design”. *IEEE Transactions on Microwave Theory and Techniques*, **44** (1997), pp. 2333–2343.
- [227] S.J. Leary, A. Bhaskar, and A.J. Keane. “A knowledge-based approach to response surface modelling in multifidelity optimisation”. *Journal of Global Optimization*, **26** (2003), pp. 297–319.
- [228] H.S. Kim, M. Koc, and J. Ni. “A hybrid multi-fidelity approach to the optimal design of warm forming processes using a knowledge-based artificial neural network”. *International Journal of Machine Tools & manufacture*, **47** (2007), pp. 211–222.
- [229] G.N. Coleman and R.D. Sandberg. “A primer on direct numerical simulation of turbulence-methods, procedures and guidelines”. *Tech. Rep AFM-09/01a* (2010).
- [230] J.X. Xiao H.and Wang, R. Sun, and C. Roy. “Quantifying and reducing model-form uncertainties in Reynolds-averaged Navier-Stokes simulations: A data-driven, physics-informed bayesian approach”. *J. Comput. Phys.*, **324** (2016), pp. 115–136.
- [231] L.J. Voet et al. “A hybrid approach combining DNS and RANS simulations to quantify uncertainties in turbulence modelling”. *Applied Mathematical Modelling*, **89** (2021), pp. 885–906.
- [232] C. J. Brooks et al. “Multifidelity design optimisation of a transonic compressor rotor”. *9th European Conference on Turbomachinery Fluid Dynamics and Thermodynamics* (2011).

- [233] H. Shah et al. “Multi-fidelity robust aerodynamic design optimization under mixed uncertainty”. *Aerospace Science and Technology*, **45** (2015), pp. 17–29.
- [234] H. Leung and S. Haykin. “Rational function neural network”. *Neural Computation*, **5** (1993), pp. 928–938.
- [235] A. Abdelbar and G. Tagliarini. “Honest: A new high order feedforward neural networks”. *Proc. IEEE Int. Conf. Neural Networks* (1996), pp. 1257–1262.
- [236] J. Duchi, E. Hazan, and Y. Singer. “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization”. *Journal of Machine Learning Research*, **12** (2011), pp. 2121–2159.
- [237] N. Pepper. “Knowledge Based Neural Networks”.  
<https://uk.mathworks.com/matlabcentral/fileexchange/93675-knowledge-based-neural-networks>, *MATLAB Central File Exchange* (2021).
- [238] N. Pepper. “Knowledge Based Neural Networks”.  
<https://github.com/npepperUQLab/Knowledge-Based-Neural-Network->,  
*GitHub* (2021).
- [239] B.L. Happel and J.M. Murre. “Design and evolution of modular neural network architectures”. *Neural Networks*, **7** (1994), pp. 985–1004.
- [240] D. Padmanabhan et al. “A study using Monte Carlo Simulation for failure probability calculation in Reliability-Based Optimization”. *Optimization and Engineering*, **7** (2006), pp. 297–316.
- [241] B. Echard, N. Gayton, and M. Lemaire. “AK-MCS: An active learning reliability method combining Kriging and Monte Carlo Simulation”. *Structural Safety*, **33** (2011), pp. 145–154.
- [242] Z. Meng et al. “An active weight learning method for efficient reliability assessment with small failure probability”. *Structural and Multidisciplinary Optimization*, **61** (2020), pp. 1157–1170.



- 
- [243] H.M. Gomes and A.M. Awruch. “Comparison of response surface and neural network with other methods for structural reliability analysis”. *Structural Safety*, **26** (2004), pp. 49–67.
- [244] M.R. Rajashekhar and B.R. Ellingwood. “A new look at the response surface approach for reliability analysis”. *Structural Safety*, **12** (1993), pp. 205–220.
- [245] V. Vapnik. *Statistical learning theory, vol. 1*. Wiley, 1998.
- [246] V. Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 2013.
- [247] C.J.C. Burges. “A Tutorial on Support Vector Machines for Pattern Recognition”. *Data Mining and Knowledge Discovery*, **2** (1998), pp. 121–167.
- [248] C.M. Rocco and J.A. Moreno. “Fast Monte Carlo reliability evaluation using support vector machine”. *Reliability Engineering & System Safety*, **76** (2002), pp. 237–243.
- [249] J.E. Hurtado. “Filtered importance sampling with support vector margin: A powerful method for structural reliability analysis”. *Structural Safety*, **29** (2007), pp. 2–15.
- [250] M.A. Hariri-Ardebili and F. Pourkamali-Anaraki. “Support vector machine based reliability analysis of concrete dams”. *Soil Dynamics and Earthquake Engineering*, **104** (2018), pp. 276–295.
- [251] A. Basudhar et al. “Constrained efficient global optimization with support vector machines”. *Structural and Multidisciplinary Optimization*, **46** (2012), pp. 201–221.
- [252] R. Teixeira, M. Nogal, and A. O’Connor. “Adaptive approaches in metamodel-based reliability analysis: A review”. *Structural Safety*, **89** (2021), p. 102019.

- 
- [253] A. Basudhar and S. Missoum. “Adaptive explicit decision functions for probabilistic design and optimization using support vector machines”. *Computers & Structures*, **86** (2008), pp. 1904–1917.
- [254] A. Basudhar and S. Missoum. “An improved adaptive sampling scheme for the construction of explicit boundaries”. *Structural and Multidisciplinary Optimization*, **42** (2010), pp. 517–529.
- [255] Q. Pan and D. Dias. “An efficient reliability method combining adaptive Support Vector Machine and Monte Carlo Simulation”. *Structural Safety*, **67** (2017), pp. 85–95.
- [256] U. Alibrandi, A.M. Alani, and G. Ricciardi. “A new sampling strategy for SVM-based response surface for structural reliability analysis”. *Probabilistic Engineering Mechanics*, **41** (2015), pp. 1–12.
- [257] A. Roy and S. Chakraborty. “Support vector regression based metamodel by sequential adaptive sampling for reliability analysis of structures”. *Reliability Engineering & System Safety*, **200** (2020), p. 106948.
- [258] J. Shawe-Taylor and S. Sun. “A review of optimization methodologies in support vector machines”. *Neurocomputing*, **74** (2011), pp. 3609–3618.
- [259] D. López-Sánchez, J. Manuel Corchado, and A. González Arrieta. “Data-independent Random Projections from the feature-map of the homogeneous polynomial kernel of degree two”. *Information Sciences*, **436** (2018), pp. 214–226.
- [260] A. Bondu, V. Lemaire, and M. Boullé. “Exploration vs. exploitation in active learning : A Bayesian approach”. *The 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, Spain* (2010), pp. 1–7.

- 
- [261] L. Schueremans and D. van Gemert. “Benefit of splines and neural networks in simulation based structural reliability analysis”. *Structural Safety*, **27** (2005), pp. 246–261.
- [262] A. Borri and E. Speranzini. “Structural reliability analysis using a standard deterministic finite element code”. *Structural Safety*, **19** (1997), pp. 361–382.
- [263] C. Coulais et al. “Discontinuous Buckling of Wide Beams and Metabeams”. *Phys. Rev. Lett.*, **115** (2015), p. 044301.
- [264] H. Kim, B. Mallick, and C. Holmes. “Analyzing Nonstationary Spatial Data Using Piecewise Gaussian Processes”. *Journal of the American Statistical Association*, **100** (2005), pp. 653–668.
- [265] J.D. Jakeman, R. Archibald, and D. Xiu. “Characterization of discontinuities in high-dimensional stochastic problems on adaptive sparse grids”. *Journal of Computational Physics*, **230** (2011), pp. 3977–3997.
- [266] Y. Gal and Z. Ghahramani. “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning”. *Proceedings of The 33rd International Conference on Machine Learning*, **48** (2016), pp. 1050–1059.
- [267] A. Okubo and S.A. Levin. “Diffusion and ecological problems: modern perspectives”. *Springer* (2013).
- [268] S. Salsa. “Partial differential equations in action. From modelling to theory”. *Springer* (2008).
- [269] M. Bonneaua, F. Johnsonb, and C. Romagosaa. “Spatially explicit control of invasive species using a reaction–diffusion model”. *Ecological Modelling*, **337** (2016), pp. 15–24.
- [270] A. Ern and J.L. Guermond. “Theory and Practice of Finite Elements”. *Springer* (2006).

- [271] L. Formaggia, F. Saleri, and A. Veneziani. “Solving Numerical PDEs”. *Springer* (2012).
- [272] Y. Saad. “Iterative Methods for Sparse Linear Systems”. *PWS* (1996).

# Appendices

## A1: Example application of aPC

In this section we provide an example of how arbitrary Polynomial Chaos (aPC) may be used to propagate uncertainty through a computational model. In this case, we consider a continuous advection-diffusion-reaction model of parabolic type that was used to simulate the dispersal of an invasive species over a realistic landscape. While not directly relevant to aeronautics, the numerical technique used to spatially discretise the equations governing the dispersal of the species, the Finite Element Method, is commonly used in structural modelling in aeronautics. The advection-diffusion-reaction model presented here is capable of accounting for the effects of geographical features on the dispersal and incorporating observational evidence into the model. By conducting an uncertainty study using arbitrary Polynomial Chaos (aPC) it is possible to account for the effect of the uncertainty in the observational evidence on the predictions of the model.

The spread of an invasive species in a region denoted by  $\Omega \subset \mathbb{R}^2$  can be modeled by a modified Fisher's equation for the density  $u(\mathbf{x}, t)$  of individuals in location  $\mathbf{x}$  at time  $t$  (see [267] for an introduction to diffusion in ecological problems). In the more general form the model is a nonlinear advection-diffusion-reaction equation of parabolic type:

$$\partial_t u - \operatorname{div}(\nu(\mathbf{x}, t) \nabla u) + \mathbf{b}(\mathbf{x}, t) \cdot \nabla u + \alpha(\mathbf{x}, t) u (u - \gamma(\mathbf{x}, t)) = f(\mathbf{x}, t) \quad \text{in } \Omega \times [0, T]. \quad (6.1)$$

In equation (6.1), the diffusion coefficient  $\nu(\mathbf{x}, t)$  accounts for the static landscape heterogeneities such as inhospitable regions and the presence of natural or artificial barriers within the computational domain, like major waterways and roads. The carrying capacity  $\gamma(\mathbf{x}, t)$  expresses the birth and death rates of the species at a given spatial and temporal location. Their spatial dependence allows to include peculiar landscape characteristics in terms of classification of the terrain (town, meadow, rock, wood, elevation). The convective term  $\mathbf{b}(\mathbf{x}, t) \cdot \nabla u$  allows for transport effects within the region to be modeled. In the case of terrestrial amphibious animals, the transport effects are associated with the presence of rivers, that can act both as an accelerator or as a contrasting agent, according to their flow direction with respect to the direction of the propagation front; outside of waterways the transport effects vanish. The temporal dependence of the coefficients allows for various types of seasonality: variations in the level of water and flow velocity of rivers, breeding for animals or sprouting for plants. Equation (6.1) is completed by a suitable initial value  $u^0(\mathbf{x})$  and boundary conditions on  $\partial\Omega$  [268]. In general, homogeneous Neumann boundary conditions are used to model an isolated environment, but other boundary conditions can be considered as well: an homogeneous Dirichlet boundary condition would model an hostile environment, while a Robin boundary condition would model migratory dynamics [269, 267].

## Numerical approximation

The finite element method is particularly suitable for complex geometries like the ones represented by real geographical regions. We thus discretize equation (6.1) in space by means of finite elements, while a classical finite difference discretization is carried out for the time discretization.

### Spatial discretization

For a regular triangulation (also called mesh) of the domain  $\Omega$ , a finite element space of order  $k$  consists of globally continuous functions that are locally a polynomial of degree at most  $k$  on every triangle of the mesh. Each function of a finite element space can be represented as a linear combination of a suitable finite basis. In the numerical simulations presented here we will use first order elements: although higher order polynomials can be considered [270], linear ones feature a sufficient level of accuracy for the applications being considering here. A generic element of the first order basis, denoted with  $\varphi_j(\mathbf{x})$  ( $j = 1, \dots, N$ ,  $N$  being the total number of mesh points), is the piecewise linear function equal to 1 on the  $j^{th}$  node of the mesh and equal to 0 in all the other nodes. The finite element approximation of the solution of equation (6.1) is given by

$$\mathbf{u}_h(\mathbf{x}, t) = \sum_{j=1}^N u_j(t) \varphi_j(\mathbf{x}). \quad (6.2)$$

The unknown time-dependent vector  $\mathbf{u}(t) = [u_1(t), \dots, u_N(t)]^T$  solves the  $N$ -dimensional nonlinear ordinary differential system

$$\mathbf{M} \frac{d\mathbf{u}(t)}{dt} + \mathbf{A} \mathbf{u}(t) + \mathbf{B} \mathbf{u}(t) = \mathcal{F}(\mathbf{u}(t)), \quad (6.3)$$

where  $\mathbf{M}$ ,  $\mathbf{A}$ , and  $\mathbf{B}$  are the mass, stiffness and transport matrices, whose  $(i, j)$ -th entries are

$$\mathbf{M}_{ij} = \int_{\Omega} \varphi_j \varphi_i d\mathbf{x}, \quad \mathbf{A}_{ij} = \int_{\Omega} (\nu(\mathbf{x}) \nabla \varphi_j) \nabla \varphi_i d\mathbf{x}, \quad \mathbf{B}_{ij} = \int_{\Omega} (\mathbf{b}(\mathbf{x}) \cdot \nabla \varphi_j) \varphi_i d\mathbf{x}, \quad (6.4)$$

while  $\mathcal{F}(\mathbf{u}(t)) = \int_{\Omega} (\gamma(\mathbf{x}, t) - \mathbf{u}_h(t)) \mathbf{u}_h \cdot \varphi_i d\omega$  is the discretization of the nonlinear term. All of the above integrals are computed by means of suitable quadrature rules.

### Time discretization

Let  $\Delta t$  be a time step and  $t^n = n\Delta t$  be the discretization of the time interval under study. We denote by  $\mathbf{u}^n = [\mathbf{u}_l(t^n)]_{l=1,\dots,N}$  the vector of the nodal values at time  $t^n$ . The incremental ratio  $(\mathbf{u}^{n+1} - \mathbf{u}^n)/\Delta t$  is an approximation of the time derivative either in  $t^{n+1}$  or  $t^n$ , with an associated numerical error proportional to  $\Delta t$ . As a good trade-off between numerical stability and computational efficiency, equation (6.3) is advanced in time by a mixed implicit/explicit (IMEX) approximation scheme, where the stiffness and the transport are treated implicitly while the nonlinear term is treated explicitly. Knowing the approximation  $\mathbf{u}^n$  of the solution at time step  $t^n$ , the solution at time step  $t^{n+1}$  is obtained by solving

$$(\mathbf{M} + \Delta t \mathbf{A} + \Delta t \mathbf{B})\mathbf{u}^{n+1} = \mathbf{M}\mathbf{u}^n + \Delta t \mathbf{F}(\mathbf{u}^n). \quad (6.5)$$

Equation (6.5) is a linear system of dimension  $N$ , where the right hand side is easily built since  $\mathbf{u}^n$  is known. For a more extensive discussion and application of the IMEX method see [271].

### Simulation protocol

The numerical simulations of Equations (6.1) are performed with a self-developed code in Matlab (MathWorks Inc., Natick, MA) with a uniform time step of  $\Delta t = 0.1$  months. Several techniques to efficiently solve linear systems like (6.5) are available based on iterative methods: at every time step we solve (6.5) with the conjugate gradient method, preconditioned by an incomplete Cholesky factorization (see, for instance book by Y. Saad [272]).

### Data driven mesh generation

The computational domain is discretized by a triangular mesh, to exploit its higher accuracy (compared to a typical cartesian grid for finite differences approximations) in



representing both complex boundaries and peculiar structures in the interior of the domain. For best approximation properties of the finite element method, adaptive grids are preferred: these feature smaller triangles in the areas that require higher levels of accuracy in the description, like the region of interest and the heterogeneities of the landscape. The geographical details described in the previous sections can be effectively implemented on a mesh whose generation is outlined by the following procedure:

1. Identify the region of interest in the computational domain and its main geographical features (rivers, lakes, urban areas..)
2. Generate an initial grid  $\mathcal{T}_0$ , uniformly refined over the region of interest.
3. Include elevation information from NASA SRTM data by interpolation on the grid  $\mathcal{T}_0$ .
4. Refine  $\mathcal{T}_0$  in the surroundings of the main geographical features identified in step (1).
5. Implement a gradient-driven refinement of  $\mathcal{T}_0$  where the elevation gradients are steeper (like along the sides of a valley).

To illustrate the characteristics of our method, we simulate the dispersal of an invasive species in the territory of the Basque Country in Northern Spain. The region itself is an excellent testbed, as the heterogeneity of its landscape encompasses various types of environment from the sea shores of the Bay of Biscay along the Atlantic Ocean to the mountainous region in the interior, to the higher peaks of the nearby Pyrenees range. The computational domain  $\Omega$  is the region comprised between  $4^\circ\text{W}$  and  $1^\circ\text{W}$  in Longitude, and between  $42^\circ\text{N}$  and the Bay of Biscay (or  $44^\circ 2' \text{N}$ ) in Latitude (see Figure 6.1, panel A). The domain  $\Omega$  contains the region of interest and is triangulated according to Steps (1)-(5) described in the previous Section. The mesh consists of 78,832 points and 155,478 triangular elements. Note that throughout co-ordinates will be expressed in Longitude and Latitude.

In the simulated scenario the invasive species is initially absent from the region of interest and is concentrated in a circle of radius 0.5 centred in the Aquitaine region in south-western France  $(-1.2349, 43.8631)$ . In a perspective of environmental conservation, the natural interest is in the population density and speed of propagation. Some more specific quantities of interest may be the arrival time to specific, sensitive, locations (like a natural reserve). As an example we consider the three major cities in Basque Country: Vitoria, Bilbao and San Sebastian. Their locations and the initial position of the invasive species are highlighted in Figure 6.1, panel A, that shows also the elevation of the region. Being this an illustrative case, for numerical simplicity the impact of rivers was assumed to be negligible compared to the extended geographical heterogeneities induced by the mountain ranges. We assume constant diffusion coefficient  $\nu = 5 \times 10^{-4} \text{ deg}^2 \text{ years}^{-1}$  while the carrying capacity for the species is assumed to depend solely on elevation:

$$\gamma(\mathbf{x}, t) = \left( 10 - \frac{z(\mathbf{x})}{100} \right). \quad (6.6)$$

The carrying capacity is thus maximum at sea level and is linearly decreasing with elevation. The species is assumed not to survive at elevations greater than 1000 meters above sea level. Results of the dispersal simulation are shown in Figure 6.1, panels B to F. As expected, higher densities are observed along the coastline. At the same time, the impact of elevation is evident: in particular, the barrier effect played by the Pyrénées ridge running along the 43<sup>rd</sup> parallel significantly affects the dispersal, and forces the species to move west (panel B and C) until the presence of a valley in the N-S direction allows the species to move south as well (panel D and E). Panel G in Figure 6.1 shows the arrival time ( $t_{arrive}$ ) of the species in the whole computational domain:  $t_{arrive}$  is defined in any point  $\mathbf{x}$  of the domain  $\Omega$  as the time at which the population density reaches 1 i.e.  $u(\mathbf{x}, t_{arrive}) = 1$ . Black areas are inhospitable for the species due to their elevation and will not be invaded. Finally, panel H in Figure 6.1 shows the temporal evolution of the

population density in the three cities of interest. Naturally, San Sebastian is the first city reached by the species. Nevertheless, Bilbao is reached before Vitoria, although the second one is closer to the initial distribution of the invasive species. Finally, the population density in Vitoria is significantly lower than in Bilbao and San Sebastian due to Vitoria's higher elevation (as it can be inferred from Figure 6.1, panel A).

## Uncertainty analysis for the Basque country test case

An uncertainty analysis was performed for the Basque country test case. As an illustrative example we explored uncertainty in both diffusion (affecting propagation, thus the speed of the invasion) and carrying capacity (playing a key role in settling of the species in a given location). To this end, we considered a modified version of the carrying capacity, scaled by a random parameter  $\alpha$ :

$$\gamma(\mathbf{x}, t) = \alpha \left( 10 - \frac{z(\mathbf{x})}{100} \right). \quad (6.7)$$

The uncertain parameters are thus the diffusion coefficient,  $\nu$ , and the scaling factor  $\alpha$ . Histograms for the two uncertain parameters were generated synthetically and are shown in panel A of Figure 6.2. The histogram for  $\nu$  was generated from a normal distribution  $N(5 \times 10^{-4}, 1.2 \times 10^{-4})$ , while  $\alpha$  was sampled from the uniform distribution  $U(0, 2)$ . Having generated synthetic histograms for  $\nu$  and  $\alpha$ , data driven aPC was used to propagate the uncertainties in these parameters through the invasive species model. Panel A in Figure 6.2 shows also the locations of the 1D collocation points for the two input distributions. Having found the collocation points and weights from the data, Smolyak's rule is applied to generate a level 1 grid. In this case the resulting grid consists of 5 points, shown in panel B of Figure 6.2. The invasive species model was evaluated at each of the points and based on the results, PDFs of  $\mathbf{u}(\mathbf{x}, t)$  were calculated, at each time step, in the whole

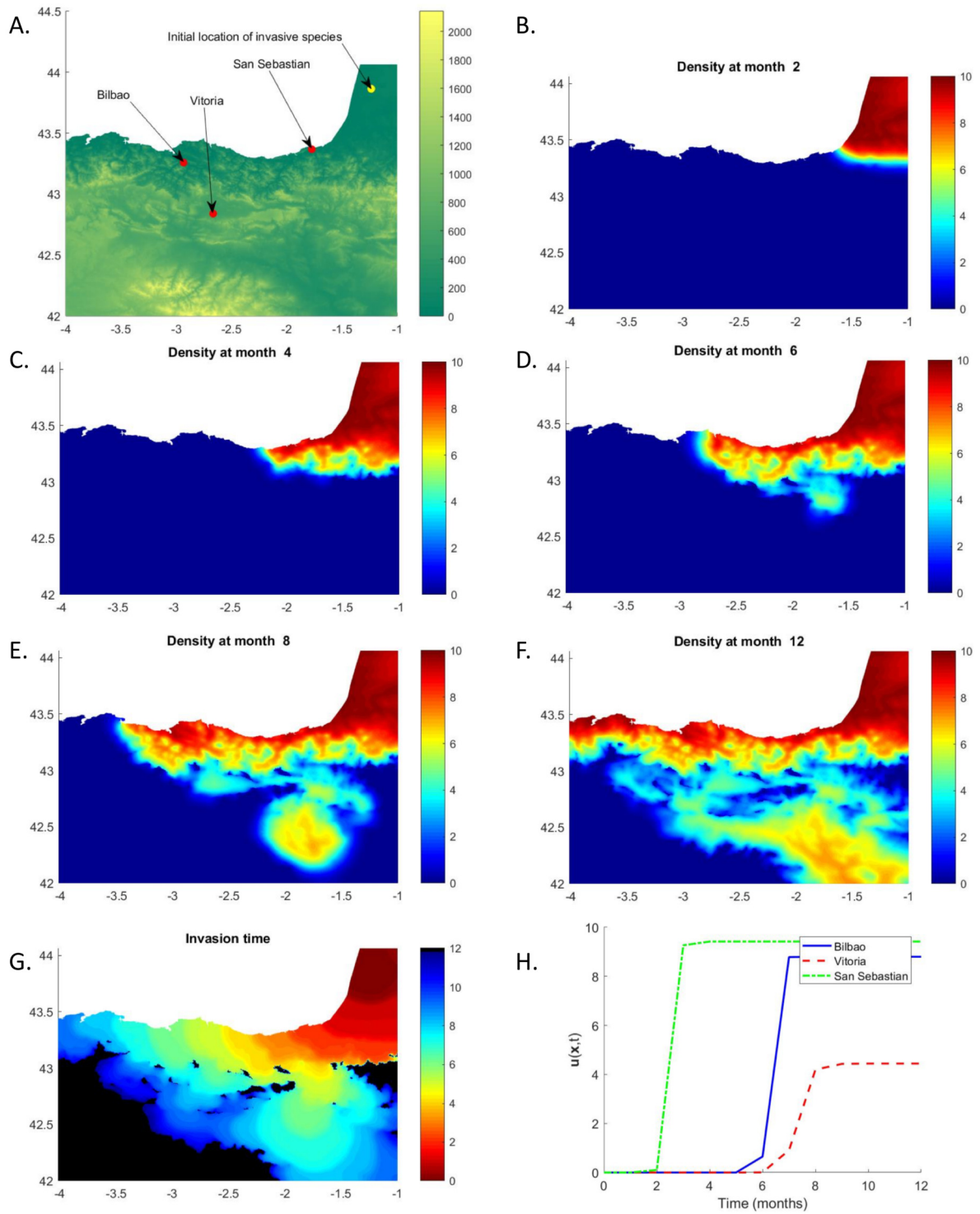


Figure 6.1: Basque Country test case. **Panel A.** Computational domain with elevation, initial location of the invasive species and location of the three cities under study. **Panel B to F.** Density of the invasive species at different times of the invasion process. The temporal dynamics highlights how extended heterogeneities due to the mountain ranges favors dispersal to the West. **Panel G.** Invasion time in the whole computational domain. Black areas are inhospitable for the species and will not be invaded. **Panel H.** Temporal dynamics of the population density  $u(\mathbf{x}, t)$  for Bilbao (solid blue), Vitoria (dashed red) and San Sebastian (dot-dashed green).

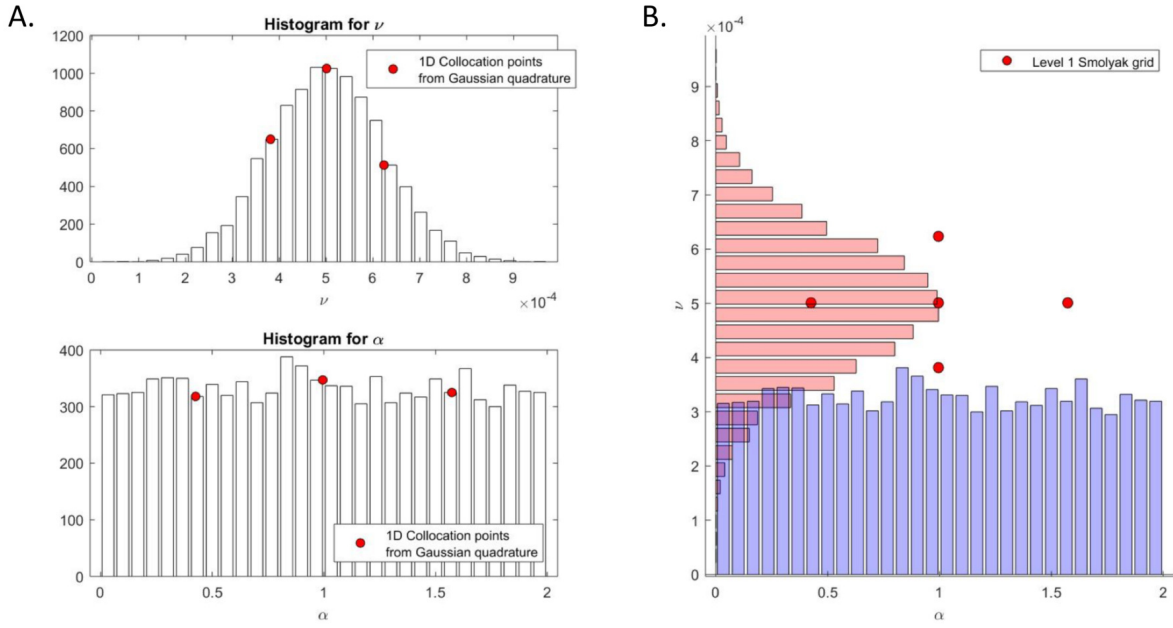


Figure 6.2: **Panel A.** Histograms for the uncertain parameters  $\nu$  and  $\alpha$ , and the 1D collocation points for the input distributions. **Panel B.** The sampling grid calculated through application of Smolyak's rule at level 1.

computational domain. Figure 6.3 collects the results. Panel A to D show the mean ( $\mu$ , A and C) and the standard deviation ( $\sigma$ , B and D) of the PDFs on the whole region at different time steps. The bottom row of Figure 6.3 show the results for Bilbao, Vitoria and San Sebastian. Panel E represent uncertainty in the arrival times of the species in the three cities, while panel F shows the temporal dynamics of the standard deviation of the population density. Panels B, D, and F highlight how uncertainty is mainly concentrated in the vicinity of the propagation front and drops significantly in its wake.

Predictive modeling of the spatio-temporal spread of an invasive species is a very useful tool to help environmental conservation policymakers in their decision-taking process for devising appropriate countermeasures. As model parameters for invasive species are often inferred from field measurements, they are imbued with uncertainty. If the species is poorly known, the uncertainty may be so significant to hinder the reliability of the model predictions. We presented a continuous reaction-diffusion model for the population density of an invasive species, coupled with an arbitrary Polynomial Chaos (aPC) method

to assess how the uncertainty of poorly known parameters affects the predictions of the deterministic model. The coupled model has several advantages. The arbitrary Polynomial Chaos is far less computational expensive with respect to a classical Monte Carlo sampling method and, more important, is able to handle scarce data. The deterministic model is equipped with a data-driven mesh generator which is capable of both accurately model the region of interest and take into account landscape heterogeneities in a natural way. Moreover, the individual runs of the deterministic model needed to set up the aPC procedure are easily computed on a laptop in few minutes. We illustrated the practical applicability of the proposed method by quantifying the uncertainty in the spread of a generic invasive population in the Basque Country area in northern Spain.

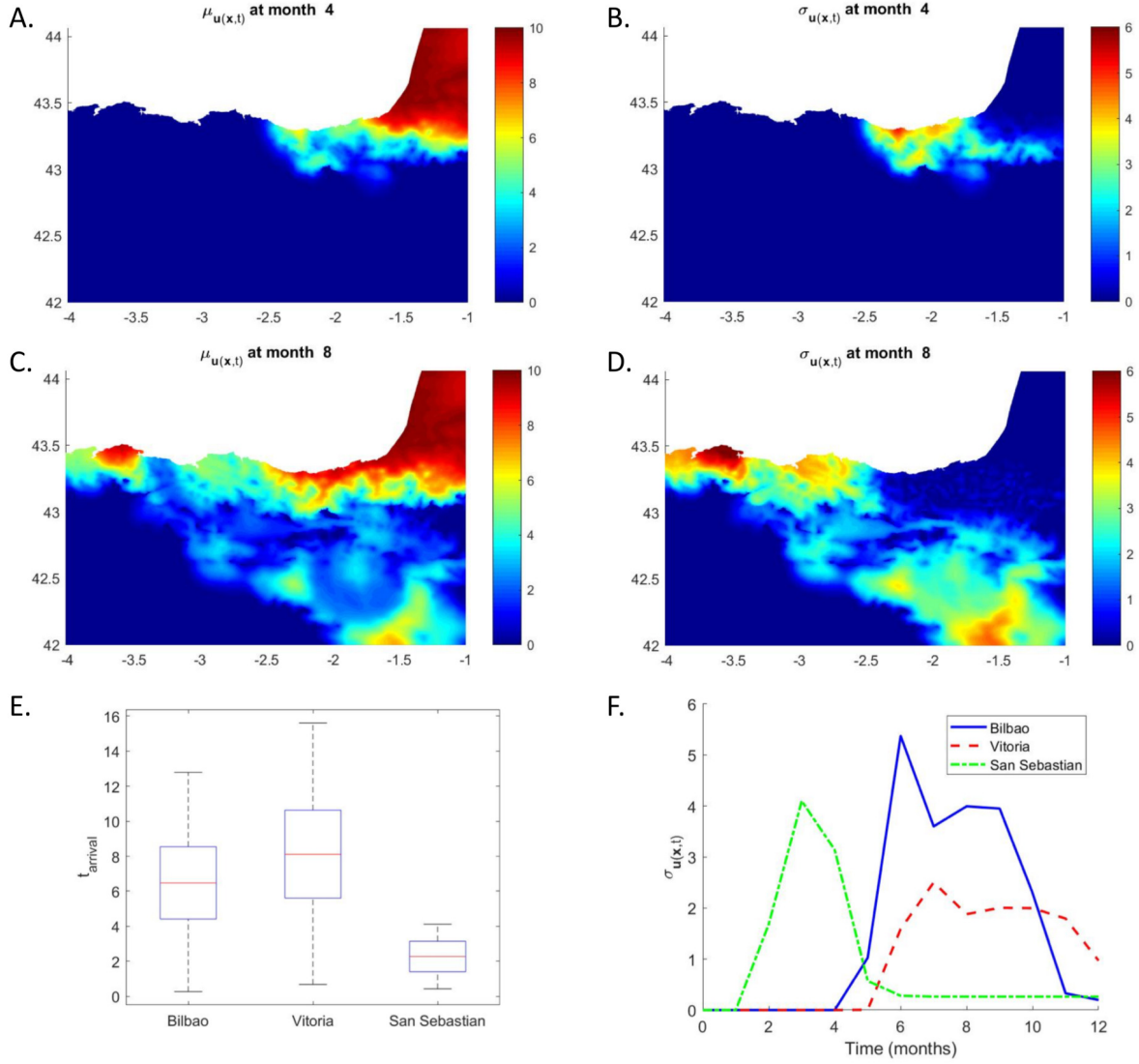


Figure 6.3: Uncertainty analysis for the Basque country test case. **Panel A to D.** Temporal dynamics of mean and standard deviation of the population density in the whole computational domain highlights the higher level of uncertainty in the surrounding of the wavefront. **Panel E.** Uncertainty in arrival time in Bilbao, Vitoria and San Sebastian. **Panel F.** Temporal dynamics of the standard deviation for the population density in Bilbao (solid blue), Vitoria (dashed red) and San Sebastian (dot-dashed green) shows how uncertainty drops significantly in the wake of the propagation wavefront.

## A2: Training KBaNNs through back-propagation

The KBaNN is trained through error back-propagation, with the network parameters determined through a gradient based optimisation algorithm. This required analytic expressions for the derivatives of the KBaNN prediction error,  $\epsilon$ , with respect to the network parameters. Expressions for these derivatives in a KBaNN architecture were first derived in Wang et al (1997) [226]. However, the architecture presented here differs in that the correction applied is additive rather than multiplicative. A further difference is that the knowledge layer presented in the original paper have been replaced with a coarse model which is not necessarily analytic. As a consequence we do not adjust the parameters of the coarse model through back-propagation. In our KBaNN formulation there is only a single path for error back-propagation through the KBaNN which simplifies the formulae for the derivatives. Finally, the term added to the error term to penalise KBaNN predictions that deviate from the coarse model output mean that there is an additional term to the error derivatives. The error function derivative for the output layer may be derived from (4.18):

$$\frac{\partial \epsilon}{\partial \mathbf{y}_j} = \mathbf{y}_j - F_{e_j}(\mathbf{x}), \quad j = 1, 2, \dots, n_y. \quad (6.8)$$

Derivatives of the error function for the connections between the output neurons, the coarse model, and the normalised region neurons may then be derived from (4.17):

$$\frac{\partial \epsilon}{\partial \beta_j} = (\mathbf{y}_j - F_{e_j}(\mathbf{x}))F_{c_j}(\mathbf{x}) + 2\delta(\beta_j - 1), \quad j = 1, 2, \dots, n_y, \quad (6.9)$$

$$\frac{\partial \epsilon}{\partial \beta_{0_j}} = (\mathbf{y}_j - F_{e_j}(\mathbf{x})) + 2\delta\beta_{0_j}, \quad j = 1, 2, \dots, n_y, \quad (6.10)$$

$$\frac{\partial \epsilon}{\partial \rho_{ij}} = (\mathbf{y}_j - F_{e_j}(\mathbf{x}))\mathbf{r}'_i + 2\delta\rho_{ij}, \quad i = 1, 2, \dots, n_r, \quad j = 1, 2, \dots, n_y. \quad (6.11)$$

The chain rule may then be employed to evaluate the gradients in the normalised region layer, the region layer, and the boundary layer respectively:



$$\mathbf{g}_{\mathbf{r}'_i} = \sum_{j=1}^{n_y} \frac{\partial \epsilon}{\partial \mathbf{y}_j} \frac{\partial \mathbf{y}_j}{\partial \mathbf{r}'_i} = \sum_{j=1}^{n_y} (\mathbf{y}_j - F_{e_j}(\mathbf{x})) \rho_{ij}, \quad i = 1, 2, \dots, n_r, \quad (6.12)$$

$$\mathbf{g}_{\mathbf{r}_i} = \sum_{j=1}^{n_r} \frac{\partial \epsilon}{\partial \mathbf{r}'_j} \frac{\partial \mathbf{r}'_j}{\partial \mathbf{r}_i} = \mathbf{g}_{\mathbf{r}'_i} \frac{1}{\sum_{k=1}^{n_r} \mathbf{r}_k} - \frac{\sum_{j=1}^{n_r} \mathbf{g}_{\mathbf{r}'_j} \mathbf{r}_j}{(\sum_{k=1}^{n_r} \mathbf{r}_k)^2}, \quad i = 1, 2, \dots, n_r, \quad (6.13)$$

$$\mathbf{g}_{\mathbf{b}_i} = \sum_{j=1}^{n_r} \frac{\partial \epsilon}{\partial \mathbf{r}_j} \frac{\partial \mathbf{r}_j}{\partial \mathbf{b}_i} = \sum_{j=1}^{n_r} \mathbf{g}_{\mathbf{r}_j} \mathbf{r}_j (1 - \sigma(\alpha_{ji} \mathbf{b}_i + \theta_{ji})) \alpha_{ji}, \quad i = 1, 2, \dots, n_b, \quad j = 1, 2, \dots, n_r. \quad (6.14)$$

(4.16) may then be used, together with these derivatives to find the error function gradient for the region layer parameters:

$$\frac{\partial \epsilon}{\partial \alpha_{ij}} = \frac{\partial \epsilon}{\partial \mathbf{r}_i} \frac{\partial \mathbf{r}_i}{\partial \alpha_{ij}} = \mathbf{g}_{\mathbf{r}_i} \mathbf{r}_i (1 - \sigma(\alpha_{ij} \mathbf{b}_j + \theta_{ij})) \mathbf{b}_j + 2\delta \alpha_{ij}, \quad i = 1, 2, \dots, n_r, \quad j = 1, 2, \dots, n_b, \quad (6.15)$$

$$\frac{\partial \epsilon}{\partial \theta_{ij}} = \frac{\partial \epsilon}{\partial \mathbf{r}_i} \frac{\partial \mathbf{r}_i}{\partial \theta_{ij}} = \mathbf{g}_{\mathbf{r}_i} \mathbf{r}_i (1 - \sigma(\alpha_{ij} \mathbf{b}_j + \theta_{ij})) + 2\delta \theta_{ij}, \quad i = 1, 2, \dots, n_r, \quad j = 1, 2, \dots, n_b. \quad (6.16)$$

Finally, the derivatives for the parameters in the boundary layer may be calculated using (4.15):

$$\frac{\partial \epsilon}{\partial \mathbf{v}_i^{(j)}} = \mathbf{g}_{\mathbf{b}_j} \mathbf{x}_i + 2\delta \mathbf{v}_i^{(j)}, \quad i = 1, 2, \dots, n_x, \quad j = 1, 2, \dots, n_b. \quad (6.17)$$

Adagrad is a gradient based optimisation algorithm used to update the KBaNN parameters with a variable learning rate. The set of KBaNN parameters,  $\Phi$ , are initialised with random values and the training set iterated through. At each iteration the KBaNN parameters are adjusted according to:

$$\Phi_{i,t+1} = \Phi_{i,t} - \frac{\eta}{\sqrt{G_{t_{ii}} + \tau}} \frac{\partial \epsilon}{\partial \Phi_{i,t}}, \quad (6.18)$$

where  $G_t$  is a diagonal matrix, in which the element  $G_{t_{ii}}$  contains the sums of the squares of the past error function gradients for the  $i^{\text{th}}$  KBaNN parameter.  $\eta$  represents the learning rate and the constant  $\tau$  is a smoothing constant used to ensure numerical stability when the gradients are small. After each complete pass through the training data, the data is shuffled to prevent the KBaNN learning spurious patterns.