Uncertainty Propagation Through Large Nonlinear Models

William Becker

A thesis submitted to the University of Sheffield for the degree of Doctor of Philosophy in the faculty of Engineering.

Department of Mechanical Engineering

January 2011

University of Sheffield

Abstract

Uncertainty analysis in computer models has seen a rise in interest in recent years as a result of the increased complexity of (and dependence on) computer models in the design process. A major problem however, is that the computational cost of propagating uncertainty through large nonlinear models can be prohibitive using conventional methods (such as Monte Carlo methods). A powerful solution to this problem is to use an emulator, which is a mathematical representation of the model built from a small set of model runs at specified points in input space. Such emulators are massively cheaper to run and can be used to mimic the "true" model, with the result that uncertainty analysis and sensitivity analysis can be performed for a greatly reduced computational cost. The work here investigates the use of an emulator known as a Gaussian process (GP), which is an advanced probabilistic form of regression, hitherto relatively unknown in engineering. The GP is used to perform uncertainty and sensitivity analysis on nonlinear finite element models of a human heart valve and a novel airship design. Aside from results specific to these models, it is evident that a limitation of the GP is that non-smooth model responses cannot be accurately represented. Consequently, an extension to the GP is investigated, which uses a classification and regression tree to partition the input space, such that non-smooth responses, including bifurcations, can be modelled at boundaries. This new emulator is applied to a simple nonlinear problem, then a bifurcating finite element model. The method is found to be successful, as well as actually reducing computational cost, although it is noted that bifurcations that are not axis-aligned cannot realistically be dealt with.

Contents

Acknowledgements

This thesis is the product of some great ideas from some very clever people, and some hard work on my part, particularly inspired by very tight deadlines towards the end. First of all, I would like to thank Keith Worden, who provided the ideas for the vast majority of the work presented here, took the patience to explain them to me when necessary (quite often, in fact!) and was always kind enough to take me and other students for curries, meals and pub visits in Sheffield and beyond. It truly has been a privilege to work with such a hugely intelligent but extremely friendly and approachable man such as Keith, who will always do his utmost to help out ailing PhD students. I will very much miss working with him when I move away from the group. I also express my thanks to Graeme Manson, who as my other supervisor was always available for help when needed.

On the academic front, thanks must go to Jeremy Oakley, who was one of the main creators of much of the theory in Chapter 3, also for his patience in attempting to explain Bayesian statistics to a newly-graduated student with an A-level statistics education. I also gratefully received much help from James Hensman concerning Gaussian processes, although many times it went over my head! For the latter stages of my PhD I would like to thank the members of "Bayes Club", which also included Rob Barthorpe and Lizzy Cross. These discussions helped me enormously with the latter chapters of this thesis and were also very enjoyable.

For help on the FE side of things, thanks go to Alaster Yoxall for his advice on heart valves and modelling, and Jen Rowson for help with LS-Dyna. I must also thank James Kennedy from KBS2 Inc. for his extremely generous assistance via the LS-Dyna forum on numerous aspects of the use of LS-Dyna, which, incidentally, is woefully under-documented.

A special thanks goes to Jem Rongong, who firstly helped with funding in the latter stages of the PhD, and secondly was very understanding about my time restrictions in the final two months.

Thanks must inevitably go to all my friends in the department who have helped me over the years with endless tips and assistance with various software packages and matters relating to general PhD life. But much more valuably, they have made working in this research group a wonderful experience. I can only hope that my next working environment will be comparable. Specific thanks must go to the members of Lunch Club: the leaders

Vaggelis (Blue) and Giovanni (Pink) for being my best friends over the past three years, plus the more peripheral members Pete, Rob and Dorit for plenty of good trips to the pub. Lunch Club alumni must also be mentioned — Frank Stolze provided endless help and jokes in the office and is very much missed, as well as *gli Italiani.* All these people have made working in the DRG a great experience. Thanks to all the people in my office for putting up with my grumpiness in the last months of thesis-writing!

I must also thank the people who I met in Torino. Cecilia Surace and David Storer for their warm welcome and endless help, including several excellent ski outings. All the Italians for inviting me for excellent dinners of vast proportions, particularly Marco Gherlone and Maria De Stefano. I must also thank Melegcim for being such an excellent friend throughout the last few months of the visit and beyond.

Thanks also go to all the people who were not directly involved in the thesis, but have been great friends and family throughout, which inevitably helps. My Mum and Dad have always been there for me and I love them dearly. My brother Alex and Dave (members of Magiklegz) are my oldest and best friends and I can count on them for anything. I should particularly thank Valeria Amenta, who was probably the only thing stopping me going crazy at a few points in the last months. Grazie mille Tortigliona!

I would like to thank Ove Arup for the use of LS-Dyna. The activity presented in Chapters 6 and 7 was part of the project entitled "Innovative Solutions for Control Systems, Electric Plant, Materials and Technologies for a Non-Conventional Remotely-Piloted Aircraft" funded by the Piedmont Region of Italy. A large part of the work here was funded by a Platform Grant for uncertainty management in nonlinear systems. Funding also came from the "Bridging the Gaps" initiative.

—p.s. Although I can't exactly thank my dog for any academic input, I would like to dedicate this thesis to Bess, who is sorely missed.

Acronyms

ALE Arbitrary-Lagrangian-Eulerian

ANN Artificial Neural Network

APDL Ansys Parametric Design Language

BBA Basic Belief Assignment

CART Classification And Regression Tree

cdf Cumulative distribution function

COV Coefficient Of Variation

CV Cross-Validation

DOE Design Of Experiments

DST Dempster-Shafer Theory of evidence

EOS Equation Of State

ETF The *Elettra Twin Flyers* (The airship design from Chapter 3

FE Finite Element

FSI Fluid-Structure Interaction

FRF Frequency Response Function

GP Gaussian Process

iid Independent and Identically Distributed

KL Karhunen-Loève (expansion)

LHS Latin Hypercube Sampling

LLM Limiting Linear Model

LTA Lighter Than Air

MAP Maximum *A Posteriori*

MCMC Markov Chain Monte Carlo

MEI Main Effect Index

MH Metropolis-Hastings

MLE Maximum Likelihood Estimation

OAT One-At-a-Time

PCE Polynomial Chaos Expansion

pdf Probability density function

RJ-MCMC Reversible-Jump Markov Chain Monte Carlo

RMS Root-Mean-Squared

SA Sensitivity Analysis

SFEM Stochastic Finite Element Method

SSFEM Spectral Stochastic Finite Element Method

ST Soft Tissue (material model)

TSI Total Sensitivity Index

UA Uncertainty Analysis

Nomenclature

$\boldsymbol{\mu}$	Vector of means of a multivariate probability distribution
$\boldsymbol{\phi}(\boldsymbol{x})$	Vector of basis function(s) of $\boldsymbol{x}$
$\boldsymbol{\phi}_r$	An integral occurring in the inference of sensitivity analysis quantities from the GP
$\boldsymbol{\Theta}$	The vector of the full set of model parameters
$\boldsymbol{X}$	Matrix of inputs of training data
ϵ	True strain
η	A leaf of a classification and regression tree
$\hat{\sigma}^2$	Best estimate of σ^2
λ	Deviatoric stretch (equal to $1 + \epsilon$)
λ^*	Stretch at which collagen fibres are fully straightened in ST model
$\hat{\mathbf{w}}$	Mean of posterior distribution of $\hat{\mathbf{w}}$
$\mathbf{e}$	A constant in the posterior mean function of the GP
$\mathbf{F}$	The deformation gradient tensor
$\mathbf{t}(\boldsymbol{x})$	Vector of the covariance of $\boldsymbol{x}$ with all training data
$\mathbf{t}_r$	An integral occurring in the inference of sensitivity analysis quantities from the GP
$\mathbf{w}$	Vector of basis function coefficients or weights
$\mathbf{y}$	Vector of model output values of training data
$\boldsymbol{x}$	Vector input of a model or function
$\boldsymbol{x}'$	An input point in the vicinity of $\boldsymbol{x}$
$\boldsymbol{x}^*$	Input of unknown function value
$\boldsymbol{y}$	Vector output of a model or function
$\mathcal{D}$	Training data
$\mathcal{GP}$	Gaussian process (distribution notation)

$\mathcal{IG}$	The inverse-gamma distribution
$\mathcal{N}$	The normal distribution
$\mathcal{U}$	The uniform distribution
$\mathcal{W}$	The Wishart distribution
$\mathcal{X}$	The sample space, or support, of a (possibly multi-dimensional) random variable
$\mathcal{X}_r$	The support of the joint distribution of the variables $\boldsymbol{x}_r$ where $r \subseteq d$
μ	Membership function of a fuzzy set or angle of incident gust on the airship model
Φ	Matrix of basis functions for each sample point
ϕ	Angle of leaflet coaptation
ρ	Density
Σ_w	Covariance matrix of posterior distribution of $\mathbf{w}$
Σ	Covariance matrix of a multivariate probability distribution
σ^2	Scaling factor of the GP covariance function
Θ	The universal set, or frame of discernment
$\tilde{I}_1, \tilde{I}_2$	First and second deviatoric invariants of the right Cauchy deformation tensor
$\tilde{A}$	A fuzzy set
$\varnothing$	The empty set
A	An event, or the covariance matrix of a GP
B	An event, or the matrix of roughness coefficients in the GP covariance function
b_i	The (i,i)th element of B, corresponding to roughness with respect to the input i
$c(\cdot,\cdot)$	Covariance function of a GP
$C1S$	Neo-Hookean coefficient of sinus/aorta material model
$C5L$	Post-transition modulus of leaflet material model
$c^*(\cdot,\cdot)$	Predictive covariance function of GP (before marginalisation)
$c^{**}(\cdot,\cdot)$	The posterior covariance function of the GP (after hyperparameter marginalisation)
$C_1, C_2, \ldots$	Coefficients of the Mooney-Rivlin material model and ST model
d	Number of model inputs or function dimensions
E	The expected value
E^*	The posterior expected value

El	Elastic modulus of valve leaflet
Es	Elastic modulus of sinus/aorta
$Fltrad$	Leaflet separation in relaxed state (ratio)
g	Monotone measure
i	An index referring to a particular input variable
J	Volume ratio (equal to det(**F**))
j	An index referring to the training data points
K	Effective bulk modulus
k	Number of divisions in a factorial experiment
$m^*(\boldsymbol{x})$	Predictive mean function of GP (before marginalisation)
$m^{**}(\cdot)$	The posterior mean function of the GP (after hyperparameter marginalisation)
n	Number of training data points
n_v	Number of training data in the region r_v
$P(x)$	Cumulative probability distribution of x
$p(x)$	Probability of x or probability density function of x
P_r	An integral occurring in the inference of sensitivity analysis quantities from the GP
$PDIF$	Pressure difference scaling factor
q	Number of basis functions of $\boldsymbol{x}$ (equal to $n+1$ in the examples used)
Q_r	An integral occurring in the inference of sensitivity analysis quantities from the GP
R	Number of regions of input space for treed GPs and importance sampling
r	A subset of d, i.e. a subset of model inputs or input dimensions
r_v	The vth leaf (terminal node) of a classification and regression tree
Rb	Radius of valve base
s	Split value at a node in a regression tree
S_i	Sensitivity index of scaled main effect index of an input x_i
S_r	An integral occurring in the inference of sensitivity analysis quantities from the GP
S_{Ti}	Scaled total effect index of x_i
T	A classification and regression tree
Tc	Thickness of commissure region
Tl	Thickness of valve leaflet

Ts	Thickness of sinus/aorta
U_r	An integral occurring in the inference of sensitivity analysis quantities from the GP
v	Index of regions in sample space
V_i	Unscaled main effect index of x_i
V_{Ti}	Unscaled total effect index of x_i
W	Strain energy
X	Random variable
x	Scalar input of a model or function
$x^{\sharp}$	A column vector comprising of x_r and x'_{-r}
Y	Random variable of (scalar) model output
y	Scalar output of a model or function

Chapter 1

Introduction

> True wisdom consists in knowing that you know nothing.
>
> *(Bill S. Preston Esq., paraphrasing Socrates - see above)*

Perhaps one of the most profound changes to engineering in the 20th century was caused by the advent of computers. Computers have enabled calculations to be performed of a complexity and scale that would have been inconceivable in previous centuries. Furthermore, as processing power has increased exponentially, so has the ability to perform increasingly complicated calculations. Finite element (FE) models are a case in point. A very complicated structure may be divided into much smaller and simpler elements, each of which is governed by well-known equations describing displacement and stress as a result of applied forces and boundary conditions. As the availability of computing power has dramatically increased, so the complexity and scale of FE models has followed, to the point where models with millions of elements are no longer seen as exceptional. This trend is reflected in every type of computer model used in engineering today.

Since computer models offer enormous insight into the workings of complicated systems, for a fraction of the cost of prototypes and tests, they are very attractive to engineering

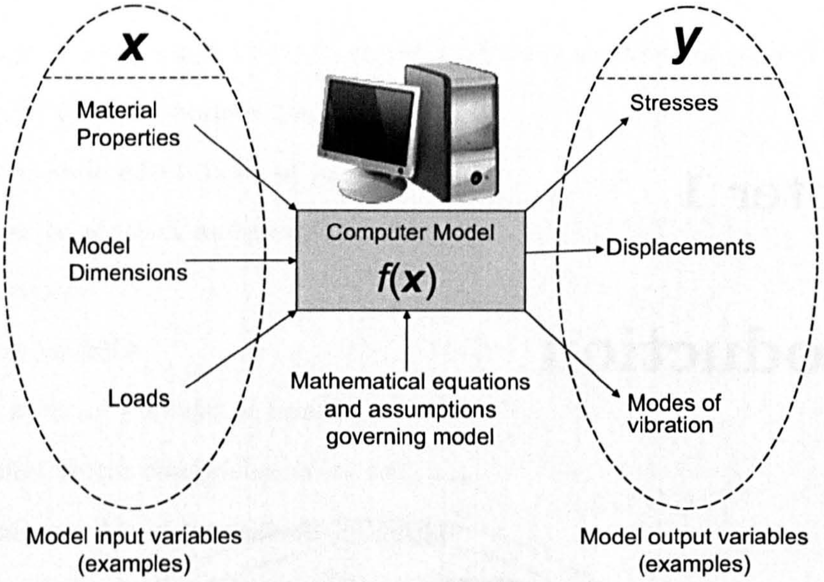

x
Material Properties
Model Dimensions
Loads
Computer Model
f(x)
Mathematical equations and assumptions governing model
y
Stresses
Displacements
Modes of vibration
Model input variables (examples)
Model output variables (examples)

Figure 1.1: A "black box" view of a computer model.

industries. Large engineering projects now rely heavily on computer models, which have become an integral part of the design process — examples of such "virtual prototyping" can be found in all major engineering companies.

1.0.1 A Black Box View

All computer models can be considered as systems that take a number of input parameters $\boldsymbol{x} = \{x_i\}_{i=1}^{d}$ (such as material properties, loads, boundary conditions etc. for an FE model), and return a number of outputs $\boldsymbol{y}$ (displacements, stresses, velocities etc.), i.e.

$$\boldsymbol{y} = f(\boldsymbol{x}) \tag{1.1}$$

Figure 1.1 illustrates this "black box" view. The mapping of the inputs to the outputs is controlled by a number of mathematical relationships, which can be considered as a function $f(\boldsymbol{x})$. There is little doubt that with the increase of processing power it has been possible to greatly increase the sophistication of the equations that dictate the model outputs given some set of inputs — *multiphysics* simulations can now closely simulate interactions between solids and fluids, with magnetic and thermal considerations if required. However, the quality of the model output is also highly dependent on the quality of the input values themselves, all the more so since more sophisticated models typically require more information to be defined than simpler ones. Consider for example the material definition in an FE model: only a few years ago many materials were approximated as

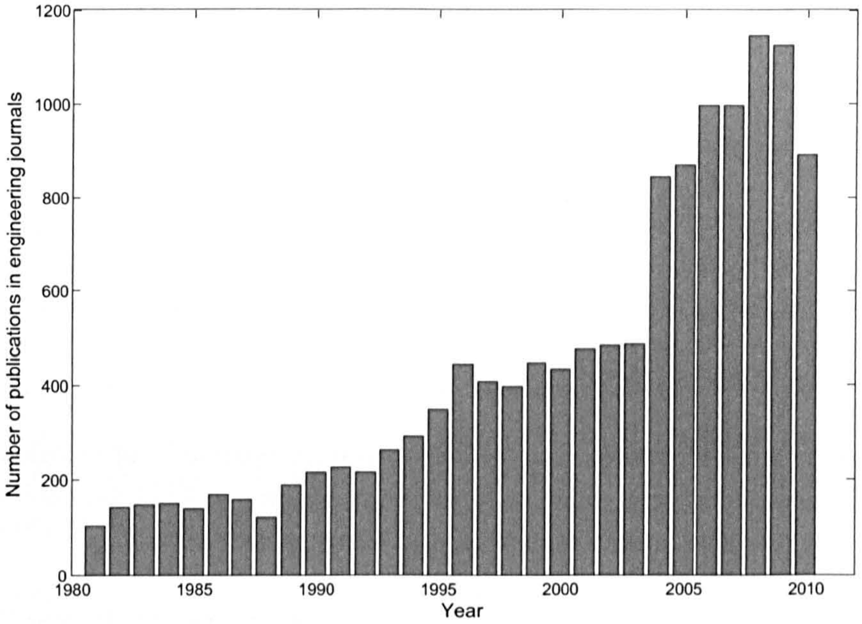


Figure 1.2: Number of publications per year with "uncertainty" in the title (source: Scopus [1]).

being linear due to restrictions in processing power, which requires the definition of only one or two parameters. Modern material models describing hyperelastic and viscoelastic materials now require a great number of parameters to fully specify the material alone.

1.1 Uncertainty in Computer Models

Given the increased sophistication and reliance on computer models, it is becoming increasingly apparent that *uncertainties* in the model inputs create uncertainty in the outputs and results of the model that cannot be reasonably discounted. This has given rise to the discipline known as *Uncertainty Analysis* (UA), which has seen a considerable surge of interest in recent years. UA seeks ultimately to quantify the uncertainty in the output of a model, given the uncertainty in the model inputs, which allows much more informed decisions to be taken based on model results. Figure 1.2 shows a histogram of publications with the word "uncertainty" in the title in engineering journals from 1980 to the present day. This illustrates (somewhat crudely, given the concurrent increase in publications in general) the rise in interest over recent years, reflecting the increasing awareness of the problem as industry relies more heavily on computer models.

As Bill S. Preston Esq. asserts at the beginning of this chapter, "true wisdom consists in knowing that you know nothing". As engineers, we do not have the luxury of such extreme assertions, but we are responsible for fully accounting for what we do and don't know.

This can be a much more difficult task than might first be thought — see Chapter 2. The difficulty of describing “knowns” and “unknowns” was famously described by Donald Rumsfeld:

> Reports that say that something hasn’t happened are always interesting to me, because as we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns – the ones we don’t know we don’t know.
>
> *(Donald Rumsfeld, 2003)*

Despite earning almost universal ridicule, this statement is actually quite logical and begins to make distinctions between different types of uncertainty. A brief discussion on types of uncertainty follows here, which is continued in Chapter 2.

1.2 Sources of Uncertainty

Uncertainties can occur for a number of reasons. Broadly speaking, they can be divided into two categories: *epistemic* uncertainties, and *aleatoric* uncertainties. Aleatoric uncertainties are those which arise from natural variability — for example, dimensional uncertainties due to machining tolerances, or inherent geometric variability between similar components. Biomechanical models are an excellent example of aleatoric uncertainty: consider a model of loading on a human bone — the dimensions and material properties vary from person to person, so it is essential to consider this variability if the model is to be generally applicable, rather than applicable to a specific individual.

Epistemic uncertainties, on the other hand, are due to the difference between a model’s theoretical approximation and reality; an example of this might be approximating a viscoelastic material by a simple time-independent model. These can be intentional, to reduce computational expense, or due to lack of knowledge. Some uncertainty can be considered as a mixture from both epistemic and aleatoric sources: it might be known that the material properties of a component vary from one component to the next (aleatory), but a small sample size may not permit the distribution of this parameter to be known with great accuracy (epistemic uncertainty). Generally aleatoric uncertainties cannot be reduced, whereas epistemic uncertainties can, by further research to improve the correlation of the simulation to reality.

1.3 Uncertainty Analysis

Uncertainty analysis is the discipline of accounting for uncertainty in systems, which often occurs in two steps. First of all, the uncertainty in model inputs must be *quantified.* Second, the uncertainty is *propagated* through the model, resulting in a quantification of output uncertainty. Sometimes it is necessary to combine information about uncertainty from different sources and frameworks; this is known as the *fusion* problem. Regarding quantification, the most popular framework is probability theory, since it is well-understood and easily interpreted by anyone with a basic mathematical background. Fusion is also an area of research that is still largely confined to mathematical research groups. As such, the majority of interest from an engineering perspective has been generated by the problem of propagating uncertainty. A much more extensive discussion of these aspects of UA is given in Chapter 2.

1.3.1 Computational Expense

Perhaps the main sticking-point of uncertainty propagation is however that it can be enormously time-consuming from a computational perspective. A large FE simulation may take hours or even days to run for a single set of input parameters. With the consideration of uncertainty, the input $\boldsymbol{x}$ can now be considered as a point in a d-dimensional input hyperspace which is bounded by the upper and lower limits of each uncertain parameter. The propagation of uncertainty involves exploring this input space in order to find how the value of y varies with variations of $\boldsymbol{x}$. Clearly, the computational cost of doing this rises dramatically with the number of uncertain inputs, to the point where it may be unfeasible without enormous amounts of processing power, or even completely impossible. A good example of this is a fairly recent calculation performed at Los Alamos National Laboratories to propagate uncertainty through a nonlinear FE model of a weapon component under blast loading. The analysis took over 72 hours on a 3968-processor cluster, using nearly 4000 Abaqus/Explicit licences [2]. Obviously this amount of processing power is not available to many, and even when it is available it may not suffice.

1.3.2 Solutions

In order to alleviate this problem, many approaches to propagating uncertainty for reduced computational cost have been proposed. A large class of these are based on the idea of using a small number of model runs at points in the input space to build an *emulator* of the model that imitates the function $f(\boldsymbol{x})$, but at a greatly reduced computational cost. The emulator is then used to explore the input space and propagate uncertainty. This approach has been shown to work well, but is only effective if the emulator accurately reproduces the input/output relationship of the model over the full range of input space. The problem of uncertainty propagation is therefore very closely associated with the disciplines of *data modelling* and *machine learning.* In particular, nonlinear models generate *response surfaces* (the hypersurface of the model output as a result of varying the inputs) that are difficult or impossible to characterise with "conventional" regression models. In the extreme case, the response surface may even bifurcate. There is therefore a necessity to develop ways of performing UA with emulators that can emulate as wide a class of computer models as possible. This is the main motivation of this thesis.

1.3.3 Sensitivity Analysis

A furtherance of UA, known as *Sensitivity Analysis* (SA), is the concept of finding how sensitive the model output is to each input or set of inputs. SA is very closely tied to UA because it forms part of the prognosis resulting from an UA: often the majority of output uncertainty is caused by only a small set of input parameters, so it is of great interest to focus efforts to reduce input uncertainty on these parameters. SA will therefore feature extensively in the following chapters.

1.4 Objectives of this Thesis

The work in this thesis can be considered from two viewpoints. From the point of view of uncertainty analysis, it aims to outline in detail and investigate the use of new emulator-based approaches that are novel to the field of engineering research. The focus here is on the ability to propagate uncertainty through large nonlinear computer models and perform sensitivity analysis for a reasonable computational cost. The work aims to apply these methods to real FE models in order to test and demonstrate their use on real engineering problems, since these nascent approaches are generally developed in the machine learning and statistics communities, and have been applied very little outside of trivial problems. This first objective may therefore be summarised as,

1. To investigate and apply new techniques for propagating uncertainty through large nonlinear engineering models at a reduced computational expense.

From another point of view, the models developed in the work here are themselves novel, particularly given their consideration of uncertainty. Therefore the application of the emulator techniques is not merely a case study or trivial problem, but offers new insight into the respective problems of each model. This second aim can be stated as,

2. To use the emulator propagation techniques from (1) to offer new insight into the problems of each case study.

Furthermore, it is intended that by examining and using these techniques in detail, the efficacy of each can be assessed and extensions to the methods suggested where appropriate.

1.5 Summary of Chapters

This thesis is structured as follows:

Chapter 2

Uncertainty analysis is explained in depth, with a thorough review of contemporary approaches to the quantification, fusion and propagation of uncertainty. Sensitivity analysis is then introduced and some common measures of sensitivity discussed. Next, the issue of computational expense is addressed, and some solutions in the engineering literature are outlined. The ideal qualities of an emulator are presented, and the Gaussian process (GP) is introduced as an emulator that fulfils many of these criteria.

Chapter 3

Gaussian processes are introduced in depth, with a step-by-step walkthrough of the Bayesian process of conditioning the prior distribution on training data to produce a posterior distribution-over-functions (i.e. training the GP). Next, the marginalisation and estimation of hyperparameters is presented in detail. Finally, the process of obtaining analytical sensitivity and uncertainty measures is given. Overall, this chapter aims to present one of the most complete descriptions of GP-based UA and SA available.

Chapter 4

The first case study for Bayesian UA/SA is presented here, which is the modelling of the aortic valve in the human heart. Two FE models are presented that are both novel in different ways. The first model is a "dry" pressure-loaded model that is specified almost entirely by geometric parameters. A second, "wet" model is a sophisticated fluid-structure interaction model that is at least comparable to contemporary models in the biomechanics literature, using anisotropic hyperelastic material models and considering geometric nonlinearities.

Chapter 5

Uncertainty and sensitivity analyses are performed on both aortic valve models using the GP emulator approach. Geometric, material and loading parameters are investigated and conclusions are drawn from a biomechanical perspective, given that the work here represents the only consideration of uncertainty in heart valve models to date. Observations are also made on the success of the GP emulator: while in general it performs extremely well, the possibility of model bifurcations motivates further work in Chapters 8 and 9.

Chapter 6

A second case study is outlined here, which is a model of an new airship design in production in Italy. The concept of the design and its applications are explained. A series of FE models of the airship are presented, which are to be used to assess stress and displacement in the design under normal loading from propellers and buoyancy, with the final model ultimately including fluid loading from a gust impact.

Chapter 7

The airship models from Chapter 6 are subjected to uncertainty analyses, again using the Bayesian approach from Chapter 3. The work here is used to assess the effect of uncertainties in the models in order to make better decisions regarding the design of the ship. Further conclusions are also drawn regarding the use of the GP emulator.

Chapter 8

Given the issue of possible model bifurcations raised in Chapter 5, this chapter introduces an extension to the GP emulator which uses classification and regression trees (CARTs) to divide the input space into more homogeneous regions. The Bayesian approach to CART is demonstrated on a simpler regression tree, then the method is shown to be extendable to the "tree-structured GP" with several alterations. Finally it is demonstrated on a simple case study of a Duffing oscillator that the this new emulator is capable of modelling bifurcating data by creating divisions of input space over bifurcations, whereas a single GP introduces unwanted fluctuations in the posterior mean.

Chapter 9

A case study of a bifurcating FE model is presented to investigate the ability of the tree-structured GP to model data from a real bifurcating engineering model. The model investigates the movement and stress in a rigid-stent prosthetic heart valve, which is known to bifurcate. The data is used to construct both tree-structured and "standard" GP emulators, and the results compared. Finally, a brief study is performed on the effect of reducing training data for the tree-structured emulator. Conclusions about the comparative abilities of both emulators are drawn, where it is noted that the tree-structured GP has additional advantages as well as its ability to model bifurcations.

Chapter 10

Conclusions from all chapters are summarised and further work is suggested. The bibliography and appendices follow.

Chapter 2

Uncertainty Analysis: An Overview

This chapter aims to outline in more detail the steps and difficulties involved in UA, and presents an overview of the various approaches that have been developed, with a view to putting the work in the rest of this thesis into context. The main aspects and issues of UA are outlined in Section 2.1, including a review of many of the various frameworks that exist for quantifying uncertainty, the relations between them and the practical details surrounding their implementation. In Section 2.3 the specific problem of propagating probabilistic uncertainty is examined, with recent developments described. The concept of *sensitivity analysis* is then introduced in Section 2.2 as a natural extension of UA, being a tool to explore in more depth the effects of uncertainties. It will be seen that one of the main stumbling blocks of uncertainty and sensitivity analysis is the problem of computational expense, therefore a class of methods known as "emulator-based" methods is introduced in Section 2.4. This provides the motivation for the work in Chapters 3 and 8.

2.1 Issues in Uncertainty Analysis

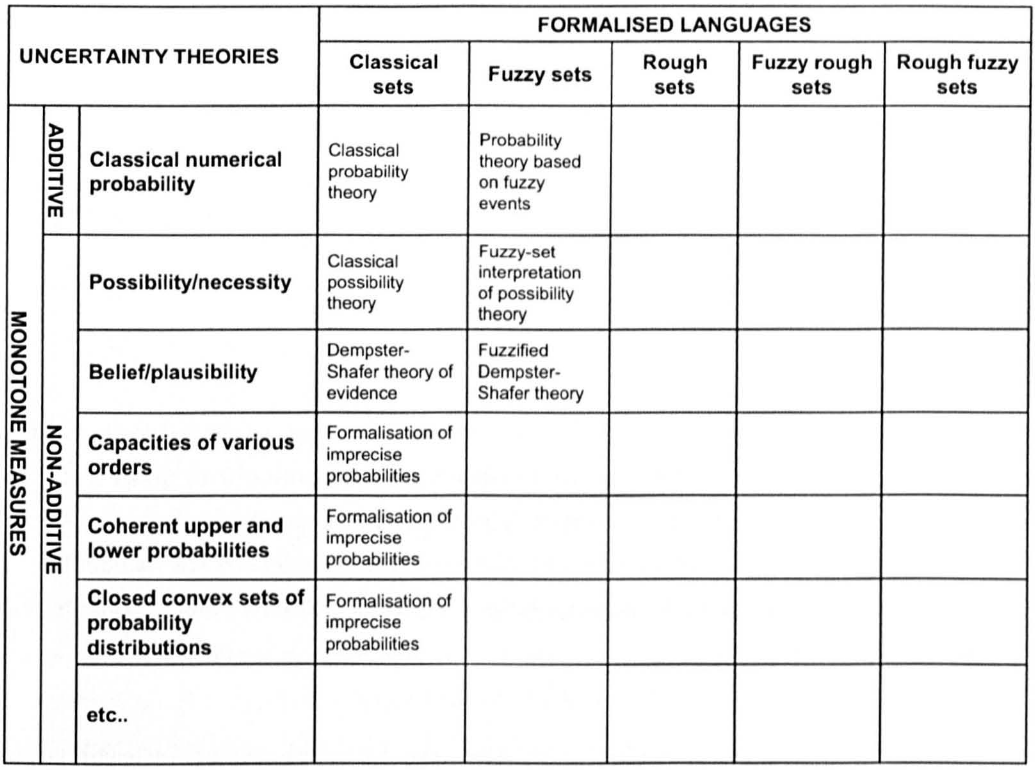
It usually agreed that the process of uncertainty analysis raises three key issues. At the most fundamental level, the magnitude and nature of the uncertainty about a given quantity must be expressed in some mathematical way. This is known as the problem of *quantification.* Probability theory is undoubtedly the most well-known framework, but there are in fact many alternatives, each with its own advantages and disadvantages. Indeed, some offer considerably more general and flexible methods of expressing uncertainty, in particular the ability to express subtle shades of partial ignorance. However many of these frameworks are relatively recent and some have made only a limited impact outside of largely theoretical examples. An intrinsic problem is that uncertainty should ideally be precisely expressed, which means that a framework should be able to express the vagueness that exists about a quantity, without adding any unnecessary assumptions, and equally, without being over-conservative. A discussion of the main methods of quantification is provided in Section 2.1.1.

Given that for a particular problem a number of different types of uncertainty may be present, each could conceivably be expressed in the uncertainty framework that is the most suitable for that particular quantity. Alternatively, for a single quantity, two measures of uncertainty may exist from different frameworks. This then creates the second problem surrounding UA: that of *fusion.* This is the problem of translating from uncertainty expressed in one framework to another in order to *fuse* the data from the two frameworks into a single, more informative expression about uncertainty in that variable. It is evident that the various frameworks discussed here are related to each other in many ways; in fact, some may be seen as special cases of other more general theories. Although the fusion problem is almost certainly the least-investigated problem of the three mentioned here, a short overview of recent developments is given in Section 2.1.2.

Finally, the problem known as *propagation* is usually the subject of the most interest in the field of engineering. In short, if the uncertainty in some model inputs has been quantified in some way, what will be the uncertainty in the outputs and results of the model? This problem is undoubtedly the most studied because it gives the end product of uncertainty analysis: the output uncertainty. For a system that can be expressed in terms of a set of tractable equations, uncertainty can be propagated analytically. In most cases however, models are not analytically tractable and can be viewed even as unknown functions of their inputs — unknown in the sense that for a particular set of input values the output is not known until the model has been run. The problem is further compounded by nonlinearities in the model response: small variations in inputs can cause disproportionately large variations in any of the model outputs and even bifurcations. This leads to the general class of propagation techniques known as sampling-based approaches. As will be seen, the main hurdle to be overcome is that the accuracy of propagation is strongly related to computational expense, so much so that standard methods can quickly become unviable for large problems. This is discussed at length in Section 2.4. A discussion of methods of propagation is given in Section 2.1.3.

2.1.1 Quantification

A brief outline is given here of some of the main frameworks for dealing with uncertainty. Although probability theory will be used throughout this thesis for reasons stated later, to put this work in context in the wider field of uncertainty analysis it is useful to briefly examine the alternative theories that are available for dealing with quantification of uncertainty. Klir and Smith compare various theories and categorise them in two "dimensions"



UNCERTAINTY THEORIES			FORMALISED LANGUAGES				
			Classical sets	Fuzzy sets	Rough sets	Fuzzy rough sets	Rough fuzzy sets
MONOTONE MEASURES	ADDITIVE	Classical numerical probability	Classical probability theory	Probability theory based on fuzzy events			
	NON-ADDITIVE	Possibility/necessity	Classical possibility theory	Fuzzy-set interpretation of possibility theory			
		Belief/plausibility	Dempster-Shafer theory of evidence	Fuzzified Dempster-Shafer theory			
		Capacities of various orders	Formalisation of imprecise probabilities				
		Coherent upper and lower probabilities	Formalisation of imprecise probabilities				
		Closed convex sets of probability distributions	Formalisation of imprecise probabilities				
		etc..					

Figure 2.1: Classification of uncertainty theories according to Klir and Smith [3]

[3] as shown in Figure 2.1. The first dimension deals with the classification of sets. In classical crisp probability theory, for example, a probability is assigned to a subset of events in some universal set. This concept can be extended to assign probabilities to *fuzzy sets*, known as *probability theory based on fuzzy events* [4]. Aside from fuzzy sets, *rough sets* (sets where the upper and lower bounds are themselves defined by sets) [5] and fusions of the two approaches (*rough fuzzy sets* and*fuzzy rough sets* [6]) are alternative ways of defining uncertainty on a set. Fuzzy sets are discussed later in this section, but rough sets will not be further discussed, since they are a relatively recent development. The interested reader can however refer to [7].

The other dimension deals with the measurement of uncertainty assigned to a given set of events (be it a crisp, fuzzy or rough set etc.). These are defined as *monotone measures*, because they satisfy the condition of monotonicity with respect to subsethood ordering, i.e. for all subsets A and B in some universal set Θ, if $A \subseteq B$, then a monotone measure g is such that $g(A) \leq g(B)$.

Monotone measures may be divided into *additive* and *non-additive* measures. For example, classical probability theory uses an additive measure, i.e. for two events A and B,

$$g(A \cup B) = g(A) + g(B) \tag{2.1}$$

whereas the more general non-additive branch of monotone measure theory allows for cases where $g(A \cup B) \neq g(A) + g(B)$, therefore capturing types of uncertainty that cannot be dealt with by probability theory alone. Not all of the theories in Figure 2.1 will be discussed here, since a number of them have made little progress beyond the mathematics literature. Instead, only a selection of the main frameworks that have been applied to engineering will be discussed. The subject of quantification is however discussed in much more detail in a recent book by Klir [8].

Interval Theory

One of the most basic expressions that can be made about uncertainty in a variable is that it is within an certain interval. For example, given an uncertain quantity x, it might be known that it lies between an upper bound $\overline{x}$ and a lower bound $\underline{x}$, but no other information is available. Such expression of uncertainty underlies the concept of *interval analysis*, which was outlined to a large extent by Moore [9] in 1966. More formally, an interval x^I is expressed as,

$$x^I : x \rightarrow [\underline{x}, \overline{x}] \tag{2.2}$$

The concept of intervals on scalar quantities can be extended to interval vectors and interval matrices, for example the d-dimensional interval vector $\mathbf{x}^I$ is expressed as,

$$\mathbf{x}^I = \begin{pmatrix} x_1^I \\ x_2^I \\ \vdots \\ x_d^I \end{pmatrix} : x_i \rightarrow [\underline{x}_i, \overline{x}_i] \tag{2.3}$$

Thus, whereas a crisp (deterministic) vector will describe a point in d-dimensional space, an interval vector describes a d-dimensional hypercube which is bounded by the upper and lower limits of each component of the vector. Perhaps the most significant limitation of interval analysis is that it is an extremely crude method of specifying uncertainty, merely between one value and another. However, this could also be viewed as an advantage in the case where upper and lower limits are the only information available (which is not an uncommon situation). Interval theory will be seen to be closely related to fuzzy set theory — in fact, perhaps the main application of interval analysis in recent literature is as a basis for implementing fuzzy set theory (see later).

Probability Theory

Probability theory is certainly the most widely-used and well understood method of quantifying uncertainty. An uncertain *random variable* X in some sample space $\mathcal{X}$ (the set of all possible outcomes of X) is assigned probabilities $p(X)$ for each value of $X \in \mathcal{X}$, such that,

$$\begin{aligned} &p(x) \rightarrow [0,1] \quad \forall\, x \in \mathcal{X} \\ &\sum_{x\in\mathcal{X}} p(x) = 1 \end{aligned} \tag{2.4}$$

In the case when X is continuous, probabilities cannot be assigned to point values of x. Therefore it is convenient to define the *probability distribution* of x in a number of ways, for example, the *cumulative distribution function* (cdf) $P(x)$ describes the probability of X being equal to or below x. More commonly however, a *probability density function* (pdf) $p(x)$ is used. It is a function that defines the probability of X falling inside a given interval $[x, x + \delta x]$, such that,

$$p\{X \in [x, x + \delta x]\} = \int_{x}^{x+\delta x} p(x)dx \tag{2.5}$$

The univariate case can be generalised to a multivariate pdf for a d-dimensional random variable (hereafter pdf will refer also to the multivariate pdf) denoted $p(\boldsymbol{x})$, which gives the individual pdfs of each variable X_i and the dependencies between them. An example of this is the multivariate Gaussian distribution, which is given here (since it will feature significantly in this thesis),

$$p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}}\exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})^T\right\} \tag{2.6}$$

where $\boldsymbol{\mu}$ is a d-dimensional vector of means and Σ is a $d{\times}d$ *covariance matrix* that gives the covariance between inputs. An important aspect of multivariate pdfs that will be stated here is the idea of marginalisation, which is stated as,

$$p(\boldsymbol{x}_r) = \int_{\mathcal{X}_{-r}} p(\boldsymbol{x})d\boldsymbol{x}_{-r} \tag{2.7}$$

which states that the distribution of $\boldsymbol{x}_r \subset \boldsymbol{x}$ can be obtained by integrating with respect to the variables in the complementary set $\boldsymbol{x}_{-r}$ over their sample space $\mathcal{X}_{-r}$.

An important distinction in probability theory that should be mentioned here (and the source of ongoing dispute) is the difference between the *frequentist* and *Bayesian* interpretations of probability. From the frequentist perspective, probability is defined as the limit of frequency of occurrence of an event over a large number of trials. This of course introduces the limitation that probabilities of events that have not been witnessed yet cannot be formally specified. In the alternative Bayesian viewpoint, probability is considered to be a degree of belief. Probabilities and distributions can therefore be assigned to events about which there is some subjective opinion or prior knowledge. Bayesian inference, which is the process of inferring unknown quantities through prior assumptions and further evidence, is used extensively in this thesis. Briefly, inference is performed using an updating procedure known as *Bayes' Theorem*, such that the probability of an event B given some evidence A is given as,

$$p(B|A) = \frac{p(A|B)p(B)}{p(A)} \tag{2.8}$$

where $p(B|A)$ is called the *posterior probability* (of B), $p(A|B)$ the *likelihood*, $p(B)$ the *prior probability* and $p(A)$ is a normalising factor. Bayes' theorem is often used to estimate model parameters Θ through Bayesian inference, sometimes by finding the *posterior mode* of Θ, a process known as *maximum a-posteriori* (MAP) estimation (see [10]).

A probability distribution is considerably more informative than the intervals described previously. The pdf $p(x)$ can take any number of forms so long as the pdf integrates to 1 over $\mathcal{X}$. This means that potentially any distribution of probability can be expressed, as long as a suitable pdf can be defined. Much like interval analysis, the main strength of probability theory is also its weakness, although in this case the issue is that uncertainty *must* be expressed in some detail. Even the simplest pdf, the uniform distribution, implies that probability is uniformly distributed within a given range. Interval theory, in contrast, makes no assumption whatsoever about the distribution of uncertainty inside the interval. In the case where detailed information is available about the uncertainty of a parameter, probability theory is an excellent choice. However, quite often the distribution of a variable is unknown and must be *elicited*. This is itself an active field of research — a full treatment is given in [11]. Even then, a lack of information can force assumptions to be made about the nature of uncertainty in a variable that cannot be justified by the available data.

Possibility Theory

First outlined in detail by Lotfi Zadeh [12], *possibility theory* is viewed as an alternative to probability theory that allows for more subtle expressions of uncertainty and partial ignorance. Whereas probability theory allows only a single measure of uncertainty (the probability of an event), possibility theory uses two measures, known as the *possibility* and the *necessity*. The possibility measure of an event A from the universal set Θ is defined such that $\text{pos}(A) \rightarrow [0,1]$, in much the same way as probability. However, a probability measure of 1 suggests that an event is certain to happen, whereas a possibility measure of 1 only suggests that it is completely possible that this event occurs, or in other words, one would not be at all surprised if the event occurs. This can be stated as:

- $\text{pos}(A) = 0$ implies that A is completely impossible
- $\text{pos}(A) = 1$ implies that A is completely possible, plausible or unsurprising

Possibility can of course take any measure in the interval $[0,1]$, expressing varying degrees of belief about the possibility of A occurring. To complement the possibility, the necessity

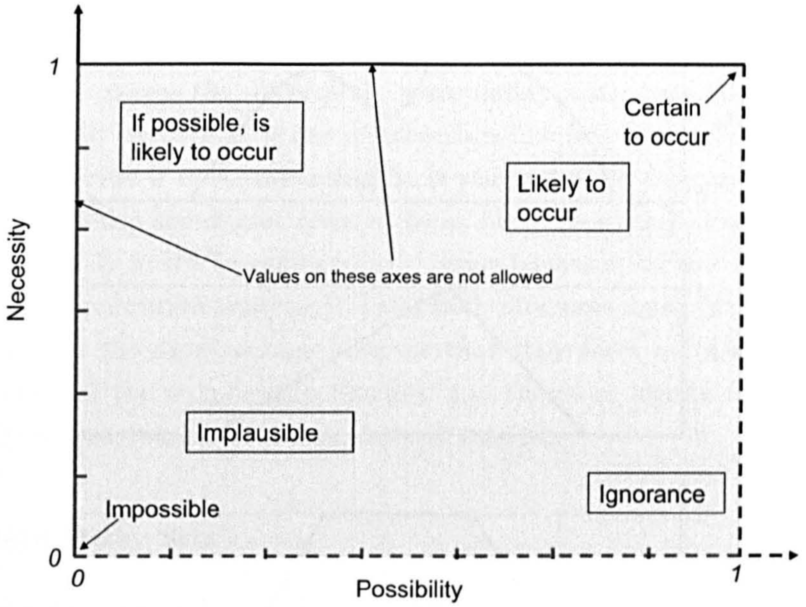


Figure 2.2: Interpretations of various regions in possibility/necessity.

must also be expressed. This is defined as,

$$\text{nec}(A) = 1 - \text{pos}(\bar{A}) \tag{2.9}$$

where $\bar{A}$ here denotes the complement of A in the universal set. Between the two measures, it is possible to express more subtle levels of uncertainty than with probability theory. Consider three situations:

- $\text{pos}(A) = 0$ implies that A is impossible. $\text{nec}(A)$ must also be zero.
- $\text{pos}(A) = 1$ implies that A is completely possible. $\text{nec}(A)$ can take any value. If $\text{nec}(A) = 0$, this implies complete ignorance about A, i.e. A is definitely possible, but it would be completely unsurprising if it did not occur.
- $\text{nec}(A) = 1$ implies that A is completely necessary and therefore will definitely occur. This requires that $\text{pos}(A) = 1$.

An illustration of this is given in Figure 2.2. Possibility measures follow a series of rules, much like in the probabilistic framework, that allow unions of possibilities, intersections and conditional possibilities. For further information, see [13].

Fuzzy Sets

Fuzzy set theory was first introduced by Lotfi Zadeh [14] (also one of the principle developers of possibility theory). In contrast to conventional set theory, where an element

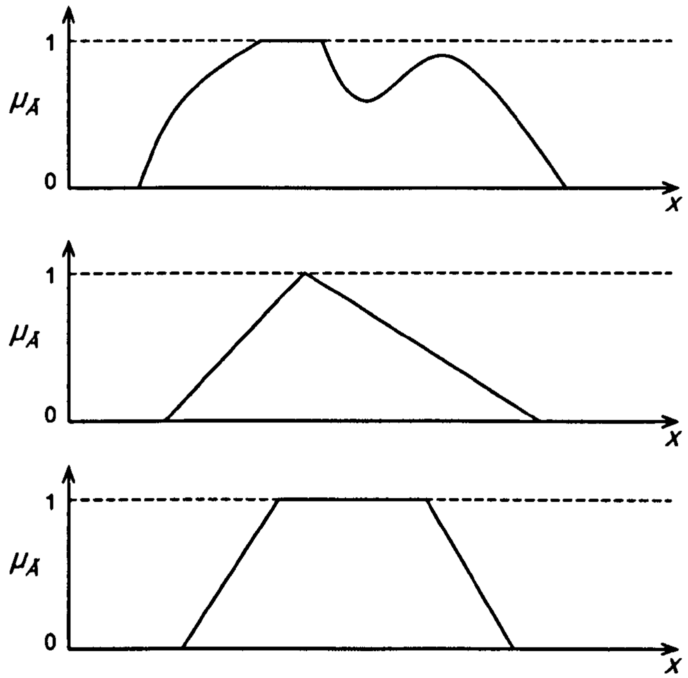


Figure 2.3: Illustration of membership functions of fuzzy quantities: general fuzzy quantity (top); fuzzy number (middle); fuzzy interval (bottom).

either definitely belongs or definitely does not belong to a given set, a fuzzy set has an associated *membership function* that states the degree of membership or belonging to the set as a value in the interval $[0,1]$. Thus a "crisp" (conventional) set can be seen as a special case of a fuzzy set, where an element's membership is only allowed to take a value of either 0 (it is not in the set) or 1 (it is in the set). A fuzzy set $\tilde{A}$ is written as,

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) : \mu_{\tilde{A}} \to [0,1]\} \tag{2.10}$$

where $\mu_{\tilde{A}}$ is the membership function that assigns a membership value to any x. As with other frameworks for uncertainty, fuzzy sets have a system of rules governing operations such as unions and intersections of sets. The reader is referred to Klir's book [15] for further information. It is important to point out here that the membership function of a fuzzy set is *not* equivalent to the probability distribution of a random variable (which is a common misunderstanding). The difference is subtle but important: whereas a fuzzy set assigns levels of membership to known and fixed elements of the universal set Θ, probability theory assumes a crisp set, but it is the values of the elements of Θ that are uncertain. Therefore, fuzzy sets should not be viewed as an alternative to probability theory; rather, the two can complement each other to a large extent. This is discussed in detail in [16].

A concept which is of particular interest in the context of uncertainty in modelling is that of *fuzzy quantities*. Fuzzy quantities are a special case of fuzzy sets, since a fuzzy quantity can be thought of as a fuzzy subset of $\mathbb{R}$. A membership function can be defined over some range of values to express the uncertainty surrounding some particular input parameter (see Figure 2.3). In particular, if the membership function is triangular, it is termed a *fuzzy number*, whereas if it is trapezoidal it is referred to as a *fuzzy interval* (although fuzzy intervals are also sometimes referred to as fuzzy numbers). Fuzzy quantities have been used extensively in UA in engineering — some instances include fuzzy finite element analysis applied to vibration analysis [17] and fault diagnosis using fuzzy logic [18]. Fuzzy quantities do have the disadvantage however that they force quite strong assumptions about the nature of the membership function and therefore ideally require considerable knowledge of the distribution of the quantity of interest.

Non-Standard Fuzzy Sets

It is worth briefly mentioning some of the extensions of fuzzy sets that have been proposed. In the first instance, *interval-valued fuzzy sets* are fuzzy sets where the membership function is itself uncertain, and assigns an *interval* of membership rather than a crisp membership value. Another way of looking at this is to define the membership function as an interval,

$$\mu_{\tilde{A}} = [\underline{\mu_{\tilde{A}}}, \overline{\mu_{\tilde{A}}}] \tag{2.11}$$

If this is not uncertain enough, there exist *type-2 fuzzy sets*, in which the membership function is itself defined on fuzzy intervals. Therefore, for a given x, the membership function will return a fuzzy quantity describing the membership level of x. In fact, this concept can be extended to further nested levels of membership functions, although the practical value of this is questionable.

Another extension, known as a *level-2 fuzzy set*, is to define a number of different level-1 (standard) fuzzy sets, which are themselves grouped together into a fuzzy set. This nesting of fuzzy sets allows higher-order concepts to be represented by lower-level ones. As with type-2 fuzzy sets, level-2 fuzzy sets can naturally be extended to further levels as required, but again these higher orders have seen little investigation as yet. Some further manifestations of fuzzy sets are discussed in [8].

Dempster-Shafer Theory of Evidence

Dempster-Shafer Theory (DST) is a theory of evidence based on work by Dempster [19] and Shafer [20] in the 1960s and 1970s. Like possibility theory, DST allows a slightly more general framework than probability theory since it is non-additive and therefore allows more subtle expressions of partial ignorance. A particular distinction is that instead of only assigning probability mass to single elements, probability mass can be assigned to *subsets* of elements.

Let the *frame of discernment* Θ (otherwise known as the *universal set* in other frameworks) be defined as the exhaustive set of possible events. The elements of $\{2^\Theta\}$, which are all the possible subsets A of Θ, are given a probability mass m known as the *basic belief assignment* (BBA), such that

$$\begin{aligned} &m(A) \rightarrow [0,1] \quad \forall A \subseteq \Theta \\ &m(\varnothing) = 0 \\ &\sum_{A \subseteq \Theta} m(A) = 1 \end{aligned} \tag{2.12}$$

where $\varnothing$ is the empty set. Any subset A of Θ where $m(A) > 0$ is called a *focal element.* If all focal elements are singletons (i.e. contain only one element), this is the special case of Bayesian probability theory - i.e. assigning belief to single events or non-overlapping subsets of events. DST therefore extends probability theory by allowing a degree of uncertainty or ignorance, because belief in events can "overlap" between different subsets. Partial ignorance is often represented by assigning a proportion of the unit belief mass to the entire frame Θ, since this implies no special knowledge about any subset of Θ.

The *belief* of an event B is related to the BBA of subsets of A by,

$$\text{Bel}(B) = \sum_{A \subseteq B} m(A) \tag{2.13}$$

which sums all the probability mass that is in support of B occurring. A further measure, known as the *plausibility*, is defined as,

$$\text{Pl}(B) = \sum_{A \cap B \neq \varnothing} m(A) \tag{2.14}$$

which expresses the total belief mass that could be assigned to B if all unknown facts were found to support B. Finally, a third measure is sometimes used, known as the *doubt*, which is related to the belief such that,

$$\text{Dou}(B) = \text{Bel}(\bar{B}) \tag{2.15}$$

i.e. the total probability mass in support of B *not* occurring. It should be emphasised that since belief and doubt do not necessarily sum to unity, DST can represent an interval of uncertainty that is beyond the scope of probability theory (see Figure 2.4). In fact, belief and plausibility actually represent the upper and lower bounds of the "true" belief in an event. The probability mass assigned to the belief and doubt is fixed, but if the ignorance were somehow removed (by gaining further information) the remaining mass

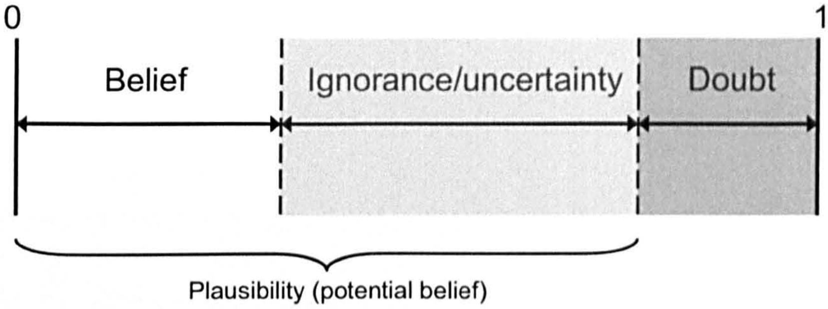


Figure 2.4: Dempster-Shafer assignment of belief and uncertainty across the unit interval (adapted from [21]).

could be assigned to either further doubt in an event, or further support, so that the true belief of B can be anywhere in the interval $[\mathrm{Bel}(B), \mathrm{Pl}(B)]$.

DST has been successfully applied in the context of damage location in aircraft [21], fault diagnosis in engines [22] and fault diagnosis in induction motors [23]. However it still remains largely within the confines of simple theoretical applications.

2.1.2 Fusion

As stated previously, fusion between the various theories of uncertainty is the least investigated of the three problems of UA. The Klir diagram in Figure 2.1 does however show that, in particular, there has been some work done to apply the theory of fuzzy sets to non-additive measures of uncertainty. For example, Zadeh generalised probability theory in 1968 to apply to events bounded by fuzzy sets [4]. The relations and misunderstandings between fuzzy sets and probability theory are discussed in some detail by Dubois [16]. Zadeh similarly interpreted classical possibility theory in a fuzzy context [12] a little later, and more recently has suggested a *Generalised theory of uncertainty* [24]. DST has also recently been generalised to deal with fuzzy sets, preserving the concept of upper and lower probabilities (which had previously not been acheived). More on this can be found in [25].

There are also some cross-overs between various theories of monotone measures. Ross has done work to create a bridge between probability and possibility theory in the context of "Total Uncertainty" [26], which allows an expression of both epistemic uncertainty and aleatory. Dubois also discusses at length the various ways in which possibilistic measures can be interpreted in a probabilistic sense [27]. In fact, possibility theory has been proposed as a kind of bridge between probability theory and fuzzy set theory by Dubois and Prade [16]. Finally, it has already been stated that DST is a generalisation of probability theory to allow non-additive measures of uncertainty. A much fuller discussion of uncertainty measures and the various associations between them is given in [8].

2.1.3 Propagation

Intervals

The method of propagation is strongly dependent on the framework of quantification of uncertainty. In the case of interval analysis a system of interval arithmetic parallels many of the arithmetic operations available to crisp numbers. For example, the addition of two intervals is achieved by,

$$[a,b] + [c,d] = [(a+c),(b+d)] \tag{2.16}$$

and multiplication is defined by,

$$[a,b] * [c,d] = [\min(ac,ad,bc,bd), \max(ac,ad,bc,bd)] \tag{2.17}$$

Intervals can therefore replace crisp numbers in the specification of model inputs, and theoretically all the calculations that are involved in the solution of the model with the crisp input could be replaced by interval calculations, thus yielding an interval vector that specifies all the outputs of a given model in terms of intervals.

One problem with propagating intervals through a model is that in order to precisely propagate them it is required that the equations governing the model can be expressed in a closed form, but in practise this is rarely possible. Instead, research focuses on approximations of the exact solution set. Additionally, interval analysis introduces conservatism, which means that it tends to over-estimate the uncertainty in the model output as a result of neglecting dependencies between the inputs. Some work has been done to address this last issue by Manson in the context of *affine arithmetic* [28]. An extensive discussion of the use and application of interval analysis is found in [29].

Fuzzy Quantities

Fuzzy quantities that are assigned to uncertain model parameters can be propagated through a standard FE model by dividing the membership functions into a series of "α-cuts", such that the membership function for each fuzzy input $\{\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n\}$ is intersected at a level $\mu_{\tilde{x}_i} = \alpha$ (see Figure 2.5). This results in number of nested crisp sets at each α-cut that are typically propagated through the model by an interval analysis, producing a realisation of the fuzzy quantities that describe the model outputs. The problem with this is that since it involves interval analysis, it introduces conservatism into the estimates of the output uncertainties (as discussed earlier). A way round this is to use global optimisation to search for the global maximum and minimum of the output at a given α-cut, which means that the practice of propagating fuzzy numbers is closely tied to the discipline

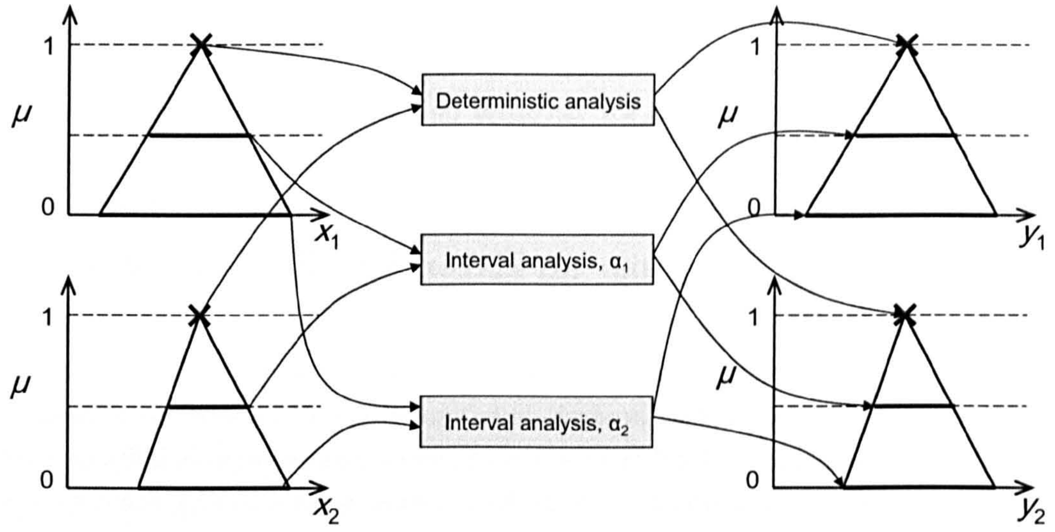


Figure 2.5: An example of α-cut propagation of fuzzy numbers through a two-input, two-output model using two α-cuts.

of optimisation — an example is that by Moens and Vandepitte [30]. Propagation of fuzzy quantities via α-cuts has been applied extensively to dealing with uncertainty in FE models — some examples of this are [17, 31].

Probability Distributions

Since propagation of probabilistic uncertainty is one of the central themes of this thesis, a description of available approaches is left until Section 2.3, where it is treated in greater detail.

Other frameworks

The methods available for propagating possibility theory and DST have seen very little investigation at all — in fact, it is difficult to find virtually any references in the literature. Therefore, propagation of these frameworks will not be discussed here.

2.1.4 Summary

Examining the uncertainty frameworks available, it is apparent that each method has its own advantages and disadvantages and can be useful in particular situations. One of the defining features is the degree of ignorance that can be expressed. Probability theory is perhaps the perfect characterisation of uncertainty, in the situation where enough information is available to characterise the pdf of an uncertain quantity. If this quantity of information is not available, assigning a pdf requires some assumptions that may not be justified. In this case, possibility theory and DST can offer more subtle expressions of ignorance. However, the frameworks for using such theories are considerably less developed. For this reason, uncertainty will hereafter be considered from a purely probabilistic perspective.

2.2 Sensitivity Analysis: an Extension of Uncertainty Analysis

The end product of uncertainty analysis is to quantify the uncertainty of some model outputs in order to know the robustness of the model, or assess the probability of exceeding some critical value in the model output. If the uncertainty surrounding the output of the model is unacceptable, this can be the start of a wider iterative process to attempt to reduce it to within some specified limit. The question then arises: "what can be done to reduce uncertainty?". In the case of aleatory uncertainty, the answer is "not a lot". As discussed, aleatory uncertainty is fundamentally irreducible since it involves natural variability. Epistemic uncertainty, on the other hand, is caused by insufficient knowledge, therefore if extra information were gained about a quantity of interest, the uncertainty of the model output could potentially be reduced.

According to a heuristic suggested by Pareto [32] (often known as the *Pareto Principle*) roughly 80% of "effects" in many systems are a result of only 20% of the "causes". Although a somewhat vague observation, this ratio tends to hold roughly true in many situations, from business (80% of business comes from 20% of clients) to software engineering (80% of errors can be eliminated by fixing the top 20% of bugs). In the context of uncertainty analysis, the implication is that quite often a small subset of model inputs is causing the majority of the output uncertainty — to put it another way, even if equal uncertainty were assigned to all model inputs, it would be seen that the output is considerably more *sensitive* to uncertainty in certain inputs, and practically *insensitive* to others. Therefore in order to reduce the uncertainty of the model output, it is of great interest to identify the inputs that are most important in influencing the output. This is one of the main motivations of what is known as *Sensitivity Analysis* (SA). Since SA is a natural extension of UA, it features at length in this thesis. An outline of some of the main motivations and techniques follows here.

2.2.1 Motivations

Saltelli lists the various motivations of SA from a number of different perspectives [33]. Amongst these are three which are particularly interesting from the point of view of an engineer:

1. To identify parameters that contribute significantly to output variability with a view to reducing the uncertainty of these parameters.

2. To identify insignificant parameters that can be eliminated from the final model (i.e. model simplification) and disregarded in uncertainty analysis.

3. To understand interactions between parameters and therefore gain deeper insight into the working of the model.

The first of these has already been discussed. If an uncertainty analysis reveals that model uncertainty is unacceptably high, SA can be used to identify the important parameters, and effort can be directed to reducing the uncertainty in these inputs. Equally, by identifying unimportant parameters, a model can be considerably simplified, thereby reducing computational expense. These parameters can then be ignored in further UA, SA or optimisation procedures. Finally, it will be seen in later chapters that SA can provide a remarkably useful way of investigating and understanding the interactions and processes at work in a complex model.

2.2.2 Levels of Sensitivity Analysis

SA is often classified into three levels of depth (see Figure 2.6). At the most basic level, *screening* ranks the inputs in order of importance in affecting the output. This can help the modeller identify the set of most important inputs, as well as any inputs that contribute very little to the output (and can thus be eliminated from subsequent uncertainty analyses), but it offers little or no quantification. The next level, *local SA*, analyses and quantifies the effects of varying input parameters, but only around their immediate locality. This does not allow for full exploration of the input space unless responses are linear, so is of limited use in complex models. The most informative analysis is *global SA*, which investigates and quantifies uncertainties over the complete range of input space. Of course, the cost of extra information inevitably increases computational expense, which will be seen to be one of the most constraining issues in UA/SA, and is discussed further in Section 2.4.

Screening

Most screening analyses are based around the concept of a *Design of Experiments* (DOE). This involves running the model at a carefully chosen set of input points to investigate the basic response of the output to simple variations in the input. An example of this is a *one-at-a-time* (OAT) experiment [34]. In this approach, the model is evaluated at maximum and minimum values of each input in turn. The results are then compared to a

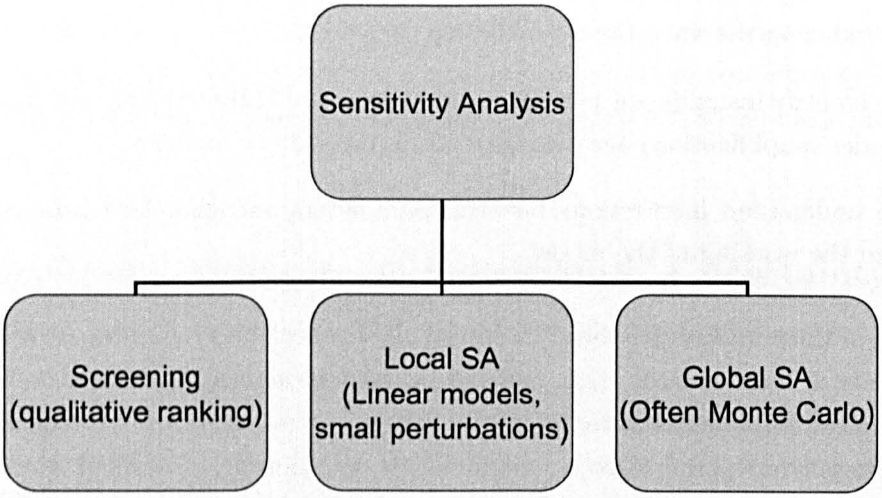

Sensitivity Analysis
Screening (qualitative ranking)
Local SA (Linear models, small perturbations)
Global SA (Often Monte Carlo)

Figure 2.6: The three levels of sensitivity analysis, as suggested by Saltelli [33].

baseline run (i.e. all parameters at their nominal values) and the residuals used to estimate sensitivity.

A clear limitation of OAT designs however is that first, they assume monotonicity, and second, they ignore the possibility of interactions between parameters. For example, two parameters x_1 and x_2 may be largely uninfluential to individual variations, but when both are simultaneously varied, the output may vary substantially. In fact, the proportion of input space partially explored in an OAT design decreases rapidly as the number of input dimensions increases. One simple solution to this problem is the concept of *factorial experimentation* [35], where the range of input parameters is divided into k levels. The input parameters are then varied simultaneously across all permutations of these levels. The clear drawback is computational cost, which is proportional to k^d, where d is the dimension of input space. A *fractional factorial design* alleviates this problem to some extent by only investigating a limited subset of input interactions. Notably, all screening methods disregard the distribution of uncertainty for input variables, so are of no use in propagating uncertainty (unless perhaps if the uncertainty is uniformly distributed and the model is linear). They may however be of use to identify non-influential variables that could then be discounted from a more detailed UA.

Local Sensitivity Analysis

Local SA is an analytical approach that involves taking partial derivatives $\partial y/\partial x_i$ of the model around some "locality" x_0, usually the nominal or mean values of the parameters, giving a first-order sensitivity measure S_i such that,

$$S_{i,\text{local}} = \left.\frac{\partial y}{\partial x_i}\right|_{x_i=x_0} \tag{2.18}$$

Local SA, is (as its name suggests) only useful in the locality of x_0, since estimates about local sensitivity are performed by making small perturbations about the nominal parameter values and therefore contain no information about the model response outside of this locality. Nevertheless, if a model is known to have a linear response, or nearly linear, it can be a useful SA tool requiring little computational expense. A variety of methods are compared by Saltelli [33] that compute the partial derivative in (2.18). The most simple approach is the finite-difference approximation, where,

$$\frac{\partial y}{\partial x_i} \approx \frac{f(x_i + \Delta x_i) - f(x_i)}{\Delta x_i} \tag{2.19}$$

although of course this can require a substantial number of model runs if d is large. Other, more sophisticated methods include the method of Miller and Frenklach [36], which approximates the solution with an empirical model, and the *polynomial approximation method* [37], which approximates the sensitivity differential equations by polynomials. One additional drawback of local SA is that it does not account for non-uniform distribution of uncertainty in the inputs. Therefore it is more applicable to SA alone, rather than propagating uncertainty.

Global Sensitivity Analysis

Global SA is the most informative level of sensitivity analysis, since (at least ideally) it investigates the entire range of input space and can handle nonlinear responses. In the case of most FE models it is required to run SA at the global level because nonlinearity is often present, or at least can rarely be ruled out. For this reason, global SA and UA will be the main theme of this thesis. Although Global UA/SA could be considered from within several of the frameworks of uncertainty discussed earlier, this work will focus on the probabilistic point of view. From this perspective then, a global SA can be seen as consisting of five levels:

1. Elicitation of input distributions
2. Building a sampling strategy (DOE)
3. Model realisations $\{y_1 = f(\mathbf{x}_1), y_2 = f(\mathbf{x}_2), ..., y_n = f(\mathbf{x}_n)\}$
4. Uncertainty analysis
5. Sensitivity analysis

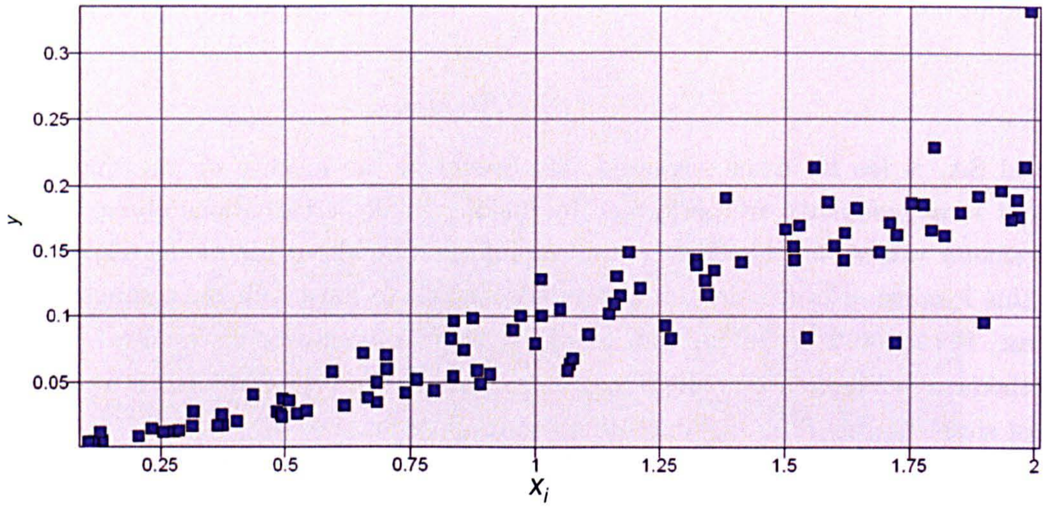


Figure 2.7: An example of a scatter plot of a chosen variable x_i and an output y.

The first four of these have been discussed previously, although still ignoring the problem of computational expense, which is addressed in Section 2.4. There are many approaches for dealing with the third step, of which many are summarised in [38].

The crudest measures of global sensitivity can be obtained by scatter plots and correlation coefficients. For example, a plot of a given output y_i against an input x_i can be drawn. Although it is a very informal method, it can serve to give a good initial impression about underlying trends. Figure 2.7 shows as an example of a number of model samples — the roughly linear trend is immediately evident, as is the increasing scatter with increase in x_i. An obvious problem however is that it is only a qualitative indication, and plots must be generated for every input. This can be addressed by the use of correlation coefficients such as the *Pearson product moment correlation coefficient* and the *Spearman coefficient*, which can be used to assess correlation in linear and nonlinear models respectively [39]. Another strategy is to fit a linear model to the sampled data and use the regression coefficients as sensitivity measures. This kind of regression analysis could be criticised by the fact that it assumes linearity, therefore is not much improvement on a local SA. A nonlinear model could perhaps be fitted, but the results might be hard to interpret.

One useful way of visually assessing the sensitivity of the model output to particular inputs is based around a functional decomposition of $f(\boldsymbol{x})$ into orthogonal functions of individual inputs and subsets of inputs (see [40]). This requires an assumption of independence. Taking expectations of this decomposition yields the following (see e.g. [41]) where the function (model) is broken down into a series of expected values $E(\cdot)$ and conditional expectations,

$$y = f(\boldsymbol{x}) = E(Y) + \sum_{i=1}^{d} \varepsilon_i(x_i) + \sum_{i<j} \varepsilon_{i,j}(\boldsymbol{x}_{i,j}) + \sum_{i<j<k} \varepsilon_{i,j,k}(\boldsymbol{x}_{i,j,k}) + \ldots + \varepsilon_{1,2,\ldots,d}(\boldsymbol{x}), \quad (2.20)$$

$$\varepsilon_i(x_i) = E(Y|X_i) - E(Y)$$

$$\varepsilon_{i,j}(\boldsymbol{x}_{i,j}) = E(Y|X_{i,j}) - \varepsilon_i(x_i) - \varepsilon_j(x_j) - E(Y)$$

and so on for higher interactions. $\varepsilon_i(x_i)$ is known here as the *main effect* of x_i, which is the effect of varying x_i, averaged out over the uncertainty in the other parameters. $\varepsilon_{i,j}(\boldsymbol{x}_{i,j})$ is known as the *interaction effect* of x_i and x_j. Typically in a global SA the main effects of input parameters are plotted on a single graph, normalised over the range of each parameter. Plotting main effects serves as a visual indication of the influence of particular inputs and interactions, showing (albeit qualitatively) the variance of the output with respect to individual input parameters and the nonlinearities associated with those responses.

If the main effect of an input produces little output variation over its range (i.e. the response is flat), it can be said that the output is insensitive to the main effect of that parameter. Conversely, if the main effect has a strong gradient, the output is clearly sensitive to variations in that input. A quantitative measure of this property is to take the variance V_i of the main effect, i.e.

$$V_i = \mathrm{var}\{E(Y|X_i)\} \quad (2.21)$$

Another way to interpret this is that it is the amount by which the uncertainty (variance) in the model output would be reduced if the "true" value of x_i were suddenly known. This can of course be extended to subsets of inputs to give the variance contribution of interactions of variables. For easy interpretation, these variance measures are typically standardised by dividing by the total output variance, i.e.

$$S_i = \frac{\mathrm{var}\{E(Y|X_i)\}}{\mathrm{var}\{E(Y)\}} \quad (2.22)$$

where S_i is known as the *main effect index* (MEI) of x_i, a widely-used global sensitivity measure proposed by Sobol' [40, 42] with the property that,

$$\sum_{i=1}^{d} S_i + \sum_{i<j} S_{i,j} + \ldots + S_{1,2,\ldots,d} = 1 \quad (2.23)$$

where $S_{i,j}$ denotes the sensitivity index of the interaction between x_i and x_j. One drawback of this measure is that to account for all variance it is necessary to calculate all $2^d - 2$ sensitivity indices, which for a model with many inputs could be prohibitively time-consuming. An alternative sensitivity measure [43], describes the output variance that would remain if one were to learn the true values of all inputs except x_i,

$$V_{Ti} = \text{var}(Y) - \text{var}\{E(Y|X_{-i})\} \tag{2.24}$$

This measure, called the *total sensitivity index* (TSI), measures the variance caused by an input x_i and any interaction of any order including x_i, since after standardisation, this yields,

$$S_{Ti} = 1 - S_{-i} \tag{2.25}$$

where S_{-i} is the sum of all the main effects and interactions that *do not* include x_i. S_{Ti} therefore allows a more holistic view of the uncertainty attributed to each input, but does not give any details as to how it is distributed between main effects and interactions. Note that since the TSIs "overlap" each other, they do not typically sum to unity.

2.2.3 Summary

Of the three levels of SA presented here, global SA is the most applicable to modern engineering models, since nonlinearities are often present or can not usually be ruled out. Screening and local SA techniques may be used as an initial analysis to filter out unimportant parameters, but to give an accurate representation of sensitivity in a nonlinear model, the variance-based measures given here are the best option. The drawback of global SA is of course that it is computationally much more expensive than simpler techniques. This problem is discussed further in the following sections.

2.3 Propagating Probabilistic Uncertainty

Methods of propagating probabilistic uncertainty through a model vary in their approach and applicability. This section will first describe general approaches for propagating probability distributions through an arbitrary system (Section 2.3.1), followed by a review of some recent approaches which are designed specifically for structural models and FE (Sections 2.3.2 and 2.3.3).

2.3.1 General Approaches

In order to exactly propagate probabilistic uncertainty through a model $y = f(x)$, an analytical solution can be found if the function f can be written explicitly. However, in the majority of practical cases, this is not possible. The most common solution to this problem is to use a sampling-based approach — in short, the joint pdf of the inputs is sampled at a number of points, then each point is run through the model to generate a crisp output value. With enough sample points, an approximation of the distribution of any output of interest may be obtained. Such an approach is known as the *Monte Carlo Method* [44].

Monte Carlo propagation works well as long as the sampling resolution is sufficiently high, but a clear problem is that if the number of input dimensions is large, and/or the simulation has significant run time, the computational expense of building up a good estimate of output distributions quickly becomes prohibitively high. It should be noted however, that that the great strength of the Monte Carlo method is that it can estimate means and variances of outputs with an accuracy that is *independent of the number of inputs* (the same is not true if building up a histogram of the output distribution). Despite this, it can still be computationally expensive if a number of separate estimates (e.g. means, variances and conditional variances) are required from a single large model, to the point of infeasibility. As such many methods have been proposed increase the efficiency of the Monte Carlo method.

One aspect of this is known as *design of experiments* (DOE), which is the strategy used for selecting the number and location of input points when building a sample set. Many types of sampling procedures exist — one of the most obvious of these is *random sampling*. Given a joint pdf $p(\boldsymbol{x})$, samples are randomly drawn from the sample-space $\mathcal{X}$, weighted by the probability distribution; i.e. more samples will be drawn from regions of high probability and less from regions of low probability. This can be done by generating random numbers across the interval $[0,1]$ and using them to draw samples of $p(\boldsymbol{x})$ from the cdf of $\boldsymbol{x}$ — a univariate example of this is shown in Figure 2.8. However, the main drawbacks with random sampling are that first, it is not possible to generate *truly* random numbers, and second, random numbers do not guarantee that the sample space will be explored evenly for small samples. Additionally, random samples may occur very close to one another, thus making the sampling inefficient.

An alternative method, known as *importance sampling*, divides the sample space $\mathcal{X}$ into R non-overlapping regions such that $\mathcal{X} = \{\mathcal{X}_v\}_{v=1}^{R}$. Random samples are drawn from each region $\mathcal{X}_v$, weighted to the probability of that region to provide meaningful results (often there is only one sample drawn per region, resulting in a total of R samples). The advantage of this method is that the sample space is guaranteed to be explored to a resolution controlled by the specification of R. However, a problem may be that the range of each input variable may not be fully explored.

One extension of importance sampling is known as *Latin hypercube sampling* (LHS) [45]. In this method, the range (support) of each variable is divided into n_{LHS} intervals of equal probability. For each interval in the first input variable x_1, a value is randomly sampled. This is then randomly paired with the n_{LHS} similarly randomly sampled values from x_2, without replacement. The process is repeated with all variables up to x_d, resulting in a total of n_{LHS} samples that are distributed in a multi-dimensional generalisation of a

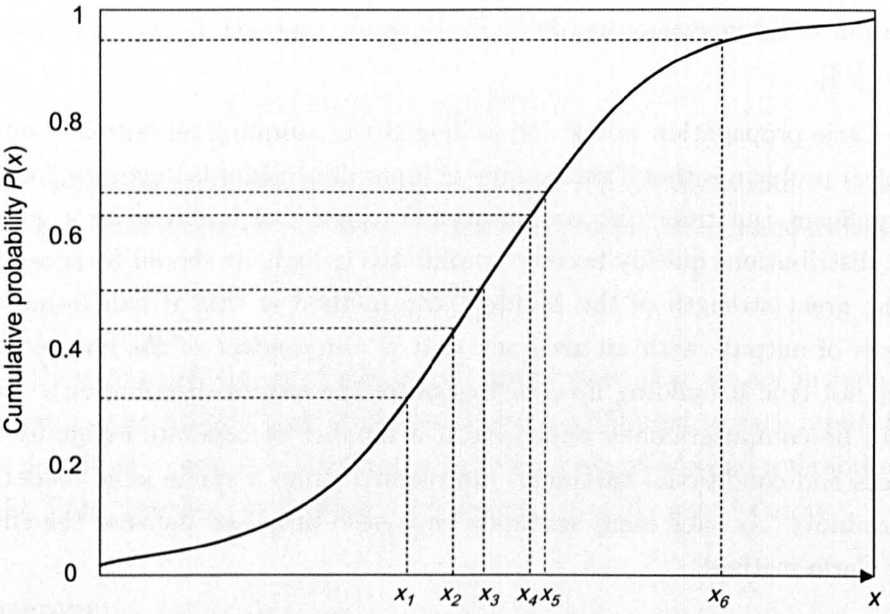


Figure 2.8: Random samples drawn from the unit $[0, 1]$ used to sample from the cumulative distribution of a normal distribution over x.

Latin square — i.e. there is exactly one sample in every division of each variable. An illustration of LHS for two input variables is shown in Figure 2.9.

Although random sampling is the "ideal" sampling method, since it will naturally provide unbiased estimates of statistical moments, the number of samples required for propagation of pdfs can be unfeasibly high. LHS designs are generally thought to be more suitable when there is a practical limit on the number of samples available and there is no need to accurately estimate high quantiles (such as 99th percentiles and above), such as in the majority of uncertainty analyses. Importantly, since LHS designs are guaranteed to sample across the full range of each variable, they will include the effects of varying every parameter between its maximum and minimum: this is particularly desirable when performing sensitivity analysis (see Section 2.2).

LHS designs can be further improved by adjusting the sample points to optimise some criterion of interest. One example of this is known as *maximin LHS*. Since points are randomly sampled within each region of the LHS grid, it may occur that points are clustered, reducing the efficiency of the design. Maximin designs seek to maximise the minimum distance between sample points, therefore providing an LHS design that has improved space-filling properties. This may not be suitable for direct Monte Carlo sampling, but can be very useful for a class of methods known as emulator-based approaches — see Section 2.4.

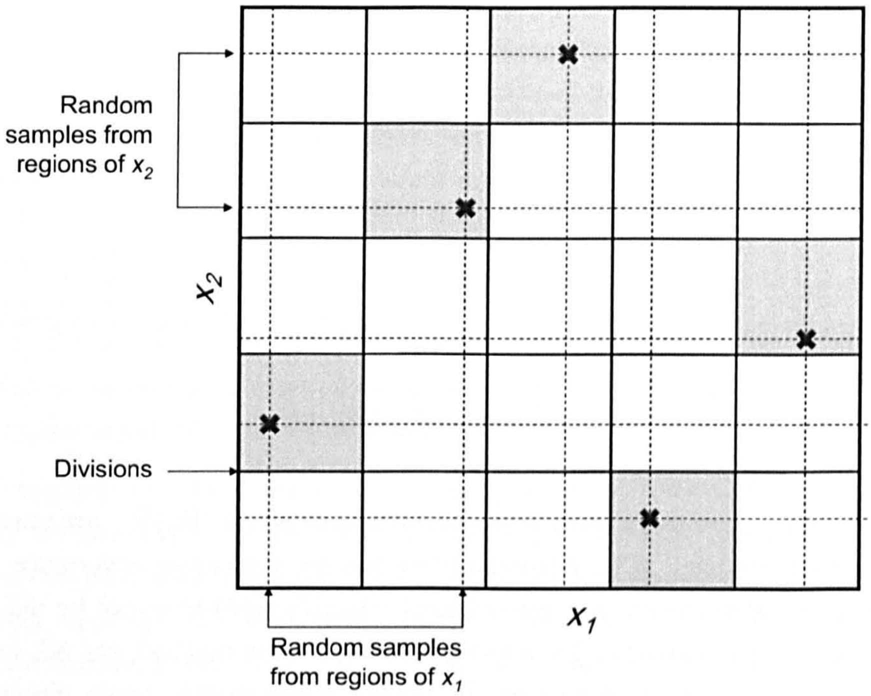

Random samples from regions of x_2
x_2
Divisions
x_1
Random samples from regions of x_1

Figure 2.9: 5-sample LHS over 2 input variables. Samples are shown as crosses.

2.3.2 The Stochastic Finite Element Method

It is worth briefly exploring an alternative approach to propagating uncertainty specifically through FE models, known as the Stochastic Finite Element Method (SFEM). Rather than viewing the model as a black box, SFEM proposes a specific method for calculating the response of FE models with uncertain element properties and loading conditions. In this method material properties (for example) may be represented as a *stochastic field* that varies spatially over the mesh, therefore allowing non-homogeneous properties of a structure. SFEM is performed in two steps: the first is to quantify the random inputs to the model — this requires the representation of a stochastic process or field. The second is to propagate these through the model to provide a realisation of the random output. These will be dealt with separately here. A good review of the subject can be found in [46].

Representing Stochastic Processes (SFEM)

Typically when quantifying input uncertainty in SFEM, an assumption is made that random quantities are distributed joint-normally. Two main approaches exist for quantifying a Gaussian stochastic field in this context — the first is called the spectral representation method and the second is known as the Karhunen-Loève expansion. A thorough review and comparison of these approaches is found in [47].

The spectral representation method expresses a stochastic field $\eta(x)$ (where x represents a spatial measure, such as a mesh location) as a sum of trigonometric functions (much like the Fourier series), but with random phase angles and amplitudes. As the number of terms approaches infinity, this representation tends to a Gaussian stochastic field. It can also be used for multi-dimensional variables.

The Karhunen-Loève (KL) expansion is similar, but uses a sum of orthogonal functions with random coefficients to represent the stochastic process. This can be expressed as,

$$\eta(x) = \eta^{(0)}(x) + \sum_{j=1}^{m} \theta_j \eta^{(j)}(x) \tag{2.26}$$

where the $\{\eta^{(j)}(x)\}_{j=0}^{m}$ are pairwise orthogonal functions and the $\{\theta_j\}_{j=1}^{m}$ are uncorrelated Gaussian random variables. The KL expansion has an associated covariance measure between points in the stochastic process (elements, in this case) that can be defined as a function of distance. Compared to the spectral representation method, the KL expansion is thought to be more accurate for lower numbers of terms under certain circumstances [46], but it is computationally less efficient and may not produce fully homogeneous sample functions when required [47]. More on the KL expansion is found in [48].

A further method that can also represent *non-Gaussian* and non-stationary stochastic processes is known as Polynomial Chaos Expansion (PCE), which was first introduced by Wiener in 1938 [49] and later reintroduced by Ghanem and Spanos [48]. Like the KL expansion, PCE uses an infinite linear combination of orthogonal functions. In this case however, the functions are orthogonal polynomials which are defined by the property that for any series of polynomials $\mathcal{T} = \{T_i(x), i \in \mathbb{N}\}$, the integral of the product of any pair T_i, T_j and a weight function $w(x)$ (equivalent to a version of the inner product) over some *interval of orthogonality* $[\underline{x}, \overline{x}]$ is zero, i.e.

$$\int_{\underline{x}}^{\overline{x}} T_i(x)T_j(x)w(x)dx = 0 \quad \forall i \neq j \tag{2.27}$$

Many types of orthogonal polynomials exist, of which a great many are discussed in detail in [50]. The *Legendre* polynomials are one example of a series of orthogonal polynomials that have been used in the context of UA and SA (see [51]), which are orthogonal with respect to the uniform distribution over $[-1, 1]$. Perhaps more importantly, the *Hermite* polynomials are orthogonal with respect to the Gaussian distribution, therefore the Hermite or Legendre polynomials are typically chosen for use in PCE depending on the distribution of interest. Some limitations of PCE have been noted however: PCE approximation is not always improved as the number of terms is increased [52], and the method can be computationally demanding due to the large number of coefficients that need to be estimated, particularly in problems involving strong nonlinearities and bifurcations [52, 53].

Propagating Uncertainty (SFEM)

Propagating uncertainty in the SFEM framework is achieved in one of three ways. The first is known as the perturbation approach, which uses a Taylor series expansion of the response vector. The second uses PCE to quantify the output; in this case this is the method of spectral stochastic finite elements (SSFEM) [48]. The Monte Carlo method may also be used, but is generally used as a validation tool for small test models due to the computational expense required.

SFEM requires that the random field describing input uncertainty be discretised into a random vector. Typically this discretisation is performed at element centroids or nodes, but it is computationally less demanding to specify an independent "stochastic mesh" that is coarser than the FE mesh, which requires less random variables. Aspects of this are discussed further in [46]. Once the random field has been defined at discrete points, it is possible to assemble stochastic matrices for each element, and ultimately for the whole model.

To solve the stochastic matrices to find (for example) nodal displacements, the simplest method is to use Monte Carlo sampling, where the stochastic matrix is sampled many times and statistical properties of output quantities are evaluated in the usual way. The Monte Carlo approach is computationally expensive but it is thought that Monte Carlo methods may be the only way of dealing with truly complex models in the SFEM approach [46].

An alternative approach is to use the perturbation method. In this approach, the stochastic field is discretised into a number of zero-mean random variables, and the stochastic system matrix (stiffness matrix) is represented as a Taylor series expansion, consisting of derivatives of the system matrix and random variable coefficients. Calculating displacements based on this stochastic matrix results in a further Taylor series expansion of the displacement vector, i.e. the stochastic output. The means and (co)variances of the displacements can be calculated using first- and second-order approximations. However since the derivatives of the system matrix must be calculated, this method can be computationally demanding when many random variables are involved [46].

The SSFEM method, in contrast, uses a KL expansion to express the stochastic spatial variation of a random input parameter, for example the elastic modulus (see e.g. [48]). Assuming deterministic loading, a PCE is used to approximate the displacement response, which may be calculated using a Galerkin approach [48].

Comments on SFEM

SFEM and SSFEM has been shown to be successful, with attractive advantages such as the ability to deal with complex specifications of input uncertainty, such as time- and spatially-varying input parameters, as well as the ability to simultaneously evaluate the uncertainty in all outputs of the FE model. However, the method has a number of limitations. First, it is applicable only to linear systems in general, and requires a considerable amount of work and modification to FE algorithms — it would therefore be difficult to implement on models built in commercial FE packages. Furthermore it is only applicable to FE, and has a computational cost that scales poorly with the number of degrees of freedom. In contrast, the methods presented in this thesis in Chapters 3 and 8 will be shown to be capable of dealing with very large models (not only FE models), which would be outside the reach of SFEM. In conclusion, SFEM is a powerful method, but seems to have some drawbacks concerning its applicability to large models.

2.3.3 Random Matrices

A further approach that will be very briefly discussed is that of random matrices, developed largely by Christian Soize [54]. In this approach, uncertainties due to structural and loading problems are considered, as well as uncertainties due to mathematical assumptions made in the construction of the model. The random structural matrix of the model is described by a "dispersion parameter" that controls the random properties of the matrix, which can be calculated by comparing experimental data with the model run at nominal parameter values. This method has seen little application so far, and again is applicable only for structural models. It has also only been applied to very simple problems. One issue is that the output of the mean model is based on arbitrary assumptions, which calls into question the meaning of the dispersion parameter. It is clear that this method requires more investigation — a summary of recent developments is given in [55]. Some very recent developments are however given in [56, 57] in which the method is demonstrated to be applicable to larger models.

2.3.4 Summary of Propagation Methods

Despite the development of efficient sampling strategies, direct Monte Carlo methods can still be computationally prohibitive. It is also evident that spectral methods and random matrix approaches are promising but only applicable to specific problems, and have other limitations, often requiring direct intervention in the FE code. For this reason, an additional class of methods exists where *emulators* or *metamodels* are built from a small number of samples based on a DOE, then propagation is performed on the (computationally much cheaper) emulator, rather than the model itself. These data modelling approaches do not require any knowledge of the model itself; only a set of model runs to use as training data. A full discussion of this problem is left to Section 2.4.

2.4 Emulators for Uncertainty Analysis

A recurring problem that has been mentioned is that, since both UA and SA involve exploration of the input space of a large model, the computational cost can often become prohibitive. The problem of estimating output pdfs in UA by Monte Carlo propagation has been discussed, and the difficulty is compounded in the case of SA. Consider for example the task of computing the MEI of a single input variable: a conditional expected value must be calculated,

$$E(Y|x_i) = \int_{\mathcal{X}_{-i}} f(\boldsymbol{x}) p_{-i|i}(\boldsymbol{x}_{-i}|x_i) d\boldsymbol{x}_{-i} \tag{2.28}$$

where the subscript $-i$ refers to the complement of i — e.g. $\mathcal{X}_{-i}$ is the range of input space in all dimensions except i. In the majority of cases this must be performed numerically since $f(\boldsymbol{x})$ is rarely tractable. Numerical Monte Carlo integration, for example, requires sampling the function and marginal pdf a large number of times in order to accurately evaluate the integral. This must be repeated for every new sensitivity measure. It is easy to see how the computational cost of numerical integration for every sensitivity measure can become prohibitive, even for relatively small models.

2.4.1 About Emulators

One solution to the problem of computational expense is to replace the model with an *emulator* (also known variously as a *metamodel*, *surrogate model* or *response surface*) that accurately imitates the response of the model over the range of input space. A simple example of an emulator might be a polynomial function. If the emulator is much cheaper to run than the original model (but still gives sufficiently similar results) the measures discussed previously can be estimated at a greatly reduced computational cost using Monte Carlo sampling of the emulator. Even better than that, if the emulator is also mathematically tractable, the integrals representing means, variances and sensitivity indices can be evaluated *analytically*, thus bypassing numerical integration completely. A restriction with many emulators is that they necessarily consider only univariate outputs, although of course a typical model will have a large number of outputs. To examine multiple outputs, typically a separate emulator is constructed for each output. For a very good book on emulators and data modelling, the reader is referred to [58].

Building an emulator is essentially an exercise in data modelling or regresssion. First of all, a metamodelling approach must be proposed that is deemed to be suitable for emulating the original model. The original model is then sampled at n points in the input space according to a sampling strategy (LHS is often used for emulator construction — see Section 2.1.3), giving a set of *training data* $\mathcal{D}$ which consists of n input-output pairs, i.e.

$$\mathcal{D} = \{(y_1, x_1), (y_2, x_2), ..., (y_n, x_n)\} \tag{2.29}$$

which can also be expressed as the matrix of training data inputs $\boldsymbol{X}$ and the corresponding vector of ouputs $\mathbf{y}$,

$$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n) \tag{2.30}$$

$$\mathbf{y} = (y_1, y_2, ..., y_n)^T \tag{2.31}$$

Typically an emulator will be defined by a set of tunable parameters or higher-level *hyperparameters.* The training data is therefore used to train the model by adjusting these parameters so that the model fits the data as well as possible. There are many methods of achieving this, usually dependent on the type of model under consideration.

2.4.2 Types of Emulator

Potentially any form of regression or machine learning can be used to emulate a model. However there are a number of desirable emulator properties that may influence the choice. Three of the key properties may be summarised as,

- Generality
- Efficiency
- Tractability

The first of these specifies that an emulator should be as general, flexible or applicable as possible. Consider for example a simple linear regression (e.g. [59]): if the model is linear this is likely to provide a good fit once the regression parameters are tuned to the training data. However, if the model is strongly nonlinear, the regression model will not be able to fit the data to an acceptable level of accuracy no matter what values of parameters are chosen. Since in the majority of cases little or no prior knowledge is available concerning the nature of the model response, the emulator should be able to model as wide a class of problems as possible.

Secondly, the efficiency of the emulator is important since it dictates the amount of training data that is required for a specific problem. Clearly for a large model of considerable run-time, the less training data the better. Last of all, with particular applicability to UA/SA, is the tractability of the emulator. As discussed before, this may allow analytical estimates of uncertainty and sensitivity, giving significant computational savings.

Parametric and Nonparametric Emulators

The choice of emulator is a field of very active research in the data modelling community, and is also closely linked to optimisation. Broadly speaking, they can be divided into two categories: parametric emulators, and semi- or non-parametric emulators. Parametric emulators make an *a priori* assumption about the order or type of model to be fitted to the data, and the corresponding emulator coefficients (parameters) are chosen to provide an optimum fit to the data — this is often done by least-squares analysis (see for example [35]). Although computationally simple, parametric emulators are considered to be quite inflexible due to the fact that an assumption must be made about the order of the function to be fitted, for example the order of a polynomial function — some comparisons of polynomials with other methods are given in [60, 61]. This can lead both to a model that does not fit the data at all, as well as to *over-fitting*, which is when the model is too complicated to describe the underlying data-generating mechanism and introduces fluctuations in the response that should not exist.

In contrast, non-parametric emulators use a higher class of governing *hyperparameters* that can represent an infinite number of parameters in a parametric model, thus removing the necessity for assumptions about the model order. Semi-parametric models are a combination of the two; for example, a parametric model with residuals modelled by a non-parametric approach. Given the limitations of parametric emulators, the majority of recent emulator-based UA and SA literature has focused on the use of semi- and non-parametric emulators.

Artificial Neural Networks

One well-established example of a non-parametric emulator is an *Artificial Neural Network* (ANN). An ANN is an arrangement (similar, but on a much reduced scale, to the design of the human brain) of simple processing units called *neurons* that are inter-connected by *synapses.* The ANN takes an input vector $\boldsymbol{x}$ and maps it to an output y by propagating the values of the various x_i through layers of neurons. Layers apart from the output and input layer are called *hidden layers* (see Figure 2.10). Each neuron consists of a summing junction that adds the weighted values of its inputs and produces an output y_n that is is some function $g(\cdot)$ of this sum, i.e.

$$y_n = g\left(\sum_{i=1}^{N} w_i x_i\right) \tag{2.32}$$

where the w_i are weights to each of the N inputs to the neuron (not necessarily equal to the number of inputs to the ANN). The function g may be a threshold function that gives an output only when the weighted sum of the inputs exceeds a certain value, or may be a continuous function of the sum, such as a hyperbolic tangent function. The weights are the parameters of the emulator, which can be optimised to a set of training data by a number of different approaches (see [62]).

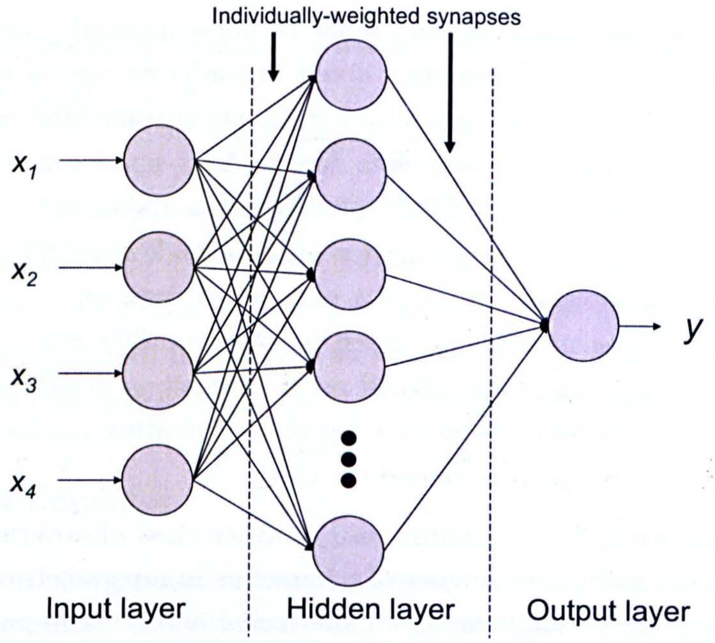


Figure 2.10: An example of an ANN with four inputs, one output and one hidden layer. The circles represent neurons and the arrows represent synapses.

Once the weights are correctly optimised, the ANN is capable of emulating a wide class of models, so long as the number of layers of neurons is sufficiently large. ANNs have been applied to calculate first-order sensitivity measures by analytically constructing partial derivatives of the output with respect to each input [63]. This approach has been extended to calculate global variance-based sensitivity measures [64], although in this case the sensitivity indices were necessarily evaluated by Monte Carlo methods applied to the trained ANN. This intractability could be seen as a drawback to ANNs. Furthermore, ANNs require specification of their structure, might introduce a certain loss of generality.

Gaussian Process Regression

One recent approach to emulation that has seen enormous interest in recent years is the use of *Gaussian processes* (GPs) (see [65]). In this method, the values of model outputs are assumed to be distributed in a multivariate Gaussian probability distribution, so that predictions may be made by conditioning the prior distribution of an unknown point on the set of training data. Inferences about the model at unobserved points return estimates that are Gaussian distributions, rather than crisp points. GPs are very well-suited to emulating computer models since they are very computationally efficient, semi-

or non-parametric (depending on prior assumptions) and can provide analytical estimates of sensitivity measures. Furthermore, very few specifications need be made about the response of the data — certainly fewer than the methods presented above. For these reasons, GP emulators will be used extensively in this thesis. A full description of GPs applied to UA/SA follows in Chapter 3, with an extension in Chapter 8 that increases the generality of GPs still further.

2.5 Conclusions

The discipline of uncertainty analysis has been outlined here, showing that it can be divided into three problems: those of quantification, fusion and propagation. The methods of quantification are various, but this thesis will focus on the probabilistic methods, since they are by far the most well-established and are widely applicable. The elicitation of probability distributions is recognised as a difficulty here but this problem is a subject on its own and is outside the scope of this thesis. Instead, it is the propagation problem that will be the focus of the remaining chapters. A furtherance of uncertainty analysis, known as sensitivity analysis, has also been introduced as a way of quantifying each input parameter's contribution to the output uncertainty — this may then be used as a starting point in reducing uncertainty and a method to achieve a deeper understanding of the model itself. Sensitivity analysis is intrinsically linked with uncertainty analysis and so will feature at length in this work.

It has been shown that the Monte Carlo method is the ideal method for propagating uncertainty, but that it is computationally very expensive when estimating detailed statistical quantities for large models, which are increasingly common in engineering. As such, there is a need for propagation methods that reduce this computational burden. This chapter has described some emerging methods such as the spectral stochastic FE method and random matrix theory that can be used to propagate probabilistic uncertainty through structural models, but it is noted that they generally require direct intervention in the FE code, and have some limitations when dealing with very large models. A different angle of attack is to use a data-modelling approach, which uses a small number of model runs to build an *emulator* of the model that is much cheaper to run, then propagates uncertainty through the emulator, either by Monte Carlo or analytical methods. This emulator-based approach requires no knowledge of the model, therefore it is applicable to any kind of model and able to deal with very large models as long as sufficient training data can be obtained. However, there has been shown to be a need for emulators that are as widely-applicable and efficient as possible, since basic emulators may fail to effectively capture the response of the model, or require an infeasibly-large number of training data.

A brief discussion of emulators has been provided, and a recent method known as Gaussian process regression has been introduced as an emulator that is efficient and analytically tractable, with the ability to emulate a wide class of computer models. This approach will be shown to be capable of analytically estimating uncertainty and sensitivity measures, therefore it provides an informative and widely-applicable method of propagating uncertainty that is perhaps out of the reach of other methods discussed here. As such, this method will be thoroughly investigated in Chapters 3 and applied in Chapters 5 and 7. Finally, extensions to the method are discussed in Chapters 8 and 9.

Chapter 3

Gaussian Processes and Bayesian Sensitivity Analysis

It has been explained previously that an inherent problem in UA/SA is the difficulty of performing a comprehensive global analysis without increasing the computational cost to an unacceptable level. Emulator-based approaches can alleviate the computational burden significantly, offering (at least partially) a solution to this problem. However, a key problem is the selection of the emulator itself. Simple emulators such as polynomials require an assumption about the order of the model when usually little or nothing is known about the response of the model in advance. The key to successful emulation is then to choose an emulator that is as general as possible, and assumes as little as possible about the function to be modelled. An additional requirement is that the emulator be able to accurately mimic the true model for as few training data points as possible.

An approach proposed by Oakley and O'Hagan [41] considers the emulation problem from a Bayesian perspective. A Gaussian Process (GP), which is a probabilistic method of regression, is used as the emulator. GPs are well-suited to emulation since they are semi- or non-parameteric (they are not specified by parameters, but rather by more general *hyperparameters*, see later), which means that they are sufficiently general to model a wide class of problems. Additionally, under certain circumstances, they are tractable enough to infer sensitivity measures analytically. This means that the GP (emulator) can be constructed from a set of runs, then used to estimate sensitivity indices, main effects, means and variances without the necessity for additional runs (even of the emulator itself). As will be seen, this involves the construction of some sizeable integrals, but once this framework is built it can be re-applied to emulate any model, so long as the assumptions about the model still hold. The GP-based SA will be shown in this chapter to provide an efficient and flexible method for detailed UA and SA of the complex nonlinear models that are found in engineering and elsewhere. The method is applied to several models in Chapters 5 and 7.

An explanation of GPs in the context of regression follows in Section 3.1, including a detailed description of how to train the GP by first conditioning on a set of training data and then analytically marginalising hyperparameters. Next, the inference of various sensitivity measures is discussed in Section 3.2, where the necessary integrals are derived for an assumed joint pdf, giving a complete description of the steps involved in this form of Bayesian SA. A more practical guide to implementing Bayesian UA/SA with a partial example is also provided in Appendix B.

3.1 Gaussian Processes

The concept of the Gaussian Process has been around for many years, but was first proposed to be used in the context of emulation of computer codes by Sacks *et al.* [65] in 1989. Since then, the power of GPs has been exploited to model many varied types of computer code — Kennedy *et al.* provide a summary of recent applications [66]. Most recently, the attraction of GPs as a sophisticated tractable emulator has been exploited in the context of UA and SA by Oakley and O'Hagan [41]. It is this approach that will be outlined in some detail here. For a very comprehensive treatment of GPs in a wider context see Rasumussen [67].

A GP can be thought of as a *distribution over functions*, i.e. the random variable of the distribution is a function rather than a single number or fixed-length vector. Rather than returning a crisp output value $f(\boldsymbol{x})$ for any given input point $\boldsymbol{x}$ (as in a standard regression), the GP returns a specification for a Gaussian probability distribution. This means that for any given set of input points, the corresponding output values are distributed joint-normally. Importantly, this applies to combinations of known points and unknown points, which provides the mechanism for training the GP — unobserved points can be *conditioned* on known points (see Section 3.1.2). The fact that any output has an associated variance means that predictions made using a GP automatically have a mean and associated confidence limits, making them particularly suitable for UA applications; furthermore, the quality of the emulator fit can be easily assessed. An example of a fitted GP is shown in Figure 3.1 for illustration. Observe that the confidence limits are naturally much smaller in the vicinity of training data, while further away (and particularly in areas of extrapolation) they become significantly wider.

One of the key assumptions made by the GP is that the model is a smooth function of its inputs. A high degree of smoothness implies that if $f(\boldsymbol{x})$ is known at $\boldsymbol{x}$, then $f(\boldsymbol{x}')$ can be predicted with a high degree of confidence if $\boldsymbol{x}'$ is close to $\boldsymbol{x}$. The specification of smoothness is the mechanism that allows a GP emulator to accurately mimic a complex model for a greatly reduced computational cost *so long as the assumption of smoothness is valid* (the problem of when this assumption no longer holds is addressed in Chapter 8).

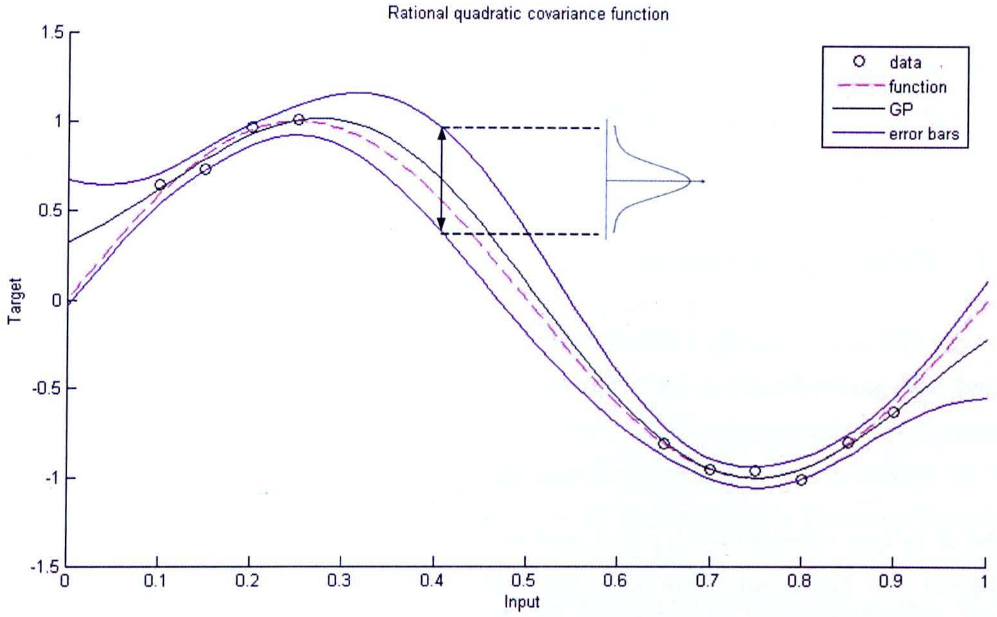


Figure 3.1: A one-dimensional example of a GP fitted to training data. Confidence limits and the true function are shown.

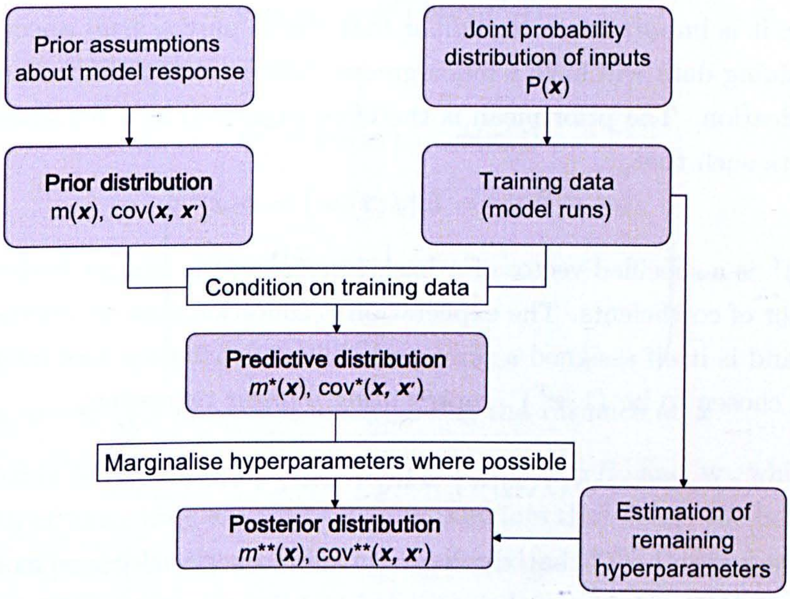

Prior assumptions about model response
Joint probability distribution of inputs P(x)
Prior distribution m(x), cov(x, x′)
Training data (model runs)
Condition on training data
Predictive distribution m*(x), cov*(x, x′)
Marginalise hyperparameters where possible
Posterior distribution m**(x), cov**(x, x′)
Estimation of remaining hyperparameters

Figure 3.2: The process of fitting a GP emulator to a set of training data

The GP is trained according to the principles of Bayesian inference, which means that prior assumptions are made about the model to give a *prior distribution* (Section 3.1.1), then a set of training data is used to update the prior assumptions, resulting in a *predictive distribution* (Section 3.1.2. The training is completed by marginalising hyperparameters in Section 3.1.3, yielding the *posterior distribution.* This process is illustrated in Figure 3.2.

3.1.1 Prior Specification

Since the GP must specify a mean and variance for any given input point, it is completely defined by a mean function $m(\boldsymbol{x})$ and a scaled covariance function $c(\boldsymbol{x}, \boldsymbol{x}')$, i.e. for a single point $\boldsymbol{x}$,

$$f(\boldsymbol{x})|\sigma^2, \mathbf{w} \sim \mathcal{GP}\left(m(\boldsymbol{x}), \sigma^2 c(\boldsymbol{x}, \boldsymbol{x}')\right) \tag{3.1}$$

where $\boldsymbol{x}'$ is any neighbouring point and σ^2 is a scaling factor (the prior variance of data). Following the Bayesian paradigm, any set of model outputs are assigned a joint prior distribution, defined by a prior mean function and prior covariance function. These will be subsequently be conditioned on the training data to provide a predictive distribution. Starting with the mean function then, the most simple assumption would be to assume that the prior mean is zero for any value of $\boldsymbol{x}$, in the absence of any other knowledge. Perhaps a better assumption however, is to guess that the data is approximately linear in $\boldsymbol{x}$. In the experience of the author, this is quite a reasonable assumption since many engineering models display roughly linear behaviour with respect to at least some of the model inputs. In any case, it is important to remember that this is only a *prior* specification of beliefs, and the training data will have a much greater influence over the final emulator than the prior specification. The prior mean is therefore expressed as a regression fit through the training data such that,

$$m(\boldsymbol{x}) = E\{f(\boldsymbol{x})|\mathbf{w}\} = \phi(\boldsymbol{x})^T\mathbf{w} \tag{3.2}$$

where $\phi(\boldsymbol{x})^T$ is a specified vector of q basis functions of $\boldsymbol{x}$, and $\mathbf{w}$ is the corresponding q-length vector of coefficients. The expectation is conditional on $\mathbf{w}$ because $\mathbf{w}$ is treated as unknown, and is itself assigned a prior probability distribution (see later). For simplicity, $\phi(\boldsymbol{x})^T$ was chosen to be $(1, \boldsymbol{x}^T)$, representing a linear regression,

$$E\{f(\boldsymbol{x})|\mathbf{w}\} = w_0 + w_1x_1 + \ldots + w_dx_d \tag{3.3}$$

This represents the belief that the data is close to a global linear model with residuals modelled by a GP. $\phi(\boldsymbol{x})^T$ can be extended to other functions if there is information to suggest that this is appropriate. The covariance between outputs is specified as a function of any two input points such that,

$$\operatorname{cov}\{f(\boldsymbol{x}), f(\boldsymbol{x}')|\sigma^2\} = \sigma^2 c(\boldsymbol{x}, \boldsymbol{x}') \tag{3.4}$$

where $c(\boldsymbol{x}, \boldsymbol{x}')$ is a covariance function that decreases as $|\boldsymbol{x} - \boldsymbol{x}'|$ increases, and satisfies $c(\boldsymbol{x}, \boldsymbol{x}) = 1$ for all $\boldsymbol{x}$.

The covariance function is the most important part of the GP: although the prior mean can potentially be set to zero (and is often used in this manner), the covariance function is the source of the GP's predictive power. It specifies the smoothness of a function by defining the covariance of output points as a function of the distance between their corresponding inputs. Here a squared-exponential function is used, of the form,

$$c(\boldsymbol{x}, \boldsymbol{x}') = \exp\{-(\boldsymbol{x} - \boldsymbol{x}')^T B(\boldsymbol{x} - \boldsymbol{x}')\} \tag{3.5}$$

where B is a $d \times d$ diagonal matrix of inverse length-scales, representing the roughness of the output with respect to each input parameter, which is treated as "known" since it cannot be dealt with analytically. Estimation of B is however addressed in Section 3.1.4. That B is diagonal requires an assumption of independence between inputs. The squared-exponential covariance function here is used since it is one of the most simple formulations and results in tractable integrals for estimation of SA measures. However, the GP is by no means limited to this form and there exist many other possible covariance functions with a variety of properties, a discussion of which is provided in [67]. With the aid of numerical integration these could also be used in the GP emulator for SA.

Equations 3.2, 3.4 and 3.5 therefore imply a prior mean for any vector of outputs $\mathbf{y} = (f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), ...f(\boldsymbol{x}_n))$ as follows,

$$\mathbf{y}|\mathrm{w}, \sigma^2 \sim \mathcal{N}\left(\Phi\mathrm{w}, \sigma^2 A\right) \tag{3.6}$$

where,

$$\Phi^T = (\phi(\boldsymbol{x}_1), ..., \phi(\boldsymbol{x}_n))$$

$$A = \begin{pmatrix} 1 & c(\boldsymbol{x}_1, \boldsymbol{x}_2) & \dots & c(\boldsymbol{x}_1, \boldsymbol{x}_n) \\ c(\boldsymbol{x}_2, \boldsymbol{x}_1) & 1 & & \vdots \\ \vdots & & \ddots & \\ c(\boldsymbol{x}_n, \boldsymbol{x}_1) & \cdots & & 1 \end{pmatrix}$$

Note that when $\boldsymbol{x} = \boldsymbol{x}'$, 3.5 reduces to unity, giving the variance at $\boldsymbol{x}$.

Note that Equation 3.6 is conditional on the quantities σ^2, B, and $\mathbf{w}$, which are known as *hyperparameters*, since they are controlling parameters that define the behaviour of the model (a much fuller explanation of this can be found in [67]). These hyperparameters must be estimated from the training data, and are instrumental in avoiding pitfalls such as over-fitting. Figure 3.3 shows an example of the result of incorrectly fitting the hyperparameters defining the smoothness of the function. Although by definition the GP must pass through all training data points whatever the value of the hyperparameters (assuming no noise on the training data), when the length-scales are incorrectly defined ($l = 0.7$)

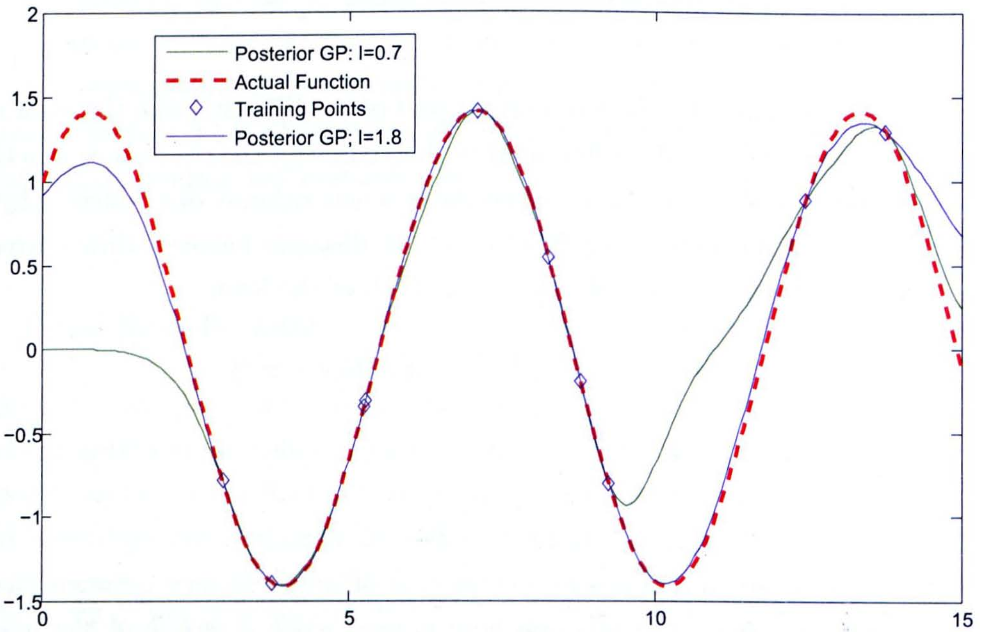


Figure 3.3: GPs of varying length scale (l) conditioned on training data from a sine wave.

the GP does a very bad job of fitting the sine wave since the covariance between points is too large. The problem of estimating the hyperparameters is dealt with in Section 3.1.3.

At this point, the hyperparameters must be assigned prior distributions to complete the prior specification of the GP. σ^2 and $\mathbf{w}$ are assigned *weak priors* (otherwise known as noninformative priors — see for example [10] for further information), which do not give any weight to any range of values over any other. This is done when there is no prior information whatsoever about the hyperparameters, although other prior distributions can be, and are, used by Oakley and O'Hagan in [41]. The prior is expressed in an *improper* form, meaning that it is only described as a proportional relationship, i.e.

$$p(\mathbf{w}, \sigma^2) \propto \frac{1}{\sigma^2} \tag{3.7}$$

which implies an infinite prior variance on $\mathbf{w}$, and a diminishing probability with increasing σ^2. The other remaining hyperparameters in B will need a different treatment as explained shortly (Section 3.1.4). The prior distribution of the function and its hyperparameters is now complete — the next step is to condition the GP on the training data.

3.1.2 Conditioning on the Training Data

In order to make in an informed prediction about some unknown function value $f(\boldsymbol{x}^\star)$ at a given input point $\boldsymbol{x}^\star$, the prior distribution of $f(\boldsymbol{x}^\star)$ must be conditioned on the training data $\mathbf{y}$. Since it is defined by the GP that all data (including the as-yet unknown point $f(\boldsymbol{x}^\star)$) are distributed joint-normally, the joint normal distribution can be expressed by partitioning the mean vector $\boldsymbol{\mu}$ and covariance matrix Σ into parts representing the distribution of $f(\boldsymbol{x}^\star)$ and parts representing the distribution of $\mathbf{y}$, i.e.

$$\boldsymbol{\mu} = \begin{pmatrix} \phi(\boldsymbol{x}^\star)^T\mathbf{w} \\ \Phi\mathbf{w} \end{pmatrix} \tag{3.8}$$

$$\Sigma = \begin{pmatrix} \sigma^2 c(\boldsymbol{x}^\star, \boldsymbol{x}^\star) & \sigma^2 \mathbf{t}(\boldsymbol{x}^\star)^T \\ \sigma^2 \mathbf{t}(\boldsymbol{x}^\star) & \sigma^2 A \end{pmatrix} \tag{3.9}$$

where,

$$\mathbf{t}(\boldsymbol{x}^\star)^T = (c(\boldsymbol{x}^\star, \boldsymbol{x}_1), ..., c(\boldsymbol{x}^\star, \boldsymbol{x}_n)), \tag{3.10}$$

A lemma exists that states that, for a joint-Gaussian distribution, the marginal distribution of $f(\boldsymbol{x}^\star)$ may be conditioned on the marginal distribution of $\mathbf{y}$, giving the *predictive distribution* (the * notation here indicates a posterior distribution) by the following result:

$$p(f(\boldsymbol{x}^\star)|\mathrm{w}, \sigma^2, \mathcal{D}) \sim \mathcal{GP}\left(m^*(\boldsymbol{x}^\star), \sigma^2 c^*(\boldsymbol{x}^\star, \boldsymbol{x}')\right) \tag{3.11}$$

where,

$$m^*(\boldsymbol{x}^\star) = \phi(\boldsymbol{x}^\star)^T\mathbf{w} + \mathbf{t}(\boldsymbol{x}^\star)^T A^{-1}(\mathbf{y} - \Phi\mathbf{w}) \tag{3.12}$$

$$c^*(\boldsymbol{x}^\star, \boldsymbol{x}') = c(\boldsymbol{x}^\star, \boldsymbol{x}') - \mathbf{t}(\boldsymbol{x}^\star)^T A^{-1}\mathbf{t}(\boldsymbol{x}^\star) \tag{3.13}$$

Note that Equation (3.11) is still conditional on the three hyperparameters. Following the principles of Bayesian inference, it is possible to *marginalise* these parameters by integrating the predictive distribution with respect to each one — this is outlined in Section 3.1.3. In the case of $\mathbf{w}$ and σ^2, they can be analytically marginalised due to the prior distributions that have been assigned to them. B, on the other hand, cannot be analytically marginalised and so must be treated as known (i.e. it is not assigned a prior probability distribution). Instead, it is estimated by *maximum likelihood estimation* (MLE), which involves expressing the likelihood of parameters given the training data and optimising the parameters to maximise the probability of the data [68] (this must be done numerically in this case). It is also possible to numerically marginalise B using *Markov Chain Monte Carlo* (MCMC) estimation, but this is not considered here. It is however referred to in Chapter 8.

3.1.3 Marginalising Hyperparameters

Considering first the problem of marginalising w, the distribution that is required is $p(f(\mathbf{x}^{\star})|\sigma^2, \mathbf{y})$. This can be expressed as,

$$\begin{aligned} p(f(\cdot)|\sigma^2,\mathbf{y}) &= \int p(f(\cdot),\mathbf{w}|\sigma^2,\mathbf{y})d\mathbf{w} \\ &= \int p(f(\cdot)|\mathbf{w},\sigma^2,\mathbf{y})p(\mathbf{w}|\sigma^2,\mathbf{y})d\mathbf{w} \end{aligned} \tag{3.14}$$

Where $f(\cdot) = f(x^{\star})$. The first term in the integral is the predictive distribution, which has already been defined in Equation 3.11. The second term can be derived from the likelihood of the data, i.e.

$$\begin{aligned} p(\mathbf{y}|\mathbf{w},\sigma^2) &= \prod_{i=1}^{n} p(y_i|\mathbf{w},\sigma^2) \\ &= \prod_{i=1}^{n} p(f(\mathbf{x}_i)|\mathbf{w},\sigma^2) \\ &\sim \mathcal{N}(\Phi\mathbf{w},\sigma^2 A) \end{aligned} \tag{3.15}$$

In order to get $p(\mathbf{w}|\sigma^2,\mathbf{y})$ it is necessary to express this distribution in terms of w. The distribution can be expanded out, i.e.

$$\begin{aligned} \mathcal{N}(\Phi\mathbf{w},\sigma^2 A) &= (2\pi\sigma^2)^{\frac{-n}{2}} \exp\left\{\tfrac{-1}{2\sigma^2}(\mathbf{y}-\Phi\mathbf{w})^T A^{-1}(\mathbf{y}-\Phi\mathbf{w})\right\} \\ &= (2\pi\sigma^2)^{\frac{-n}{2}} \exp\left\{\tfrac{-1}{2\sigma^2}\left(\mathbf{y}^T A^{-1}\mathbf{y} - 2\mathbf{y}^T A^{-1}\Phi\mathbf{w} + \mathbf{w}^T\Phi A^{-1}\Phi\mathbf{w}\right)\right\} \end{aligned} \tag{3.16}$$

Through some judicious rearrangement it is possible to extract a normal distribution over $\mathbf{w}$ and an inverse-gamma distribution over σ^2, therefore splitting the distribution into two distributions which can be used separately for marginalising each hyperparameter. The aim is therefore to express the exponent in 3.16 in the form,

$$\begin{aligned} &\tfrac{1}{2\sigma^2}\left\{(\mathbf{w}-\hat{\mathbf{w}})^T\Sigma_w^{-1}(\mathbf{w}-\hat{\mathbf{w}}) + (n-q-2)\hat{\sigma}^2\right\} \\ = \;&\tfrac{1}{2\sigma^2}\left\{\hat{\mathbf{w}}^T\Sigma_w^{-1}\hat{\mathbf{w}} - 2\hat{\mathbf{w}}^T\Sigma_w^{-1}\mathbf{w} + \mathbf{w}^T\Sigma_w^{-1}\mathbf{w} + (n-q-2)\hat{\sigma}^2\right\} \end{aligned} \tag{3.17}$$

where the $(n-q-2)\hat{\sigma}^2$ has been inserted as the constant in the inverse-gamma distribution in a form that allows a convenient form after marginalisation (see Equation 3.39 later). Comparing coefficients of $\mathbf{w}(\cdot)\mathbf{w}^T$ between Equations 3.16 and 3.17 it is evident that the covariance of matrix of $\mathbf{w}$ is $\Sigma_w^{-1} = \Phi^T A^{-1}\Phi$. Further, if the coefficients of $\mathbf{w}$ are compared, the mean $\hat{\mathbf{w}}$ of $\mathbf{w}$ can be found,

$$\begin{aligned} \hat{\mathbf{w}}^T\Sigma_w^{-1} &= \mathbf{y}^T A^{-1}\Phi \\ \hat{\mathbf{w}}^T &= \mathbf{y}^T A^{-1}\Phi\Sigma_w = \mathbf{y}^T A^{-1}\Phi(\Phi^T A^{-1}\Phi)^{-1} \end{aligned} \tag{3.18}$$

This expression may now be used as a "best-estimate" of $\mathbf{w}$, since it represents the value of $\mathbf{w}$ for which the probability of the observed data is the highest, given that the mean of a Gaussian distribution is also its mode — this is therefore a *Maximum a Priori* (MAP) estimate (the equivalent of a MLE estimate, but with a prior distribution included). A prior distribution has been specified for $\mathbf{w}$; since the prior is uninformative, however, the MAP estimate gives the same result as the MLE equivalent. The distribution of weights may now be stated,

$$p(\mathbf{w}|\sigma^2, \mathbf{y}) \sim \mathcal{N}(\hat{\mathbf{w}}, \sigma^2\Sigma_w) \tag{3.19}$$

Now to find the estimate of σ^2 the constants are compared between Equations 3.16 and 3.17 to find that,

$$(n-q-2)\hat{\sigma}^2 + \hat{\mathbf{w}}^T\Phi^T A^{-1}\Phi\hat{\mathbf{w}} = \mathbf{y}^T A^{-1}\mathbf{y}$$

Substituting the expression for $\hat{\mathbf{w}}$ and rearranging gives,

$$\begin{aligned} (n-q-2)\hat{\sigma}^2 &= \mathbf{y}^T A^{-1}\mathbf{y} - \mathbf{y}^T A^{-1}\Phi(\Phi^T A^{-1}\Phi)^{-1}\Phi^T A^{-1}\Phi(\Phi^T A^{-1}\Phi)^{-1}\Phi^T A^{-1}\mathbf{y} \\ &= \mathbf{y}^T A^{-1}\mathbf{y} - \mathbf{y}^T A^{-1}\Phi(\Phi^T A^{-1}\Phi)^{-1}\Phi^T A^{-1}\mathbf{y} \end{aligned}$$

$$\hat{\sigma}^2 = \frac{1}{n-q-2}\left\{\mathbf{y}^T\left(A^{-1} - A^{-1}\Phi(\Phi^T A^{-1}\Phi)^{-1}\Phi^T A^{-1}\right)\mathbf{y}\right\} \tag{3.20}$$

Similar to 3.18, this provides the best estimate of σ^2.

Turning back now to the integral in Equation 3.14, the first term may be expressed as,

$$p(f(\cdot)|\mathbf{w}, \sigma^2, \mathbf{y}) \propto \exp\left\{\frac{-1}{2c^*\sigma^2}\left(f(\cdot) - \mathbf{k}_m^T\mathbf{w} + \alpha\right)^2\right\} \tag{3.21}$$

where $\mathbf{k}_m^T = \phi(\boldsymbol{x})^T - \mathbf{t}(\boldsymbol{x})^T A^{-1}\Phi$ and $\alpha = \mathbf{t}(\boldsymbol{x})^T A^{-1}\mathbf{y}$. $c^*(\boldsymbol{x}^\star, \boldsymbol{x}')$ is denoted here as c^* for convenience. The second term is,

$$p(\mathbf{w}|\sigma^2, \mathbf{y}) \propto \exp\left\{\frac{1}{2\sigma^2}(\mathbf{w} - \hat{\mathbf{w}})^T\Sigma_w^{-1}(\mathbf{w} - \hat{\mathbf{w}})\right\} \tag{3.22}$$

Therefore (denoting the distribution in 3.14 as p_f) the combined result is,

$$p_f \propto \int \exp\left\{\frac{-1}{2c^*\sigma^2}\left(\left(f(\cdot) - \mathbf{k}_m^T\mathbf{w} + \alpha\right)^2 + (\mathbf{w} - \hat{\mathbf{w}})^T c^*\Sigma_w^{-1}(\mathbf{w} - \hat{\mathbf{w}})\right)\right\} d\mathbf{w} \tag{3.23}$$

Note that the aim here is to find the posterior distribution of $f(\cdot)$, which will be Gaussian. Given that this distribution must necessarily integrate to unity, the normalising constant can be evaluated at the end of the integration, and constants (terms that are not a function of $f(\cdot)$) that appear throughout the calculation can be ignored.

In order to integrate 3.23 it is necessary to expand the terms in the exponent and to complete the square. Skipping some rearrangement, this becomes a quadratic in $\mathbf{w}$,

$$p_f \propto \int \exp\left\{\frac{-1}{2c^*\sigma^2}\left(\mathbf{w}^T P\mathbf{w} + 2\mathbf{q}^T\mathbf{w} + r\right)\right\} d\mathbf{w} \tag{3.24}$$

where,

$$\begin{aligned} P &= K + c^*\Sigma_w^{-1} \\ \mathbf{q}^T &= (\alpha - f(\cdot))\,\mathbf{k}_m^T - \hat{\mathbf{w}}^T c^*\Sigma_w^{-1} \\ r &= (f(\cdot) - \alpha)^2 + \hat{\mathbf{w}}^T c^*\Sigma_w^{-1}\hat{\mathbf{w}} \\ K &= \mathbf{k}_m\mathbf{k}_m^T \end{aligned} \tag{3.25}$$

Completing the square gives,

$$p_f \propto \int \exp\left\{\frac{-1}{2c^*\sigma^2}\left(\mathbf{w} - \mathbf{a}\right)^T P\left(\mathbf{w} - \mathbf{a}\right) + b\right\} d\mathbf{w} \tag{3.26}$$

$$\begin{aligned} \mathbf{a} &= P^{-1}\mathbf{q} \\ b &= r - \mathbf{q}^T P^{-1}\mathbf{q} \end{aligned}$$

The integral therefore becomes,

$$p_f \propto \exp\left\{\frac{-1}{2c^*\sigma^2}\left(r - \mathbf{q}^T P^{-1}\mathbf{q}\right)\right\} \int \exp\left\{\frac{-1}{2c^*\sigma^2}\left(\mathbf{w} - \mathbf{a}\right)^T P\left(\mathbf{w} - \mathbf{a}\right)\right\} d\mathbf{w} \tag{3.27}$$

since b is not a function of $\mathbf{w}$, and can be taken out of the integral. It is useful at this point to carefully consider the integral in 3.27. This is the integral of a d-dimensional Gaussian function which has P, $\mathbf{a}$ and $c^*\sigma^2$ as parameters. Of these, only $\mathbf{a}$ is a function of $f(\cdot)$, which is the variable of interest. The Gaussian integral in Equation 3.27 can be solved by a standard result I_G,

$$I_G = \sqrt{\frac{2\pi^d}{\left|\frac{P}{c^*\sigma^2}\right|}} \tag{3.28}$$

which notably is *not* dependent on the $\mathbf{a}$ term (and hence not on $f(\cdot)$). This means that the entire integral can be discarded as a proportional constant, leaving only the b term. The remaining terms are restated here explicitly in their original form for clarity,

$$p_f \propto \exp\left\{\frac{-1}{2c^*\sigma^2}\left(r - \mathbf{q}^T P^{-1}\mathbf{q}\right)\right\} \tag{3.29}$$

$$\begin{aligned} P &= K + c^*\Sigma_w^{-1} \\ \mathbf{q}^T &= \left(\mathbf{t}(\cdot)^T A^{-1}\mathbf{y} - f(\cdot)\right)\mathbf{k}_m^T - \hat{\mathbf{w}}^T c^*\Sigma_w^{-1} \\ r &= \left(f(\cdot) - \mathbf{t}(\cdot)^T A^{-1}\mathbf{y}\right)^2 + \hat{\mathbf{w}}^T c^*\Sigma_w^{-1}\hat{\mathbf{w}} \end{aligned}$$

Remembering that $f(\cdot)$ is the variable of interest, it is convenient to define $\varpi = f(\cdot) -$

$\mathbf{t}(\cdot)^T A^{-1}\mathbf{y}$, giving,

$$\begin{aligned} p_f &\propto \exp\left\{\tfrac{-1}{2c^*\sigma^2}\left(\varpi^2 - \left[-\mathbf{k}_m^T\varpi - \hat{\mathbf{w}}^T(P-K)\right]P^{-1}\left[-\mathbf{k}_m^T\varpi - \hat{\mathbf{w}}^T(P-K)\right]^T\right)\right\} \\ &\propto \exp\left\{\tfrac{-1}{2c^*\sigma^2}\left(\varpi^2 - \varpi^2\mathbf{k}_m^T P^{-1}\mathbf{k}_m^T + 2\varpi\mathbf{k}_m^T P^{-1}(P-K)\hat{\mathbf{w}}\right)\right\} \\ &\propto \exp\left\{\tfrac{-1}{2c^*\sigma^2}\left(\varpi^2(1-\mathbf{k}_m^T P^{-1}\mathbf{k}_m^T) - 2\varpi(1-\mathbf{k}_m^T P^{-1}\mathbf{k}_m)\mathbf{k}_m^T\hat{\mathbf{w}}\right)\right\} \end{aligned} \tag{3.30}$$

Where the $\hat{\mathbf{w}}^T c^*\Sigma_w^{-1}\hat{\mathbf{w}}$ term has been discarded as a constant. Now defining $Q = 1 - \mathbf{k}_m^T M^{-1}\mathbf{k}_m$ this leaves,

$$\begin{aligned} p_f &\propto \exp\left\{\tfrac{-1}{2c^*\sigma^2}\left(Q\varpi^2 - 2Q\varpi\mathbf{k}_m^T\hat{\mathbf{w}}\right)\right\} \\ &\propto \exp\left\{\tfrac{-1}{2c^*\sigma^2}Q\left(\varpi^2 - 2\varpi\mathbf{k}_m^T\hat{\mathbf{w}}\right)\right\} \\ &\propto \exp\left\{\tfrac{-1}{2c^*\sigma^2}Q\left(\varpi - \mathbf{k}_m^T\hat{\mathbf{w}}\right)^2\right\} \end{aligned} \tag{3.31}$$

Where once again a constant is discarded. This is now starting to resemble a Gaussian distribution. Replacing the original terms yields, after a little rearrangement,

$$p_f \propto \exp\left\{\frac{-1}{2c^*\sigma^2}Q\left(f(\cdot) - \left[\mathbf{t}(\cdot)^T A^{-1}\mathbf{y} + \left(\phi(\cdot)^T - \mathbf{t}(\cdot)^T A^{-1}\Phi\right)\hat{\mathbf{w}}\right]\right)^2\right\} \tag{3.32}$$

From which it can be recognised that the distribution is proportional to a Gaussian over $f(\cdot)$, such that,

$$p(f(\boldsymbol{x}^\star)|\sigma^2, \mathbf{y}) \sim \mathcal{N}\left(m^{**}(\boldsymbol{x}^\star), \sigma^2 c^{**}(\boldsymbol{x}^\star, \boldsymbol{x}^\star)\right) \tag{3.33}$$

The mean is easily identified from the exponent and is given as,

$$m^{**}(\boldsymbol{x}^\star) = \phi(\boldsymbol{x}^\star)^T\hat{\mathbf{w}} + \mathbf{t}(\boldsymbol{x}^\star)^T A^{-1}(\mathbf{y} - \Phi\hat{\mathbf{w}}) \tag{3.34}$$

Concerning the posterior covariance, it is a little trickier to get it into the correct form. First it is recognised that the posterior covariance c^{**} is found by comparing to a Gaussian distribution, giving,

$$\begin{aligned} c^{**} &= c^*\sigma^2 Q^{-1} \\ &= c^*\sigma^2(1 - \mathbf{k}_m^T M^{-1}\mathbf{k}_m^T)^{-1} \end{aligned} \tag{3.35}$$

This can be shown to be equal to the following after some rearrangement,

$$c^{**} = c^*\sigma^2(1 + \mathbf{k}_m^T(M-K)^{-1}\mathbf{k}_m^T) \tag{3.36}$$

This simplifies to,

$$c^{**} = c^*\sigma^2(1 + \mathbf{k}_m^T(c^*)^{-1}\Sigma_w\mathbf{k}_m^T) \tag{3.37}$$

Now substituting back in all original terms, the final expression is given as,

$$c^{**}(\boldsymbol{x}^{\star},\boldsymbol{x}') = c^{*}(\boldsymbol{x}^{\star},\boldsymbol{x}') + (\phi(\boldsymbol{x}^{\star})^T - \mathbf{t}(\boldsymbol{x}^{\star})^T A^{-1}\Phi)(\Phi^T A^{-1}\Phi)^{-1}(\phi(\boldsymbol{x}')^T - \mathbf{t}(\boldsymbol{x}')^T A^{-1}\Phi)^T \tag{3.38}$$

Note that the mean function is the same as the predictive mean in 3.12, except the random variable $\mathbf{w}$ is replaced by the best-estimate $\hat{\mathbf{w}}$. The covariance function, however, has an extra term to accommodate the additional uncertainty surrounding the estimate of $\mathbf{w}$.

Finally, the marginalisation of σ^2 is performed by combining the normal distribution in 3.33 with the inverse gamma distribution of σ^2 found previously to give a normal-scaled inverse gamma distribution over $f(\boldsymbol{x})$ and σ^2. It is a standard result that the marginal distribution of $f(\boldsymbol{x})$ (i.e. integrating out σ^2) in this case is a multivariate student's t-distribution (see e.g. [69]) conditional only on B and $\mathbf{y}$, such that,

$$f(\boldsymbol{x})|B,\mathbf{y} \sim t_{n-q}\{m^{**}(\boldsymbol{x}), \hat{\sigma}^2 c^{**}(\boldsymbol{x},\boldsymbol{x}')\} \tag{3.39}$$

The GP is now fully fitted to the training data. Examining the posterior mean function in (3.34) (which would be used in predictions made with the GP), it is composed of two terms — the prior mean from (3.2) and the GP's correction to this prior "guess" after conditioning on the data:

$$m^{**}(\boldsymbol{x}) = \phi(\boldsymbol{x})^T\hat{\mathbf{w}} + \Psi(\boldsymbol{x})\hat{\epsilon} \tag{3.40}$$

where $\Psi = \mathbf{t}(\boldsymbol{x})^T A^{-1}$ and $\hat{\epsilon} = (\mathbf{z} - H\mathbf{w})$. This latter term can be viewed as a vector of residual errors between the linear regression model and the training data. If the training data were perfectly linear, this term would become zero and (3.40) would collapse to the prior mean, i.e. the linear model. When the training data deviates from the linear model, however, this term modifies the linear prior model to fit the training data with a smoothness (covariance) dictated by the hyperparameters that have been estimated. It is this combination of regression and interpolation in the posterior mean that means that the GP has much greater predictive power than a linear regression alone. In fact, the GP is even more sophisticated than this; as one also has a posterior variance for the model, one can establish natural confidence bounds on predictions.

It can be seen then that the GP is a very appropriate emulator for use in UA/SA. It is an efficient form of regression, thus requiring many fewer runs than estimation using MC methods. The structure is sufficiently flexible to fit a wide class of models, and the smoothness parameters can be estimated from the training data. Furthermore, confidence intervals are given for predictions — the GP quantifies its own uncertainty in modelling the data and can pass this model uncertainty on in the UA/SA to give a fuller overall account of uncertainty. Finally, the structure of the posterior mean and covariance functions are sufficiently tractable to analytically provide many of the required sensitivity estimates, since a squared-exponential covariance function has been selected and this will be shown to allow analytical evaluation of integrals when normal distributions are assigned to the input parameters.

3.1.4 Estimating Roughness Parameters

The roughness matrix B cannot be analytically marginalised in the same way as the other hyperparameters. For this reason, it is estimated from the data by MAP estimation, which involves assigning prior distributions to each element b_i (in this case uniform weak priors) and expressing the probability of the observed data as a function of B. The value of B that gives the highest probability of the data (i.e. the mode of the posterior distribution) is known as the MAP estimate and will be used as the best-estimate of B.

The derivation of the likelihood function for B will not be given in detail here, but further details can be found in [70]. However, the principle is as follows. The prior distribution of the data conditional on all the hyperparameters is given in 3.6. By combining this with the improper prior distribution $p(\sigma^2, \mathbf{w})$ given in 3.7 and improper priors over each element of B, the posterior distribution of parameters given the data can be expressed as,

$$f(\mathbf{w}, \sigma^2, B|\mathbf{y}) = \frac{1}{\sigma^{n+2} 2\pi^{n/2} |A|^{1/2}} \exp\left\{ \frac{-1}{2\sigma^2} (\mathbf{y} - \Phi\mathbf{w})^T A^{-1} (\mathbf{y} - \Phi\mathbf{w}) \right\} \tag{3.41}$$

It is now necessary to marginalise $\mathbf{w}$ and σ^2. Integrating first with respect to $\mathbf{w}$ gives (recognising that proportional constants can be ignored since this will not affect the value of the posterior mode),

$$f(\sigma^2, B|\mathbf{y}) \propto \frac{1}{\sigma^{n+2-q} |A|^{1/2} |\Sigma_w|^{1/2}} \exp\left\{ \frac{-1}{2\sigma^2} (\mathbf{y} - \Phi\hat{\mathbf{w}})^T A^{-1} (\mathbf{y} - \Phi\hat{\mathbf{w}}) \right\} \tag{3.42}$$

Next, σ^2 may be integrated out to give,

$$f(B|\mathbf{y}) \propto \hat{\sigma}^{\frac{q-n}{2}} |A|^{\frac{-1}{2}} |\Sigma_w|^{\frac{-1}{2}} \tag{3.43}$$

With a global optimisation algorithm, the maximum of this function (or more conveniently, the logarithm of the function) can be found by searching the parameter space of B. Of course, this is another manifestation of the curse of dimensionality, so optimising B for models with a large number of input dimensions can be extremely time-consuming. In fact, the estimation of B may often represent the most computationally-intensive part of the process of UA with a GP emulator. It would however still tend to represent only a small fraction of the cost of obtaining the training data in the first place.

3.2 Inference for Sensitivity Analysis

At this point it is assumed that a joint pdf $p(\boldsymbol{x})$ exists, expressing the uncertainty of the input parameters to the emulator. This necessarily forces the inputs to be defined by the same class of probability distribution. Additionally, an assumption of independence is made, first to improve the tractability of the integrals that follow, and second, to allow the variance decomposition given previously in Section 2.2.2. To ensure that the integrals for inference are tractable it is further required that the joint pdf is either uniform or Gaussian; the integrals defined here are derived for the joint-Gaussian case. This is a limitation of sorts; however in the experience of the author, it is not usually the case that enough data concerning the input uncertainties are available to define distributions of a more complex form than Gaussian or uniform. However, in the event that other forms or a mixture of forms is essential, a Beta distribution (or similar flexible distribution) could be used which allows the definition of a wide class of pdf shapes. In this case however, numerical integration would be required.

3.2.1 Mean, Variance and Main Effects

With the joint pdf defined as Gaussian or uniform, all of the uncertainty measures discussed in Section 2.2.2 can be analytically derived from the posterior distribution of the GP. For notational clarity, throughout the rest of this chapter any marginal pdf such as $p_{-r|r}(\boldsymbol{x}_{-r}|\boldsymbol{x}_r)$ will be expressed as $p(\boldsymbol{x}_{-r})$. A shorthand integral notation will also be adopted so that, for example,

$$\int_{\mathcal{X}_{-r}} \phi(\boldsymbol{x})^T p(\boldsymbol{x}_{-r}) d\boldsymbol{x}_{-r} \equiv \int_{\mathcal{X}_{-r}} \phi(\boldsymbol{x})^T dP(\boldsymbol{x}_{-r}) \tag{3.44}$$

where $\mathcal{X}_r$ is the support of the marginal distribution of $\boldsymbol{x}_{-r}$ and $-r$ represents the complimentary set of r.

Of immediate interest are the expected value and variance of any outputs of the model, given the uncertainty in the inputs. The expected value can be shown to be a special case of the definition of main effects and interactions for a subset of inputs r, since,

$$E(Y|\boldsymbol{x}_r) = \int_{\mathcal{X}_{-r}} f(\boldsymbol{x}) dP(\boldsymbol{x}_{-r}) \tag{3.45}$$

from which it is evident that if r is the empty set, (3.45) reduces to the expected value of Y. This expression can be therefore used to define both the main effects of Y, as well as the expected value. Remember here that the function $f(\boldsymbol{x})$ is a random variable and therefore has associated mean and variance; $E(Y|\boldsymbol{x}_r)$ is therefore *also* a random variable. To find the posterior mean of this conditional expectation, it suffices to insert the posterior mean of the GP in place of $f(\boldsymbol{x})$ such that,

$$E^*\left\{E(Y|\boldsymbol{x}_r)\right\} = \int_{\mathcal{X}_{-r}} m^{**}(\boldsymbol{x})dP(\boldsymbol{x}_{-r}) \tag{3.46}$$

(with E^* denoting the posterior expectation over the random variable $f(\cdot)$), which can be expressed as,

$$E^*\left\{E(Y|\boldsymbol{x}_r)\right\} = \boldsymbol{\phi}_r(\boldsymbol{x}_r)^T\hat{\mathbf{w}} + \mathbf{t}_r(\boldsymbol{x}_r)^T\mathbf{e} \tag{3.47}$$

where,

$$\boldsymbol{\phi}_r(\boldsymbol{x}_r) = \int_{\mathcal{X}_{-r}} \boldsymbol{\phi}(\boldsymbol{x})dP(\boldsymbol{x}_{-r}) \tag{3.48}$$

$$\mathbf{t}_r(\boldsymbol{x}_r) = \int_{\mathcal{X}_{-r}} \mathbf{t}(\boldsymbol{x})dP(\boldsymbol{x}_{-r}) \tag{3.49}$$

$$\mathbf{e} = A^{-1}(\mathbf{y} - \Phi\hat{\mathbf{w}}) \tag{3.50}$$

Main effects result from considering the case when r consists of only one input, while interactions can be found when r is extended to larger subsets. Notice that the expression for the main effect (3.47) is a function of $\boldsymbol{x}_r$ and therefore to plot the main effect response it is evaluated over the range of uncertainty of $\boldsymbol{x}_r$ and typically standardised for convenient comparison with other main effect plots. Clearly, plotting interactions of third order and higher interactions becomes difficult but is rarely of interest.

In order to find the variance of the posterior distributions of main effects (and hence the posterior variance of Y), a general result can be used which is given as,

$$\begin{aligned}\mathrm{cov}^*\left\{E(Y|\boldsymbol{x}_r), E(Y|\boldsymbol{x}'_p)\right\} &= \hat{\sigma}^2 \int_{\mathcal{X}_{-r}}\int_{\mathcal{X}_{-p}} c^{**}(\boldsymbol{x},\boldsymbol{x}')dP(\boldsymbol{x}'_{-p})dP(\boldsymbol{x}_{-r}) \\ &= \hat{\sigma}^2\begin{bmatrix} U_{r,p}(\boldsymbol{x}_r,\boldsymbol{x}'_p) - \boldsymbol{t}_r(\boldsymbol{x}_r)A^{-1}\boldsymbol{t}_p(\boldsymbol{x}_p)^T + \\ \left\{\boldsymbol{r}_r(\boldsymbol{x}_r) - \boldsymbol{t}_r(\boldsymbol{x}_r)A^{-1}\Phi\right\}\Sigma_w\left\{\boldsymbol{r}_p(\boldsymbol{x}_p) - \boldsymbol{t}_p(\boldsymbol{x}_p)A^{-1}\Phi\right\}^T\end{bmatrix}\end{aligned} \tag{3.51}$$

where,

$$U_{r,p}(\boldsymbol{x}_r,\boldsymbol{x}'_p) = \int_{\mathcal{X}_{-r}}\int_{\mathcal{X}_{-p}} c(\boldsymbol{x},\boldsymbol{x}')dP(\boldsymbol{x}'_{-p})dP(\boldsymbol{x}_{-r}) \tag{3.52}$$

The evaluation of the integrals in Equations 3.48 and 3.49 is now considered. Since the joint distribution is Gaussian, integrating over the support $\mathcal{X}_r$ or $\mathcal{X}_{-r}$ is equivalent to integrating between the limits $-\infty$ to $+\infty$. It is supposed from here that r is an integer denoting the last input dimension in the monotonically increasing set $\{x_1, x_2, ..., x_r\}$. This is to simplify the illustration of the matrices that result from these integrals, but r can of course be any permutation of input dimensions.

Integral ϕ_r

Starting with $\phi_r(\boldsymbol{x}_r)$ and remembering that $\phi(\boldsymbol{x})^T = (1, x_1, x_2, ..., x_d)$, the resulting vector is composed of two separate integrals. A notation will be used from here where integrals are split into manageable terms denoted as, for example, I^{ϕ}_{α}, where ϕ denotes the vector or matrix to be evaluated, and α is simply an index for reference. The first integral of $\phi_r(\boldsymbol{x}_r)$ is the integral of 1,

$$I^{\phi}_1 = \int_{\mathcal{X}_{-r}} dP(\boldsymbol{x}_{-r}) = 1 \tag{3.53}$$

Since the integral of a pdf is by definition equal to 1. Next, for any $x_i \notin \boldsymbol{x}_r$ we have,

$$\begin{aligned} I^{\phi}_2 &= \int_{\mathcal{X}_{-r}} x_i dP(\boldsymbol{x}_{-r}) \text{ for } x_i \notin \boldsymbol{x}_r \\ &= \int_{\mathcal{X}_{-(r\cup i)}} p\left(\boldsymbol{x}_{-(r\cup i)}\right) d\boldsymbol{x}_{-(p\cup i)} \int_{\mathcal{X}_i} x_i p\left(x_i\right) dx_i \\ &= 1.E\left(x_i\right) = E\left(x_i\right) \end{aligned} \tag{3.54}$$

Finally, for the case of $x_i \in \boldsymbol{x}_r$,

$$\begin{aligned} I^{\phi}_3 &= \int_{\mathcal{X}_{-r}} x_i dP(\boldsymbol{x}_{-r}) \qquad \text{for} \quad x_i \in \boldsymbol{x}_r \\ &= x_i \int_{\mathcal{X}_{-r}} p(\boldsymbol{x}_{-r}) d\boldsymbol{x}_{-r} \\ &= x_i \end{aligned} \tag{3.55}$$

since x_i is treated as constant. This results in the vector,

$$\phi_r(\boldsymbol{x}_r) = (1, x_1, ..., x_r, \mu_{r+1}, ..., \mu_d) \tag{3.56}$$

where $\mu(x_i) = E(x_i)$.

Integral $\mathbf{t}_r$

Turning now to $\boldsymbol{t}_r(\boldsymbol{x}_r)$, the jth element T_j of $\boldsymbol{t}_r(\boldsymbol{x}_r)$ is the integral,

$$\begin{aligned} T_j &= \int_{\mathcal{X}_{-r}} c(\boldsymbol{x}, \boldsymbol{x}_j) dP(\boldsymbol{x}_{-r}) \\ &= \int_{\mathcal{X}_{-r}} \exp\left\{-(\boldsymbol{x} - \boldsymbol{x}_j) B (\boldsymbol{x} - \boldsymbol{x}_j)\right\} p(\boldsymbol{x}_{-r}) d\boldsymbol{x}_{-r} \end{aligned} \tag{3.57}$$

Since B is a diagonal matrix this breaks down into,

$$\begin{aligned} T_j &= \int_{\mathcal{X}_{-r}} \left[\prod_{i=1}^{d} \exp\left\{-b_i(x_i - x_{i,j})^2\right\} \right] p(\boldsymbol{x}_{-r}) d\boldsymbol{x}_{-r} \\ &= \prod_{i=1}^{r} \exp\left\{-b_i(x_i - x_{i,j})^2\right\} \prod_{i=r+1}^{d} \left[\int_{\mathcal{X}_{-r}} \exp\left\{-b_i(x_i - x_{i,j})^2\right\} p(x_i) dx_i \right] \end{aligned} \tag{3.58}$$

since the joint distribution is also separable. For any of the integrals in second product operator of 3.58, we have,

$$I_1^t = \int_{\mathcal{X}_i} \exp\left\{-b_i(x_i - x_{i,j})^2\right\} \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left\{ -\frac{(x_i - \mu_i)^2}{2\sigma_i^2} \right\} dx_i \tag{3.59}$$

By completing the square this results in a Gaussian integral, which can be shown (after some rearrangement) to be equal to,

$$I_1^t = \frac{\exp\left\{ \frac{\left(2b_i x_{i,j} + \frac{\mu_i}{\sigma_i^2}\right)^2}{4b_i + \frac{2}{\sigma_i^2}} - b_i x_{i,j}^2 - \frac{\mu_i}{2\sigma_i^2} \right\}}{\sqrt{2b_i^2 + 1}} \tag{3.60}$$

$\boldsymbol{t}_r(\boldsymbol{x}_r)$ can now be constructed from 3.60 and 3.58.

3.2.2 Sensitivity Indices

The same procedure applies for derivation of the Sobol' sensitivity indices. In order to evaluate V_r, the variance of a main effect or interaction, a variance identity is used (see e.g. [10]),

$$V_r = \mathrm{var}\{E_{-r}(Y|\boldsymbol{x}_r)\} = E_r\{E_{-r}(Y|\boldsymbol{x}_r)^2\} - E_r\{E_{-r}(Y|\boldsymbol{x}_r)\}^2 \tag{3.61}$$

This reduces to the following using a property of conditional expectation known as *consistency* which states that $E(E(Y|X)) = E(Y)$ [10],

$$\mathrm{var}\{E_{-r}(Y|\boldsymbol{x}_r)\} = E_r\{E_{-r}(Y|\boldsymbol{x}_r)^2\} - E_r(Y)^2 \tag{3.62}$$

Hence,

$$E^*(V_r) = E^*[E_r\{E_{-r}(Y|\boldsymbol{x}_r)^2\}] - E^*(E_r(Y)^2) \tag{3.63}$$

$E^*(E_r(Y)^2)$ can be found from the result that,

$$E^*(E_r(Y)^2) = \mathrm{var}^*\{E_r(Y)\} + \{E^*(E_r(Y))\}^2 \tag{3.64}$$

using a rearragement of the same variance identity (see again [68]). This leaves $E^*[E_r\{E_{-r}(Y|\boldsymbol{x}_r)^2\}]$ the equation for which is presented here, using the notation that $x^\sharp$ is the vector comprising of x_r and x'_{-r}, i.e.

$$\boldsymbol{x}^{\sharp} = \begin{pmatrix} x_1 \in \boldsymbol{x}_r \\ \vdots \\ x_r \in \boldsymbol{x}_r \\ x'_{r+1} \in \boldsymbol{x}_{-r} \\ \vdots \\ x'_r \in \boldsymbol{x}_{-r} \end{pmatrix} \text{ and } \boldsymbol{x} = \begin{pmatrix} x_1 \in \boldsymbol{x}_r \\ \vdots \\ x_r \in \boldsymbol{x}_r \\ x_{r+1} \in \boldsymbol{x}_{-r} \\ \vdots \\ x_r \in \boldsymbol{x}_{-r} \end{pmatrix} \tag{3.65}$$

This can then be represented as the following,

$$\begin{aligned} & E^*[E_r\{E_{-r}(Y|\boldsymbol{x}_r)^2\}] \\ &= \int_{\mathcal{X}_r}\int_{\mathcal{X}_{-r}}\int_{\mathcal{X}_{-r}} E^*\{f(\boldsymbol{x})f(\boldsymbol{x}^{\sharp})\}dP(\boldsymbol{x}_{-r})dP(\boldsymbol{x}'_{-r})dP(\boldsymbol{x}_r) \\ &= \int_{\mathcal{X}_r}\int_{\mathcal{X}_{-r}}\int_{\mathcal{X}_{-r}} E^*\{\hat{\sigma}^2 c^{**}(\boldsymbol{x},\boldsymbol{x}^{\sharp}) + m^{**}(\boldsymbol{x})m^{**}(\boldsymbol{x}^{\sharp})\}dP(\boldsymbol{x}_{-r})dP(\boldsymbol{x}'_{-r})dP(\boldsymbol{x}_r) \end{aligned} \tag{3.66}$$

which can equivalently be expressed as,

$$\begin{aligned} & E^*[E_r\{E_{-r}(Y|\boldsymbol{x}_r)^2\}] \\ &= \hat{\sigma}^2[U_r - \mathrm{tr}(A^{-1}P_r) + \mathrm{tr}\{\Sigma_w(Q_r - S_rA^{-1}\Phi - \Phi^TA^{-1}S_r^T + \Phi^TA^{-1}P_rA^{-1}\Phi)\}] \\ & + \mathrm{tr}(\mathrm{e}^TP_r) + 2\mathrm{tr}(\hat{\mathrm{w}}^TS_r\mathrm{e}) + \mathrm{tr}(\hat{\mathrm{w}}^TQ_r\hat{\mathrm{w}}), \end{aligned} \tag{3.67}$$

where,

$$Q_r = \int_{\mathcal{X}_r}\int_{\mathcal{X}_{-r}}\int_{\mathcal{X}_{-r}} \phi(\boldsymbol{x})\phi(\boldsymbol{x}^{\sharp})^TdP(\boldsymbol{x}_{-r})dP(\boldsymbol{x}'_{-r})dP(\boldsymbol{x}_r) \tag{3.68}$$

$$S_r = \int_{\mathcal{X}_r}\int_{\mathcal{X}_{-r}}\int_{\mathcal{X}_{-r}} \phi(\boldsymbol{x})\mathrm{t}(\boldsymbol{x}^{\sharp})^TdP(\boldsymbol{x}_{-r})dP(\boldsymbol{x}'_{-r})dP(\boldsymbol{x}_r) \tag{3.69}$$

$$P_r = \int_{\mathcal{X}_r}\int_{\mathcal{X}_{-r}}\int_{\mathcal{X}_{-r}} \mathrm{t}(\boldsymbol{x})\mathrm{t}(\boldsymbol{x}^{\sharp})^TdP(\boldsymbol{x}_{-r})dP(\boldsymbol{x}'_{-r})dP(\boldsymbol{x}_r) \tag{3.70}$$

$$U_r = \int_{\mathcal{X}_r}\int_{\mathcal{X}_{-r}}\int_{\mathcal{X}_{-r}} c(\boldsymbol{x},\boldsymbol{x}^{\sharp})dP(\boldsymbol{x}_{-r})dP(\boldsymbol{x}'_{-r})dP(\boldsymbol{x}_r) \tag{3.71}$$

and "tr" denotes the trace of a matrix. The derivation of these integrals follows in Section 3.2.3. The variance of V_r could also be derived but is complex and also rarely of interest. Note that (3.63) can be used to express both S_i and S_{Ti} by considering the cases were $\boldsymbol{x}_r = x_i$ and $\boldsymbol{x}_r = \boldsymbol{x}_{-i}$ respectively. To standardise, the estimates can be divided by $E^*\{\mathrm{var(Y)}\}$, although this is not exactly equal to the posterior expectation of the sensitivity indices. It is however the closest analytical expression available.

An implementation of the method outlined here is available in a package called GEM-SA [71]. The equivalent was also written in Matlab by the author. For the analyses performed in this thesis a mixture of the two codes was used.

3.2.3 Derivation of Integrals for Sensitivity Indices

The remaining integrals are here presented in a convenient order based on complexity. The notation is consistent with that used previously in the chapter.

Integral Q_r

$$Q_r = \int_{\mathcal{X}_r} \int_{\mathcal{X}_{-r}} \int_{\mathcal{X}_{-r}} \phi(\boldsymbol{x})\phi(\boldsymbol{x}^\sharp)^T dP(\boldsymbol{x}_{-r}) dP(\boldsymbol{x}'_{-r}) dP(\boldsymbol{x}_r) \tag{3.72}$$

The product of $\phi(\boldsymbol{x})\phi(\boldsymbol{x}^\sharp)^T$ creates the matrix,

$$\phi(\boldsymbol{x})\phi(\boldsymbol{x}^\sharp)^T = \begin{pmatrix} 1 & x_1 & \cdots & x_r & x'_{r+1} & \cdots & x'_d \\ x_1 & x_1^2 & \cdots & x_1 x_r & x_1 x'_{r+1} & \cdots & x_1 x'_d \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ x_r & x_r x_1 & \cdots & (x_r)^2 & x_r x'_{r+1} & \cdots & x_r x'_d \\ x_{r+1} & x_{r+1} x_1 & & x_{r+1} x_r & x_{r+1} x'_{r+1} & \cdots & x_{r+1} x'_d \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ x_d & x_d x_1 & \cdots & x_d x_r & x_d x'_{r+1} & \cdots & x_d x'_d \end{pmatrix} \tag{3.73}$$

This results in several different integrals, which are listed here:

$$\begin{aligned} I_1^Q &= \int_{\mathcal{X}_r} \int_{\mathcal{X}_{-r}} \int_{\mathcal{X}_{-r}} dP(\boldsymbol{x}_{-r}) dP(\boldsymbol{x}'_{-r}) dP(\boldsymbol{x}_r) \\ &= 1 \end{aligned} \tag{3.74}$$

since once again, it is the integral of a pdf. The second case is,

$$\begin{aligned} I_2^Q &= \int_{\mathcal{X}_r} \int_{\mathcal{X}_{-r}} \int_{\mathcal{X}_{-r}} x_i dP(\boldsymbol{x}_{-r}) dP(\boldsymbol{x}'_{-r}) dP(\boldsymbol{x}_r) \text{ for } x_i \in \boldsymbol{x}_r \\ &= \int_{\mathcal{X}_r} x_i dP(\boldsymbol{x}_r) \\ &= \int_{\mathcal{X}_i} x_i p(x_i) dx_i \\ &= E(x_i) = \mu_i \end{aligned} \tag{3.75}$$

Considering now the case when $x_i \notin \boldsymbol{x}_r$,

$$\begin{aligned} I_3^Q &= \int_{\mathcal{X}_r}\int_{\mathcal{X}_{-r}}\int_{\mathcal{X}_{-r}} x_i dP(\boldsymbol{x}_{-r})dP(\boldsymbol{x}'_{-r})dP(\boldsymbol{x}_r) \text{ for } x_i \notin \boldsymbol{x}_r \\ &= \int_{\mathcal{X}_{-r}} x_i dP(\boldsymbol{x}_{-r}) \\ &= \int_{\mathcal{X}_i} x_i g(x_i)dx_i \\ &= E(x_i) \end{aligned} \tag{3.76}$$

This result also holds for integrals of $x'_i \in \boldsymbol{x}'_{-r}$. In the case of x_i^2,

$$\begin{aligned} I_4^Q &= \int_{\mathcal{X}_r}\int_{\mathcal{X}_{-r}}\int_{\mathcal{X}_{-r}} x_i^2 dP(\boldsymbol{x}_{-r})dP(\boldsymbol{x}'_{-r})dP(\boldsymbol{x}_r) \text{ for } x_i \in \boldsymbol{x}_r \\ &= \int_{\mathcal{X}_i} x_i^2 \tfrac{1}{\sigma_i\sqrt{2\pi}} \exp\left\{-\tfrac{(x_i-\mu_i)^2}{2\sigma_i^2}\right\} dx_i \end{aligned} \tag{3.77}$$

Using a substitution combined with the standard result for a Gaussian integral, and after a little rearrangement, this yields the satisfyingly simple result,

$$I_4^Q = \sigma_i^2 + \mu_i^2 \tag{3.78}$$

Turning now to the other diagonal elements,

$$\begin{aligned} I_5^Q &= \int_{\mathcal{X}_r}\int_{\mathcal{X}_{-r}}\int_{\mathcal{X}_{-r}} x_i x'_i dP(\boldsymbol{x}_{-r})dP(\boldsymbol{x}'_{-r})dP(\boldsymbol{x}_r) \text{ for } x_i \in x_{-r} \\ &= \int_{\mathcal{X}_i} x_i dp(x_i) \int_{\mathcal{X}_i} x'_i dp(x'_i) \\ &= \mu_i^2 \end{aligned} \tag{3.79}$$

since in this case the integrals are separable. For all other elements, the variables are also treated separately, and the integrals are again separable. The matrix can now be constructed as,

$$U_p(\boldsymbol{x}_p) = \begin{pmatrix} 1 & \mu_1 & \cdots & \mu_r & \mu_{r+1} & \cdots & \mu_d \\ \mu_1 & \mu_1^2+\sigma_1^2 & \cdots & \mu_1\mu_r & \mu_1\mu_{r+1} & \cdots & \mu_1\mu_d \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ \mu_r & \mu_r\mu_1 & \cdots & \mu_r^2+\sigma_r^2 & \mu_r\mu_{r+1} & \cdots & \mu_r\mu_d \\ \mu_{r+1} & \mu_{r+1}\mu_1 & \cdots & \mu_{r+1}\mu_r & \mu_{r+1}^2 & \cdots & \mu_{r+1}\mu_d \\ \vdots & \vdots & & \vdots & \vdots & \ddots & \vdots \\ \mu_d & \mu_d\mu_1 & \cdots & \mu_d\mu_r & \mu_d\mu_{r+1} & \cdots & \mu_d^2 \end{pmatrix} \tag{3.80}$$

Integral S_r

This is expressed as,

$$S_r = \int_{\mathcal{X}_r} \int_{\mathcal{X}_{-r}} \int_{\mathcal{X}_{-r}} \phi(\boldsymbol{x}) \mathrm{t}(\boldsymbol{x}^\sharp)^T dP(\boldsymbol{x}_{-r}) dP(\boldsymbol{x}'_{-r}) dP(\boldsymbol{x}_r) \tag{3.81}$$

where,

$$\phi(\boldsymbol{x})\mathrm{t}(\boldsymbol{x}^\sharp)^T = \begin{pmatrix} c(\boldsymbol{x}^\sharp, \boldsymbol{x}_1) & c(\boldsymbol{x}^\sharp, \boldsymbol{x}_2) & \cdots & c(\boldsymbol{x}^\sharp, \boldsymbol{x}_n) \\ x_1 c(\boldsymbol{x}^\sharp, \boldsymbol{x}_1) & x_1 c(\boldsymbol{x}^\sharp, \boldsymbol{x}_2) & \cdots & x_1 c(\boldsymbol{x}^\sharp, \boldsymbol{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ x_d c(\boldsymbol{x}^\sharp, \boldsymbol{x}_1) & x_d c(\boldsymbol{x}^\sharp, \boldsymbol{x}_2) & \cdots & x_d c(\boldsymbol{x}^\sharp, \boldsymbol{x}_n) \end{pmatrix} \tag{3.82}$$

Considering first the jth element in the top row only,

$$I_1^S = \int_{\mathcal{X}_r} \int_{\mathcal{X}_{-r}} \int_{\mathcal{X}_{-r}} c(\boldsymbol{x}^\sharp, \boldsymbol{x}_j) dP(\boldsymbol{x}_{-r}) dP(\boldsymbol{x}'_{-r}) dP(\boldsymbol{x}_r) \tag{3.83}$$

This is in fact equal to 3.57, and the result can therefore be expressed by 3.60. Considering now an element in a row for $x_i \in \boldsymbol{x}_r$,

$$\begin{aligned} I_2^S &= \int_{\mathcal{X}_r} \int_{\mathcal{X}_{-r}} \int_{\mathcal{X}_{-r}} x_i c(\boldsymbol{x}^\sharp, \boldsymbol{x}_j) dP(\boldsymbol{x}_{-r}) dP(\boldsymbol{x}'_{-r}) dP(\boldsymbol{x}_r) \\ &= \int_{\mathcal{X}_i} x_i \exp\left\{-b_i (x_i - x_{i,j})^2\right\} dp(x_i) \int_{\mathcal{X}_{-r}} x_i c(\boldsymbol{x}'_{-r}, \boldsymbol{x}_j) dP(\boldsymbol{x}'_{-r}) \end{aligned} \tag{3.84}$$

The expression for the first integral can be evaluated by completing the square and using the standard result of a Gaussian integral, resulting in the following,

$$I_{2,\alpha}^S = \frac{\exp\left\{ -\left(\frac{2b_i\sigma_i^2 x_{i,j} + \frac{\mu_i}{\sigma_i^2}}{-2b_i\sigma_i^2 - 1} \right)^2 - b_i x_{i,j}^2 - \frac{\mu_i}{2\sigma_i^2} \right\} \left\{ 2b_i\sigma_i^2 x_{i,j} + \mu_i \right\}}{2b_i\sigma_i^2 + 1} \tag{3.85}$$

The result of the second integral, for a single input variable, is given in 3.60, and will be denoted here as $I_{2,\beta}^S$ The combined result is therefore,

$$I_2^S = I_{2,\alpha}^S \prod_{i=r+1}^{d} I_{2,\beta}^S \tag{3.86}$$

So that $I_{2,\alpha}^S$ is evaluated for inputs in the set r, and $I_{2,\beta}^S$ for the remaining inputs.

The final case occurs for the remaining rows, where $x_i \notin \boldsymbol{x}_r$,

$$I_3^S = \int_{\mathcal{X}_r} \int_{\mathcal{X}_{-r}} \int_{\mathcal{X}_{-r}} x_i c(\boldsymbol{x}^\sharp, \boldsymbol{x}_j) dP(\boldsymbol{x}_{-r}) dP(\boldsymbol{x}'_{-r}) dP(\boldsymbol{x}_r) \text{ for } x_i \notin \boldsymbol{x}_r \tag{3.87}$$

Since $x_i \notin \boldsymbol{x}_r$, it is only integrated with respect to $\boldsymbol{x}_{-p}$ and can be separated from the other terms. This results in the product of three integrals,

$$\begin{aligned} I_3^S = & \int_{\mathcal{X}_i} x_i dp(x_i) \prod_{i=1}^{r} \left\{ \int_{\mathcal{X}_r} \exp\left\{-b_i(x_i - x_{i,j})^2\right\} dG(\boldsymbol{x}_r) \right\} \\ & \prod_{i=r+1}^{d} \left\{ \int_{\mathcal{X}_{-r}} \exp\left\{-b_i(x_i' - x_{i,j})^2\right\} dP(\boldsymbol{x}_{-r}) \right\} \end{aligned} \tag{3.88}$$

The product of the last two integrals here is equal to the product of $I_{2,\beta}^S$ over all input dimensions. The first integral is simply the expected value of x_i, therefore,

$$I_3^S = \mu_i \prod_{i=1}^{d} I_{2,\beta}^S \tag{3.89}$$

This results in the final solution for S_r, given as,

$$S_r = \begin{pmatrix} I_{1,j=1}^S & I_{1,j=2}^S & \cdots & I_{1,j=n}^S \\ I_{2,i=1,j=1}^S & I_{2,i=1,j=2}^S & \cdots & I_{2,i=1,j=n}^S \\ \vdots & \vdots & \cdots & \vdots \\ I_{2,i=r,j=1}^S & I_{2,i=r,j=2}^S & \cdots & I_{2,i=r,j=n}^S \\ I_{3,i=r+1,j=1}^S & I_{3,i=r+1,j=2}^S & \cdots & I_{3,i=r+1,j=n}^S \\ \vdots & \vdots & \cdots & \vdots \\ I_{3,i=d,j=1}^S & I_{3,i=d,j=2}^S & \cdots & I_{3,i=d,j=n}^S \end{pmatrix} \tag{3.90}$$

Integral P_r

This is defined as,

$$P_r = \int_{\mathcal{X}_r} \int_{\mathcal{X}_{-r}} \int_{\mathcal{X}_{-r}} \mathbf{t}(\boldsymbol{x})\mathbf{t}(\boldsymbol{x}^\sharp)^T dP(\boldsymbol{x}_{-r}) dP(\boldsymbol{x}'_{-r}) dP(\boldsymbol{x}_r) \tag{3.91}$$

where,

$$\mathbf{t}(\boldsymbol{x})\mathbf{t}(\boldsymbol{x}^\sharp)^T = \begin{pmatrix} c(\boldsymbol{x},\boldsymbol{x}_1)c(\boldsymbol{x}^\sharp,\boldsymbol{x}_1) & c(\boldsymbol{x},\boldsymbol{x}_1)c(\boldsymbol{x}^\sharp,\boldsymbol{x}_2) & \cdots & c(\boldsymbol{x},\boldsymbol{x}_1)c(\boldsymbol{x}^\sharp,\boldsymbol{x}_n) \\ c(\boldsymbol{x},\boldsymbol{x}_2)c(\boldsymbol{x}^\sharp,\boldsymbol{x}_1) & c(\boldsymbol{x},\boldsymbol{x}_2)c(\boldsymbol{x}^\sharp,\boldsymbol{x}_2) & \cdots & c(\boldsymbol{x},\boldsymbol{x}_2)c(\boldsymbol{x}^\sharp,\boldsymbol{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ c(\boldsymbol{x},\boldsymbol{x}_n)c(\boldsymbol{x}^\sharp,\boldsymbol{x}_1) & c(\boldsymbol{x},\boldsymbol{x}_n)c(\boldsymbol{x}^\sharp,\boldsymbol{x}_2) & \cdots & c(\boldsymbol{x},\boldsymbol{x}_n)c(\boldsymbol{x}^\sharp,\boldsymbol{x}_n) \end{pmatrix} \tag{3.92}$$

All the elements in this matrix can be treated in the same way. The integral of the (j,k)th element is given as,

$$P_{r(j,k)} = \int_{\mathcal{X}_r} \int_{\mathcal{X}_{-r}} \int_{\mathcal{X}_{-r}} c(\boldsymbol{x},\boldsymbol{x}_j)c(\boldsymbol{x}^\sharp,\boldsymbol{x}_k) dP(\boldsymbol{x}_{-r}) dP(\boldsymbol{x}'_{-r}) dP(\boldsymbol{x}_r) \tag{3.93}$$

This splits into the product of three integrals,

$$P_{r(j,k)} = I_1^P I_3^P I_3^P \tag{3.94}$$

where,

$$\begin{aligned}
I_1^P &= \prod_{i=1}^{r}\left\{\int_{\mathcal{X}_i} \exp\left\{-2b_i\left((x_i - x_{i,j})^2 + (x_i - x_{i,k})^2\right)\right\} dP(x_i)\right\} \\
I_2^P &= \prod_{i=r+1}^{d}\left\{\int_{\mathcal{X}_i} \exp\left\{-b_i\left((x_i - x_{i,j})^2\right)\right\} dP(x_i)\right\} \\
I_3^P &= \prod_{i=r+1}^{d}\left\{\int_{\mathcal{X}_i} \exp\left\{-b_i\left((x_i' - x_{i,k})^2\right)\right\} dP(x_i')\right\}
\end{aligned} \tag{3.95}$$

Both I_2^P and I_3^P have been presented before, and the result is given in 3.60. I_1^P is found from the usual method of completing the square, using the result of a Gaussian integral and some tedious rearrangement, yielding the result,

$$I_1^P = \frac{\exp\left\{-\left(\frac{2b_i(x_{i,j}+x_{i,k})+\mu_i}{-4b_i\sigma_i^2-1}\right)^2 - b_i\left(x_{i,j}^2 + x_{i,k}^2\right) - \frac{\mu_i^2}{2\sigma_i^2}\right\}}{\sqrt{4b_i\sigma_i^2 + 1}} \tag{3.96}$$

Integral U_r

This is defined as,

$$U_r = \int_{\mathcal{X}_r}\int_{\mathcal{X}_{-r}}\int_{\mathcal{X}_{-r}} c(\boldsymbol{x}, \boldsymbol{x}^\sharp) dP(\boldsymbol{x}_{-r}) dP(\boldsymbol{x}'_{-r}) dP(\boldsymbol{x}_r) \quad , \tag{3.97}$$

This integral is slightly different from the previous ones, since it is the covariance between two *variables*, $\boldsymbol{x}$ and $\boldsymbol{x}^\sharp$. Integrating first with respect to $\boldsymbol{x}_r$,

$$\begin{aligned}
\int_{\mathcal{X}_r} c(\boldsymbol{x}, \boldsymbol{x}^\sharp) dP(\boldsymbol{x}_r) &= \prod_{i=1}^{r}\left\{\int_{\mathcal{X}_i} \exp\left\{-b_i(x_i - x_i)^2\right\} dP(x_i)\right\} \\
&= 1
\end{aligned} \tag{3.98}$$

since $c(x_i, x_i) = 1$ and this leaves the integral of a pdf. This then leaves the integral of all $-r$ terms, i.e,

$$\begin{aligned}
U_r &= \int_{\mathcal{X}_{-r}}\int_{\mathcal{X}_{-r}} \left\{\prod_{i=r+1}^{d} \exp\left\{-b_i(x_i - x_i')^2\right\}\right\} dP(\boldsymbol{x}_{-r}) dP(\boldsymbol{x}'_{-r}) \\
&= \prod_{i=r+1}^{d} \int_{\mathcal{X}_i}\int_{\mathcal{X}_i} \left\{\exp\left\{-b_i(x_i - x_i')^2\right\}\right\} dP(x_i) dP(x'_{-i})
\end{aligned} \tag{3.99}$$

since as usual, the diagonal covariance expression can be split into multiples of integrals. For any ith integral I_i^P of this product of integrals where $x_i \notin \boldsymbol{x}_r$, integrating first with respect to x_i is accomplished by remembering the result from 3.60, i.e.

$$I_i^P = \int_{\mathcal{X}_i} \left\{ \frac{\exp\left\{ \frac{\left(2b_i x_i' + \frac{\mu_i}{\sigma_i^2}\right)^2}{4b_i + \frac{2}{\sigma_i^2}} - b_i \left(x_i'\right)^2 - \frac{\mu_i}{2\sigma_i^2} \right\}}{\sqrt{2b_i^2 + 1}} \right\} dP(x'_{-i}) \tag{3.100}$$

This integral is solved by collecting terms in $(x_i')^2$, x_i, and constants, giving

$$I_i^P = \psi \int_{\mathcal{X}_i} \{\omega(x'_{-i})^2 + \xi x'_{-i} - \zeta\} d(x'_{-i}) \tag{3.101}$$

where,

$$\begin{aligned} \omega &= -\left\{ \tfrac{1}{2\sigma_i^2} + \left(\tfrac{2b_i\sigma_i^2}{2b_i\sigma_i^2+1}\right)^2 - b_i \right\}; \\ \xi &= \tfrac{\mu_i}{\sigma_i^2} - \tfrac{4\mu_i b_i \sigma_i^2}{\left(2b_i\sigma_i^2+1\right)^2}; \\ \zeta &= \tfrac{\mu_i^2}{2\sigma_i^2} + \left(\tfrac{\mu_i}{2b_i\sigma_i^2+1}\right)^2; \\ \psi &= \tfrac{\exp\left(\frac{-\mu_i^2}{2\sigma_i^2}\right)}{\sqrt{4\pi b_i\sigma_i^4 + 2\pi\sigma_i^2}} \end{aligned} \tag{3.102}$$

After completing the square and applying the standard result of a Gaussian integral as usual, this gives,

$$I_i^P = \psi\sqrt{\frac{\pi}{\omega}} \exp\left\{ -\left(\frac{\xi}{2\omega}\right)^2 - \zeta \right\} \tag{3.103}$$

Which, after some very considerable rearrangement, leads to the almost suspiciously simple result,

$$U_r = \prod_{i=r+1}^{d} \frac{1}{\sqrt{4b_i\sigma_i^2 + 1}} \tag{3.104}$$

3.3 Conclusions

The Bayesian SA approach has been outlined in considerable detail in this chapter, with the intention of providing a step-by-step walkthrough of the specification of the GP, its training, and the analytical inference of UA and SA quantities with the use of a joint-Gaussian distribution for model uncertainties. The result is an emulator that is sufficiently flexible to model a wide class of problems, yet is very computationally efficient, since to estimate posterior means and (conditional) variances it requires no additional runs, even of the emulator itself. This is in contrast to Monte Carlo methods which require separate DOEs for each quantity to be estimated. The expressions given here are applicable to any model with a joint-Gaussian input distribution, so long as the assumptions about smoothness are valid. This means that once the expressions are programmed, they can be reused for any set of training data for any model.

Perhaps the only drawbacks that could be mentioned are the GP's requirement that the response of the model is smooth (although this issue is addressed in Chapter 8) and that for a large number of training points the GP can be slow, since it requires the inversion of the $n \times n$ covariance matrix A. Additionally, the matrix B of roughness parameters must be optimised, the computational expense of which increases exponentially with d. However it should be remembered that the GP emulator would typically only be used when a small set of training data is available — even for larger sets, the cost of training the GP will usually be only a small fraction of the cost of obtaining the training data.

In the following chapters, the GP emulator is used extensively on a number of complex engineering models: its practicality will be thoroughly tested and conclusions drawn at each stage. A practical walkthrough of the method described in this chapter is also provided in Appendix B.

Chapter 4

Modelling the Aortic Valve

There are many examples of models that occur in an engineering context that could require an uncertainty or sensitivity analysis. An area of modelling that is particularly hampered by the presence of large uncertainties is that of biomechanics, i.e. the study of the mechanics of biological systems. In order to more fully understand the mechanics of such systems, computational models (very often FE models) are used since detailed measurement of mechanical properties *in vivo* is extremely difficult and potentially tricky from an ethical perspective. However, the modelling of biological systems is subject to large uncertainty in material properties, loading and geometry (and many other model inputs) from both aleatoric and epistemic sources. The dimensions and material properties of a given biological component (a bone, for example) vary dramatically from one individual to the next. Furthermore, the material behaviour is often not well understood and can only be mathematically characterised with limited accuracy.

The Aortic Valve (AV) has been the subject of much interest in the biomechanics community since, of the four valves in the heart, it is the most prone to failure. Modelling the AV is motivated by two aspects: first, to understand the failure mechanism of the natural valve (and thereby hope to improve its longevity); second, to attempt to replicate the remarkable durability of the natural valve in bioprosthetic devices, which are themselves subject to failure. Many models of the AV have been created, usually using the FE approach, yet to date very little consideration has been given to uncertainties in modelling, despite the fact that the magnitude of uncertainties is large and their existence is well known. This chapter outlines the background to AV simulation, then describes in detail two models of the AV that have been constructed, both of which are novel in different ways. The following chapter (Chapter 5) presents an in-depth uncertainty and sensitivity analysis of both models with two motivations: first, to investigate the efficacy of Bayesian SA on finite element models; and second, to understand the effect of uncertainties on the modelling of the AV. The work can therefore be considered both from a statistical and a biomechanical perspective.

4.1 The Aortic Valve

The AV is one of the four valves in the human heart. It is situated between the left ventricle and the aorta (see Figure 4.1), its function being to ensure one-way flow of oxygenated blood from the heart to the rest of the body. Figure 4.1 also shows an illustration of the AV itself: it consists of three "cusps" or "leaflets" arranged in a rotationally symmetrical pattern, normal to the axis of blood flow. Opposite each leaflet are the *sinuses*, which are bulges in the aorta allowing the leaflets space to open fully. The section of aorta between the leaflets and the ventricle is known as the *aortic coronet.* When the blood pressure on the ventricular side of the valve exceeds the pressure on the aortic side (i.e. in *systole*), the valve opens by the three leaflets separating from each other and reversing their curvature; conversely, when the opposite is true (*diastole*) the valve closes and the leaflets come together or "coapt". The thicker regions of the leaflet that contact each other on closure are known as the *commissures.*

The opening mechanism is one of the most studied aspects of the AV, because in a natural lifetime the valve can open and close around 4 billion times. In the majority of cases this is achieved without failure. It is this impressive resistance to fatigue that prosthetic designers attempt to emulate.

4.1.1 Valve Failure and Prostheses

Although in the majority of cases the AV performs without failure for the entire life of the individual, of the four heart valves it is the most likely to fail. The valve can become *stenotic* (forward blood motion is impeded) or *insufficient* (the valve fails to fully close, allowing blood to leak back into the ventricle during diastole). Deformations (where the valve is not properly formed) are also reasonably common. In the case of AV failure, the patient is treated by fitting a replacement valve. There are several types of replacement, all of which have their respective advantages and disadvantages.

Perhaps the most common replacements are mechanical prostheses, which can consist of a ball-in-cage device, or an arrangement of moving plates — see [72] for a recent summary of both mechanical and bioprosthetic devices. These devices are very durable (being typically made of metal), but have major drawbacks. Blood forms clots on contact with metal, therefore anti-coagulents must be taken to prevent this for the remaining life of the patient. Furthermore, the blood flow (haemodynamics) can be turbulent around the angular surfaces of the mechanical prosthetic, which can damage the components of the blood. The alternative to a mechanical prosthetic is a bioprosthetic device — this consists of a valve constructed at least in part with biological materials. Quite often the leaflets of the aortic valves of pigs are attached to a metallic or polymer stent. This is known as a *stented porcine xenograft.* Pigs are chosen because the dimensions of the porcine

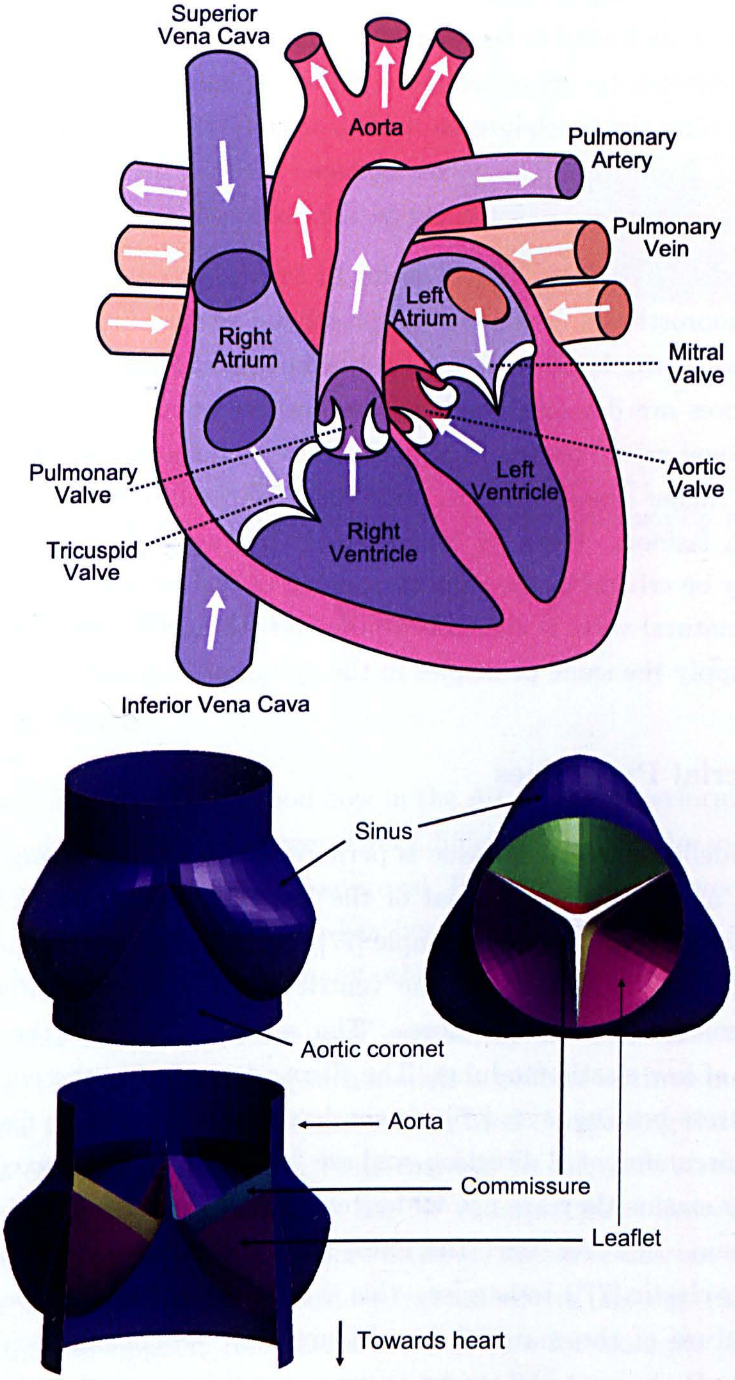


Figure 4.1: Diagram of the human heart (top); and aortic valve (bottom)

valve are very similar to the human equivalent (although some work has shown that the small differences could be responsible for reduced longevity of porcine xenografts [73]). Bioprosthetic devices do not require the use of anti-coagulents and have much improved haemodynamic performance. However, they are much less durable than the mechanical alternatives and as such tend to be used for elderly patients with a shorter life expectancy, since it is not difficult to appreciate that replacing an AV involves major surgery and therefore performing the procedure more than once is very undesirable. Overall however, there is no clear favourite in the choice between mechanical and bioprostheses [74], but there is certainly scope for improvement in the design of both.

Given the context of this work, it is useful to briefly examine the reasons for the poor durability of bioprostheses. Failure in bioprosthetic valves is usually due to calcification and degradation of the leaflets. Research has indicated that this calcification occurs in areas where fibres are damaged by high compressive stress [75]. This is thought to be due to sub-optimal valve opening in bioprostheses — in short, the leaflets of bioprosthetic valves undergo higher stresses in each cycle than the natural valve since they do not open in as smooth a fashion. Work by Yoxall *et al.* [76] suggests that the flexibility of the aortic root may be critical in the smooth opening of the natural valve. The motivation of modelling the natural valve is therefore to understand how it minimises leaflet stress and subsequently apply the same principles in the design of long-lasting prostheses.

4.1.2 Material Properties

Accurately modelling biological tissue is perhaps the most challenging part of developing biomechanical models. The material of the heart valve leaflet is known to be highly nonlinear and anisotropic (see for example [77]) and even heterogeneous [78]. It comprises of three distinguishable layers: on the ventricular side is the *ventricularis*, consisting of a layer of collagen and elastin fibres. The *spongiosa* layer in the middle consists of protoglyceans of low elastic modulus. The *fibrosa* layer facing the aorta is considered to be the main stress-bearing layer [75] - it consists mostly of collagen fibres. The fibres run largely in the circumferential direction, and are "crimped" in the relaxed state of the valve; as such, at low strains they are not under tension and resistive force is due purely to the connective tissue. However, once the fibres are un-crimped, they provide a strong (and virtually linear-elastic [77]) resistance - this is responsible for the hyperelastic, or roughly bi-modular nature of the material (see Figure 4.2). Some previous FE analyses have used a Mooney-Rivlin material model to represent this property [79]. Other alternatives include a piecewise-linear model or an Ogden model. The material has also been shown to be insensitive to strain rate and does not exhibit creep, although does experience stress relaxation [80]. The aortic tissue is also hyperelastic, but due to the relatively small strains in this region compared to the leaflet, it is often considered to be linear for the purposes of modelling.

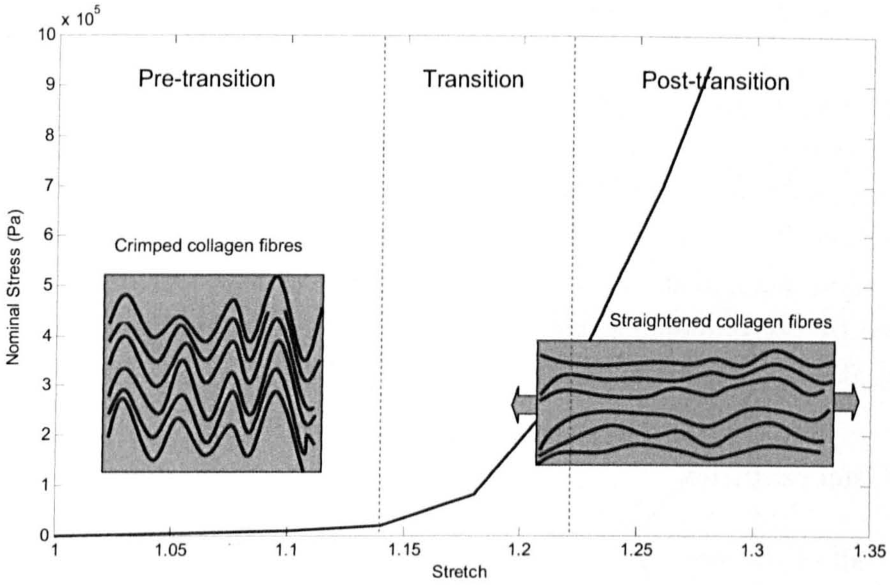


Figure 4.2: Uniaxial stress test on circumferentially-orientated AV leaflet tissue [81]. The arrangement of collagen fibres is shown in the pre- and post-transition regions.

4.2 Simulating the Aortic Valve

4.2.1 Existing Work

In vivo measurement of stress and blood flow in the AV has been performed in some cases, for example by Thubrikar [75]. It is however very difficult to get precise measurements, and requires specialist equipment and ethical approval. Instead, Finite Element (FE) modelling offers a detailed alternative and as such has been used extensively in the biomechanics literature to model the AV, as well as many other biological components.

AV models available in the literature vary extensively in their level of sophistication. Broadly speaking, the sophistication of such models is evident in three main areas: the solution method, the loading and the material models used. Regarding the solution method, early FE models considered only linear static models to calculate stress under pressure loading, usually in the closed position — see for example [82]. However, the dynamic opening of the valve is usually of much more interest, therefore current models break the simulation down into time steps and consider inertial effects (see e.g. [83, 84]). This can be done with either an implicit integration method (see for example [85]), but explicit integration is usually chosen since it is more stable when element contact is used and is suitable for fast, short-term events. Recent models have also used encompassing multi-scale approaches, considering organ-level, tissue-level and cell-level simulations ([86]).

In reality, the loading of the valve is provided by pressure exerted on the valve by the flow of blood. The actual pressure exerted on each side of the leaflet is a function of the position of the valve and the stage in the cardiac cycle, but early models prescribe pressure loading only by defining a simple pressure curve on the leaflets (e.g. [79, 87]). With an increase in available processing power and software capabilities, Fluid-Structure-Interaction (FSI) models have become increasingly common, including approaches based on the "Lagrange multiplier-based fictitious domain method" [88, 89], and several based on the use of Eulerian elements [84, 86, 90]. This allows a much more accurate representation of valve loading and also allows investigation of the blood flow itself depending on the fluid model used (see for example Morsi *et al.* [90]). Regarding the sophistication of material modelling, this is discussed in Sections 4.1.2 and 4.3.2.

4.2.2 Uncertainties

In almost all of the existing literature, little or no consideration has been given to the uncertainties prevalent in biomechanical models. In fact, the only treatment of uncertainty in AV models to the knowledge of the author is an informal investigation by Ranga *et al.* [79] of varying the elastic modulus of the aorta. This is despite the fact that the uncertainty associated with a AV model (and biomechanical models in general) is very substantial and stems from a number of sources, both epistemic and aleatoric.

One obvious source is that biological components vary in size from one individual to the next. For example, Thubrikar [75] states the dimensions of the AV only in wide ranges. Figure 4.3 illustrates these dimensions: this diagram has become the *de facto* reference for constructing the geometry of the AV. The values for each parameter are quoted as, for example, $H = 15.7$ to 19.8mm, or $\phi = 25°$ to $37°$. The vast majority of HV models in the literature pick the mean value for these parameters, yet the effects of these uncertainties on the output of the model remain unknown. The thickness of the leaflets and sinus (the aortic root) is also known to vary significantly and has in fact been quoted very differently in different AV models [73].

Aside from the geometry of the valve, the loading can vary depending on the individual — for example, the blood pressure in a healthy adult is known to be 120/80mmHg (this first value represents maximum pressure in systole; the second is the maximum pressure in diastole), but it is possible for it to be as low as 75/50mmHg and as high as 150/100mmHg in the cases of *hypotension* and *hypertension* respectively [91]. An additional source of uncertainty is from the material properties of the valve. This is both aleatory uncertainty (since material properties also vary from individual to individual) and epistemic, given that the behaviour of heart valve material is extremely complex and can only be characterised with limited fidelity. More is said on material properties in the following section.

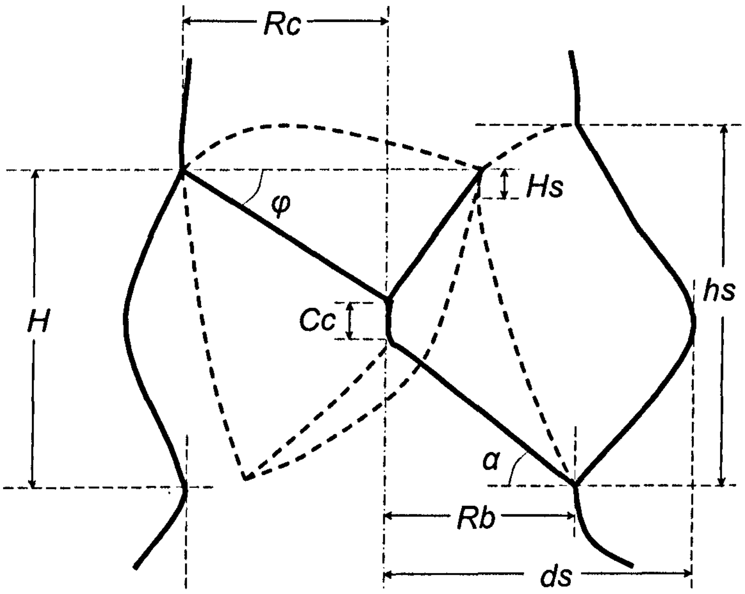


Figure 4.3: Schematic of the AV: R_b =radius of base; R_c =radius of commissures; H =valve height; α=bottom surface angle of leaflet; ϕ=free edge angle of leaflet; H_s =height of commissure; h_s =sinus height; d_s =radius of extremity of sinus; C_c =coaptation height. (adapted from Thubrikar [75])

4.3 The Finite Element Models

The solver used for all the modelling in this chapter is the explicit dynamic solver "LS-Dyna". This is a well-established commercial FE package that is capable of handling highly nonlinear dynamic events, complicated element contact and fluid-structure interaction. It has been successfully used in the literature to model the AV, e.g. [84, 86] amongst others. Two models were constructed for the purposes of this investigation. The first shall be referred to as the "dry" model, since it is loaded using only specified pressures on surfaces. The second makes use of fluid-structure interaction (FSI) and includes Eulerian elements; it will therefore be called the "wet" model. Although the wet model is substantially more sophisticated than the dry model, the dry model has its geometry defined parametrically and can thus be used to investigate geometric uncertainties such as those mentioned previously. The dry model is capable of this since it is pre- and post-processed entirely in the Ansys environment, using Ansys Parametric Design Langauge (APDL). This means that for a given set of geometric parameters it is automatically meshed, and dimensions can be treated as variables. However, the more advanced features of LS-Dyna are not available in the Ansys environment, thus the wet model was not able to be completely constructed in APDL. For this reason, the two models will be used to investigate different types of uncertainty and complement each other to some extent.

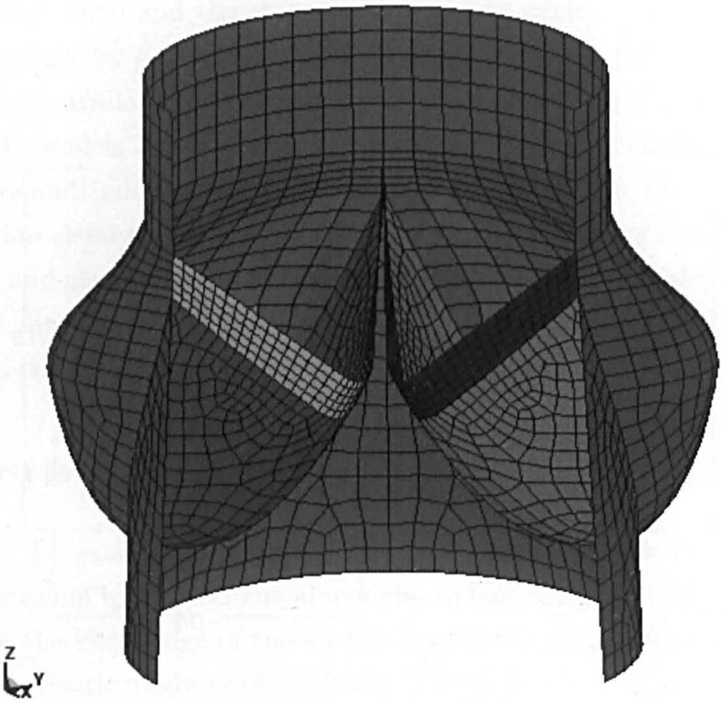


Figure 4.4: Two-thirds of aortic valve dry mesh (typical parameter set)

4.3.1 The Dry Model

The dry model was constructed parametrically in Ansys using APDL and solved using Ansys/LS-Dyna (a limited version of LS-Dyna embedded into the Ansys environment). Post-processing was also written into the APDL file to extract quantities of interest. The approach was to use Belytschko-Lin-Tsay (BLT) shell elements [92] to model the entire structure, encompassing the sinus, leaflets and a section of aorta at either end. This is an approach very commonly used in the literature (see for example [83, 87, 93] amongst many others), since the BLT element is the most computationally efficient shell element available in LS-Dyna [94] and is set as default. An illustration of the mesh for a typical parameter set is shown in Figure 4.4. The thickness of the leaflet is defined separately between the commissural region and the remainder of the leaflet. The entire model consists of some 3200 elements (dependent on the dimensions specified) and is run over a period of 0.1s, representing the opening of the valve at the beginning of systole.

Loading

The loading of the model was pressure-based— in other words, a time-varying pressure was specified on the underside of the valve leaflets in order to open the valve, as well as separate pressure specifications for the sinuses and aortic coronet (see Figure 4.5). The pressure curve was calculated from the well known pressure variations that result from

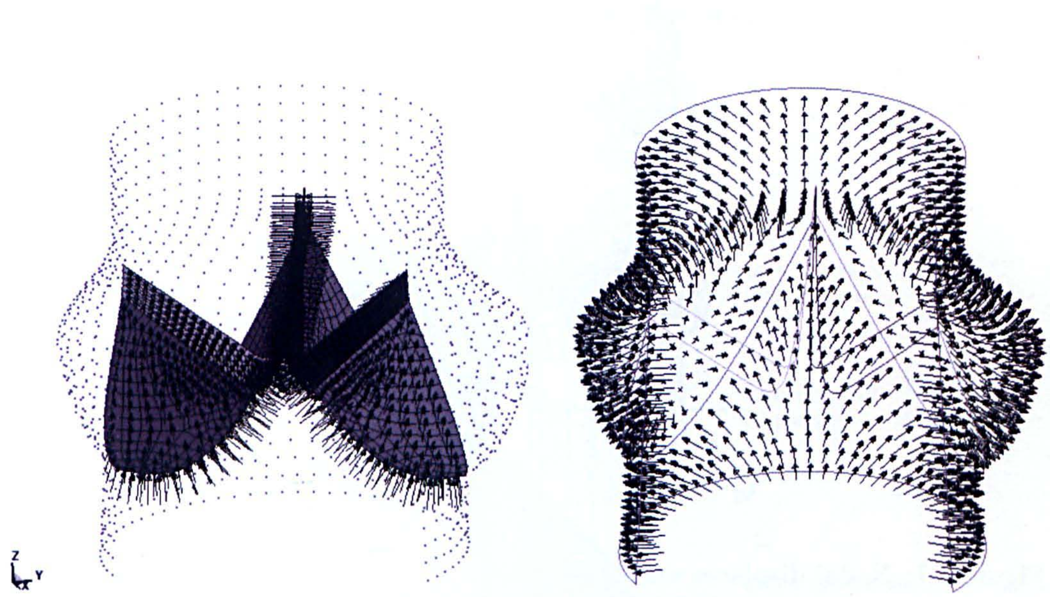


Figure 4.5: Pressure loading on the dry AV model: leaflet pressure (left); sinus and aorta pressure (right).

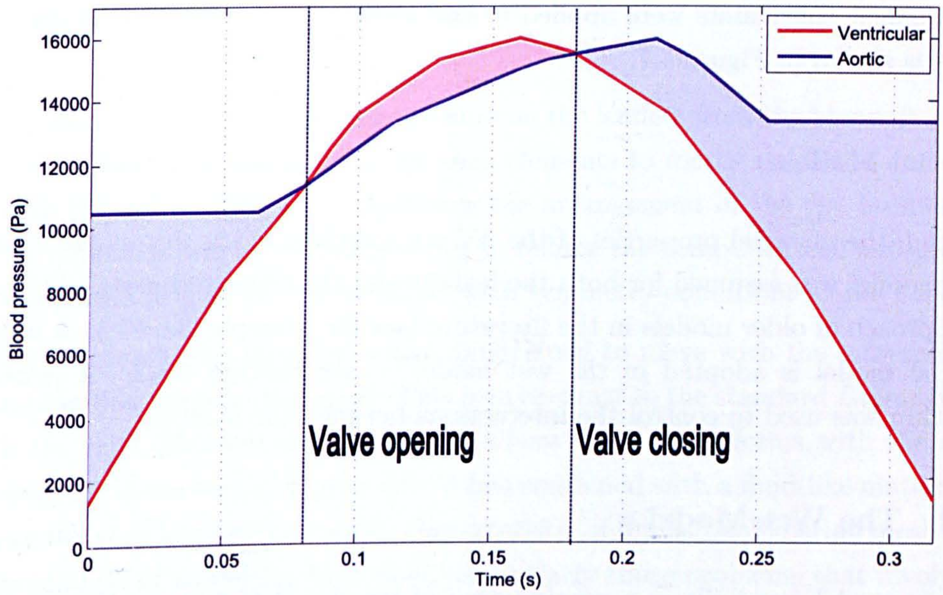


Figure 4.6: Aortic and ventricular pressure variations during the systolic phase of the cardiac cycle

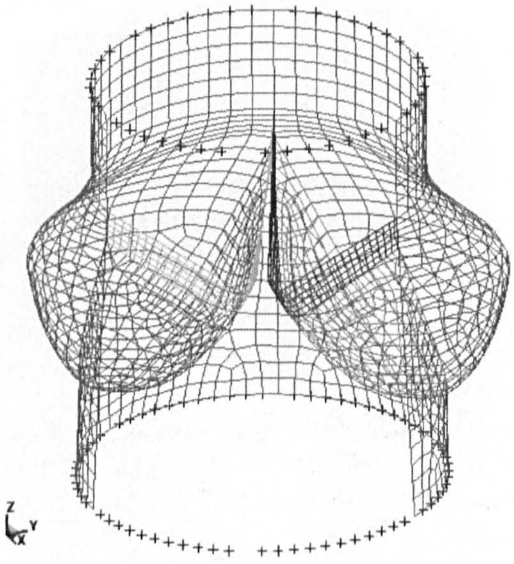


Figure 4.7: Nodal displacement constraints on the dry AV model (marked as crosses)

the cardiac cycle (see Figure 4.6). This formulation is not ideal because although the pressure difference between the aortic and ventricular side of the valve can be specified quite confidently from the pressure graph of the cardiac cycle, the pressure difference on the leaflets is only equal to the blood pressure difference (as specified by Figure 4.6) when the valve is closed. Once the valve is open or partly open, the pressure difference across the leaflets is no longer obvious. Nevertheless, this was considered an adequate first approximation for the mainly geometry-orientated uncertainty analysis of this model. Displacement constraints were applied to the nodes at the extremities in the z-direction — this is shown in Figure 4.7.

Material Models

Although the material properties of the AV are complex, in the dry model a simple linear elastic model was assumed for both the leaflets and the sinus and aorta. This is following the approach of older models in the literature (see for example [84, 87]). A more accurate material model is adopted in the wet model — see Section 4.3.2. A general contact algorithm was used to control the interactions between the leaflets.

4.3.2 The Wet Model

The wet model was built as a continuation of the dry model, using the same mesh, and the same shell element approach. As explained previously, the wet model is solved in the stand-alone LS-Dyna solver. Due to the nature of the input file, it is not possible to define global dimensions of the input file parametrically. Therefore the wet model was built using the mean values of the dimensions provided by Thubrikar [75] and other parameters were varied in the UA/SA. It improves on the more basic dry model in several ways.

Fluid Mesh

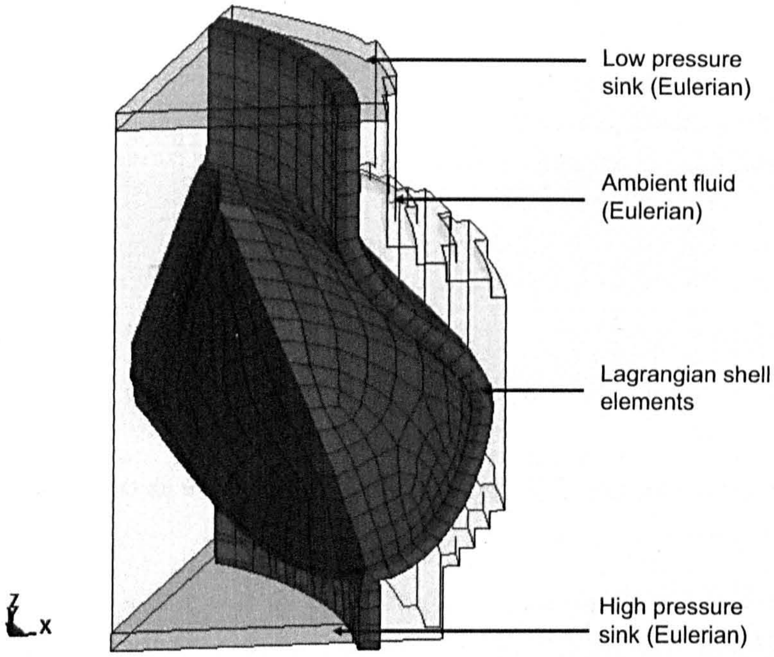

Low pressure sink (Eulerian)
Ambient fluid (Eulerian)
Lagrangian shell elements
High pressure sink (Eulerian)
z
x

Figure 4.8: Illustration of wet AV model with overlapping Eulerian mesh. Shell elements are displayed with their assigned thicknesses.

In order to more accurately model the pressure on the leaflets from the blood, it is possible to use an *Eulerian* mesh overlapping the shell elements to model the blood and its interactions with the valve. Figure 4.8 illustrates the arrangement of the two meshes. Notice that cyclic symmetry was exploited in order to reduce the computational expense of FSI; the model consists of $\frac{1}{6}$ of the entire valve, with symmetry conditions at the boundaries.

In the Eulerian mesh, the material is not constrained to move with the movement of the nodes, and can flow through the mesh. This is in contrast to the standard *Lagrangian* mesh (of which the shell elements are composed), where the mesh deforms with the material. The difference is illustrated in Figure 4.9. When combined with a fluid-like material model that allows the very large deformations characteristic of fluids, the Eulerian mesh provides a suitable method of modelling fluid flow, particularly those problems that involve short-duration, high-pressure events. This approach has indeed been successfully used before in the AV literature, for example [84] amongst others.

The Eulerian mesh is arranged into three parts: a high pressure sink, a part at ambient (unspecified) pressure, and a low pressure sink. The sinks can be constrained to follow

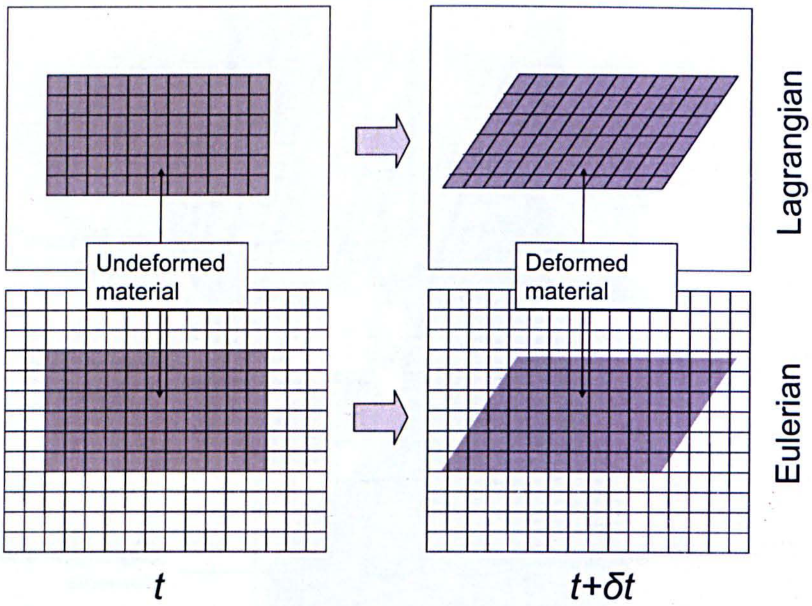


Figure 4.9: Lagrangian and Eulerian element formulations at time t and time $t + \delta t$ (after deformation)

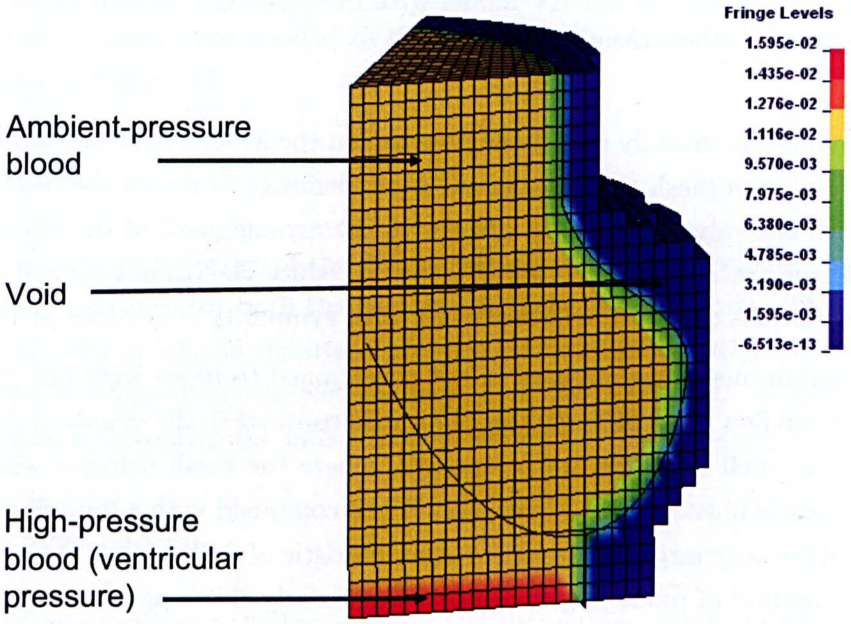


Figure 4.10: Pressure fringe plot for the wet AV model at $t = 0$. Pressure units are MPa.

the pressure variations given in Figure 4.6 — that is to say, at every timestep the pressure in the elements is reset to the pressure specified in a given pressure-time curve. In the Eulerian framework, the material occupying the mesh is assigned a material model and an *equation of state* (EOS). The EOS used here is the Gruneisen type, which consists of a number of parameters which define the pressure on a compressed element as a function of relative volume and internal energy (see [94] for more information). The pressure of Eulerian elements can therefore be controlled by adjusting these parameters. Figure 4.10 shows the pressure distribution of the model at the initial timestep. The division of materials in the Eulerian mesh is also shown. Note that the region outside the valve is assigned a "void" material model, which is used to fill the areas of empty mesh into which the valve is expected to expand. The other regions are assigned a constitutive material model which defines deviatoric stresses with a viscosity term. Coupling between the two meshes is performed using a penalty-based system: more on this can be found in [94]. Therefore, over the opening time period considered, the fluid flows from the high pressure sink to the low pressure sink, exerting a pressure on the Lagrangian elements that is much closer to that found in the natural valve.

Material Model

A further improvement in the wet model is that a nonlinear anisotropic material model was used. This is a constitutive soft-tissue (ST) model developed by Weiss *et al.* [95] which assumes that the hyperelastic response of the material is due to unidirectional fibres embedded in an isotropic matrix. The model has the added advantage that it is already implemented in LS-Dyna. To the knowledge of the author, this model has not been previously used to model AV tissue.

As with all hyperelastic material models, the stress is defined indirectly through a strain energy function W of invariants of the deformation gradient tensor $\mathbf{F}$. This ensures that the material is perfectly elastic and is a scalar function, so is mathematically easier to handle. The invariants ensure that the constitutive equation is the same under a change of basis (the reader is referred to Bower [96] for a full treatment of tensors and hyperelasticity, which is outside the scope of this thesis).

In the case of the ST model, the strain energy function is defined as,

$$W = F_1\left(\tilde{I}_1, \tilde{I}_2\right) + F_2(\lambda) + \frac{1}{2}K\left\{\ln(J)\right\}^2 \tag{4.1}$$

where $\tilde{I}_1$ and $\tilde{I}_2$ are deviatoric invariants of the right Cauchy deformation tensor, λ represents the deviatoric stretch (where $\lambda = 1 + \epsilon$, ϵ being the true strain) in the fibre direction, K is the effective bulk modulus and J is the volume ratio (the determinant of the deformation gradient tensor $\mathbf{F}$). The functions F_1 and F_2 are the (deviatoric) contributions of the isotropic background matrix and the reinforcing fibres respectively.

The last term provides the volumetric component, although heart valve tissue is usually assumed to be incompressible, rendering the term negligible. The isotropic background matrix is represented by the Mooney-Rivlin model [97], which expresses the strain energy as,

$$F_1\left(\tilde{I}_1, \tilde{I}_2\right) = C_1\left(\tilde{I}_1 - 3\right) + C_2\left(\tilde{I}_2 - 3\right) \tag{4.2}$$

where C_1 and C_2 are constants which can be determined from test data (see Section 4.3.2 below). The fibre term assumes that the fibres provide no resistance in compression, and in tension provide resistance in two stages. Below the critical stretch level λ^*, stress is described by a scaled exponential function as the fibres "un-crimp", whilst at higher stretches it is a linear function, representing the approximately elastic response of the straightened fibres. This is expressed as,

$$\lambda\frac{\partial F_2}{d\lambda} = \left\{\begin{array}{ll} 0 & \lambda < 1 \\ C_3\left[\exp\left(C_4(\lambda - 1)\right) - 1\right] & \lambda < \lambda^* \\ \left(C_5\lambda + C_6\right) & \lambda \geq \lambda^* \end{array}\right\} \tag{4.3}$$

where C_3 represents a scaling factor of the exponential region, C_4 is the exponent, C_5 is effectively the elastic modulus of the straightened fibres and C_6 is a function of the other parameters that ensures continuity at λ^*.

Fitting the Material Model

In order to fit material coefficients to test data, a nonlinear regression approach is adopted here, following that outlined by Quapp and Weiss [98]. On the assumption that fibres run only circumferentially, radial uniaxial stress/strain data can be used to fit the isotropic matrix term F_1, and the remaining coefficients can be fitted from the circumferential test data.

For the special case of an incompressible uniaxially loaded material, the Mooney-Rivlin stress σ_{MR} is expressed by differentiating the strain energy function with respect to $\mathbf{F}$, setting appropriate terms to zero (see [96] for further explanation):

$$\sigma_{MR} = 2\left(\lambda - \frac{1}{\lambda^2}\right)\left(C_1 + \frac{C_2}{\lambda}\right) \tag{4.4}$$

for which the constants can be readily computed by a linear regression algorithm. In the fibre direction, λ^* was estimated by visual inspection of the circumferential test data, it being the point at which the effective modulus sharply increases and the stress is approximately linear with respect to stretch. The test data was then divided into pre and post-transition regions, such that C_3 and C_4 were fitted to the former and C_5 to the latter using nonlinear regression.

Results of some element tests with a range of coefficients suggested that a strongly negative

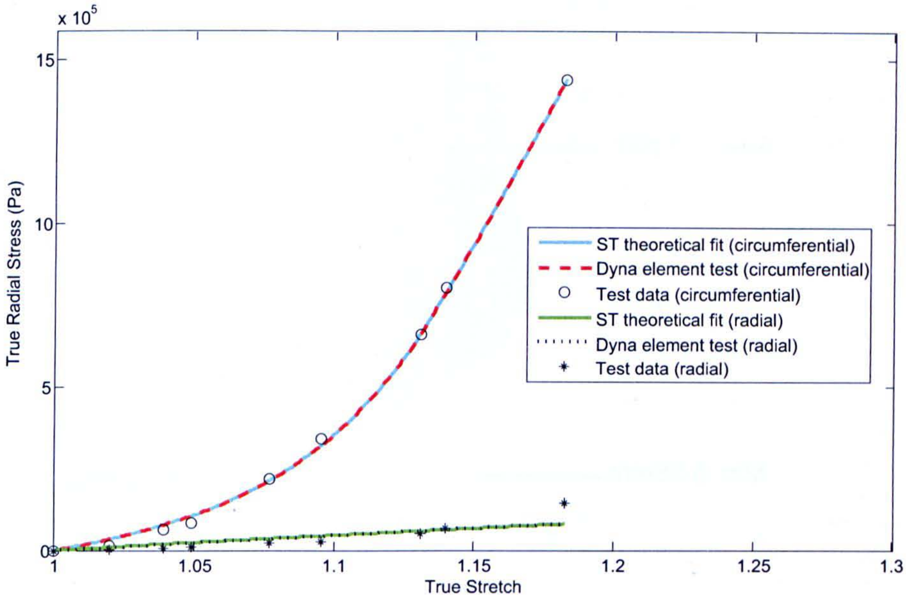


Figure 4.11: ST material model fitted to uniaxial test data [99] and compared against LS-Dyna element test

ρ	C_1	C_2	C_3	C_4	C_5	K	λ^*
1300kg/m^3	80.6kPa	0	60.1kPa	16.26	9.91MPa	2200GPa	1.14

Table 4.1: Typical parameter values for the ST model, found from data in [93]. ρ represents density.

C_2 could result in numerical instability. Although it is a requirement that $(C_1 + C_2) > 0$ (since this is a Drücker stability criterion [100]) even some negative C_2 values that satisfied this equality were observed to cause simulation failure. Therefore, C_2 was set to zero to prevent problems, considering that the model needed to be robust over the wide input space considered in the sensitivity analysis. This effectively reduced the isotropic matrix term to a Neo-Hookean model. For the same reasons, the sinus was also modelled by a single-term Mooney-Rivlin model, with the assumption of isotropy. A typical set of parameter values calculated from data in [93] is presented in Table 4.1. The resulting theoretical curves were compared against available test data for the leaflet of the AV and also against data from a single-element test in LS-Dyna. The result for a set of uniaxial data gained from Li *et al.* [99] is shown in Figure 4.11. The circumferential test data is well represented by the ST model and captures the linear post-transition region well. The radial data is not quite as well-fitted, due to the necessity of using a single-term Mooney-Rivlin model. However the fit is still acceptable. The element test results for the coefficients calculated in the ST model follow the theoretical curves exactly (as would be expected) — this confirms that the material model in LS-Dyna behaves in the same way as described here.

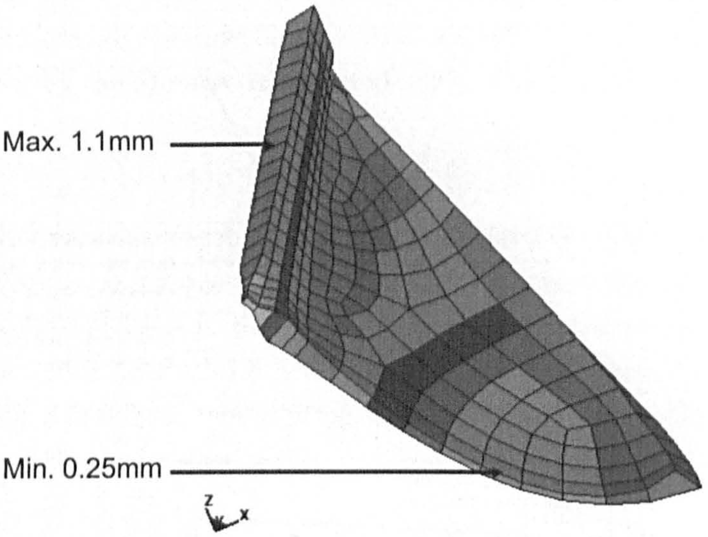


Figure 4.12: The varying thickness of the AV leaflet (half-section of single leaflet shown).

Other Aspects

The thickness of the AV leaflet is also known to be different over different areas of the surface. Thubrikar [75] gives the thickness of the leaflet as varying from 0.25mm in the belly region to 1.33mm near the leaflet edge (the coaptation region). This aspect was included in the model (see Figure 4.12). Finally, to account for contact with neighbouring leaflets, a sliding contact plane was defined on the coaptation face of the model. A reduced input deck for the wet model can be found in Appendix C.

4.4 Conclusions

A description of the aortic valve and of two finite element models representing its response to physiological loading has been given here. Modelling the aortic valve has been shown to be motivated by the desire to improve the longevity of replacement prosthetic valves. Each of the two models developed here is novel in its own way. In the first case, the dry model is a basic pressure-driven simulation with linear material properties. However, since it is specified almost entirely by adjustable parameters inside the Ansys environment, it is possible to automatically vary geometric parameters that would otherwise be impossible to investigate. Therefore, it is not intended to be a state-of-the-art heart valve model, but rather a less-sophisticated simulation that has been designed with a very high degree of flexibility. In this respect therefore, it can be regarded as novel.

In the second case, the wet model is a state-of-the-art aortic valve model that is at least comparable with any of the contemporary models available in the literature. It is fully fluid-driven, exploits cyclic symmetry and includes a sophisticated constitutive material model that has been validated on available material data. It therefore represents a typical model that would be created with the intent of investigating the aortic valve. The following chapter will investigate the effects of uncertainties in these models, which are typically disregarded in the literature.

Chapter 5

Uncertainty Analysis of the Aortic Valve Models

Chapter 4 outlined in detail two finite element models of the aortic valve. The first (dry model) was a somewhat simpler model, but was built almost entirely on a parametric basis. The second (wet model) can be considered a state-of-the-art biomechanical model, including a constitutive hyperelastic material model, anisotropy, and fluid-structure-interaction. This chapter now draws together the Bayesian UA/SA technique outlined in Chapter 3 by testing it on the two AV models. This allows evaluation of the Bayesian techniques, but can also be considered as a novel investigation from the biomechanical perspective: a formal UA/SA has not hitherto been performed on an AV model, and the awareness of uncertainty analysis is only beginning to filter into the biomechanics community. Certainly Bayesian SA is novel in its application to biomechanical models.

Results from the "dry" model were found to be somewhat problematic, due to stress concentrations. However the results are still presented here since useful lessons were learned from the perspective of the emulator.

Section 5.1 gives details of the uncertainty analyses to be performed on both models. This is followed by the results of these analyses in Section 5.2.2. Finally, a discussion of the results and the implications is given in Section 5.3, followed by conclusions.

5.1 Approach

Even with an emulator-based approach, it is impractical to consider all possible uncertainties, given the exponential increase in computational expense with the number of input parameters (the so-called "curse of dimensionality"). Strictly speaking, a FE model can have an almost unlimited number of uncertain input variables (since the properties of each element could be considered as separate variables), but it is sufficient to select only

Parameter	Name	Mean	Standard Deviation	Data source
Elastic modulus of sinus (MPa)	*Es*	2.45	0.8	[79]
Thickness of sinus shells (mm)	*Ts*	1.3	0.325	[73]
Elastic modulus of leaflet (MPa)	*El*	2.05	0.65	[73]
Thickness of leaflet shells (mm)	*Tl*	0.35	0.0085	[75]
Thickness of commissure shells (mm)	*Tc*	1	0.25	[73, 75]
Radius of base (mm)	*Rb*	12.7	0.675	[75]
Angle of coaptation (degrees)	ϕ	31	3	[75]
Leaflet separation in relaxed state (ratio)	*Fltrad*	6	2	Subjective

Table 5.1: Uncertain input parameters, dry AV model.

a subset of these inputs that are of interest: perhaps those that are suspected to cause large uncertainties in the output, or those that are known only to be within wide ranges. Accordingly, a number of input parameters were chosen to be investigated for each model, based on the capabilities of the model and available data. It is not therefore claimed that an exhaustive UA has been performed here, however the investigation could easily be extended if required.

5.1.1 Dry Model

For the dry model, eight input parameters were chosen to be examined that were found to vary substantially from one simulation to the next in a comprehensive literature review on the topic — these are presented with references in Table 5.1. Ranges could then be collected for each parameter based, in a loose sense, on the "belief" of existing literature, in the absence of formal statistical data. Other parameters were collected from studies quoted by Thubrikar [75]. Each parameter was assigned a normal distribution for mathematical tractability. It is acknowledged that this is an assumption that cannot be fully justified; however, it is required that either a uniform or normal distribution be assigned (see Chapter 3). Normal distributions appear to be more suitable in this case, since the size of biological components tends to be normally-distributed (the height of human beings is a well-known example of the normal distribution [101]). This is thought to be due to a manifestation of the central limit theorem (see e.g. [10]).

The parameters listed were chosen to reflect a range of types of uncertainty, and to have little or no correlation with each other. Several are geometric parameters, but also some material parameters are investigated, despite the fact that material parameters in the more complex model are investigated in Section 5.1.2. This is to examine the effects of the assumption of a simple material model compared to a more complex model. The elastic modulus of the sinus was assigned a distribution based on values collected in a study in [79]. The leaflet modulus was assigned a range based on the variation between the material properties in the circumferential and radial directions [73], thus examining the effect of assuming isotropy. The base radius and angle of coaptation were gained by measurements taken from a range of specimens [102]. Finally the leaflet separation of the valve is an important quantity, since it determines how "open" the valve is in its relaxed state, which can influence stress patterns later in the analysis. This quantity is an epistemic uncertainty, so it was thought to be of interest to investigate its effect — a subjective range was assigned to this parameter. It is acknowledged that the means and variances assigned to parameters here are somewhat subjective, due to the scarcity of data. Note that the fact that the thickness and elastic modulus of the leaflet and sinus have been found to vary from simulation to simulation suggests that they are epistemic uncertainties. However, additionally they are sources of aleatory, since they vary naturally from one individual to the next.

The input distributions were used to create a maximin Latin hypercube design of 250 training points (see Section 2.1.3). This input matrix was then run through LS-Dyna to evaluate the model 250 times and provide corresponding output points, which were then used to train the GP emulator. The model took around 3 minutes to solve on a 3GHz dual-core processor, giving a total analysis time of about 12 hours. Although this is not nearly as intensive as the wet model (see later), the emulator approach allows this to be run in a very manageable time (an equivalent Monte Carlo simulation for the same analysis would take days or even weeks).

Note that the UA/SA of this model is used as an example to demonstrate the practical points of Bayesian uncertainty analysis in Appendix B.

5.1.2 Wet model

For the wet model five input parameters were chosen to consider as uncertain, listed in Table 5.2. These parameters were selected because they are sources of significant uncertainty and interest, data is available concerning their upper and lower limits, and they can be easily varied within the LS-Dyna input file.

The sources of uncertainty in the wet model were as follows. Thubrikar quotes the ranges of the transition stretch λ^* and post-transition modulus *C5*, which are known to vary from person to person [75]. *C1S*, being the neo-Hookean coefficient of the background

Parameter	**Name**	**Min.**	**Max.**	**Data source**
Fibre elastic modulus in leaflet	*C5L*	3MPa	10MPa	Thubrikar [75]
Straightened fibre stretch	λ^*	1.13	1.2	Thubrikar [75]
Neo-Hookean coefficient in sinus	*C1S*	0.1MPa	1MPa	Ranga [79]
Thickness of sinus shells	*TS*	0.9mm	1.2mm	Grande [73]
Pressure difference scaling factor	*PDIF*	0.625	1.25	Coni [91]

Table 5.2: Uncertain input parameters, wet AV model.

matrix, is varied according the range specified by Ranga *et al.* in a study on uncertainty in material properties of the aorta [79] (after conversion). The pressure scaling factor *PDIF* is obtained from well-established variations in the cardiac cycle, as explained in Section 4.2.2 previously [91]. This contrasts with the normally-assumed 120/80mmHg (10.7/16kPa) healthy blood pressure (equivalent to *PDIF*=1). Finally, the thickness of the sinus and aorta is known to vary significantly, both within different parts of the same valve, and from one person to the next (see for example the study by Grande *et al.* [73]). A conservative range of 0.9-1.2mm was therefore assigned to this parameter.

A space-filling sample design based on Latin Hypercube sampling was created to generate training data, consisting of 100 model runs over the 5-dimensional input space. On a 3GHz dual-core machine, the solution time for one model run was around three and a half hours, which meant that the uncertainty analysis required around two weeks of CPU time. This immediately demonstrates the attractiveness of using a GP-based emulator; similar Monte Carlo simulations have been shown to take orders of magnitude longer in CPU time [41], which would have made this UA impractical without the use of brute processing force. The output quantities of interest were extracted after each model run.

5.2 Results

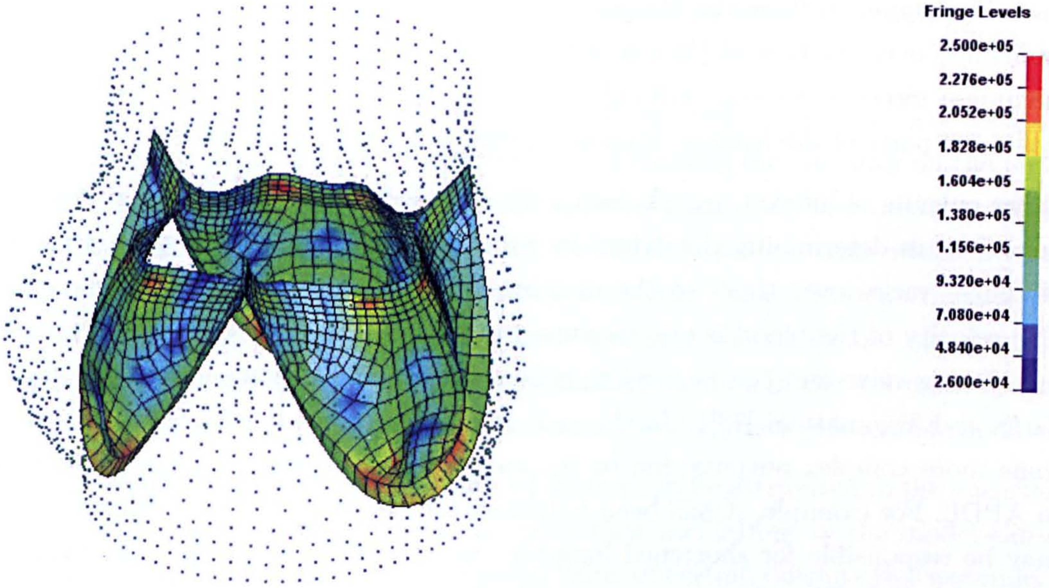


Figure 5.1: Illustration of typical Von Mises stress (Pa) distribution over dry AV model at half-open stage; sinuses removed for clarity.

Interpreting the results of a sensitivity analysis can be a difficult process, since the quantity of data is very substantial. For each run of a dynamic model, a large quantity of data is output for each requested time state — for example, in both the wet and dry models, field data is recorded at 100 equally-spaced intervals throughout the solution. This is in order to give sufficient "time resolution" to capture high-speed events in the movement of the valve. For each of the 100 states, nodal coordinates, velocities, element stress and many other quantities are recorded (see for illustration the stress plot in Figure 5.1) for each of the thousands of elements in the model. This creates a problem, namely that the GPs used here can only consider a scalar output. The issue then becomes one of dimensional reduction — in other words, it is required to find a scalar quantity for each run that represents some output of interest. The method of Bayesian SA allows us to investigate the variations of these outputs in detail and with greater accuracy than the raw data. However, plots with raw data are still used to illustrate trends that are not visible otherwise (since data is lost by reducing each run to a single number). In practice, a combination of emulator data and raw data works well.

5.2.1 Model Outputs

It is useful at this point to examine existing AV modelling literature to see which outputs are of most interest in an AV model, since these should be the outputs examined in the

SA. An obvious choice is stress in various parts of the valve — this has been investigated in almost all of the literature (see for example [85, 86, 89] amongst many others), given that stress is important in determining the durability of the valve. This must be reduced to a single quantity however, despite the fact that stress varies over the elements of the model and over the time of the simulation. The approach adopted here is to consider the maximum stress in various parts of the model, for example the maximum stress in the leaflets or parts of the leaflets, from any point in the simulation time.

Other outputs of interest include leaflet displacement (investigated in e.g. [85]), which is important in determining the extent to which the valve opens at a given pressure. This of course varies over time, so the opening speed of the valve may also be considered. The velocity of the blood is also measured in the wet model since it indicates a correctly functioning valve and can be used to validate the model to some extent (see for example Sacks and Yoganathan [78]). In the case of the dry model it was possible to investigate some more complex outputs due to the automatic post-processing capabilities available in APDL. For example, it has been indicated that buckling in the AV leaflet on opening may be responsible for shortened longevity and therefore should be minimised as much as possible [83]. Therefore, a measure of leaflet buckling was created and recorded at the point where the leaflet reverses its curvature. This was defined as a function of the angles that successive elements made with each other in the XY plane at the point of curvature reversal.

5.2.2 Dry Model Results

Output Uncertainties

Parameter	**Mean**	**Standard deviation**	**COV (%)**	**Relative RMS error (%)**
Dispmax	12.5mm	0.161mm	12.8	4.9
Thalfop	0.00194s	0.00122s	63.4	37
Sigmax	636kPa	355kPa	55.8	41
Wig	0.161	0.0157	9.8	7.24

Table 5.3: Posterior output uncertainties for dry AV model. *Dispmax*=Max. displacement of outer leaflet node; *thalfop*=time until valve is half-open; *sigmax*=max. stress anywhere in the valve at any time state; *wig*=measure of leaflet buckling.

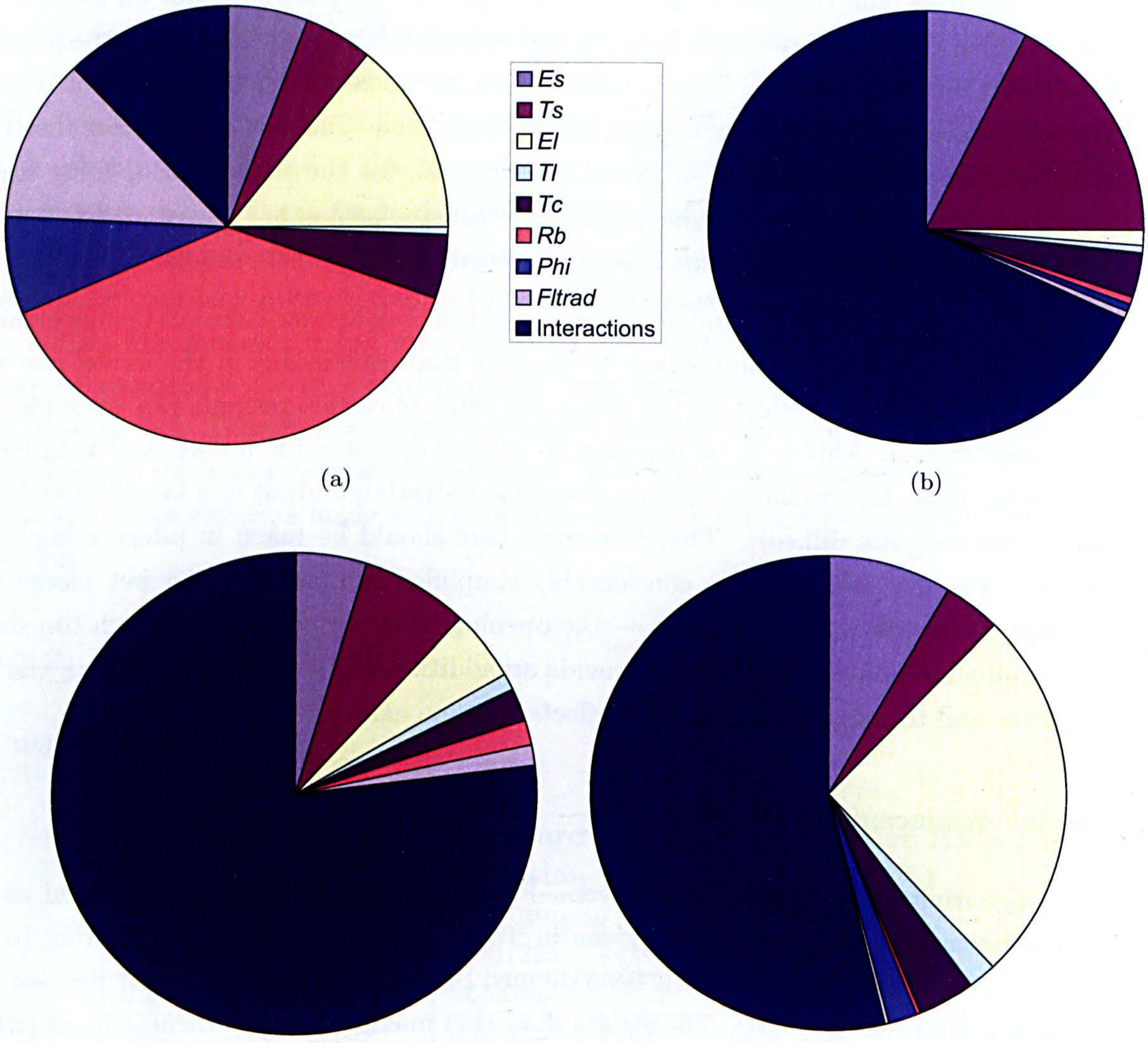
Starting first with the uncertainty measurements, the posterior expected value of the mean and standard deviation for each output parameter is shown in Table 5.3 (see Equations 3.45 and 3.51 respectively). This gives an overall idea of the uncertainty in each output, given the uncertainty assigned to each input. To illustrate the comparative uncertainty between outputs, the standard deviation has also been presented as a percentage of the mean (known as the *Coefficient of Variation* (COV)). It is evident that some of the uncertainties here are substantial - *Thalfop*, which represents the speed of opening of the valve, has a COV of 63%. Similarly, the maximum stress has a COV of about 56%. The first conclusion here is thus that the uncertainties are indeed significant and the UA seems an appropriate measure.

To check the prediction accuracy of the emulator, Cross Validation (CV) was used. This process involves removing one of the data points and training the emulator on the remaining set, then evaluating the error between the removed data point and the corresponding prediction by the emulator. This is repeated for all data points, from which a relative Root-Mean-Squared (RMS) error value can be calculated. Table 5.3 shows that the RMS error is quite high, possibly to an unacceptable level, for the stress and opening times. These outputs are however examined in a more realistic fashion in the wet model (Section 5.2.3). The results are highlighted here to illustrate the diagnostic process of the GP.

Although much of the uncertainty in the model output can be attributed to the uncertainty in the inputs, it is equally important to consider that oscillations in the model can also cause a kind of random effect on the maximum value of certain outputs (for example, the maximum stress). This is to be expected in a large dynamic FE model, and is in fact a limitation of the GP emulator: finding a sensible univariate output in a large dynamic FE model can be quite difficult. Therefore some care should be taken in interpreting these results. The dry model is also considerably simplified compared to the wet model and some of the results are not realistic — the opening time, for example, is much too short. The results are however left here to provide an additional case study for use with the GP emulator and to investigate geometric effects to some extent.

Leaflet displacement

Some sensitivity measures are now given. Figure 5.2 shows the MEIs expressed as pie charts for each of the four outputs given in Table 5.3 (for the equations relating to the MEIs, see Equation 3.63, which has been divided by Equation 3.51 to standardise; see also Appendix B for more details). The results show that maximum leaflet displacement (which controls the blood flow through the valve) is shown to be possibly the most "well-behaved" of the outputs tested, with nearly 90% of output variance explained by main effects of parameters (see Figure 5.2(a)). This is regarded as "well-behaved" because when high-order interactions explain a large proportion of the variance (see e.g. the maximum stress, Figure 5.2 (c)), in the experience of the author, this is often evidence of some "noise" in the data that cannot be easily explained by the parameter variations. Or in other words, there is no definite trend in the data, which may be caused by model oscillations overshadowing the effects of parameter variations. This does however illustrate a great advantage of SA that might often be overlooked: it is possible to deeply explore the workings of a model



(a) (b)

(c) (d)

Figure 5.2: MEIs and interactions with respect to (a) *Dispmax*; (b) *thalfop*; (c) *sigmax*; (d) *wig*.

and identify possible faults when expected trends do not appear. The low proportion of variance due to MEIs is evident in valve opening time (Figure 5.2(b)), maximum stress (Figure 5.2(c)), and to a lesser extent in the leaflet buckling parameter *wig*. All three of these model outputs should therefore be treated with a little caution. Further evidence of the tenacity of the emulator fit in these outputs is given by the CV values in Table 5.3. The emulator fit could be potentially improved by adding more data points, however if there is reason to suspect that there is a problem with the model output, this is unlikely to help greatly. The wet model eliminates many of the problems here because leaflets are damped to some extent by the fluid, and oscillations are reduced.

Returning to the MEIs of *dispmax*, the remaining variance is explained quite evenly between first-order interactions. Notably, the main effect of *Rb* accounts for nearly 40% of output variance. This is not unexpected, since a wider aortic radius creates an overall wider valve, which allows leaflets to open further. The elastic modulus of the leaflet also contributes significantly, as does the initial opening width, both of which are to be expected since they effect the shape and stiffness of the leaflet and therefore the opening of the valve.

To examine the effects of the inputs on leaflet displacement in more detail the main effect plots of each input are presented - see Figure 5.3. Note that the main effect plots are generated using Equations 3.47 and 3.51, setting $p = i$, where i is the input variable of interest (see also Appendix B). The plots here show random draws from the posterior main effects of each input (200 draws are shown for each variable). The dispersion of the main effect plots here is quite small, suggesting that there is a good emulator fit to the training data. One of the most noticeable trends is the strong, roughly linear main effect of *Rb*. As expected, a wider valve allows a greater leaflet displacement. Negative trends are evident in *Es*, *Ts*, *El*, *Tl* and *Tc* since an increase in any of these parameters will either impede the flexibility of the sinus or the leaflet. The responses are generally quite linear, with the exception of *Es* and *El*.

Stress

In contrast with maximum displacement, examination of Figure 5.4 shows that there is a much greater uncertainty surrounding the main effect plots of *sigmax*, given that the main effect draws are considerably more dispersed than those of *dispmax* - indeed, less that a quarter of output uncertainty can attributed to the main effects of input parameters (see Figure 5.2 (c)). There are however some trends that can be extracted: leaflet thickness and stiffness both cause roughly linear responses in stress, but it is largely insensitive to both ϕ and *Rb*. It is also again evident that the properties of the sinus are causing a nonlinear effect. Both *Es* and *Ts* cause quite strongly nonlinear trends as well. This pattern will be seen to be prevalent also in the wet model, and is discussed further in

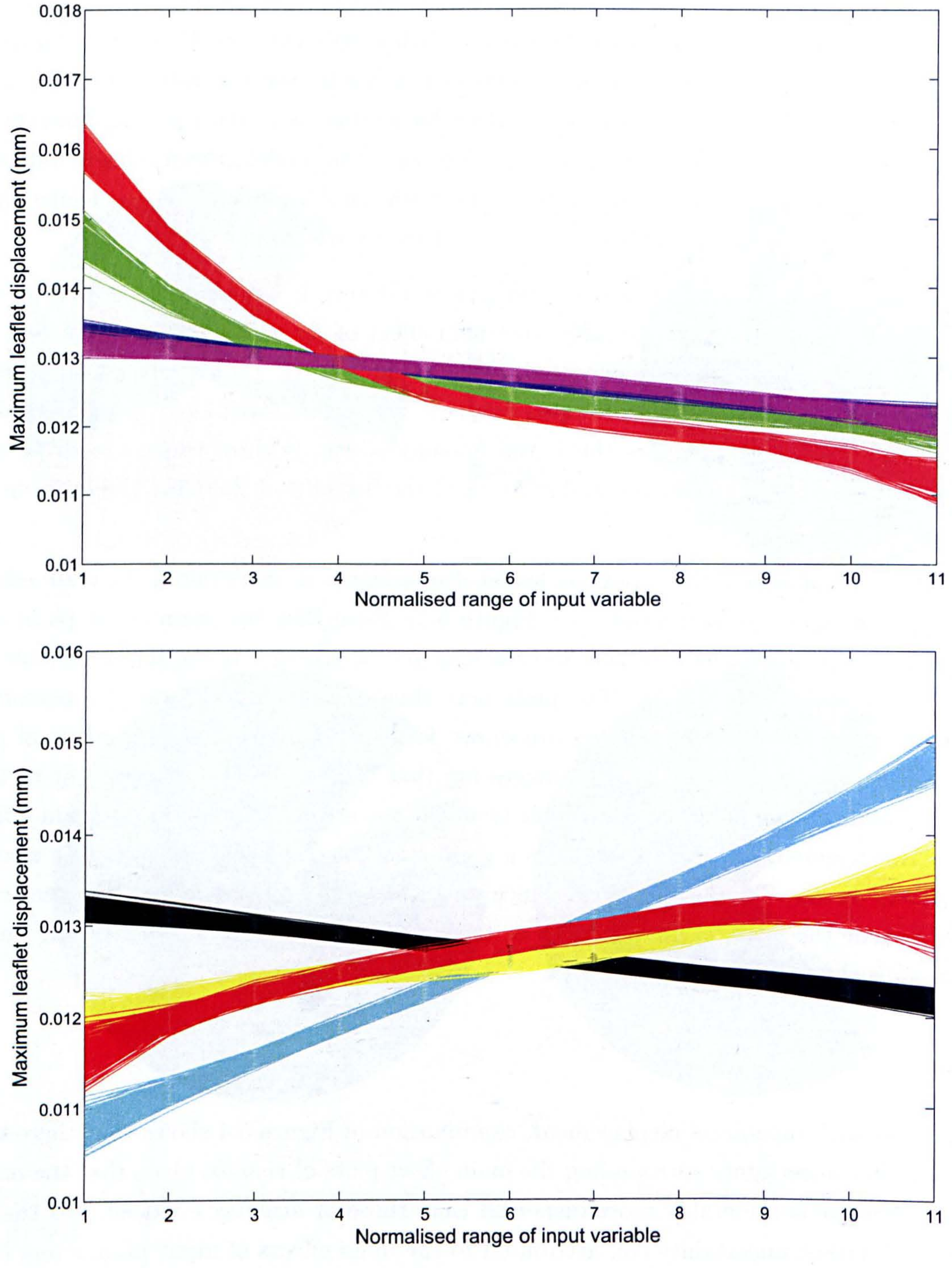


Figure 5.3: Main effect plots of inputs with respect to *dispmax*. Green=*Es*; blue=*Ts*; red (upper)=*El*; pink=*Tl*; black=*Tc*; light blue=*Rb*; yellow=ϕ; red (lower)=*Fltrad*. Input variables are normalised over the sample range for comparison.

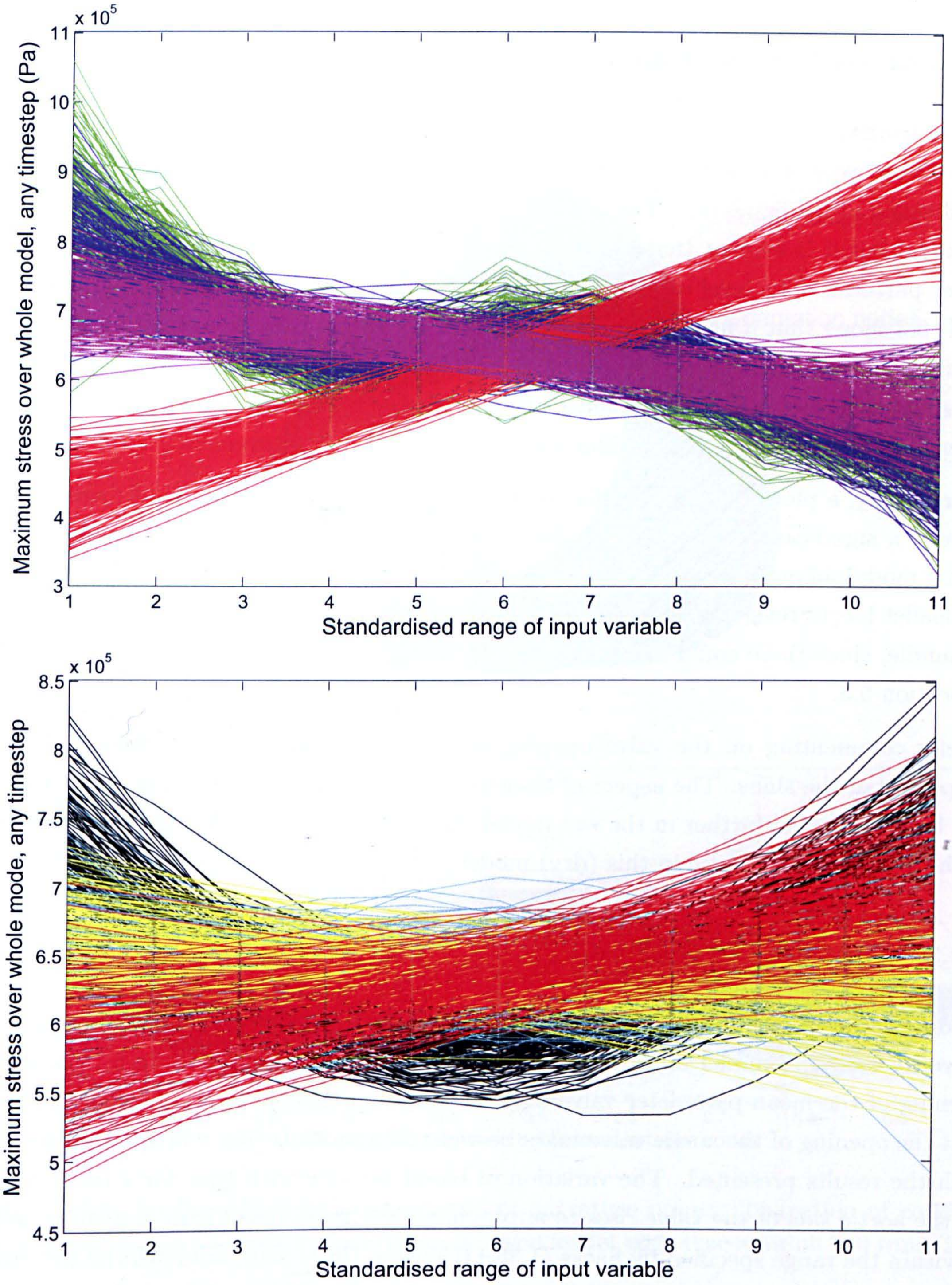


Figure 5.4: Main effect plots of inputs with respect to *sigmax*. Green=$\grave{E}s$; blue=Ts; red (upper)=El; pink=Tl; black=Tc; light blue=Rb; yellow=ϕ; red (lower)=$Fltrad$.

Section 5.3. It is inappropriate to draw further conclusions from the stress results given the level of dispersion in the main effects.

Buckling and Opening Time

To illustrate the variation in the opening mechanism of the valve, the positions of the leaflet edge nodes were recorded at the reversal-of-curvature point for each run. These curves are illustrated in Figure 5.5. The general trend is a smooth opening curve with minimal buckling, but it is evident that some values of the input parameters cause very distorted leaflet patterns. The level of buckling is quantified numerically by the *wig* parameter — Table 5.3 shows that it has the lowest COV of all, at around 10%. Investigating the causes of this, using the MEIs in Figure 5.2, the dominant parameters are the leaflet stiffness and the sinus stiffness, because the deformation of the leaflet is a function both of its own stiffness and the stiffness of the sinus that is constraining its movement.

Interestingly, a plot of the main effect of *El* with respect to *wig* (Figure 5.6) reveals that there is a significant nonlinearity in the response of *wig* to *El*. It is evident that below elastic moduli of around 500kPa the level of bucking increases dramatically. The fact that the leaflet has to reverse its curvature creates responses that might be difficult for the GP to handle, since there could be abrupt changes in the response. This is discussed further in Section 5.3.

Briefly commenting on the valve opening time, it is predominantly influenced by the properties of the sinus. The aspect of sinus material properties influencing valve opening will be investigated further in the wet model (Section 5.2.3) given that the opening speed of the valve is not realistic in this (dry) model.

5.2.3 Wet Model Results

Before the uncertainty analysis, the wet model was run at mean parameter values in order to verify that the model output agreed with experimental data. Figure 5.7 shows the opening of the mean-parameter valve over a simulation time of 20ms. Thubrikar reports that the opening of the aortic valve takes between 17 and 20ms [75], which is in agreement with the results presented. The variation of blood velocity with time for a central point on the aortic side of the valve rises to approximately 1.2m/s (see Figure 5.9 later), which is within the range specified by Sacks (1.35±0.35m/s, [78]). This validates the wet model to a certain extent, therefore it will provide a considerably more realistic simulation of the valve opening than the dry model.

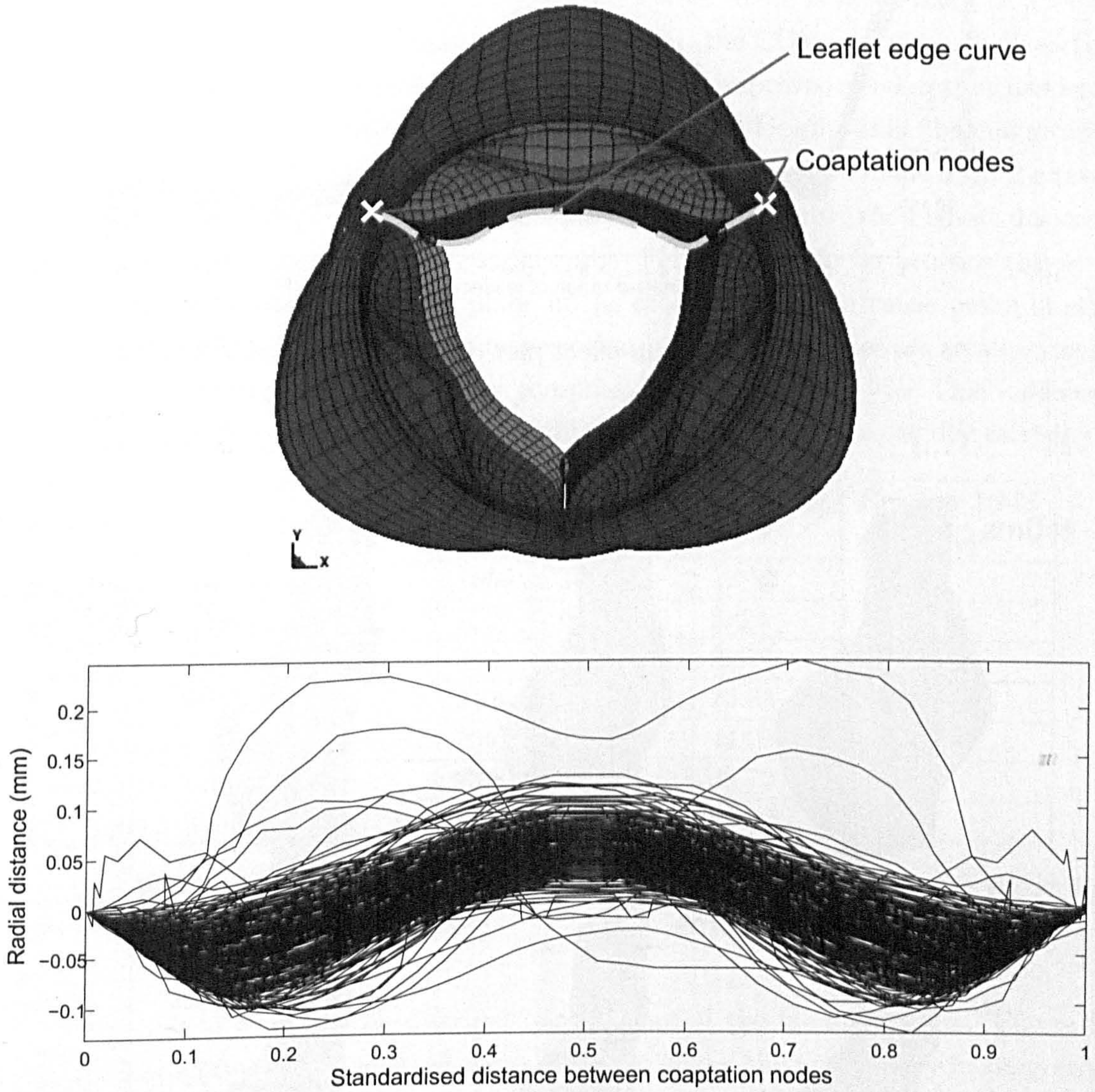


Figure 5.5: Leaflet edge curves at reversal-of-curvature point: illustration of coaptation nodes and leaflet free edge (upper); standardised leaflet edge traces for all 250 runs (lower).

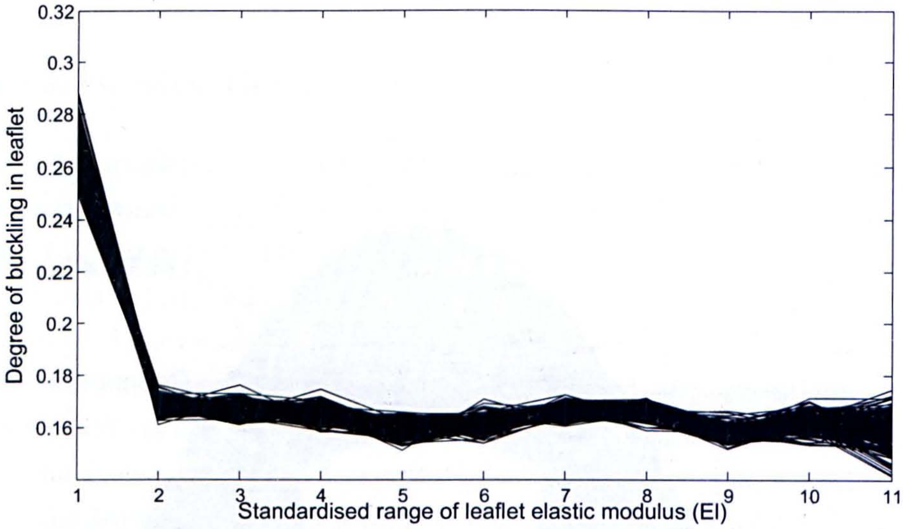


Figure 5.6: Main effect plot of *El* for output *Wig*

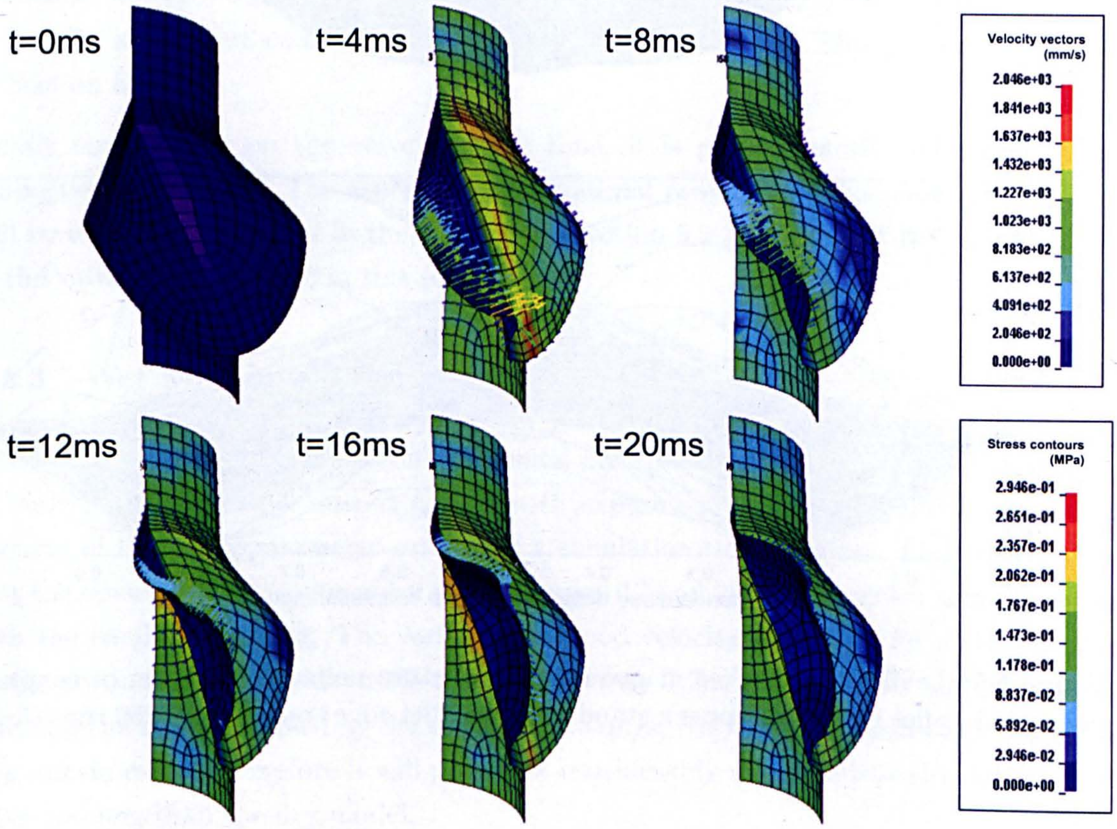


Figure 5.7: Von Mises stress and velocity vectors of wet AV model at intervals up to 20ms (fluid not shown).

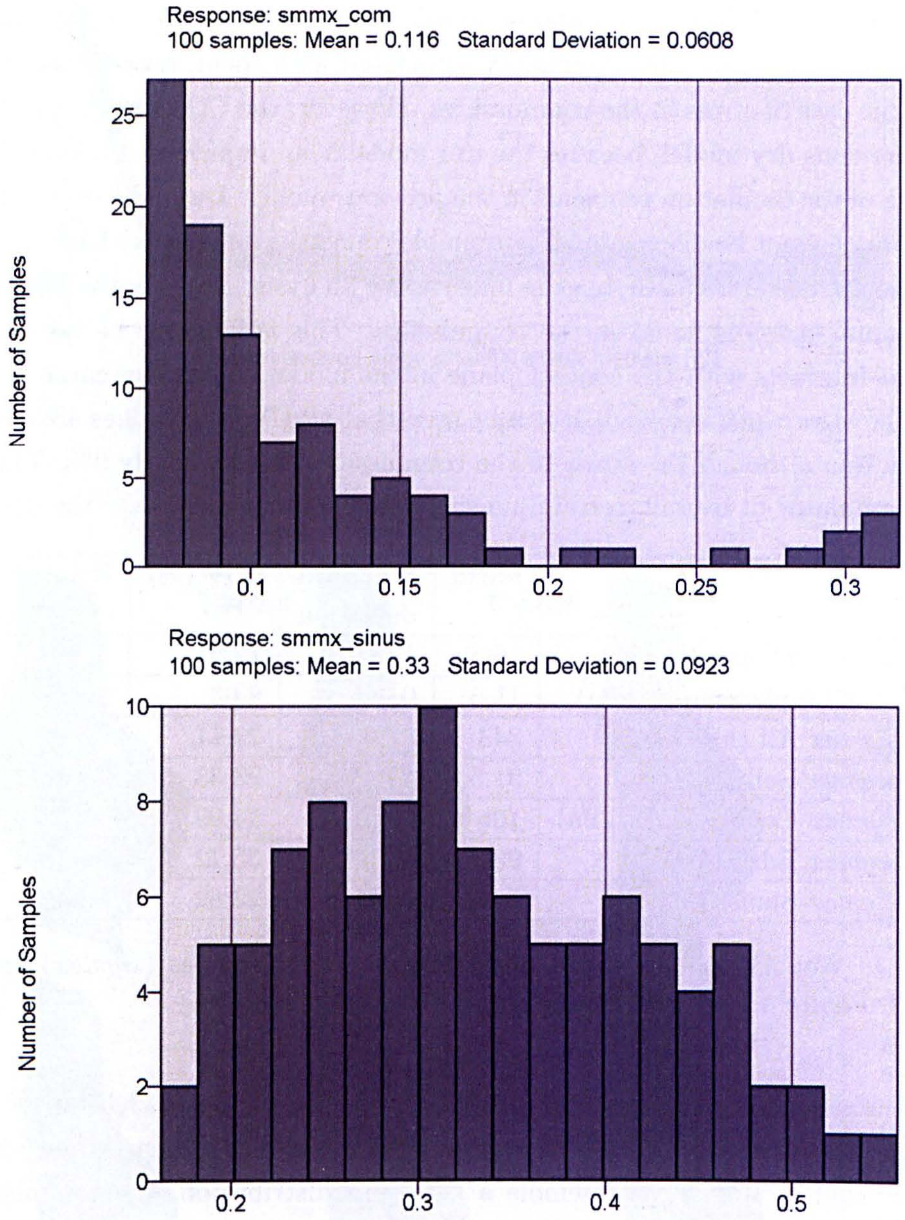
Output Uncertainties

The uncertainty in the model outputs, which are expressed as posterior means and variances in the same manner as the dry model results, are presented in Table 5.4. Note again that these quantities are based on the emulator and therefore provide a more informed estimate than the sample mean and variance. From the results it is clear that all outputs measured have a significant uncertainty associated with them, rising as high as a CoV of 44% in the case of stress in the commissures. However, the COVs are not as high as those in the previous dry model, because the wet model is an improved version that has ironed out some of the oscillation problems in the previous model. Despite this, the magnitude of uncertainty cannot be discounted if reasonable conclusions are to be made from the results of the model, therefore justifying the uncertainty analysis. Notably the highest dispersion in maximum stress is found in the commissure. This is likely to be because this is the area that interacts with the contact plane in the model, which can cause peaks in stress due to the algorithms involved. It can be seen that all CV error values are small, typically less than 5%, although the stress in the commissure rises to nearly 9%. This indicates a reliable emulator fit overall, certainly a significant improvement over the dry model.

	Mean	Standard deviation	CoV (%)	Relative RMS error (%)
Blood Velocity (mm/s)	1069	133.9	12.52	3.88
Max Displacement (mm)	11.5	0.995	8.63	1.79
sigmax All (kPa)	343	77.9	22.71	3.49
sigmax Belly (kPa)	91.6	21.4	23.33	4.17
sigmax Commissure (kPa)	105	46.2	44.09	8.57
sigmax Edge (kPa)	93.8	33.5	35.72	4.34
sigmax Sinus (kPa)	342	80.8	23.62	4.23

Table 5.4: Wet AV model output uncertainties: stress regions (*sigmax*) are those illustrated in Figure 4.1.

It is interesting to examine the distributions of some of the model outputs. Figure 5.8 illustrates histograms of two selected stress outputs: whilst the uncertainty in sinus stress could be said to somewhat resemble a Gaussian distribution, the commissure stress is highly skewed towards lower values, with a scattering of much higher values. Assuming that the sampling strategy is giving a reasonable representation of the output distribution, these distributions must be explained by the nonlinearity of the model with respect to the uncertain input parameters.



Response: smmx_com
100 samples: Mean = 0.116 Standard Deviation = 0.0608

Response: smmx_sinus
100 samples: Mean = 0.33 Standard Deviation = 0.0923

Figure 5.8: Histograms of maximum stress in commissure (top), and sinus (bottom) from raw data. Stress values in MPa.

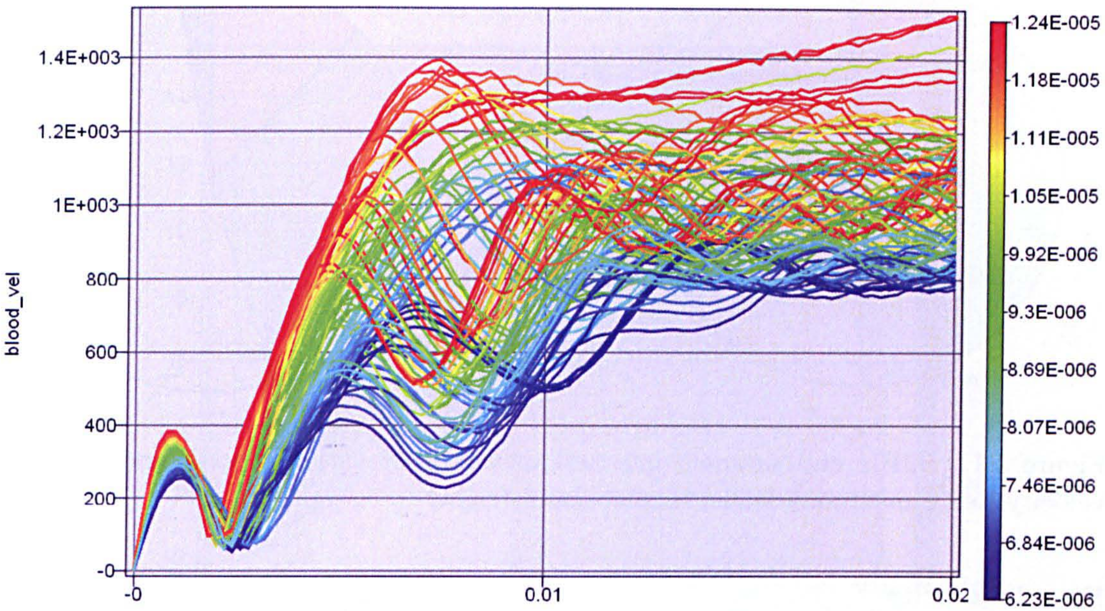


Figure 5.9: Variation of blood velocity with time for all model runs. Colours represent *PDIF* values.

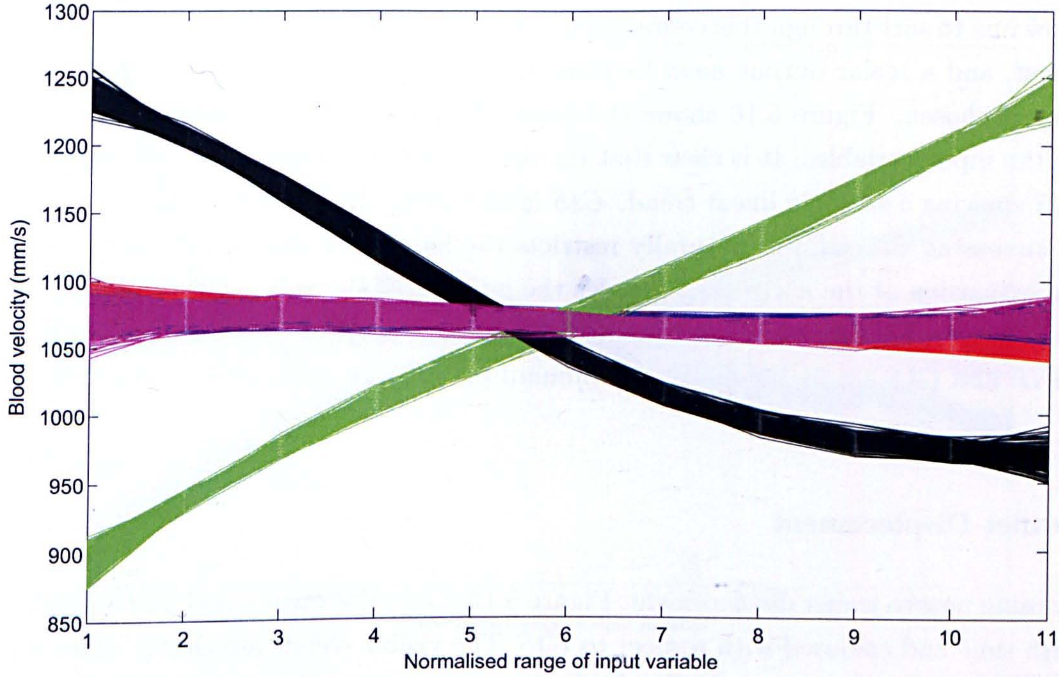


Figure 5.10: Main effects of inputs on maximum blood velocity. Red=*TS*, blue=*C5L*, green=*PDIF*, black=*C1S*, pink=λ^*.

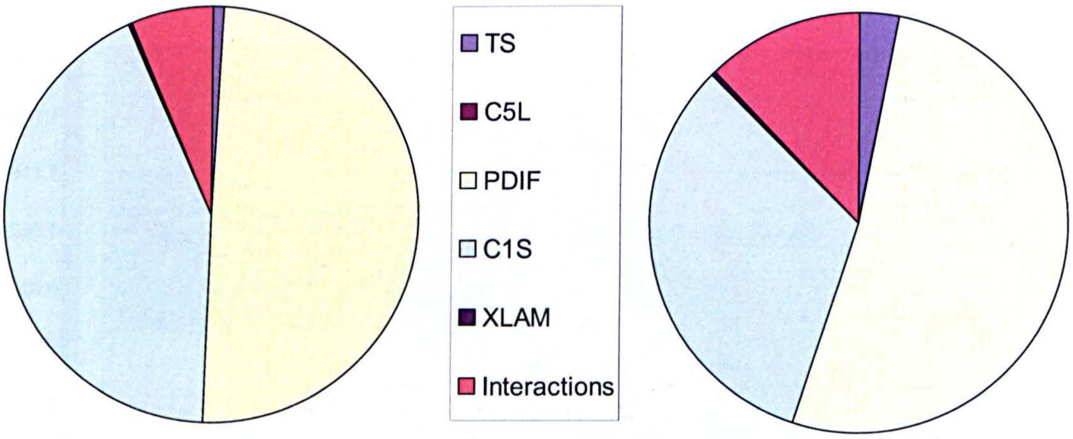


Figure 5.11: MEIs and summed interactions for input variables with respect to blood velocity (left); maximum leaflet displacement (right)

Blood Velocity

The blood velocity will now be considered in more detail: Figure 5.9 shows the variation of blood velocity with time as a result of varying *PDIF*, generated from raw data from each of the 100 model runs. The clear (and expected) trend is that increasing the pressure difference produces a higher blood velocity, though the fact that some low *PDIF* values produce high blood velocity is due to variation in other parameters. The main effect plots allow one to sort through this confusion to a large extent, although the variation with time is lost, and a scalar output must be considered. In this case the maximum velocity over time is chosen. Figure 5.10 shows the main effect plots of the maximum blood velocity for the input variables. It is clear that the dominant parameters are *PDIF* and *C1S*, with *C1S* showing a strongly linear trend. *C1S* is effectively the stiffness of the sinus and aorta, so increasing the stiffness naturally restricts the blood flow somewhat, since it constricts the expansion of the aorta and impedes the opening of the valve. Other parameters have little effect. This trend is confirmed by examining the MEI pie chart in Figure 5.11 — *PDIF* and *C1S* are by far the most dominant parameters, with little interaction variance.

Leaflet Displacement

Turning now to leaflet displacement, Figure 5.12 shows the variation of leaflet displacement with time and coloured with respect to *C1S*. The visible trends are that a stiffer sinus and aorta cause the valve to open to a lesser extent and more slowly. This is according to expectations that the flexibility of the aorta is crucial in the opening of the valve [83]. This time, however, the equivalent main effect plot (figure 5.13) reveals trends that are not evident in the plots from the raw data. There is a strong nonlinearity evident in the

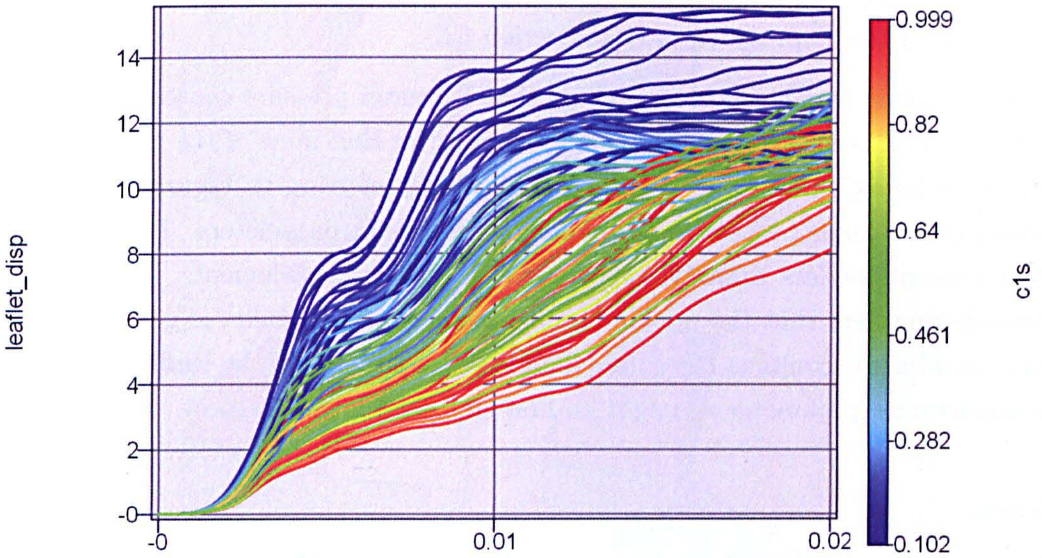


Figure 5.12: Leaflet displacement in the x-direction; variation with time and *C1S*.

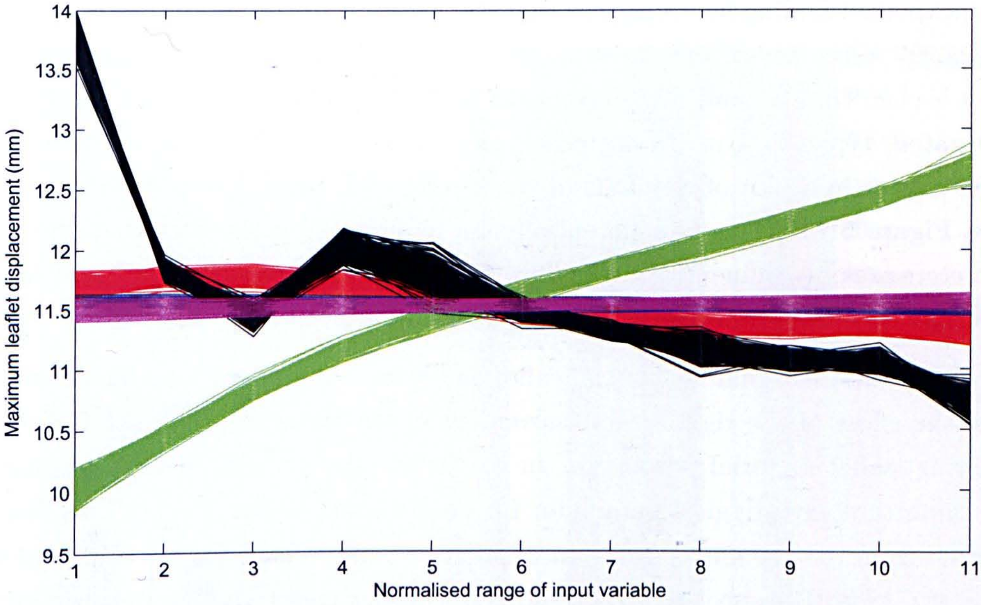


Figure 5.13: Main effects of inputs on maximum leaflet displacement. Red=*TS*, blue=*C5L*, green=*PDIF*, black=*C1S*, pink=λ^*

main effect of *C1S*, such that some intermediate values of sinus stiffness cause greater leaflet displacements than lower values, contrary to the general (and expected) trend. This may be because of the existence of more than one opening mechanism for the AV. A discussion of this is provided later in Section 5.3.

Other trends in leaflet displacement are that a greater pressure causes the leaflet to open further, which is unsurprising. It is notable however that none of the material parameters that were investigated have any significant effect; referring to Figure 5.11, the pressure difference and sinus properties are by far the dominant parameters. The material parameters account for less than 1% of variance in leaflet displacement. It is perhaps worth remembering here that the maximum displacement of the leaflet is generally dictated by point at which it contacts the sinus, therefore the stiffness of the leaflet itself may not be as important a parameter as might be first supposed in this respect.

Stress

One important model output is the stress, particularly in the leaflet, since high stresses are responsible for prosthetic valve failure. Figure 5.7 shows Von Mises stress contour plots for the valve at various stages of its opening for a typical parameter set. The highest stresses tend to occur in what will be called the "edge" region, this being the region where the leaflet joins the sinus. In this area the leaflet is forced to bend significantly on opening, creating high stress concentrations.

One unexpected finding however, was that the leaflet material properties do not appear to significantly affect any of the stress outputs. Examining pie charts of the MEIs in Figure 5.14, it is clear that λ^* and *C5L* account for very little of the variance in all of the outputs investigated, typically 1 or 2% including interactions with other parameters. The trend is even clearer in a plot of the TSIs of the inputs with respect to stress in model regions — see Figure 5.15. This can potentially be explained by the fact that these material parameters are only influential in the fibre direction, and stress perpendicular to the fibres is consequently unaffected. Furthermore, the uncertainty in the input λ^* is quite small.

It would seem then that although leaflet material parameters have some influence on stress, the effect of the rigidity and movement of the sinus vastly overshadows the effect of varying leaflet material properties. In particular, the stiffness of the sinus is easily the most important investigated parameter for determining leaflet stress. This result agrees with that from the dry model and that of previous studies [83]. Figure 5.16, a plot of main effects of *C1S* with respect to stress, shows the underlying trends in more detail: one can see that low values of *C1S* cause a marked rise in stress in the three areas shown. After some examination of stress plots of various runs, it seems that when the sinus is below a certain stiffness, the blood pressure is enough to cause the sinus to expand to the point where it starts to stretch the leaflet between its coaptation nodes and cause additional

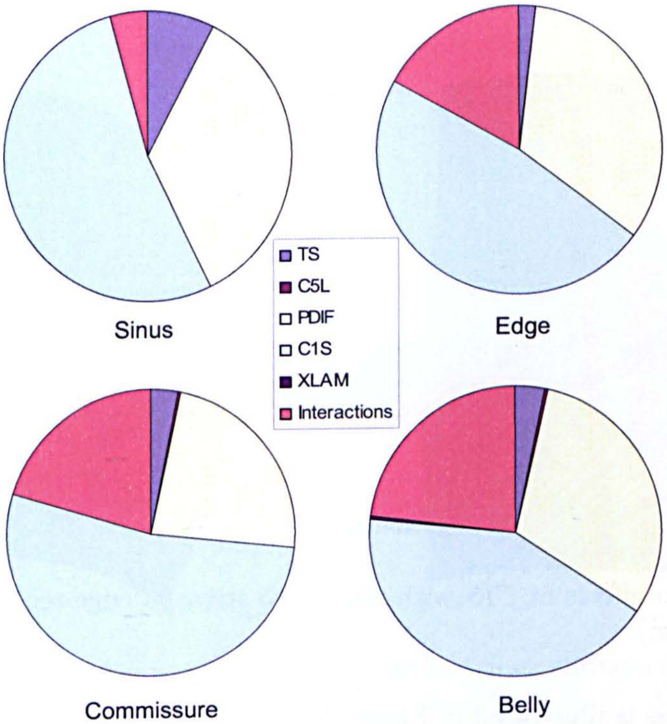


Figure 5.14: Sensitivity indices of model inputs and interactions for stress in various regions of wet AV model. "XLAM" here refers to λ^*.

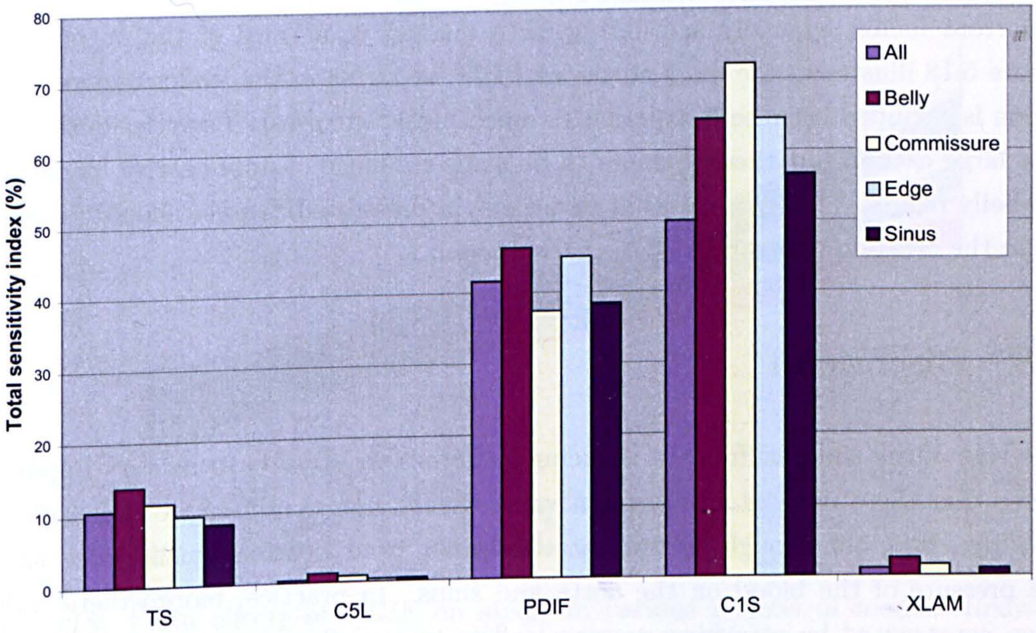


Figure 5.15: TSIs of model inputs with respect to Von Mises stress in various model regions. "XLAM" here refers to λ^*.

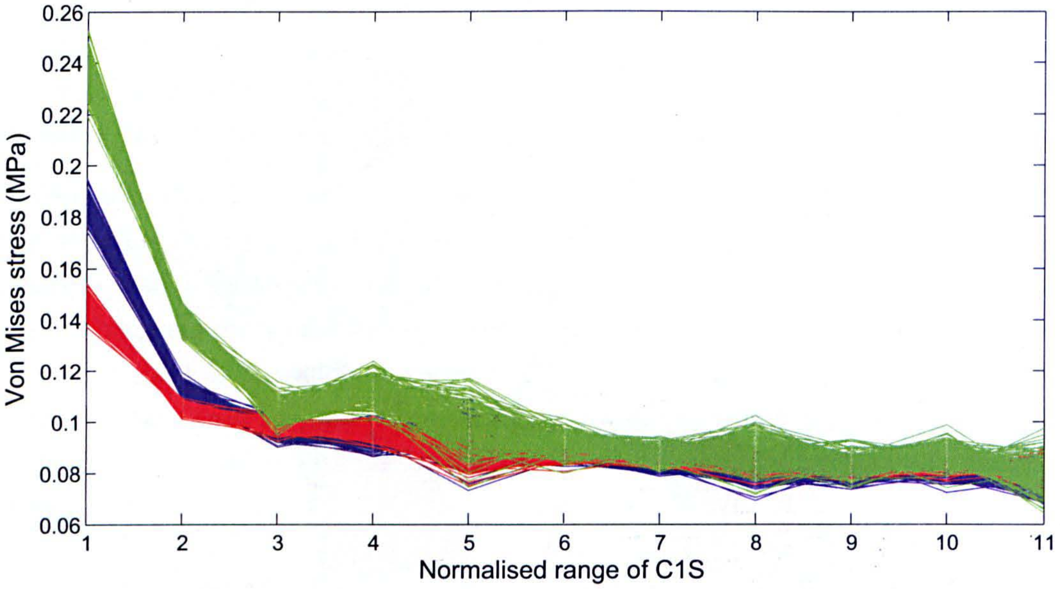


Figure 5.16: Main effects of *C1S* with respect to stress in edge region (blue); belly (red); commissure (green).

tensile stress. This is illustrated in Figure 5.17. The two plots show the distortion of the valve at roughly the same point in the opening process — the low sinus stiffness causes the valve to expand to the point where the leaflet is under tension, constraining further expansion. Conversely, the stiffer sinus expands considerably less and the leaflet is not under tension, but rather under some compression, and opens with a modest amount of buckling at the reversal-of-curvature point. Finally, the pressure difference was also an important factor, typically accounting for a quarter to a third of the output variance. Figure 5.18 illustrates the main effects of *PDIF* on stress in the leaflet regions (the sinus region is excluded because it experiences much higher stresses). The relationship is linear to a large extent, but there appears to be some evidence of nonlinearity, particularly in the belly region. This suggests that variations in pressure difference should be accounted for in the creation of a robust aortic valve model.

5.3 Discussion

The role of the sinus stiffness in reducing leaflet stress appears to be very important. It seems that there may be an optimum value that is not so stiff so as to cause extensive buckling, but stiff enough to prevent the leaflet from bearing tensile force caused by the pressure of the blood on the aorta and sinus. In practice, bioprosthetic valves are often constructed by attaching porcine leaflets to an inflexible stent - this arrangement undoubtedly causes unnecessarily high stress due to buckling. More recent valve designs have used a stentless approach, which has been shown to significantly increase survival

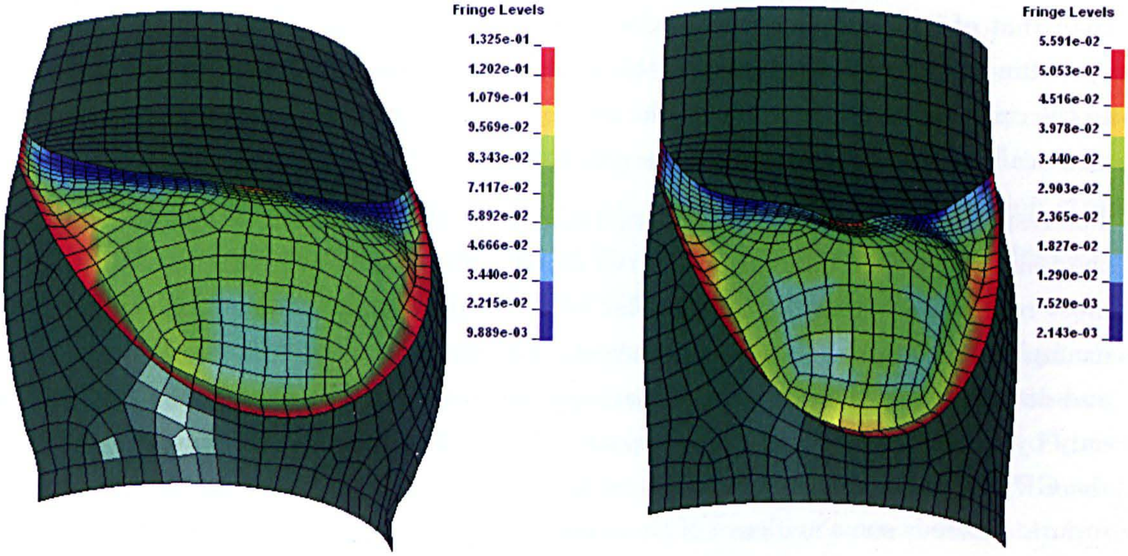


Figure 5.17: Opening mechanism in the leaflet for low sinus stiffness (left) and high sinus stiffness (right).

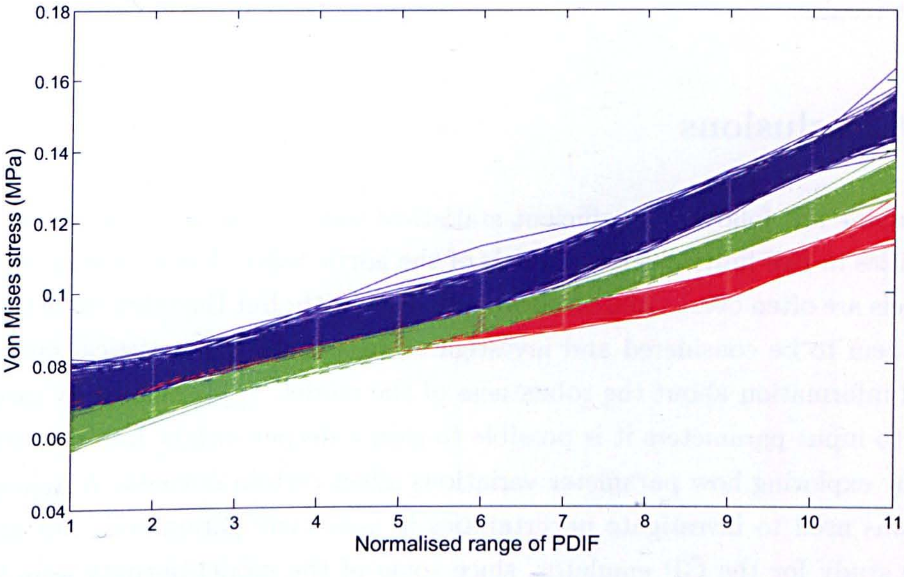


Figure 5.18: Main effects of *PDIF* on stress in various regions of model: Red=belly, blue=commissure, green=edge.

rate [103], doubtless (at least in part) for the reasons outlined above. In the stentless approach, a section of porcine aorta may also be transplanted, which should be closer to the optimum flexibility. However, the blood pressure in pigs is known to be higher than that of human beings [104], which could mean that the valve is not working under optimum conditions. A solution to this could be to specifically design the stent to expand to the optimum amount based on the average blood pressure of the patient, although the practicality of doing so might be questionable.

The response of the model to some inputs, particularly to the stiffness of the sinus, raises the issue of possible bifurcations in the model. Since the valve consists of a leaflet that must reverse its curvature, it is similar to a "snap-through" problem, which is a typical example of a bifurcation in displacement. For some responses therefore, such as stress and displacement, there may exist two separate, very different regimes that are separated only by a small difference in input space. This kind of response is inherently difficult for the GP to handle, since it must assume a smooth relationship between the output and its inputs. There is some evidence of these non-smooth responses in the model — see Figures 5.13 and 5.16. This has given motivation for further work — see Section 5.4.

From an additional angle, it is interesting to test the effect of sinus rigidity up to the point of constraining the coaptation nodes completely, since this simulates the effect of a rigid-stented bioprosthetic valve. This is interesting both from a biomechanical perspective and from the point of view of the capability of the emulator to model a bifurcating system. This problem provides the motivation for work in Chapters 8 and 9, since performing an uncertainty and sensitivity analysis using a GP on a bifurcating model may produce incorrect results.

5.4 Conclusions

This chapter has followed an efficient statistical method for analysing uncertainties and sensitivities in two finite element models of the aortic valve. Uncertainties in biomechanical models are often overlooked or informally dealt with, but Bayesian uncertainty analysis allows them to be considered and investigated in a thorough statistical fashion, yielding detailed information about the robustness of the model. Additionally, by examining sensitivity to input parameters it is possible to gain a deeper insight into the working of the model by exploring how parameter variations affect certain outputs. A less-sophisticated model was used to investigate uncertainties in geometric parameters, but also as a useful case study for the GP emulator, since some of the model outputs were shown to be somewhat unreliable. From this it was shown that sensitivity analysis can be employed as a useful diagnostic tool — little or no trends found in the data can be evidence of a poorly-constructed or unstable model.

From the biomechanical perspective, valve opening has been shown to be highly dependent on the stiffness of the aorta/sinus in both models, reinforcing the supposition that the expansion of the aorta is intrinsic in the opening mechanism of the valve. It is suggested that there is an optimum aorta/sinus stiffness and that this should be accounted for in the design of prosthesis, including the stentless variety. The effect of valve stiffness has also been shown to be very nonlinear, possibly to the point of bifurcation. This motivates further work in Chapters 8 and 9. One surprising conclusion was that some of the chosen leaflet material parameters from the constitutive material model (the transition stretch and post-transition modulus) were not found to be very influential, being overshadowed by the effect of the expansion of the sinus on the buckling of the leaflet.

It is not claimed here that an exhaustive uncertainty analysis has been performed, since the number of uncertain parameters can be extended almost without limit. However, a practical means of investigating selected uncertainties has been demonstrated. Furthermore, it has highlighted the necessity for consideration of uncertainties in AV models, since the dispersion of model outputs is significant. This necessity can presumably be extended to apply to biomechanical models in general, since they are prone to uncertainties for similar reasons to the AV.

Chapter 6

Modelling an Airship

This chapter concerns the design of a new unmanned airship, ongoing at the Department of Aerospace Engineering in the *Politecnico di Torino* in Turin, Italy. The design aims to address certain problems that limit the effectiveness of classical airship designs (see later). To reduce costs in prototypes and testing, finite element models have been integral in the development of the project to date. However, since the sophistication of the models is quite substantial, a concern is that uncertainties in model inputs may have significant effects on the outputs of interest. In the case of the airship, a great number of parameters must be input into a finite element model, such as material properties, boundary conditions and loading conditions. Any number of these can be subject to uncertainties resulting from lack of knowledge, varying operating conditions, or other reasons. As such, a series of uncertainty analyses were performed to investigate the effects of uncertainties in loading and material properties in models created to test proposed airship designs, since design decisions are taken based on results from such models (i.e. if the model outputs are uncertain then it is difficult to make a clear decision about the choice of design). Since some of the models produced here were very substantial in size, the Bayesian SA approach outlined in Chapter 3 is used to conduct the uncertainty and sensitivity analysis.

The work here is used as a further case study for the GP emulator, as well as providing useful data on uncertainty for the airship project. The field of application is briefly outlined in Section 6.1, after which the specific design is described in Section 6.2. Details of the FE models used are given in Section 6.3, followed by a short conclusion. The uncertainty analysis itself is left until Chapter 7.

6.1 Airships and Applications

Airships, a form of "lighter-than-air" (LTA) aircraft, have been in use in some form or another since the 18th century. LTA aircraft are distinguished by the characteristic that their weight is less than (or nearly equal to) the volume of air they displace; thus they are held aloft by buoyancy. Despite clear caveats such as poor manoeuvrability and enormous bulk, airships still retain certain attractive advantages, including the possibility of long flight times for a comparatively low cost, the ability to remain virtually stationary and to take off vertically. They are also relatively environmentally-friendly when compared to fixed-wing aircraft. One interesting area of application is the lifting of very heavy loads to remote locations, although a design has never been fully realised for this to date. Given the potential number of applications, there are still a number of airship projects ongoing around the world aimed at specialist applications. Liao provides a summary of various projects and designs in progress worldwide as of 2009 [105].

One set of applications in which airships perform particularly well compared to heavier-than-air aircraft is that of surveillance, monitoring and communications relaying. Such airships are often relatively small and unmanned, since the applications require lengthy flight times, yet airships are particularly well-suited to this application due to their ability to remain aloft for long periods for a very low cost, and also the stable and relatively vibration-free platform that they can provide for delicate surveillance equipment. Furthermore, since they are practically noise free and use little fuel, they are well-suited for situations where a low environmental impact is imperative, such as in environmental and oceanographic monitoring. However, there are several drawbacks in conventional designs: since airships have very large profiles, they are very susceptible to disturbance from cross-winds, a problem that is particularly undesirable in landing and takeoff. Additionally, at low velocities, the rudders and elevators that are used for manoeuvring are ineffective since they depend on lateral air movement to generate any steering force.

6.2 The *Elettra Twin Flyers*

6.2.1 Design

In order to combat the problems mentioned with conventional airship designs and expand the range of operable weather conditions, a new design has been proposed in [106], called the *Elettra Twin Flyers* (ETF). The ETF falls into the category of *semi-rigid* airships, which means that the lifting gas is contained in flexible membranes that are partially constrained by some rigid structure. The ETF is designed for use in environmental monitoring and surveillance. An early demonstrator design has been constructed and tested — this is illustrated in Figure 6.1.

In continuation of this design, an altered design was proposed, which is the subject of this investigation. Figure 6.2 illustrates the concept of the new design. Note that the design innovations of the ETF are twofold - firstly there are two gas envelopes rather than one (as found in conventional designs), positioned side by side. This allows a smaller profile for the equivalent lift, thereby reducing forces on the ship from lateral gusts. The two

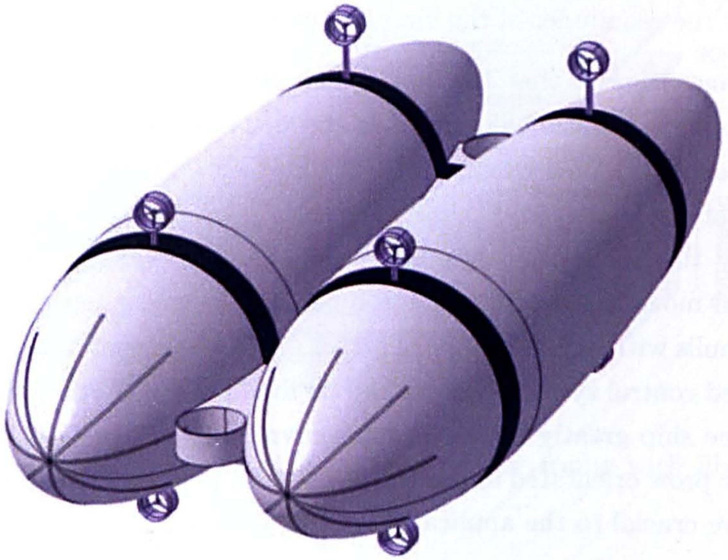


Figure 6.1: Illustration of the ETF demonstrator design

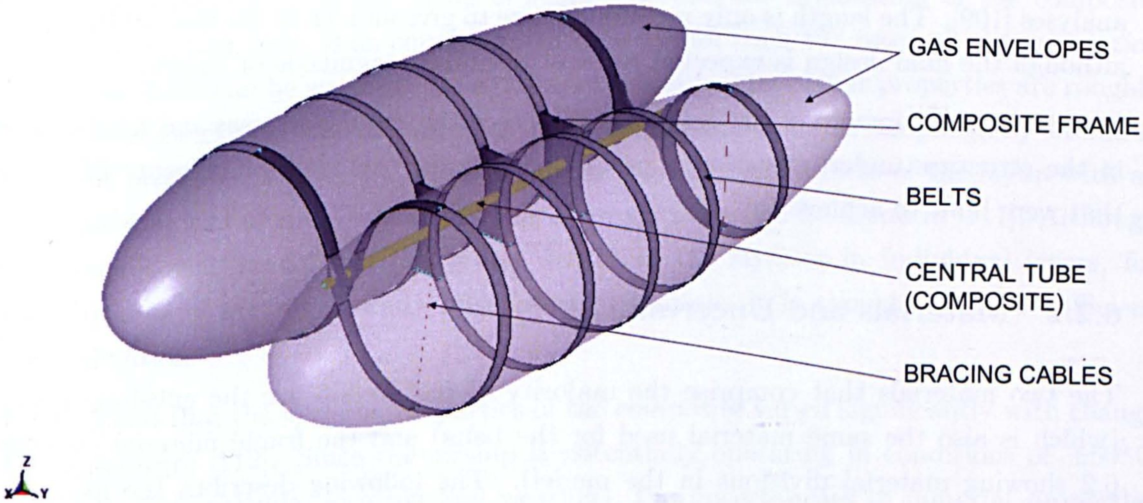


Figure 6.2: An illustration of the ETF design; one balloon is translucent for clarity.

balloons are connected by a rigid central beam that also acts as a platform for affixing the payload and operational equipment. The balloons are attached to the frame by a number of rigid "ribs" that encompass part of the circumference of each, with flexible "belts" running round the remainder of the circumference (see Figure 6.2).

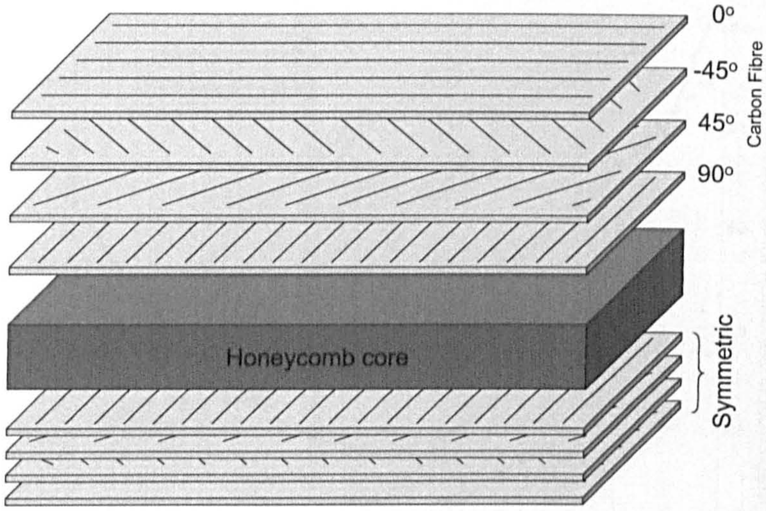
The second innovation is that the ship is moved in all six degrees of freedom by ten directional propellers (rather than movable aerodynamic surfaces) — the advantage being that the ship can be manoeuvred effectively without the requirement of forward motion. The position of the propellers is illustrated in Figure 6.1. During hovering, altitude is maintained by the helium and the propellers orientated along the vertical axis, while during forward movement additional aerodynamic lift is also generated as a result of airflow over the hulls with suitable angle of attack. This movement of the airship is controlled by a specialised control system - details can be found in [107]. Altogether, the innovations should give the ship greatly increased manoeuvrability at low velocities, and allow it to hover with the prow orientated in any direction, even in adverse weather conditions. These capabilities are crucial to the application of the airship.

A series of previous analyses were performed to ascertain the dimensions of the structure [108]. The thickness and shape of the structure is governed by the stresses due to loading, however changing the structure to withstand these stresses will inevitably change the mass, which then will require a greater or lesser volume of helium for buoyancy. This in turn will change the overall size of the airship and change the loading. The final dimensions will therefore be the result of an informal iterative process — the models in this chapter do not represent final designs but parts of the iteration. The models investigated here are 36m in length, this being the length required for sufficient buoyancy to support the total weight of the structure and payload (approximately 3 tonnes), as calculated by previous analyses [109]. The length is only mentioned here to give an idea of the scale of the airship, although the final design is expected to be of a similar magnitude of length.

Part of the development of the design involves ascertaining the stresses and displacements in the structure under operational loading conditions; this chapter presents FE models that were built to achieve this.

6.2.2 Materials and Uncertainties

The two materials that comprise the majority of the airship are the envelope material (which is also the same material used for the belts) and the frame material (see figure 6.2 showing material divisions in the model). The following describes the nature and characterisation of these materials, followed by the uncertainties that are associated with them due to temperature. The material models described here are adopted for all of the FE models in this chapter.



b

Figure 6.3: Arrangement of sandwich layup material for frame with fibre angles shown (layup is symmetric about the honeycomb layer).

Frame

Desirable material properties of the frame material include low density, whilst being as stiff as possible, since large deformations of the structure would alter the aerodynamic response and have unwanted effects on the control system. It was concluded, after some simple preliminary simulations, that the most suitable material would be a sandwich layup, consisting of layers of carbon/epoxy T300 15k/976 composite, either side of an Ultracor® honeycomb core [109, 110]. An illustration of the suggested layup is given in Figure 6.3.

Since the arrangement of the layup is symmetric and the orientation of the composite layers is such that there is an equal number of layers for each 45 ° interval, the orientation of the sandwich can be assumed to be unimportant since the planar properties are roughly equal in all directions. The definition of the material can be further simplified by assuming constant properties across the layers by approximating the stiffness of the layup with an isotropic material model — details of this approximation can be found in [111]. Although this macroscopic model will not reveal details of the stresses in individual layers, for the purposes of an initial study into frame dimensions it is thought to be a sufficient approximation.

It was found that the material properties of the composite varied significantly with change in temperature [112]. Since the airship is potentially operating in conditions of ±50°C (depending on altitude, location and weather), the uncertainties in material properties were expected to have a substantial effect on the response of the structure to loading. Given the variation in the composite material it was possible to calculate macroscopic moduli for the layup over the range of temperatures specified (no temperature-dependent data was available for the honeycomb core, so it was treated as constant). By fitting

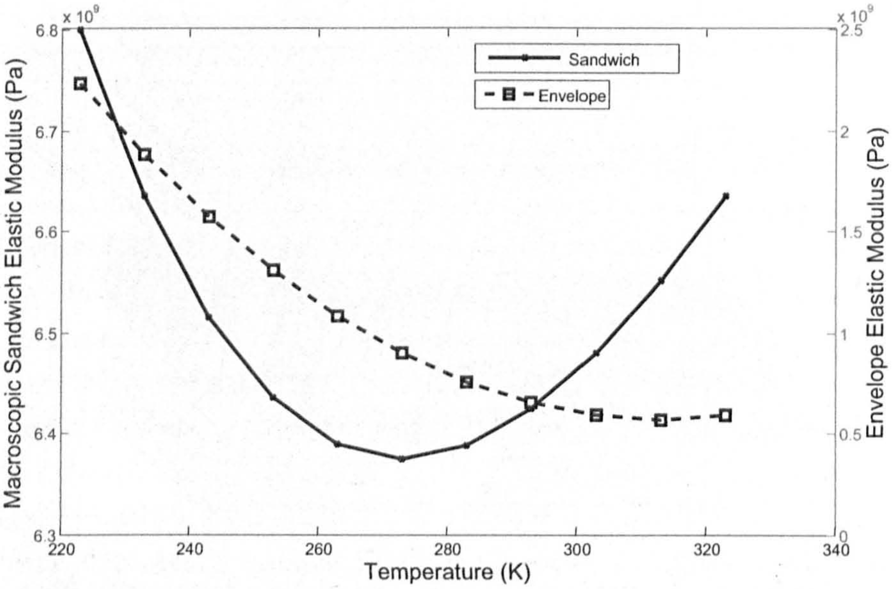


Figure 6.4: Variation of material properties of sandwich layup and envelope material with temperature

a polynomial function to these calculated values, the stiffness could be expressed as a function of temperature, so that temperature could be used as the uncertain input variable, rather than stiffness. This is expressed as,

$$\begin{aligned} E_c &= aT^2 - bT + c \\ a &= 0.137 \\ b &= 75.0 \\ c &= 16910 \end{aligned} \tag{6.1}$$

where E_c is the macroscopic elastic modulus of the sandwich (MPa), T is the temperature (in Kelvin) and a, b and c are specified constants. Figure 6.4 illustrates the variation of stiffness of the sandwich layup with varying temperature.

Envelope

The envelope material was required to possess a high tensile stiffness with minimal weight. Additionally, it should have low permeability (to helium) and be resistant to degradation from UV radiation. A number of candidate materials were under consideration, and one of these (Nylon 6) was selected and tested to failure at a range of temperatures. Figure 6.5 shows examples of material tests at two temperatures within the range. Although the material is not perfectly linear, it was considered sufficient to model it as a linear elastic material for the purposes of investigating temperature-based uncertainty. Preliminary runs of the model (in Section 6.3.1) at nominal parameter values suggested that envelope

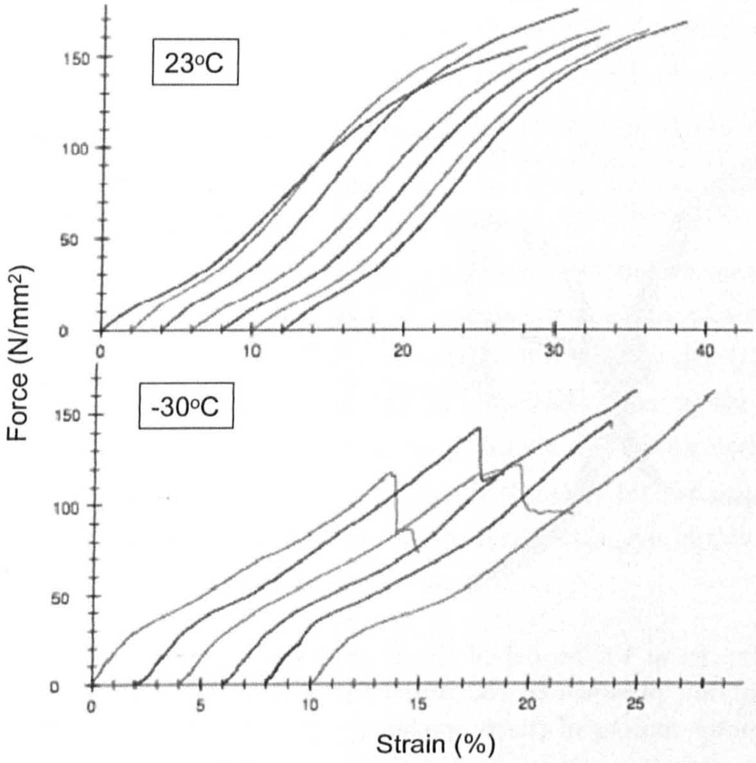


Figure 6.5: Selection of material tests performed on samples Nylon-6 envelope material. Successive tests have been shifted along the strain axis for clarity only; all tests start at zero strain.

strains were $< 5\%$, therefore it was not thought necessary to consider plastic deformation or failure.

Given the linear material model for the envelope, a relationship could be established between temperature and elastic modulus, in MPa (also presented in Figure 6.4), governed by the quadratic equation in Equation 6.1 with the coefficients $a = 0.205$, $b = 128.5$ and $c = 20680$.

6.3 The Airship Models

The modelling of the airship was performed to reveal the stress and deformation of the airship under normal and extreme loading conditions. As mentioned, control of the structure is performed by a number of directional propellers, (currently) based on the assumption that the structure is rigid. This is an important point because even minor deformations and rotations of the arms on which the propellers are located can have effects on the control system that are hard to predict. One main objective of the FE analysis was thus to quantify deformation in certain parts of the structure under a variety of loading condi-

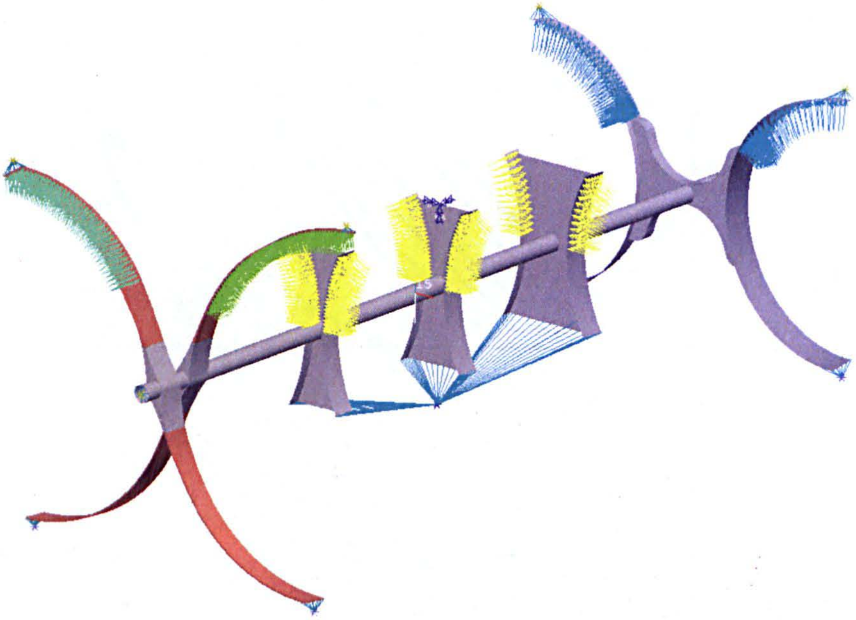


Figure 6.6: Structural FE model of the ETF frame. Loading from the balloons is considered by applying pressure to the underside of the arms (see arrows). Blue asterisks represent the point masses of the propellers.

tions, and to ensure that it would not deform to an unacceptable level. Furthermore, it was important to know whether the structure could withstand various loads without failure, and make informed decisions regarding, for example, thicknesses and types of material.

Published literature concerning airship modelling is scarce, probably due to the small number of airship design projects in existence, and the fact that the heyday of airship design was many years before the rise of computer modelling. Furthermore, the simulation of airships poses particular difficulties to the modeller, given that it involves interactions between flexible membranes and stiff structures, aerodynamic loading, and is unconstrained in all degrees of freedom. Of the few available articles, Bessert et al. [113] examined the effects of aeroelasticity in a classical airship design by coupling and iterating between structural and fluid solvers. Liu et al. [114] performed a similar analysis, concluding that aeroelasticity has an effect on the pressure distribution of a conventional airship balloon. Li et al. [115] performed an analytical analysis of the aerodynamics of a flexible airship. However, the author is not aware of any consideration of uncertainties in airship modelling; furthermore the novel twin-balloon design of the ETF is hitherto untested concerning dynamic loading.

Three models were constructed to investigate the ETF design; first, a preliminary linear static model investigated deformations on the structure without consideration of the presence of the membrane (see Figure 6.6) — this was used to estimate structural dimensions for the more sophisticated models. Although the membrane itself was not present, the buoyancy loads were applied to the frame as pressures to the underside of the upper arms.

The structure was constrained by a technique known as *inertia relief*, which has been successfully used in previous airship structural analyses [113]. In order to keep deflection to an acceptable level for the 36m configuration, the sandwich material was required to be 38mm thick. This thickness was then used in the construction of the following, more sophisticated models. The static model will not be mentioned further since it did not feature in the uncertainty analysis and is only used as a preliminary model.

The second model was a nonlinear model which included the membranes and investigated the uncertainties in pressure and material properties as a result of temperature. This will be referred to as the "dry" model. Finally, a full FSI model (the "wet" model) considered the effect of gusts of wind incident at different angles to the airship. To deal with nonlinearities in the latter two models, the solution was performed in the primarily explicit-dynamic code LS-Dyna. A similar approach was used for the wet model to that of the wet AV model in Chapter 4, where Eulerian fluid elements overlapping the Lagrangian mesh are used to model the effect of a gust impact.

6.3.1 Nonlinear Dry Simulation

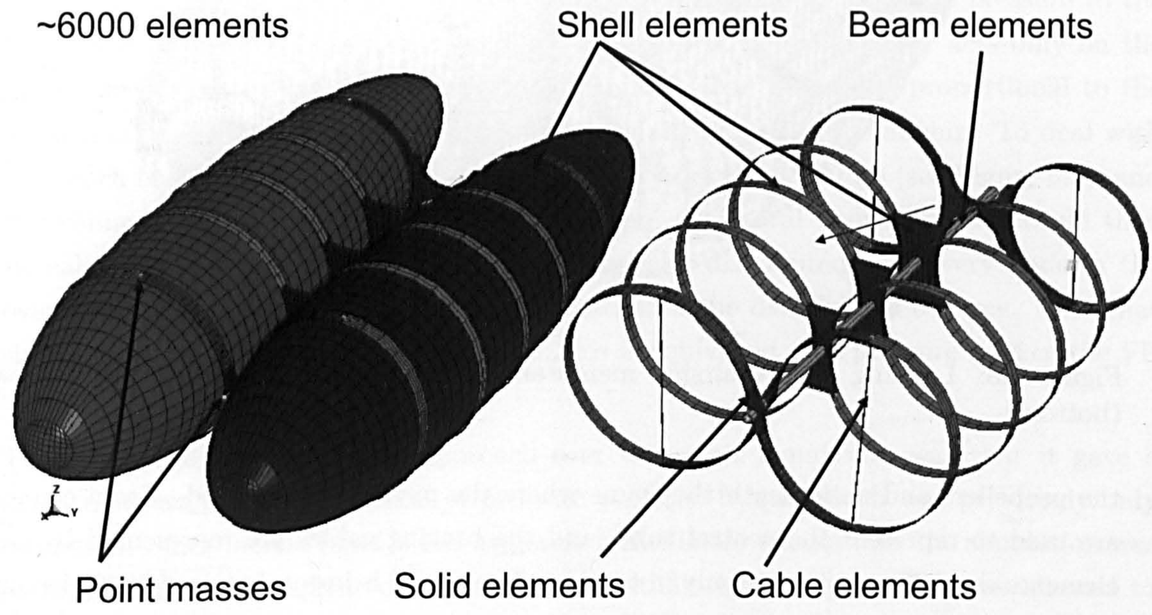


Figure 6.7: The mesh of the airship FE model, balloons removed for clarity (right).

An illustration of the dry model mesh is shown in Figure 6.7. As can be seen, the two balloons are now present, which provides a more realistic model of the rigidity of the airship. The model consists primarily of shell elements for the membrane, belts and most of the frame — the thick sections of the frame near the axis are actually hollow. Solid elements are also used where necessary. Point masses are positioned in the locations of

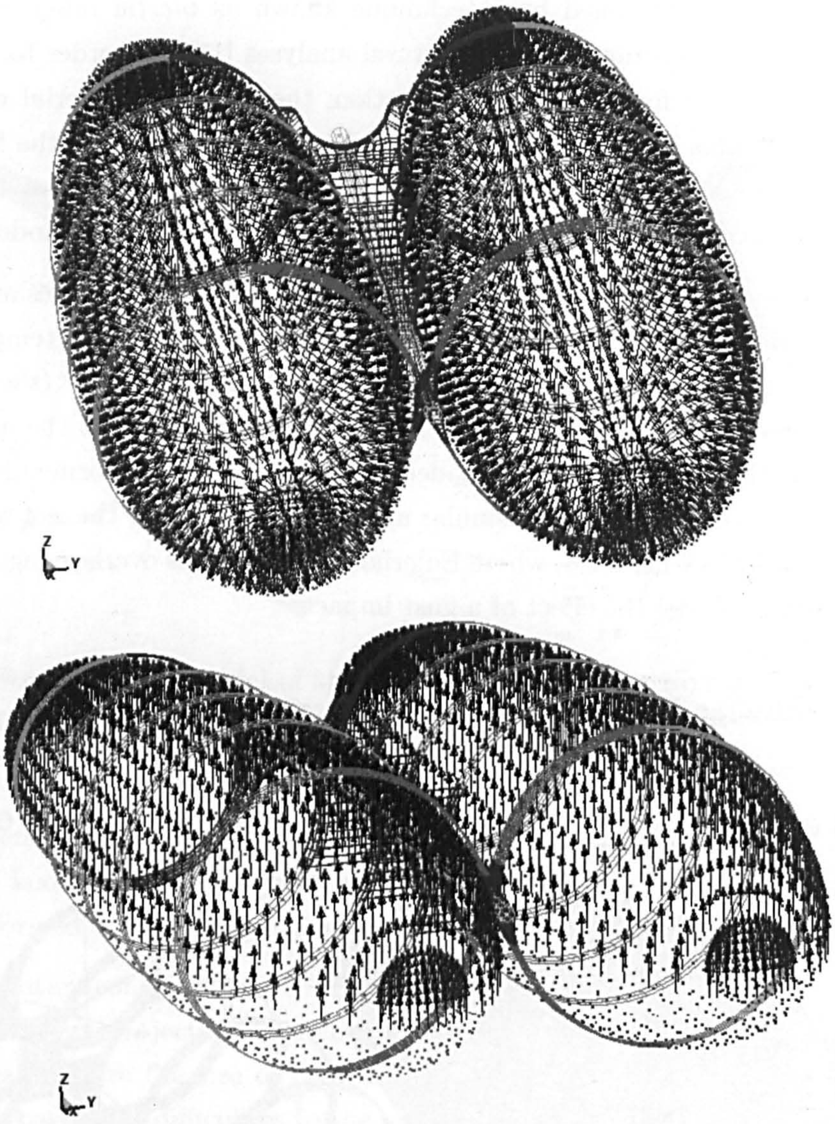


Figure 6.8: Loading on the airship membranes: pressure loads (top); buoyancy loads (bottom).

the propellers and underneath the frame where the payload is attached. Beam elements are used to represent the central tube, and the bracing cables are represented by cable elements that offer resistance only in tension. The model is in total around 6000 elements.

Although as previously mentioned, the material could be treated as linear, the inclusion of the gas envelopes required that a nonlinear solver be used. The reason for this was that in flight the envelopes are in a pre-stressed state. To determine this state prior to further simulation, it was necessary to "inflate" the membranes, that is, to estimate the un-inflated size and apply a pressure so that the balloons expanded into the frame. On contact between balloon and frame nodes, coupling occurred to unify the structure. The contact algorithms can only be performed by a nonlinear solver.

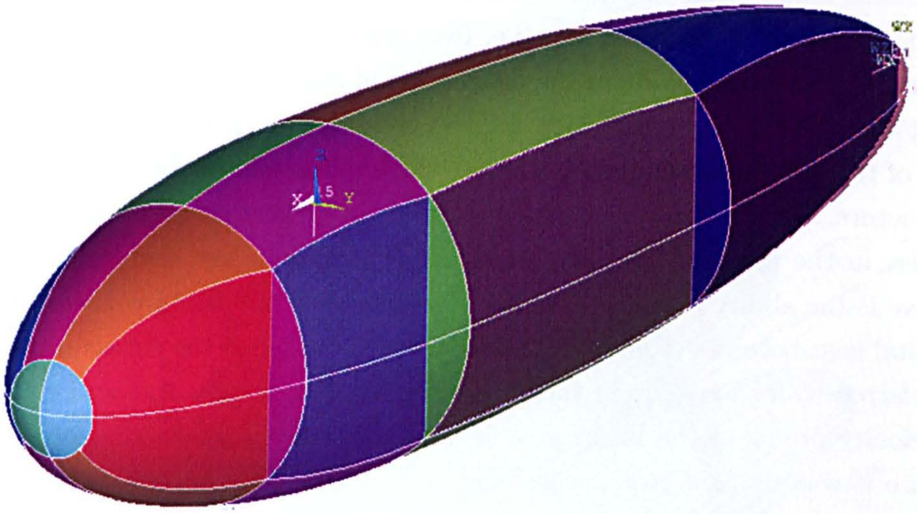


Figure 6.9: Division of balloon into discrete volumes for assignment of buoyancy

Loading on the structure was the same as the linear simulation, with the addition of the helium loads on the membranes. This loading was divided into two components (see Figure 6.8): first, a pressure acting equally over the entire inside surface of each balloon, representing the pressure due to inflation. This was applied as a surface pressure to the shell elements. Additionally, buoyancy was considered. The buoyancy acts only on the upper surface of the balloons, and is not a uniform load since it is proportional to the volume of underlying helium, which varies along the length of the airship. To deal with this, each balloon volume was considered as twenty discrete volumes (see Figure 6.9), and the volume of helium calculated for each division. The total buoyancy force could then be calculated for each volume and the force equally distributed over every node in the overlying surface, giving a discrete approximation of the distribution of force. Note that since the forces are applied to every node, this is equivalent to a pressure load in the FE model.

The advantage of this loading approach over the static simulation was that it gave a much more accurate distribution of pressure over the frame, which was previously only estimated. Since inertia relief is not available in nonlinear simulations the simulation was performed using an explicit dynamic solver, so that rigid body motion was allowed to occur. This more accurately reflects the true nature of the problem because airships are of course unconstrained when in flight, and displacements and stresses vary as a function of time.

6.3.2 Nonlinear Wet Simulation

One further problem of concern was to understand how the airship would respond under aerodynamic loading, which was hitherto not considered. The forms and magnitudes of aerodynamic loading can vary greatly, from the response of the structure in steady-state flow, to ramped or stepped loading (i.e. from gusts of wind in still air). The gust is an important aspect in the study of the interaction of any aircraft with the real atmosphere. It must be taken into consideration for two reasons: first, to consider the dynamic response of the aircraft; secondly, to evaluate the implications of the gust on the stresses in the structure. With regard to the latter, the most critical situation is the vertical gust. However, in the case of the ETF, operational velocity is low and an important design objective is the ability to hover in adverse weather conditions. In this case therefore, the horizontal gust becomes of interest. The form of gust used is the classic constant-gradient gust referred to for example in the European Aviation Safety Agency regulations [116]. The velocity profile of the linear gust is known to closely follow that of a natural gust, therefore it was thought to be reasonable to use this slight simplification, given that in the structural analysis the model nonlinearities are of much greater importance.

Since the pressure distribution on a non-trivial structure due to a particular gust of wind is far from obvious (especially at certain angles), a FSI model was created whereby an arbitrary-Lagrangian-Eulerian (ALE) mesh was overlapped with the Lagrangian mesh used for the "dry" model. The possibility of structural deformation that would change the boundary conditions of a standard CFD model necessitates this approach. The ALE mesh combines the benefits of the Lagrangian and Eulerian mesh approaches (see Section 4.3.2 and Figure 4.9): respectively, that the mesh is allowed to move in space (thus limiting the size of the mesh necessary); and that the material can flow through the mesh, allowing the very large deformations that are characteristic in fluid flow. The ALE mesh is capable of both, therefore it is an Eulerian mesh which can move in space and deform if necessary. The Eulerian mesh used in Chapter 4 can therefore be regarded as a special case of an ALE mesh. An extensive discussion on the ALE method is found in [117]. The two meshes are coupled together using a penalty-based approach in the explicit solver so that they iteratively interact with each other [94]. Figure 6.10 illustrates the arrangement of the two meshes. The ALE mesh is cylindrical because it allows investigation of different incident gust angles, as the airship could simply be rotated by altering the incident angle μ between the ship and mesh without the need for extra pre-processing. Note that the wet model is an extension of the dry model; as such it includes all the loading described in Section 6.3.1, additional to the loads from the ALE mesh.

The ALE mesh is of a multi-material type that allows mixing of different materials in individual elements. It was configured so that it was filled with stationary air surrounding the airship, then at the edge nodes of the mesh on one side (the inlet to the mesh) the air was constrained to move at a velocity ramped from 0m/s to 10m/s in the direction of the ship. These are modelled as separate materials, even though the properties are identical — Figure 6.11 illustrates this arrangement. This closely simulates the effect of a sudden gust of air hitting the airship hovering in stationary air. A key advantage of the ALE mesh

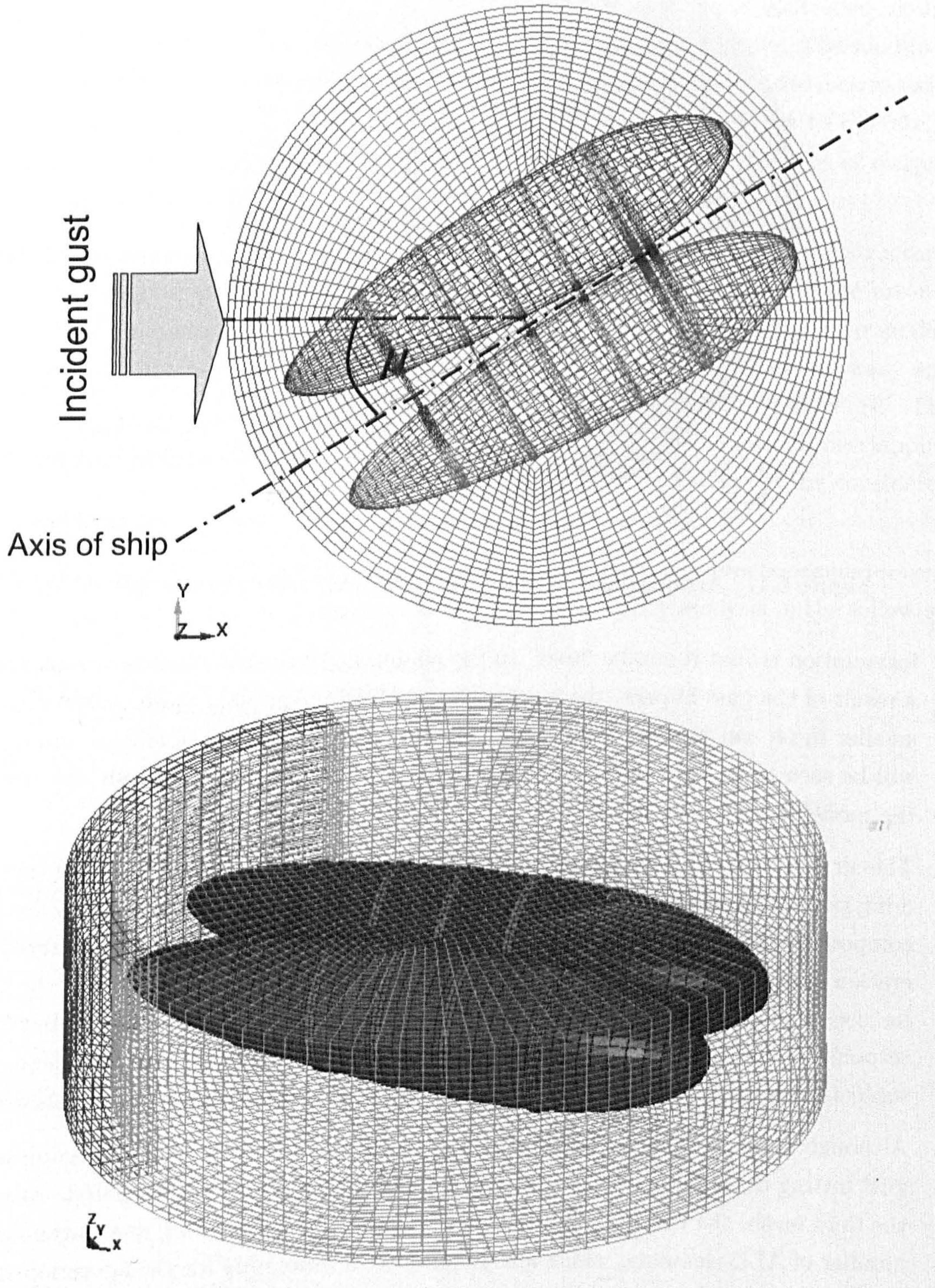


Figure 6.10: Top-view of the ALE fluid mesh overlapping the Lagrangian mesh of the airship (above); oblique view (below).

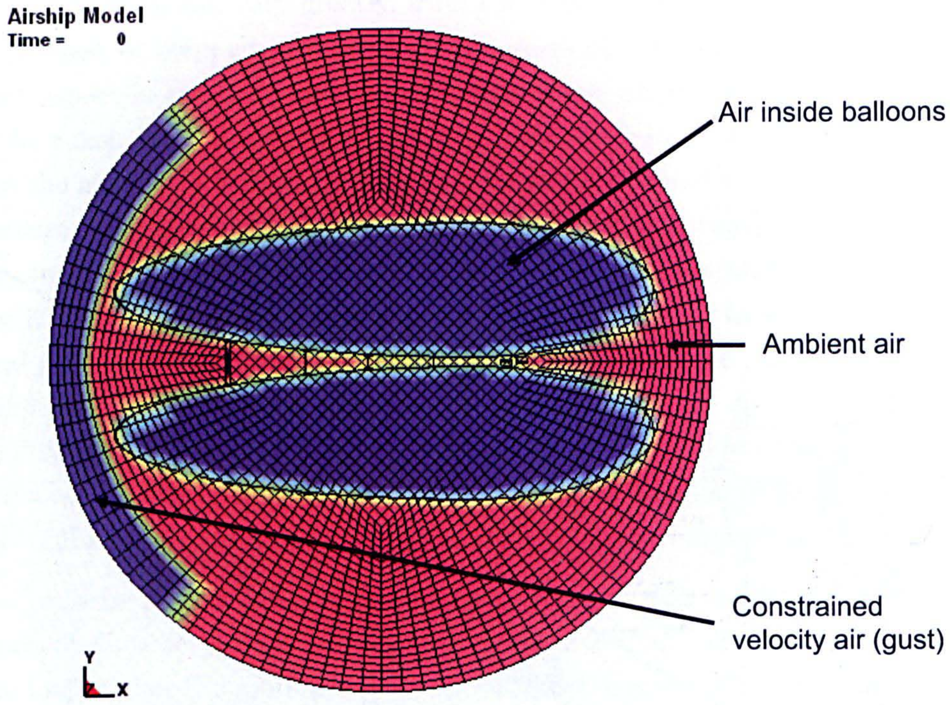


Figure 6.11: Arrangement of materials in the ALE mesh at the initial time step.

formulation is that it can be "tied" to the airship mesh so that when the airship moves as a result of the gust impact, the mesh moves with the structure, which means that a much smaller mesh can be used even if the airship is displaced by a significant amount. This will be seen to be very significant since even with the moving ALE mesh, the run-time of the model is very substantial.

The air is defined by a constitutive model, which gives the deviatoric stresses (not considering possible changes in volume); and an equation of state, which provides the pressure component. This allows modelling of compressible, viscous flow, although without consideration of turbulence, representing a first approximation of the airflow around the structure for use in uncertainty analysis. Note that this is not a full CFD analysis, but since the response of the frame is of more interest than the airflow itself, it is sufficient. A more sophisticated CFD analysis would however be an obvious continuation of this work.

Although there are many conceivable uncertainties associated with the definition of the gust hitting the airship, in this analysis only the wind angle was investigated. After adding the fluid mesh, the run time of the model became very substantial, due to the large extra number of ALE elements, which also require extra processing for the advection steps (the operation of translating the ALE nodes). As such, for the wet model only the wind angle was treated as uncertain; the material properties were assumed to take nominal values.

6.4 Conclusions

This chapter has outlined a new and innovative design for a remotely-controlled airship that is under development at the *Politecnico di Torino* in Turin, Italy. The airship is intended for use in the fields of surveillance and monitoring, and has a distinctive twin-balloon design with a series of directional propellers that should allow it to operate in adverse weather conditions, thereby overcoming some of the main weaknesses of conventional airship designs.

As part of the design process, finite element models have been built to test the structural integrity of the airship, as well as to ensure that deformations in the structure are not so extensive as to hamper the control system that governs the manoeuvres. Two models have been presented here: the first is a "dry" model that accounts for the buoyancy and pressure loading due to helium, as well as directional loads from the propellers. The second ("wet") model is an extension of the dry model and also includes an overlapping arbitrary-Lagrangian-Eulerian mesh which allows aerodynamic loading to be considered from a sudden gust of wind.

The two models will be used as further case studies for the Gaussian process emulator outlined in Chapter 3, by performing an uncertainty and sensitivity analysis in the following chapter.

Chapter 7

Uncertainty Analysis of the Airship Model

Chapter 6 presented two models that were developed for an unmanned airship design known as the *Elettra Twin Fliers* (ETF). Both of these models are large simulations that are subject to substantial uncertainty in the input parameters. Using the Bayesian technique from Chapter 3, these models will be subjected to an uncertainty and sensitivity analysis. This will be useful from the point of view of the airship design, as well as a further case study for the Gaussian process emulator.

Section 7.1 outlines the approach and results of the uncertainty analysis of the dry airship model, followed by the same for the wet model in Section 7.2. Conclusions are drawn at the end of this chapter.

7.1 Dry Model

The uncertainties of interest in the airship model are those that are due to varying operating conditions. One of the main concerns was that since the airship could conceivably be deployed in a very wide range of environments, the uncertainty in model outputs due to temperature-dependent material properties could be substantial. Furthermore, the effect of varying altitude will change the helium pressure in the balloons within quite a wide range.

7.1.1 UA Approach

A two-dimensional parameter space was investigated, varying material temperature between -50°C to +50°C (which in turn, varied the material properties of both the frame and envelope according to the relationships illustrated in Figure 6.4) and also varying he-

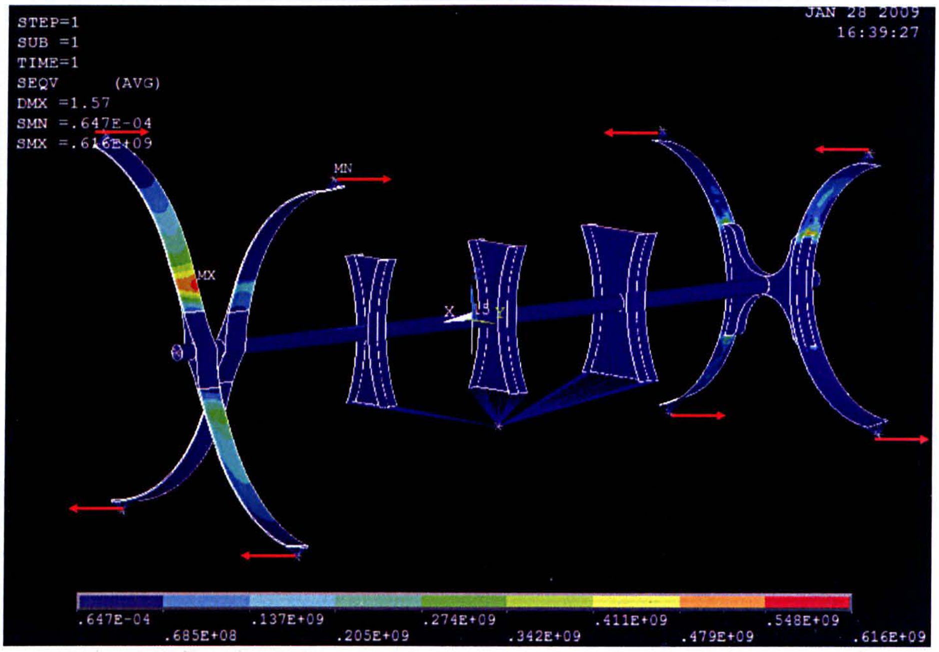


Figure 7.1: A Von-Mises stress contour plot of the linear model under a torsional loading situation. Forces are marked as red arrows.

lium pressure. This second variable (in reality the difference between internal and external pressure) was considered between 2kPa and 5kPa. This was considered to be a plausible variation of pressure given a particular mission objective or a malfunctioning regulator valve.

The motor loading of the structure was considered to be an additional uncertainty, since in the event of a control system failure the motors could conceivably be orientated in any direction. It is of course desirable that the structure be able to withstand this eventuality, however unlikely it might be. Since the consideration of eight additional motor angles would greatly increase the computing time required for the analysis, a separate static analysis was run with nominal temperature and pressure conditions for the motor loading conditions to find the most severe deformation, and this condition (representing a twisting of the frame) was used in the subsequent analysis by applying point forces at the locations of the propellers. The propeller loading is illustrated in Figure 7.1, showing the directional forces applied in 135° and 315° angles in the XY plane. All forces here are 1000N in magnitude. The simulation is therefore a worst-case scenario in terms of motor loading.

Over the two-dimensional parameter space investigated, fifteen training simulations were run by creating a space-filling design-of-experiments for each loading case. The maximin LHS design was again used to train the GP, as in previous analyses. The output of interest was the maximum stress in the structure, since this would indicate a possible failure. As such, maximum stress was extracted from each run from both the envelope and the composite frame. The stress value was taken from the final time step since this

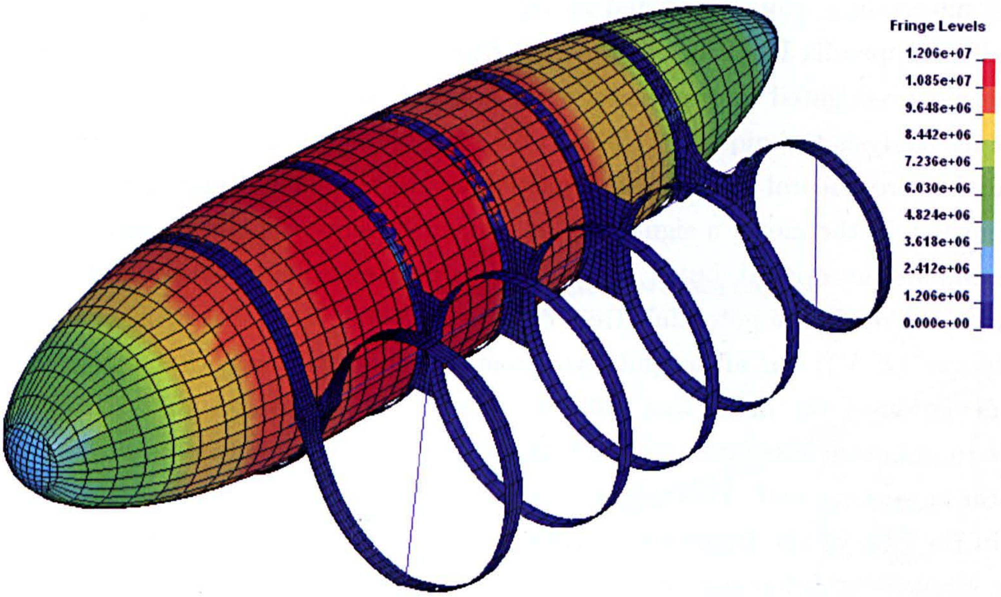


Figure 7.2: Typical Von Mises stress plot of the dry airship model

represented the most stable state of the model, after oscillations due to ramped loading were reduced to a minimum.

For each run, the maximum stress was collected at the final time step, as well as the maximum stress for any time step, for different parts of the model. The model was divided into the two balloons, the composite frame, and the belts. Of the 15 runs performed, two (corresponding to low frame stiffness) resulted in numerical failure around half way through the simulation. This is typical of an uncertainty analysis of a complex nonlinear model - it is very difficult to construct a model that is stable over a wide input parameter space. Despite this, the thirteen runs were still sufficient to conduct the analysis. For illustration, a typical final-state stress plot is shown in 7.2.

7.1.2 Results

Model region	**Mean Stress (MPa)**	**Standard deviation (MPa)**	**COV**	**RMS error**
Balloon 1	14.8	1.13	7.6%	2.75%
Balloon 2	15.0	2.83	18.9%	6.38%
Belts	9.11	3.31	36%	14.5%
Frame	10.0	3.14	31.4%	19%
All	15.3	2.65	17.3%	6.23%

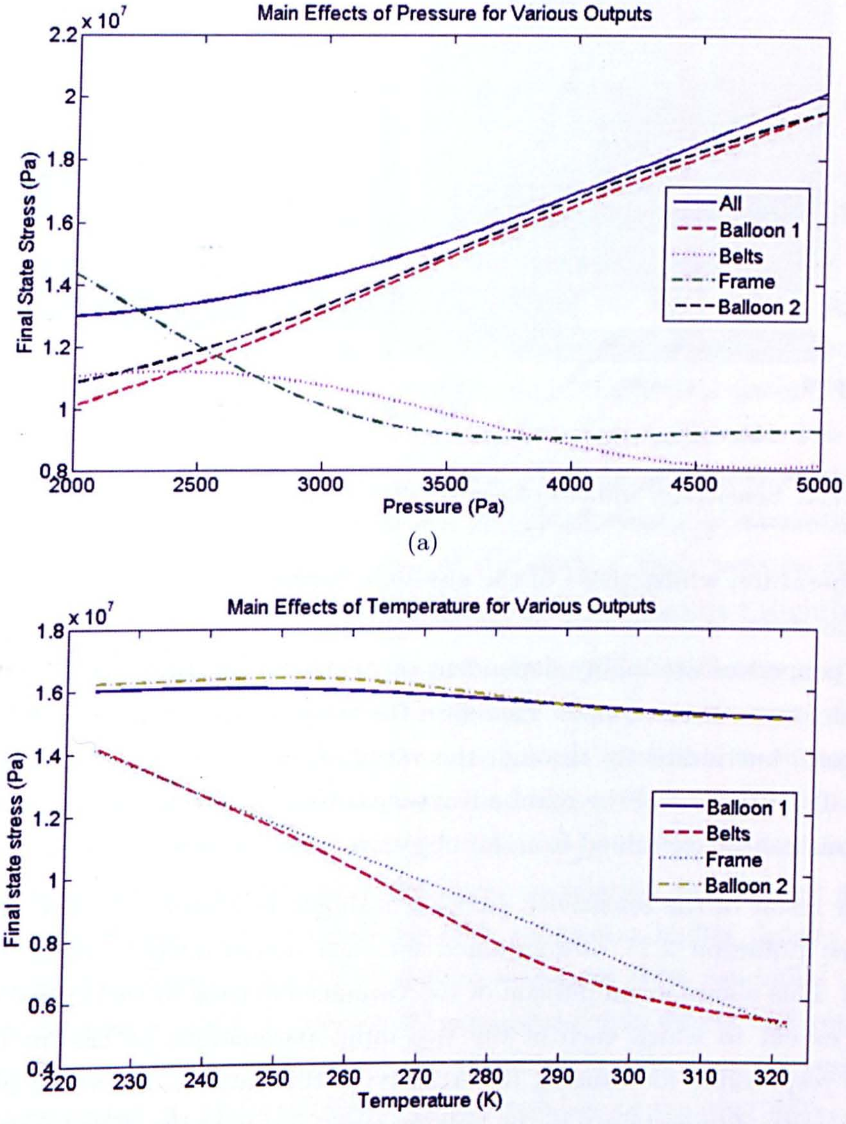
Table 7.1: Uncertainty analysis results from the dry airship model

The overall uncertainty values are listed in Table 7.1 (refer again to Equations 3.45 and 3.51, as well as Appendix B). What is apparent from these values is that the uncertainty in the outputs investigated in the model is substantial, immediately justifying the use of uncertainty analysis techniques. This is to be expected, since an airship operates in a wide range of environmental conditions. In particular, the standard deviation of the frame stress is nearly $\frac{1}{3}$ of the mean, a significant output uncertainty. Of course, stresses lower than the mean are not critical, but it is clear that taking mean temperature and pressure values would underestimate potential stress by a large extent. Gaussian process emulator uncertainty $\text{var}^*\{E(Y)\}$ for all outputs was found to be small, such that the posterior standard deviation of the mean was 100kPa at most. This indicates that the GP can interpolate to unknown simulation points with a high level of confidence; in other words, the emulator is working well. This is further confirmed by the relatively low RMS errors, although in the case of the frame stress the uncertainty is nearly 20%. If the emulator accurately mimics the behaviour of the model then the sensitivity analysis data is also of a high fidelity. Note that this $\text{var}^*\{E(Y)\}$ is distinct from $E^*\{\text{var}(Y)\}$, which is the posterior estimate of the variance of the model output, presented in Table 7.1.

Sensitivities

The main effect plots for the two inputs examined are presented in Figure 7.3, to gain a more detailed insight into the influences of these parameters on the stress in various parts of the model. It should be remembered that these lines represent mean main effect values, averaged over variations in the other parameter, i.e. the expected value of the output with respect to one parameter if we were to suddenly know the "true" value of the other parameter. Recall that the main effect plots are generated using Equations 3.47 and 3.51, setting $p = i$, where i is the input variable of interest (see also Appendix B). Examining the pressure plot first, the strongly linear relationship of stress and pressure is evident in the nearly-straight lines representing both the balloons. The small differences between the two balloons are due to the asymmetry of the frame - loading conditions are not exactly the same for both balloons. A less obvious relationship is that of the frame stress and pressure. The fact that frame stress decreases with pressure suggests that when the balloons are under greater tension due to pressure they are impeding distortion in the frame from motor loading. This relationship of pressure and frame stress supports the hypothesis of the author that a structural analysis of the frame without consideration of the balloon is inaccurate, since it plays an important part in the rigidity of the structure.

Regarding the main effects of temperature, it is apparent that the stress in the balloons is not strongly dependent on this parameter. The stress varies little between 16MPa and 17MPa. Conversely, the stress of the frame is highly dependent on temperature. This is perhaps contrary to initial intuition since the material properties of the frame vary little



(a)

(b)

Figure 7.3: Posterior mean of the main effects of: (a) helium pressure, (b) temperature.

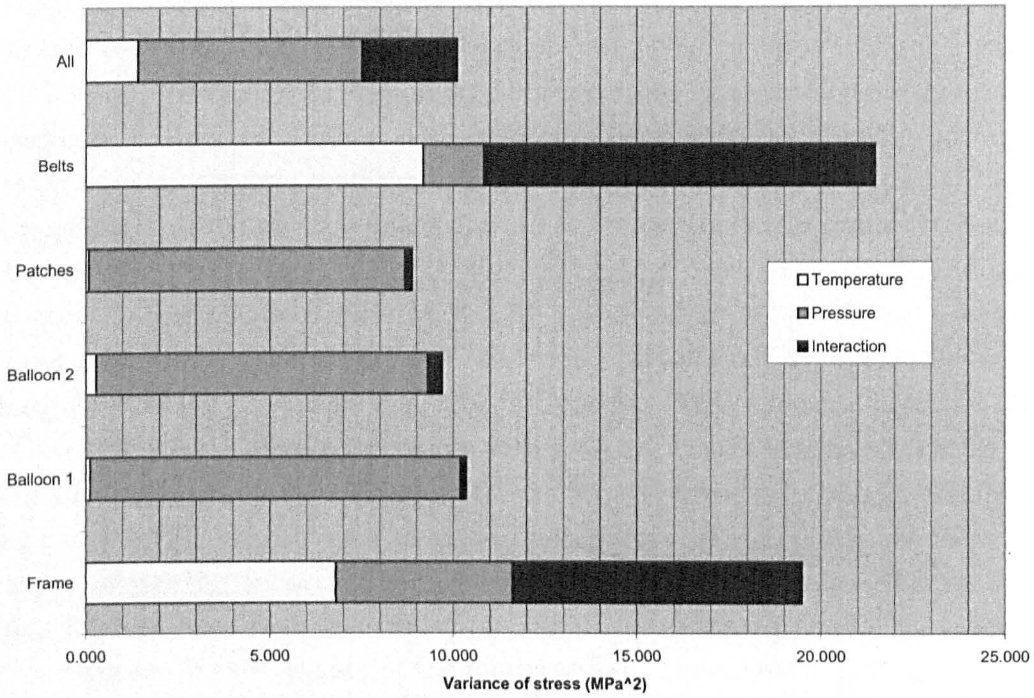


Figure 7.4: Sensitivity indices of temperature and pressure on various model regions

with temperature, whilst those of the envelope vary to a large extent. However, consider that frame stress is a function of material properties *and* the applied loads. Since the envelope properties are highly dependent on temperature, they exert large temperature-dependent forces on the frame. Therefore the stress in the frame *is* highly dependent on temperature, but indirectly through the variation of envelope material properties. This serves to illustrate one of the main advantages of sensitivity analysis - complicated model interactions can be examined from an objective point of view.

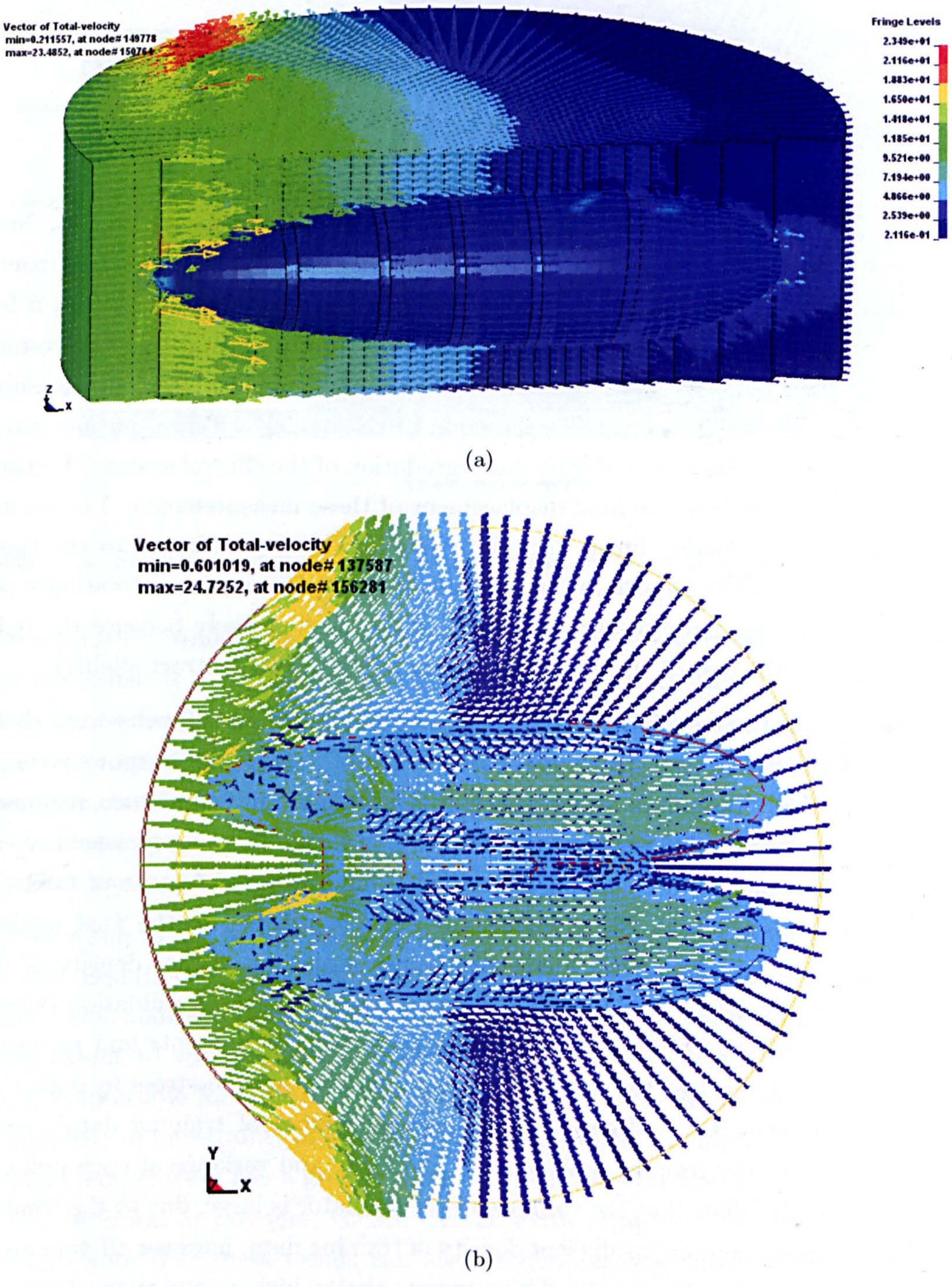
A further result of the sensitivity analysis is shown in Figure 7.4. The non-standardised MEIs (see Equation 2.21) are summed for each model output, along with the output variance. This allows a comparison of the variance of stress in various parts of the model, and the extent to which each of the two input parameters, or the interaction between them, is responsible for causing uncertainty in the output. As would be expected, the great majority of uncertainty in the balloon stress is due to the uncertainty in pressure. In the frame, temperature has a greater influence, but there are also significant interactions with the variation in pressure, caused by the extra rigidity in the structure at higher pressures. Note that these trends can also be qualitatively observed from the plots of main effects, but the MEIs quantify the sensitivity of the output to each parameter to give a precise measurement. The main effect plots are still required though to see the exact relationship between an output and the variation in an input. Between the plots and the sensitivity indices, a very detailed picture of the model response can be built up.

Examination of the maximum stress values collected over all time steps in the model shows that they are up to 50% higher than the final time step stress in some cases. This is evidence of the higher stresses caused by initial fluctuations in the model, a drawback of using a dynamic analysis, yet unavoidable due to the particular loading and constraint conditions of an airship.

7.2 Wet Model

A typical illustration of airflow around the airship hull is shown in Figure 7.5. Note that extracting a meaningful quantity from this model is not trivial because the impact of the gust causes structural oscillations, and all displacement and stress values are a function of time. Some meaningful results can however be obtained. Examining the results from the wet model, Figure 7.6 illustrates the positioning of four displacement measurements between opposite propellers in the x-direction of the model, which are a measure of the deformation of the structure and thus the degradation of the control system. Figure 7.7(a) shows the main effects of the final displacement of these measurements. The variation in displacement is reasonably linear and quite small. This is attributed to the rigidity of the structure caused by tension in the balloons. The displacement is increasingly negative as the gust is angled more towards the front of the airship, likely because the balloon is pushed towards the rear of the ship, causing the structure to contract slightly.

Other outputs of the model are less well-behaved. A significant difficulty with this model is that a small change in gust angle can drastically change the output of the model. Figure 7.7 shows the variation of stress with gust angle for four chosen regions in the model (regions are shown in Figure 7.6). Each of these regions represents an area of observed stress concentration in the frame (and thus a potential point of failure). The response of stress is shown to be highly nonlinear in relation to the gust angle. This does cause some problems since a rough response requires a higher density of training data. An investigation of this (Figure 7.8) uses leave-one-out cross validation (which tests the prediction error as a result of repeatedly removing single points and retraining the emulator, see [33]) to test the accuracy of the emulator for the stress in region 2. The emulator is constructed n times (where n is the number of training data), each time leaving out one of the training data points. The mean and variance at each point is then noted. The results show that the variance of the emulator is large, due to the combination of a rough response and an insufficient density of training data, however all data are within the predicted confidence intervals. This suggests that a higher resolution of output data is necessary to investigate more accurately the true response of the model, but that the emulator is at least quantifying its own uncertainty correctly. Therefore, the results in Figure 7.7 are ideally in need of refinement but provide an insight into the problems that can occur in drawing conclusions from a very nonlinear model.



(a)

(b)

Figure 7.5: Velocity vector plots of gust impact at $\mu = 0^{o}$: (a) oblique view; (b) top view.

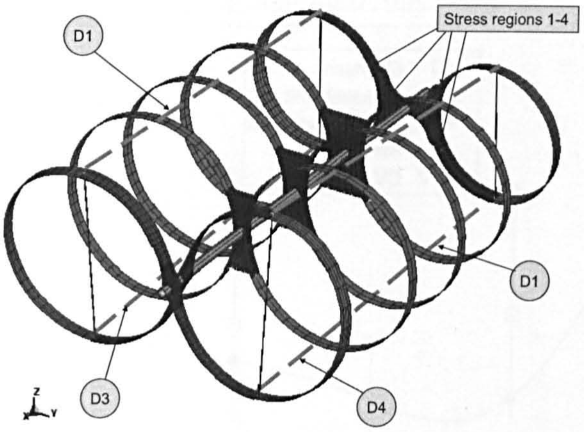


Figure 7.6: Displacement measurements 1-4 (labelled D1-D4 respectively), and stress regions 1-4, on the airship frame.

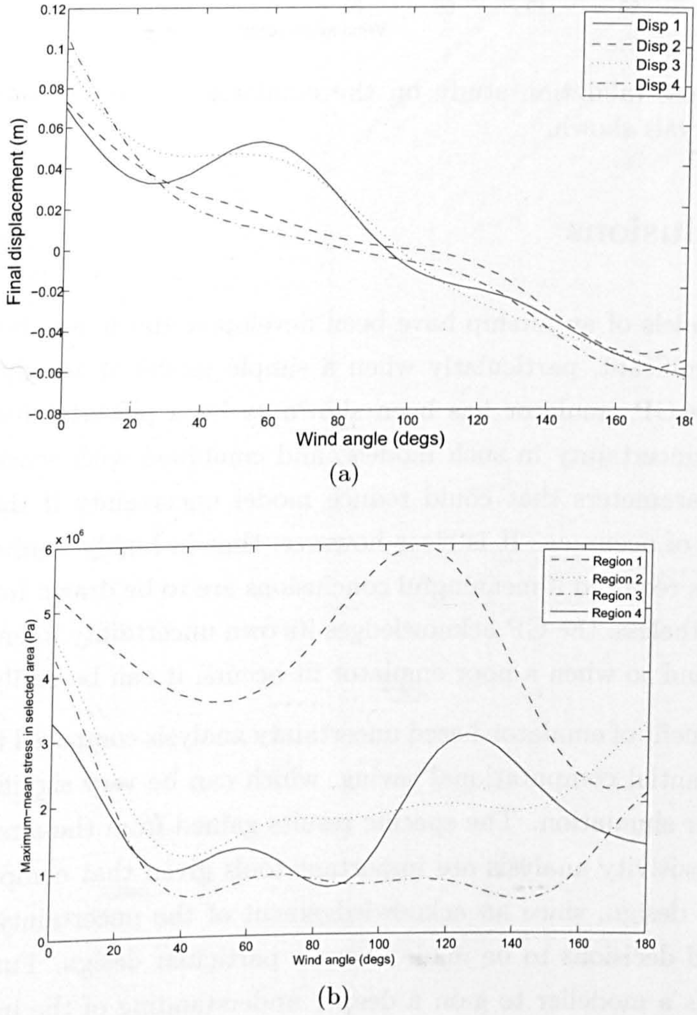


Figure 7.7: Posterior means with respect to wind angle: (a) final displacements 1-4, (b) maximum of mean stress in regions 1-4

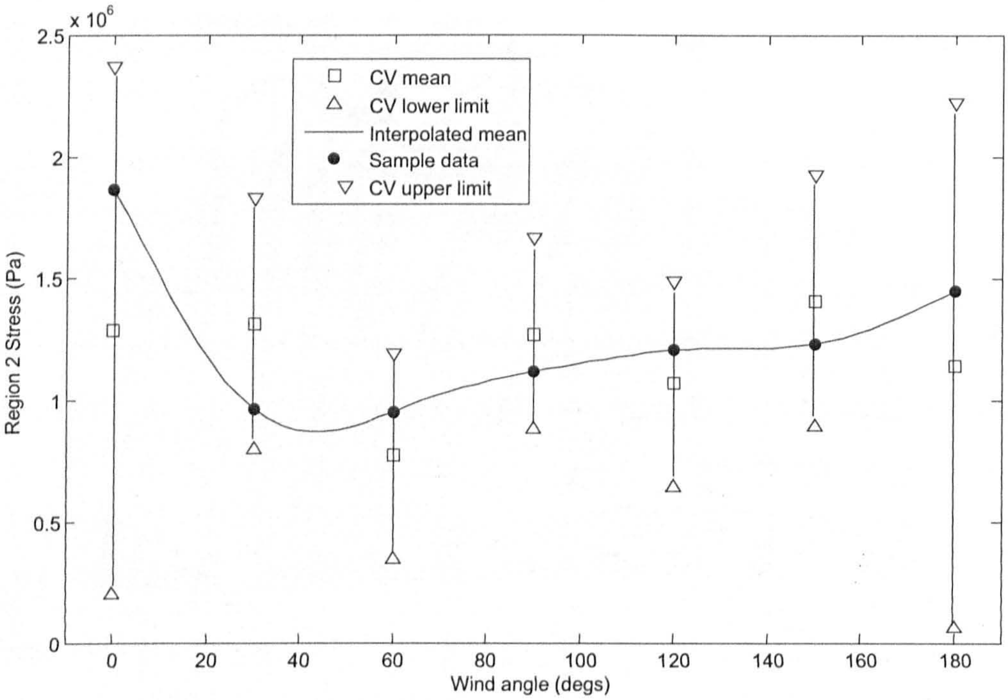


Figure 7.8: Cross-validation study on the emulator of the wet airship model — 95% confidence intervals shown.

7.3 Conclusions

Typical FE models of an airship have been developed and it was found that output uncertainty is significant, particularly when a simple model of aerodynamic loading is introduced. The GP emulator has been shown to be a powerful tool to investigate the magnitude of uncertainty in such models, and combined with sensitivity analysis data, can identify parameters that could reduce model uncertainty if they were known to a greater degree of accuracy. It is clear however, that in highly nonlinear responses, more training data is required if meaningful conclusions are to be drawn from emulators such as the GP. Nevertheless, the GP acknowledges its own uncertainty in emulating the response of the model and so when a poor emulator fit occurs, it can be easily identified.

The overall benefit of emulator-based uncertainty analysis compared to Monte Carlo methods is a substantial computational saving, which can be very significant in the case of a large computer simulation. The specific results gained from the airship show that uncertainty and sensitivity analysis are important tools given that computer models are used extensively in design, since an acknowledgement of the uncertainty in the model allows more informed decisions to be made about a particular design. Furthermore, sensitivity analysis allows a modeller to gain a deeper understanding of the interactions at work in a model, by examining how variation in certain parameters affects various model outputs. Further work is required from the perspective of the airship design to test material uncertainty in the other candidate membrane materials.

Chapter 8

Tree-Structured Gaussian Processes for Uncertainty Analysis

Chapters 3-7 have outlined in detail a sophisticated emulator-based approach for performing uncertainty and sensitivity analysis on large nonlinear models using GPs. The method has been shown to provide very detailed information at low computational cost, whilst making as few assumptions as possible about the form of the response surface of the model output.

The uncertainty analysis of the aortic valve in Chapter 5 raised an interesting issue however. It was observed that for some areas of model input space the response of the model may have been violating one of the few assumptions of the GP emulator, namely that the data is a smooth function of its inputs. This suggests the question — how well can the GP emulator cope with non-smooth data, and is there perhaps an alternative way of performing uncertainty analysis on non-smooth and even bifurcating computer models? One conceivable solution to this problem is to somehow divide the input space, fitting separate regression models at each region and using the region boundaries to model discontinuities.

A very recent approach to deal with exactly this issue has been proposed by Gramacy *et al.* [118] which essentially uses a fully Bayesian variant of *classification and regression trees* (CARTs) to perform the divisions of input space. Gaussian process (GP) emulators are fitted at each of R "leaves" of the tree (regions of input space), with the result that the emulator now consists of a number of non-overlapping GPs, rather than a single GP. The "standard GP" from Chapter 3 (as it will now be termed), could therefore be viewed as a special case of this "tree-structured GP".

Tree-structured GP emulators were originally developed for use with data that has a non-stationary covariance structure (known as *heteroskedastic* data), since the standard GP necessarily uses the same covariance function for the whole range of input space. Instead, this chapter will investigate the possibility of using the tree-structured GP emulator for

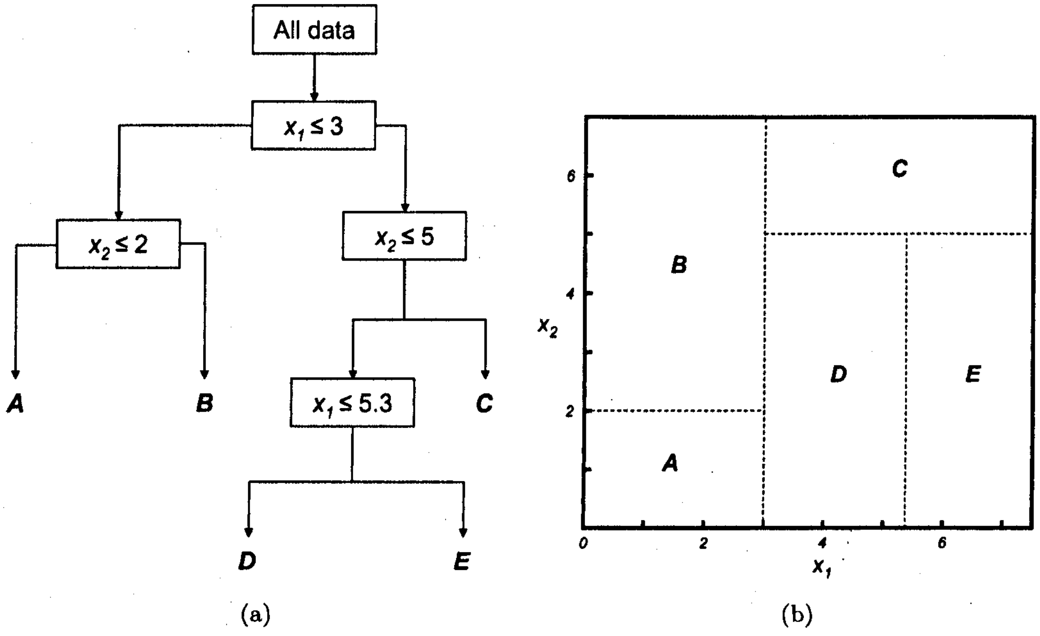


Figure 8.1: An example of a 2D CART: (a) Tree structure; (b) Corresponding divisions of input space.

bifurcating data, using the tree divisions to model bifurcations (since the emulator can be discontinuous over divisions in input space). One very nice feature of the tree-structured GP is that the computational expense is actually *reduced* compared to the standard GP, since the cost of inverting the required covariance matrices is n^3 for a single GP, whereas when the data are handled by R equally-sized GPs the cost is n^3/R^2.

The concept of CARTs will be explained in Section 8.1, after which the Bayesian approach to CART is outlined in Section 8.2 and applied to an simple example in Section 8.3. Conclusions and comments follow.

8.1 Classification and Regression Trees

A Classification and Regression Tree (CART) is a system used to recursively divide data into classes that are increasingly homogeneous within their divisions in some respect. CARTs were developed primarily by Leo Breiman, who wrote a seminal book describing methods of constructing trees and their applications [119].

An example of a CART applied to 2-dimensional data is shown in Figure 8.1(a). The tree starts with the full set of training data. At each split in the tree (known as a "node"), a rule is used of the form $x_i \leq s$ to divide the data using a single input variable. If a data point agrees with this rule, it goes to the left "branch". Otherwise, it goes to the right. This process is repeated as many times as necessary on any variables until the data is sorted according to some criterion, at which point the data reaches the "terminal node" or "leaf" of the tree and is assigned either a category (in the case of a classification tree), or some numerical value (a basic regression tree). In fact, the concept may be extended to fitting regression models at each leaf (i.e. the terminal node contains a parameter set describing an emulator) which will be of particular interest in the context of this work.

The result of this splitting of data is that the input space is divided into R non-overlapping regions, as illustrated in Figure 8.1(b). Note that since the splits are performed on one variable at a time, all the partitions are made using lines parallel to the axes. This will be seen to be a possible limitation later on because when splits are required to be non-parallel it may require a complicated tree to satisfactorily partition the data.

8.1.1 Growing Trees - Greedy Algorithms

Leaving aside for a moment the issue of what is actually *at* the leaf of the tree, the main problem to overcome is that of constructing a tree that partitions the data to some required level of accuracy, ideally by the simplest means possible. In other words, it is desirable to have a tree that is no more complicated than is necessary to partition the data, to avoid the problem of over-fitting (note that this is based on the principle of Occam's Razor [120]).

The simplest approach to "growing" the tree is to use a *greedy algorithm* that optimises some criterion at each split. Starting with the whole data set at the root node, the algorithm performs an exhaustive search of every possible split for every variable and calculates the number of correct classifications for each. Since there are a finite number of possible ways to divide the data and the calculation of classification error is very cheap, this strategy is not a problem from a computational point of view. The split that minimises the classification error is then selected and the data accordingly divided according to the new rule. The process is then repeated for the two new leaves of the tree, resulting in more divisions. The division of data continues until some global criterion has been reached, such as an overall percentage of correct classification or when the classification rate has converged.

The problem with the greedy algorithm is that although the splits are locally optimal, there is no guarantee that this will result in a *globally* optimal solution. The sub-optimality of greedy algorithms is demonstrated by the classic "travelling salesman problem" (see e.g. [121]). Some improvement may be gained by "pruning" the tree, which involves simplifying the tree by removing nodes, based on some strategy such as a cost-complexity criterion or using cross validation. Further methods for tree construction are discussed in [119].

8.2 Bayesian CART

An alternative way of growing CARTs using a Bayesian approach has been suggested by Chipman *et al.* [122], where each leaf contains a Gaussian distribution (i.e. a regression tree). This was later extended to use linear regression models at each leaf [123]. Most recently, Gramacy and Lee extended the Bayesian CART much further by using it to fit GPs over each region of input space. However, in all cases the principles of the tree construction itself remain essentially the same, being based on Chipman's original paper. For the purposes of clarity therefore, the following sections will discuss the Bayesian approach to CART construction in the context of a simple regression tree. The extension to the full GP tree will be briefly explained but not treated in full detail. A very detailed description of the tree-structured GPs can however be found in [124].

8.2.1 Overview

The Bayesian approach to CARTs considers the problem from a probabilistic perspective. Following the normal procedure, a prior distribution *over models* is created, which will ultimately be conditioned on training data to produce a posterior distribution. In order to fully specify a CART model, it is necessary to have both a tree structure (which will be called T), and a set of parameters Θ that specifies the regression parameters at each leaf, for all leaves. In the full GP tree, these would be GP hyperparameters, but in the basic CART they will be means and variances at each leaf. In particular, the following explanation will use the simple "mean-shift" model described by Chipman as an example [122]. In this model, it is assumed that the data is distributed with a separate mean μ_v for each leaf $r_v \in \{r_\nu\}_{v=1}^{R}$, but is governed by a global variance measure σ^2 that is the same for all leaves.

Likelihood

To express the model likelihood, it is assumed that the data y_j (where j is the sample index inside a given leaf) are independent and identically distributed (iid) within each of the R leaves, and independent across leaves. Defining the division of data across leaves as $\mathbf{y} = \{\mathbf{y}_v\}_{v=1}^{R}$, and the data inside each leaf as $\mathbf{y}_v = \{y_j\}_{j=1}^{n_v}$ (where n_v is the number of data points in the leaf r_v) the specification of model likelihood may be given as,

$$p(\mathbf{y}|X, \Theta, T) = \prod_{v=1}^{R} f(\mathbf{y}_v|\theta_v) = \prod_{v=1}^{R}\prod_{j=1}^{n_v} f(y_{vj}|\theta_v) \tag{8.1}$$

where y_{vj} is the jth observation at the vth leaf. In the case of the mean-shift model, a normal distribution is assigned, such that for any leaf the $f(y_{vj}|\boldsymbol{\theta}_v)$ are expressed as,

$$\{y_j\}_{j=1}^{n_v}|\boldsymbol{\theta}_v \text{ iid } \sim \mathcal{N}(\mu_v, \sigma^2) \tag{8.2}$$

This can be easily extended to the mean-variance-shift model where σ^2 can vary between leaves, but this will not be considered here.

Prior

The prior distribution over models $p(T, \Theta)$ is now outlined. This can conveniently be divided such that

$$p(\Theta, T) = p(\Theta|T)p(T) \tag{8.3}$$

which allows the tree prior to specified *independently* of the regression parameters.

The prior may now be combined with the model likelihood given the data, which is dependent on the type of model used at each leaf. If the parameter prior has a carefully chosen form, it is possible to analytically marginalise the model parameters, i.e.,

$$p(Y|X, T) = \int p(Y|X, T, \Theta)p(\Theta|T)d\Theta \tag{8.4}$$

Now using Bayes' theorem the posterior distribution over trees can be found up to a proportional constant,

$$p(T|X, Y) \propto p(Y|X, T)p(T) \tag{8.5}$$

By searching the posterior distribution in Equation 8.5, the most probable trees may be selected given the supplied training data. For example, if a single tree is required, it might be reasonable to pick the MAP tree - i.e. the tree with the highest posterior probability. The selection of trees is discussed further in Section 8.2.3. Now, the specification of priors and likelihoods for the CART model will be dealt with.

8.2.2 Prior Specification

The model prior is split into a tree prior and a conditional parameter prior, as noted in Equation 8.3. These are dealt with separately.

Tree Prior

The tree prior $p(T)$ cannot be expressed in a closed form. However, it is possible to create a stochastic process that produces a series of trees based on specified rules that successively split terminal nodes. Implicitly, this allows sampling from an indirectly-specified prior tree

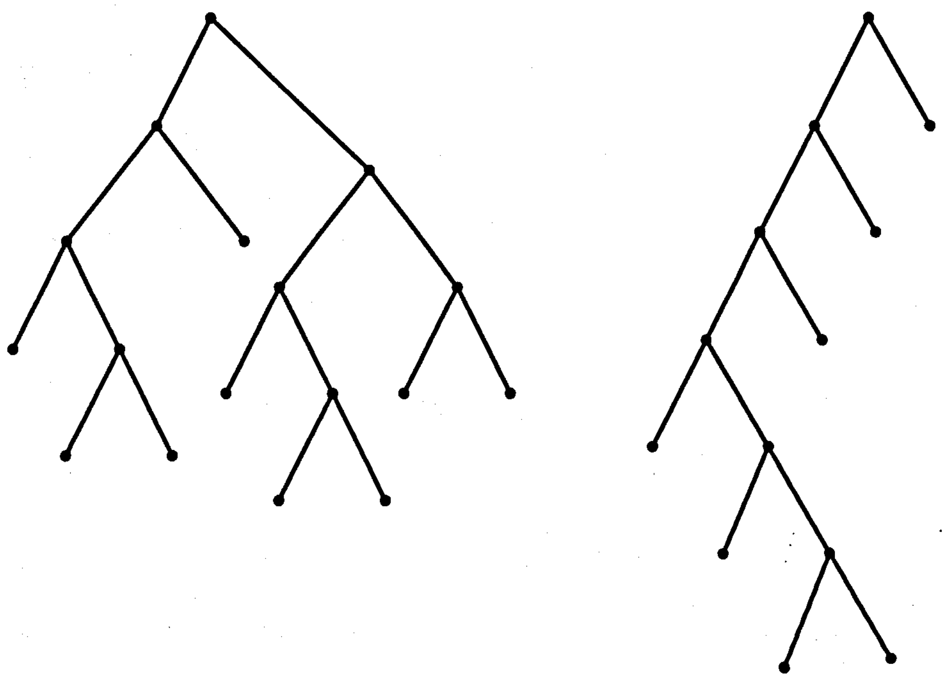


Figure 8.2: Desirable bushy trees (left) and undesirable lanky trees (right)

distribution. Since the posterior tree distribution from Equation 8.5 is explored by MCMC (see later), this definition of $p(T)$ is not a problem in practice. Importantly, the tree prior is independent of the model parameters at the leaves — it only specifies the structure of the trees and the associated splitting rules at this point.

The tree-generating process proceeds as follows. One starts with an "empty tree", which consists of a single root node. At this point, all the data is at the root node, and the input space is undivided. In order to split the data, it is necessary to choose first, the node $\eta \in T$ to be split (in the case of the empty tree there is of course no choice); second, the variable $x_i \in \boldsymbol{x}$ on which to perform the split; and last, the value s which dictates the value of x_i at which to perform the split. With the new splitting rule, two "child nodes" are created. This new tree can be considered as a sample from $p(T)$, and the same process can be applied repeatedly to generate a series of samples.

To deal with the selection of the node to be split, Chipman suggests a splitting rule p_{SPLIT} such that,

$$p_{\text{SPLIT}}(\eta, T) = \alpha(1 + d_\eta)^{-\beta} \tag{8.6}$$

where d_η is the depth of node η (the number of splits above η), and α and β are parameters that can be set to control the shape of the prior distribution. Since this rule is a decreasing function of d_η, it encourages the growth of "bushy" trees (trees whose terminal nodes do not vary too much in depth), rather than "lanky" trees (see Figure 8.2), which are more likely to over-fit the data. The shape parameters can be adjusted empirically by generating samples from $p(T)$ and plotting histograms of criteria of interest, for example, by examining the number of terminal nodes.

Now, given that a split has been made, the variable x_i on which to make the split, and the split value s must be chosen. This is done by assigning a further rule $p_{\text{RULE}}(\rho|\eta, T)$, which dictates the probability of the rule ρ by a distribution on the available x_i, and a further distribution on the available split values, conditional on each x_i. "Available", in this context, refers to split variables and split values that do not lead to empty leaves. Note that s could take any value, but there are only a finite number of possible data splits — for example, the midpoints between all the data values along a given x_i. Therefore p_{RULE} is a discrete distribution. By extension, this also implies that $p(T)$ is also a discrete distribution, albeit over a very large number of possible trees for a reasonable-size data set.

The specification of p_{RULE} is simply given by assigning a discrete uniform distribution over available x_i and a discrete uniform distribution over available split values. It could be reasoned, however, that this strategy could be improved if there were some *a priori* knowledge about the importance of individual x_i. For example, the current rule assigns lower probability to multiple splits on a single variable. If it were known that splits are more likely or less likely to occur on particular variables, it might be of interest to assign non-uniform distributions over the variables to reflect this. Furthermore, it might also be a better strategy to assign a distribution to the splitting value that puts more weight on splitting in the centre of the data, rather than the tails. This strategy would give more probability to bushy trees. However, the uniform distributions described above will be assumed to suffice for now.

Parameter Prior

In choosing the parameter prior $p(\Theta|T)$, it is convenient to choose a form that allows the analytical marginalisation described in Equation 8.4. In this case, the conjugate prior of the Gaussian likelihood in Equation 8.1 is used, which is a normal distribution over each μ_v and an inverse-gamma distribution over σ^2. This is broken down such that $p(\mu, \sigma^2|T) = p(\mu|\sigma^2, T)p(\sigma^2|T)$, where,

$$\{\mu_v\}_{v=1}^{R}|\sigma^2, T \text{ iid } \sim \mathcal{N}(\bar{\mu}, \sigma^2/a) \tag{8.7}$$

$$\sigma^2|T \sim \mathcal{IG}(v/2, v\lambda/2) \tag{8.8}$$

This now yields an analytical expression for the marginal likelihood of the trees after marginalising the parameters (by performing the integral in Equation 8.4), which is,

$$p(Y|X,T) = \frac{ca^{b/2}}{\prod_{v=1}^{R}(n_v + a)^{1/2}} \left\{ \sum_{v=1}^{R}(s_v + t_v) + v\lambda \right\} \tag{8.9}$$

where c is a constant that does not depend on T, s_v is $(n_v - 1)$ times the sample variance of $\mathbf{y}_v$, $t_v = [n_v a/(n_v + a)](\bar{y}_v - \bar{\mu})^2$, and $\bar{y}_v$ is the mean value of the $\mathbf{y}_v$. The prior parameter values may be chosen with guidance from the training data — for details of this the reader is referred to Chipman [122].

8.2.3 Searching the Posterior Distribution

Now that the prior distribution and likelihood have been specified, the posterior distribution of trees $p(T|X,Y)$ (having integrated out the model parameters) is expressed by Equation 8.5, up to a constant of proportionality. Although as mentioned, $p(T)$ is a discrete distribution, and therefore the posterior is also discrete, an exhaustive search of all trees is usually impossible due to the sheer number of possible trees. However, using the Metropolis-Hastings (MH) algorithm, it is still possible to effectively search for the most probable trees, since the MH algorithm naturally moves towards areas of higher probability. The workings of the MH algorithm will briefly be explained here, after which an overview of its use in the context of trees will be given.

The Metropolis-Hastings Algorithm

The MH algorithm, which has been described as one of the most important algorithms of the last century [125], is a form of Markov Chain Monte Carlo (MCMC) estimation. First proposed by Metropolis [126] and subsequently extended by Hastings [127], it allows sampling from a distribution even when the normalising constant is unknown.

The MH algorithm is based on the idea of generalised rejection sampling in a Markov chain. A Markov chain is a sequence of events $\{A_t, A_{t+1}, ...\}$ where the probability of each successive event is conditional only on the event immediately before it, i.e.,

$$p(A_{t+1}|A_t, A_{t-1}, A_{t-2}, ..., A_0) = p(A_{t+1}|A_t) \tag{8.10}$$

where A_t denotes the state of the chain at time t. The Markov chain is defined by an initial starting state A_0, and a transition probability distribution $q(A_i, A_j)$ which describes the probability of jumping to a state j given that the current state is i. For chains where the possible number of states is small, the probability of jumping between any permutation of steps can be explicitly defined, however quite often the transition

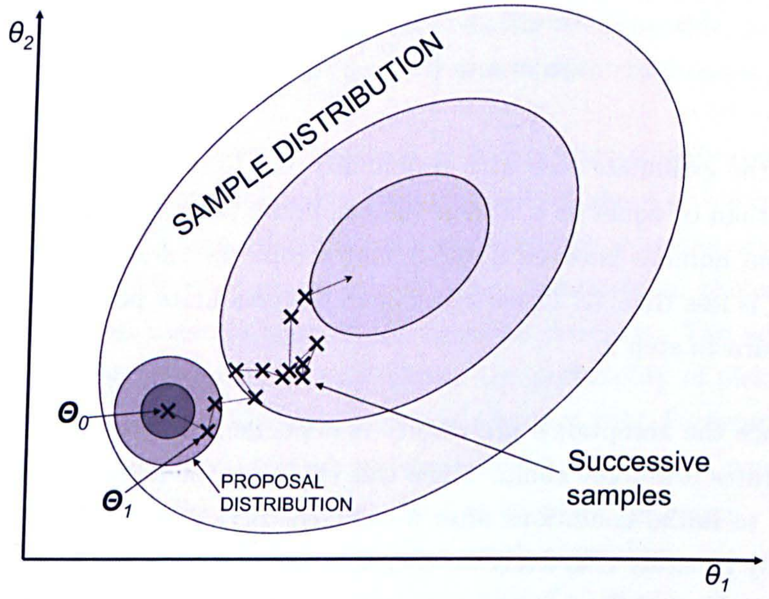


Figure 8.3: Metropolis-Hastings draws from a 2-dimensional sample distribution. Candidate points are drawn from the proposal distribution and accepted or rejected based on the MH acceptance ratio.

probability must be defined by a transition kernel that returns the probability of jumping to a new state based on some function of the present state. Typically this may be done by using a small jumping distribution (such as a Gaussian), centered on the present state, to propose new values of A.

The MH algorithm uses a Markov chain to move around the support of the sample distribution (which is very often a distribution of parameters), taking samples $\{\Theta_0, \Theta_1, \Theta_2, ...\}$ as it moves, however the jumping distribution only proposes new values, which are either accepted or rejected based on the MH *acceptance ratio*. Figure 8.3 illustrates the movement of the MH algorithm over a 2-dimensional parameter space.

In the context of the tree posterior, the variable of interest is the tree T, and it is required that samples be drawn from $p(T|X,Y)$. This is performed by generating a Markov chain of tree samples $\{T_0, T_1, T_2 ...\}$ that converges to the posterior distribution. The steps of the MH algorithm are here given in the context of trees, but the principle extends to the search of any posterior:

1. Start with an initial tree T_0. Typically this will be the empty tree, but not necessarily so (see later).
2. Generate a candidate tree $T^\star$ from a *jumping distribution* $q(T_0, T^\star)$. Note this is not the probability of acceptance. It is simply a probability distribution of possible candidate trees.

3. Evaluate the *acceptance probability* α such that,

$$\alpha = \min\left\{1, \frac{p(T^\star|X,Y)q(T^\star,T_0)}{p(T_0|X,Y)q(T_0,T^\star)}\right\} \tag{8.11}$$

4. Accept the candidate tree with probability α. In other words, if the MH ratio is greater than or equal to 1, accept the candidate point. If it is less than 1, generate a random number between 0 and 1 and accept the candidate point if the random number is less than α. Otherwise, reject the candidate point, keep the current tree and return to step 2.

Note that since the acceptance probability is dependent only on the previous state, this process generates a Markov chain. Since this chain can be shown to be *ergodic* (i.e. it is not sensitive to initial conditions after a sufficient period of time, known as burn-in), it will eventually generate effectively independent samples from the distribution of interest. The MH ratio also satisfies a criterion known as *detailed balance*, which ensures that after sufficient burn-in, the chain is stationary, and will continually sample from the distribution of interest (see [120] for more information).

The key to the success of the MH algorithm is that since the acceptance probability α is a ratio of posterior probability values, the normalising constant of the posterior actually cancels out. Therefore it is only necessary to know the posterior distribution up to a constant of proportionality, as in Equation 8.5. Additionally, since the algorithm will *always* accept jumps to points that have a higher probability in the sample distribution, the MH search naturally gravitates towards areas (trees, in this case) of higher probability.

Posterior Search Using MH

In order to calculate the acceptance ratio, it is necessary to have some jumping distribution q that specifies the probability of jumping to a new tree, given the present tree state. Chipman suggests a distribution that may be defined implicitly by a set of four rules (although other rules could easily be conceived), which are as follows,

1. GROW: Randomly pick a terminal node and use the p_{RULE} rule defined in the prior to split it into two child nodes.

2. PRUNE: Randomly pick the parent of two terminal nodes and collapse it into a terminal node by removing the splitting rule.

3. CHANGE: Randomly pick an internal (not terminal) node and randomly re-specify the splitting rule using p_{RULE} from the prior.

4. SWAP: Randomly pick a parent and child node that are not terminal nodes. Swap the splitting rules, unless the other child node has the same rule, in which case swap the splitting rule of the parent node with that of both child nodes.

Note that when selecting between these rules, it is necessary to only consider those that will not lead to an empty terminal node. The rules defined above give a reversible Markov chain, because the GROW and PRUNE rules are reversible counterparts. Furthermore, the CHANGE and SWAP rules are reversible when compared to themselves, therefore every possible move has a counterpart in the opposite direction. The actual probability of a given move can be calculated by multiplying the probability of picking a particular rule (1/4) by the probability of the alteration made by that rule. For example, in the case of the GROW move, one would multiply the probability of picking a particular terminal node ($1/R$) by the probability of assigning the rule specified by p_{RULE}. The calculation of these probabilities can be easily built into the search algorithm.

8.2.4 Comments on the Posterior Search

Chipman *et al.* note that although the MH algorithm is convergent, and can be shown to visit all possible trees given enough time, in practice the search tends to get stuck for very long periods of time in local modes of high probability. Therefore it is a good strategy to restart the chain every 500 moves or so. The restart point can be the empty tree, previously visited trees, or a start tree generated by some process, perhaps by a greedy algorithm. It has been shown that this will significantly improve the mixing of the chain and therefore high-probability trees will be located in fewer runs [122].

It is necessary to have some strategy for selecting "good" trees. Because of the huge number of possible trees, it is not possible to use the frequency of visits to a particular tree as an indicator, because it would simply take too long. An obvious approach is to pick trees with the highest posterior probability, although it is noted that this may be a misleading measure because when comparing single trees, a relative posterior probability measure will favour trees using splits on variables with fewer split points (an explanation of this is given in Chipman [122]). A much better alternative might be to use model averaging, where an overall posterior mean can be calculated by using weights proportional to the posterior probability of each tree visited. This also has the advantage of nullifying the problem of comparing trees based on posterior probability.

8.2.5 Extension to Tree-Structured GPs

The previous explanation has focused on the use of a CART coupled with a simple mean-shift model to assign means and variances at each leaf of the tree. Although from a high-level perspective the method remains essentially the same when a GP is fitted at each leaf (since a GP is defined by a set of parameters Θ in the same way as the mean-shift model), in practice the details of its implementation are somewhat more complex. Since the principles of Bayesian CART have been shown in depth, it is not considered necessary to provide a detailed treatment of the construction of tree-structured GPs, however, an outline will be given, and the main differences explained. For a full explanation, the reader is referred to [124].

The idea of tree-structured GPs was introduced by Gramacy and Lee [118], who are also the authors of the R package "tgp" [128], which incorporates all the features described here, including the ability to perform UA and SA on both tree-strucured and standard GPs. This package was used to perform all the tree-structured GP analyses in this thesis.

Prior Specification

At each of the R leaves $\{r_v\}_{v=1}^{R}$ of the tree a GP is defined. Gramacy *et al.* propose a GP for each region r_v of the same form used in Chapter 3, except further parameter uncertainty is accounted for by using a hierarchical specification of the hyperparameters. Rather than specifying an uninformative prior over the weights $\mathbf{w}$ and the variance scale σ^2 (refer back to Equation 3.7), normal and inverse-gamma distributions are used respectively, the hyperparameters of which are also assigned distributions. This hierarchical model and its priors are specified as,

$$\mathbf{y}_v|\mathbf{w}_v, \sigma_v^2, A_v \sim \mathcal{N}_{n_v}(\Phi_v \mathbf{w}_v, \sigma_v^2 A_v) \tag{8.12}$$

$$\begin{aligned}
&\mathbf{w}_v|\sigma_v^2, \tau_v^2, W, \mathbf{w}_0 \sim \mathcal{N}_m(\mathbf{w}_0, \sigma_v^2 \tau_v^2 W)\\
&\mathbf{w}_0 \sim \mathcal{N}_m(\mu, C)\\
&\tau_v^2 \sim \mathcal{IG}(\alpha_\tau/2, q_\tau/2)\\
&\sigma_v^2 \sim \mathcal{IG}(\alpha_\sigma/2, q_\sigma/2)\\
&W^{-1} \sim \mathcal{W}((\rho_v V)^{-1}, \rho)
\end{aligned} \tag{8.13}$$

where W is an $m \times m$ matrix (m is the number of covariates in the training data), $\mathcal{IG}$ and $\mathcal{W}$ are the inverse-gamma and Wishart distributions respectively and the remaining undefined hyperparameters are treated as known. The covariance function is the same as that used in Chapter 3. Rather than estimating the roughness parameters B in the covariance function however, prior distributions are assigned to the diagonal elements of B, giving a fuller account of uncertainty. Details of these priors will not be presented here, but can be found in [118].

The hyperparameter priors from Equation 8.13 together constitute the conditional hyperparameter prior $p(\Theta|T)$. Additionally, the parameter likelihood is easily expressed from Equation 8.12. The tree prior $p(T)$ is the same used in the mean-shift CART from Section 8.2.2.

Posterior Search

Since the form of $p(\Theta|T)$ is considerably more complex than that used by Chipman *et al.*, numerical integration is used to marginalise parameters. In fact, the tree-structured GP approach goes one step further, by using *Reversible-Jump Markov Chain Monte Carlo* (RJ-MCMC) sampling [129] to fully integrate out tree dependence, i.e.,

$$p(Y|X) = \int p(Y|X, \Theta, T)p(\Theta|T)p(T)d\Theta dT \tag{8.14}$$

This is now a formal model-averaging approach, since all uncertainty has been accounted for, not only in the parameter estimation, but also in the tree estimation. The nice thing about averaging over trees is that although for any given tree the posterior mean will be discontinuous over leaf boundaries, when model averaging is used it will actually smooth out at the boundary, as a result of averaging over many possible models. Importantly for this investigation, however, it still retains the ability to model bifurcating data at leaf boundaries.

In order to sample from the joint posterior of (Θ, T) (remembering that parameters have not been analytically marginalised here), MCMC is used to explore the model space, consisting of jumps between parameter sets and trees. Moves between trees and parameter sets are treated separately, so the sampling performs a move to a new set of parameters, then a new tree, then a parameter move again, and so on. In other words, samples are being drawn alternately from $p(\Theta|T)$, then $p(T|\Theta)$. In order to do this, it is necessary to use a mixture of forms of MCMC.

For moves in parameter-space, a sampling technique known as *Gibbs sampling* [130] is used where possible. A problem with the MH algorithm is that when the number of parameters to be estimated (i.e. the number of dimensions of the probability distribution to be searched) is more than a few, searching the parameter space can take a very long time. Gibbs sampling is a special case of the MH algorithm where the acceptance ratio is always 1, therefore all samples are accepted. However, instead of sampling from the joint posterior, the Gibbs algorithm samples sequentially from univariate *full conditional* distributions (distributions of a single parameter, conditional on all other parameters). This allows a faster sampling of parameters in many cases. Full conditional distributions can be derived for all parameters except those used in the covariance matrix (roughness parameters), therefore Gibbs steps may be used [124]. In the case of the roughness parameters, the MH acceptance ratio is used. The steps are performed over all parameters $\boldsymbol{\theta}_v$ for a single tree leaf, then repeated for all leaves, generating a set of parameters conditional on a tree $(\Theta|T)$.

Once a move to a new full set of tree parameters has been performed, it is necessary to move to a new tree, using a stochastic tree-jumping process similar to that used by

Chipman *et al.*. However, since the size of the parameter space (i.e. the length of the Θ vector) will change as a result of adding or subtracting leaves, in order to formally integrate out tree dependence it is necessary to use RJ-MCMC, which is a generalisation of the MH algorithm that can cope with jumps between parameter spaces of different dimensions. RJ-MCMC copes with the jumps between dimensions by using a function h that dictates the next state of the chain Θ' as a function of the current state Θ and some vector of random numbers u, generated from a density g such that,

$$h(\Theta, u) = (\Theta', u') \tag{8.15}$$

where u' is a new vector of random numbers. To make the reverse move, a set of random numbers u' is generated from a density g'. The key to RJ-MCMC is that if the transformation h is *bijective* (the sum of dimensions of Θ and u is equal to the sum of dimensions of Θ' and u'), the MH acceptance ratio can still be calculated, if appropriately modified. Crucially, although the sum of dimensions must remain the same, the size of Θ need not be the same as Θ'. The MH acceptance ratio now must include a Jacobian term to account for the change of variables,

$$\alpha_{\text{RJ-MCMC}} = \min\left\{1, \frac{p(\Theta^{\star})g'(u')}{p(\Theta)g(u)}\left|\frac{\partial h}{\partial(\Theta, u)}\right|\right\} \tag{8.16}$$

Note that given the presence of the Jacobian term, it is necessary that the function h and its inverse are differentiable — this is known as a *diffeomorphism.* Further details about RJ-MCMC will not be discussed here, however the interested reader can refer to a good recent tutorial by Green and Hastie [131], which is perhaps a little more accessible than Green's original paper [129].

Setting up a RJ-MCMC parameter search can be somewhat time-consuming, but it is a very powerful method, allowing searches across parameter and model space. The MH acceptance ratios must be calculated for each possible move, but since the move types are very few, this is not necessarily too arduous. The acceptance ratios for the moves used in the tree-structured GP search will not be presented here, but can be found in [118].

Limiting Linear Models

One further additional feature used in the *tgp* package is the concept of Limiting Linear Models (LLMs). The response of a model in part (or all) of the input space may be linear, in which case fitting a GP is unnecessary, and increases computational expense for no reason. In the case of a region where the data is sufficiently linear, the GP jumps to the LLM, which is a special case of Equation 8.12, such that,

$$\mathbf{y}_v|\mathbf{w}_v, \sigma_v^2 \sim \mathcal{N}_{n_v}(\Phi_v\mathbf{w}_v, \sigma_v^2 I) \tag{8.17}$$

where I is the $n_v \times n_v$ identity matrix. This is actually the Bayesian linear model considered by Chipman *et al.* as an extension of basic Bayesian CART [123]. The switch is implemented by a Boolean operator that chooses the GP or the LLM based on the estimated roughness matrix B. If the roughness parameters b_i for each dimension are sufficiently small, the data can be treated as linear. The full details of this will not be discussed here, but can be found in [124]. It will however be seen to be a significantly useful feature in the following case studies.

8.2.6 Application to UA/SA

The use of tree-structured regression models in UA/SA has been considered before by Pappenburger *et al.*, but only using simple regression trees and random forests [132]. The tree-structured GP method is considerably more sophisticated, yet the practical details of calculating the necessary integrals from Section 3.2 are not too difficult.

The approach of Gramacy *et al.* [133] is that at each tree (and corresponding parameter set) that is visited during the posterior search, output predictions are made over a large selected set of input locations based on an LHS design. For each set of predictions, the necessary integrals may be evaluated conditional on the predicted output by Monte Carlo integration [44]. Although this approach is less elegant than the analytical methods used in Chapter 3, it does have the advantage that UA and SA estimates now account for uncertainty over possible models, and all hyperparameters are estimated in a fully Bayesian fashion. Additionally, rather than calculating sensitivities as ratios of expected values, the numerically-integrated sensitivity indices here are true estimates of the S_i. Full details of the numerical integration are found in [133]. In short, the tree-structured GP can estimate all of the sensitivity measures discussed in Section 2.2.2 and used in previous chapters.

8.2.7 Summary of Tree-Structured GPs

The Bayesian approach to mean-shift CART models has been shown here to be extendable to tree-structured GPs, albeit with the introduction of some new features, such as the hierarchical GP model and the necessary RJ-MCMC to integrate out tree dependence. The tree-structured GP method relies heavily on numerical integration, but in terms of computational expense it has been found by the author to be rather efficient and in no way time-consuming to analyse substantial data sets. This is in part due to the useful reduction in CPU time due to using multiple smaller GPs, and perhaps the fact that roughness parameters are estimated by MCMC, rather than the (possibly) more time-consuming optimisation that is required by the standard GP. A test of the tree-structured GP now follows with a simple case study.

8.3 Example: Duffing Oscillators

In order to compare the abilities of the tree-structured GP against the standard GP, both emulators were applied to the bifurcating response of a Duffing oscillator. This is a very simple case-study of a response bifurcation that can be governed by two parameters.

8.3.1 About Duffing Oscillators

The Duffing oscillator is a very well-known example of a nonlinear system in structural dynamics. It is a sinusoidally-forced single degree of freedom system with a cubic stiffness term:

$$m\ddot{y} + c\dot{y} + ky + k_3y^3 = P\cos(\omega t) \tag{8.18}$$

where m is a mass, c is a damping coefficient, k and k_3 are stiffness constants, P is the forcing amplitude and ω is the forcing frequency. y, $\dot{y}$ and $\ddot{y}$ denote the displacement, velocity and acceleration respectively.

The frequency response function (FRF), in the case of $k_3 = 0$, is single-valued at any frequency. However, when $k_3 \neq 0$, at certain frequencies the response of Equation 8.18 can have three possible amplitudes, since they are the solutions of a cubic equation. Of these three, one amplitude is unstable and is never achieved in practice, however, as the system approaches a steady state it snaps to one of the two remaining amplitudes, either the high amplitude or the low one. Whether the system ends up in the high or low amplitude is determined by the initial conditions of the system: the initial displacement y_0 and the initial velocity $\dot{y_0}$. For a more detailed description of the response of Duffing oscillators the reader is referred to [134].

The amplitude response therefore provides a taxing system with which to test the GP and tree-structured GP methods. Essentially the steady state amplitude can be seen as a function of the two initial conditions. But since the amplitude can only take on either the high value or low value, there will be bifurcations in the response. Examples of this are illustrated in the following section. In order to investigate response surfaces of varying complexity, several areas of parameter space were sampled, which will be referred to as cases 1-3. These are explained in the following pages.

8.3.2 Results

Cases 1 and 2

Figure 8.4 shows a high-resolution amplitude plot of two of the parameter regions investigated, cases 1 and 2. This therefore represents the "true" solution. Case 1 was chosen

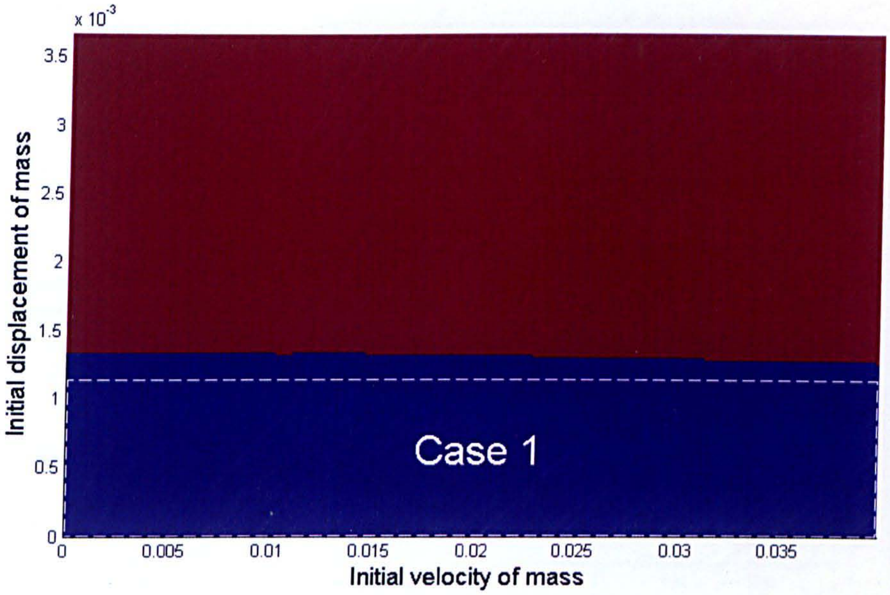


Figure 8.4: Case 2, single bifurcation (case 1 region illustrated). Red=high, blue=low.

to be the simplest possible response surface - an area where the amplitude is completely insensitive to parameter variations. As such, the sensitivity data and model fits are not illustrated here since it is a trivial problem. Case 2 provides a more interesting test. Expanding the range of initial displacement reveals a region of high amplitude response and a bifurcation between the low amplitude (blue) and high-amplitude (red) regions.

A set of 49 training points was drawn with a maximin Latin hypercube sampling strategy (see Section 2.1.3). This was then used to train both the GP and tree-structured GP models. Figure 8.5 shows surface plots of the fitted models and the training samples. It is immediately clear that the tree-structured GP models the data in a much closer way to the real response. The GP, on the other hand, introduces a type of Gibb's phenomenon in order to accommodate the steep bifurcation, because the GP's roughness parameters are necessarily constant over the whole model. The bifurcation forces the roughness to be high in the direction of initial displacement, but this value is not suitable in the perfectly smooth regions on either side, causing undulations in areas of sparse training data. Conversely, the tree-structured GP can snap to the linear model in the high and low regions, and easily model the bifurcation with a two-leaf tree. That the bifurcation does not appear perfectly vertical in Figure 8.5(b) is due only to the resolution of predictive points used to plot the surface. The MAP tree for the tree-structured GP is not illustrated here since it has only a single split at $y_0 = 0.00122$.

Figure 8.6 shows a main effect plot of the input parameters for case 2. Both models capture the variation of the output with respect to each input quite well. The tree-structured GP, however, captures the linearity of the high and low regions and the bifurcation with more accuracy. The main effect of initial velocity is virtually flat in both models (since the

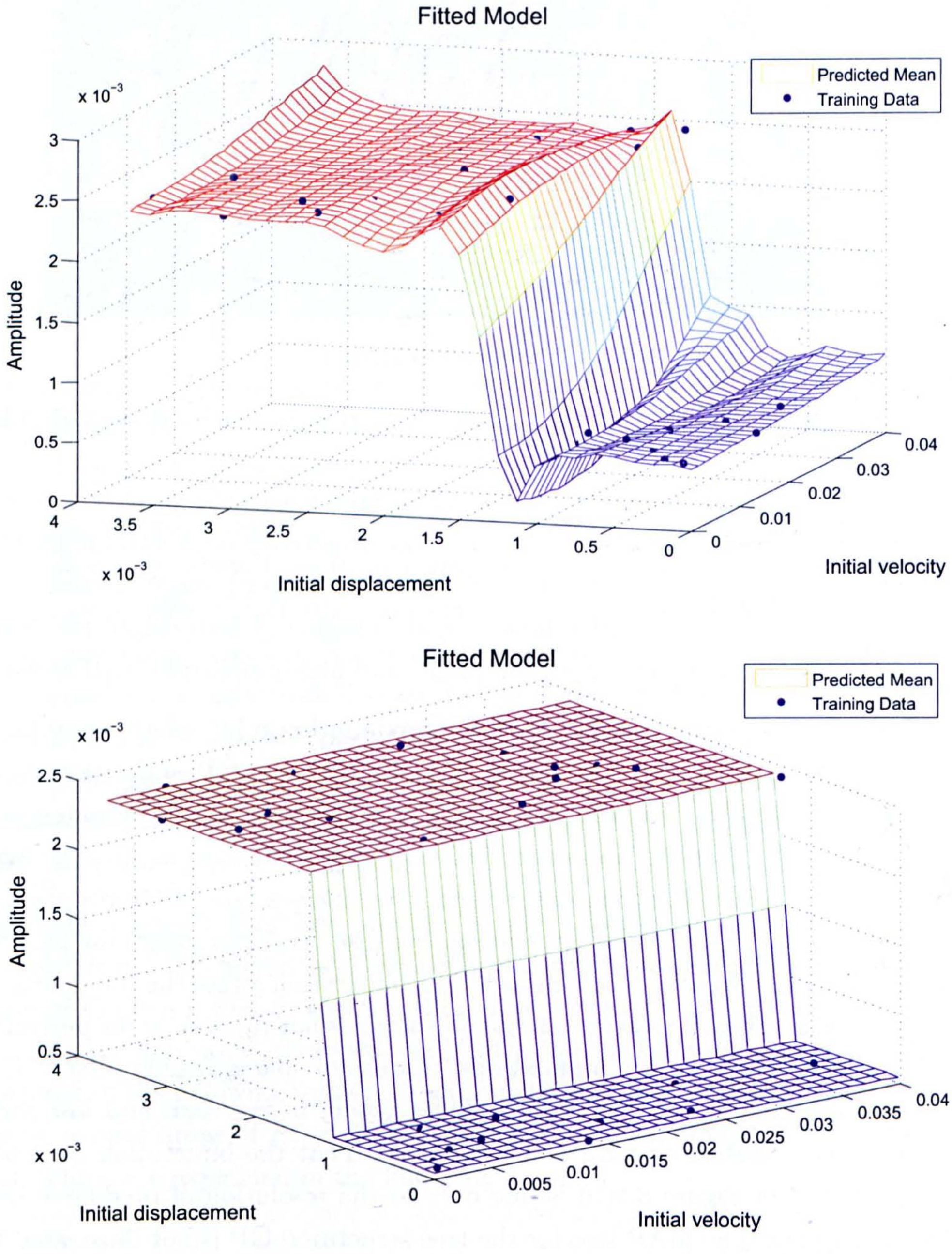


Figure 8.5: Case 2 fitted emulators: GP (above); tree-structured GP (below).

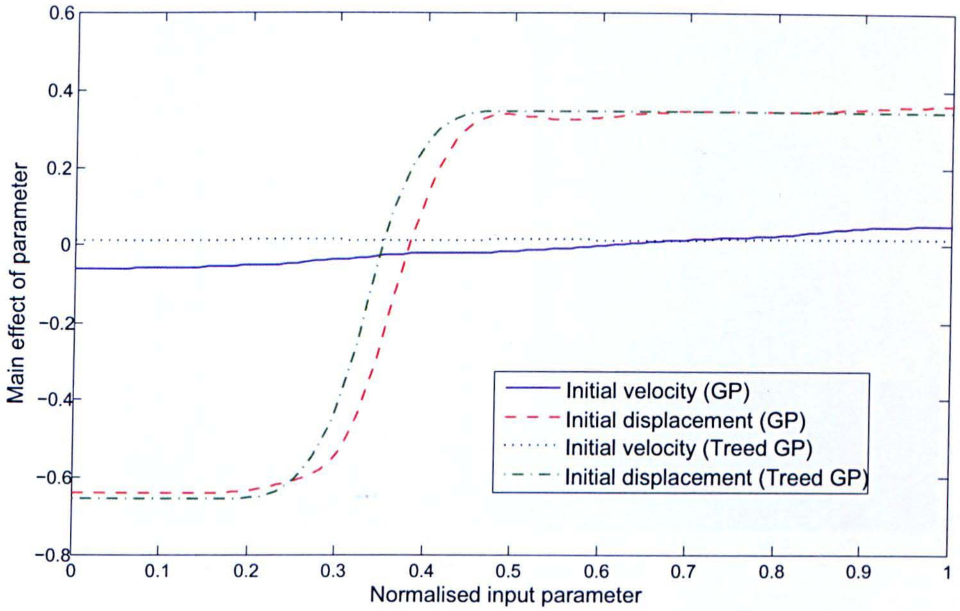


Figure 8.6: Posterior means of main effects, case 2, GP and tree-structured GP.

bifurcation is not exactly parallel to the velocity axis it would not be expected to be perfectly flat) but the GP appears to show a little too much sensitivity, which is reflected in the slight gradient in the velocity direction in Figure 8.5.

Case 3

Figure 8.7 illustrates the parameter space of case 3, which is an enlargement of the parameter space in case 2. This reveals several bifurcations, including one that is not perpendicular to either axis and also curved. This poses immediate problems for even the tree-structured GP, because divisions of input space must be made perpendicular to the axes. An initial LHS sample of 49 training points was used to train the model, but initial analyses suggested that more training data were needed to model the complex response. The sample was therefore increased to 100 training points, the results of which are shown in Figure 8.8.

As in case 2, the GP cannot faithfully reproduce the response surface, introducing fluctuations in the surface as a result of the several bifurcations. It is clear that a standard GP is not suitable for modelling this kind of surface (which is a well established fact). The tree-structured GP, on the other hand, does a reasonable job. The bifurcations in the region of $y_0 = 0.005$ are captured quite well. However, the tree-structured GP struggles with the curved bifurcation and tries to fit a linear surface in that region. This appears to be partly due to a lack of training data in that area; Figure 8.9(a) (a plot of predictive uncertainty) shows that in the lower right region there are few training points and subsequently the

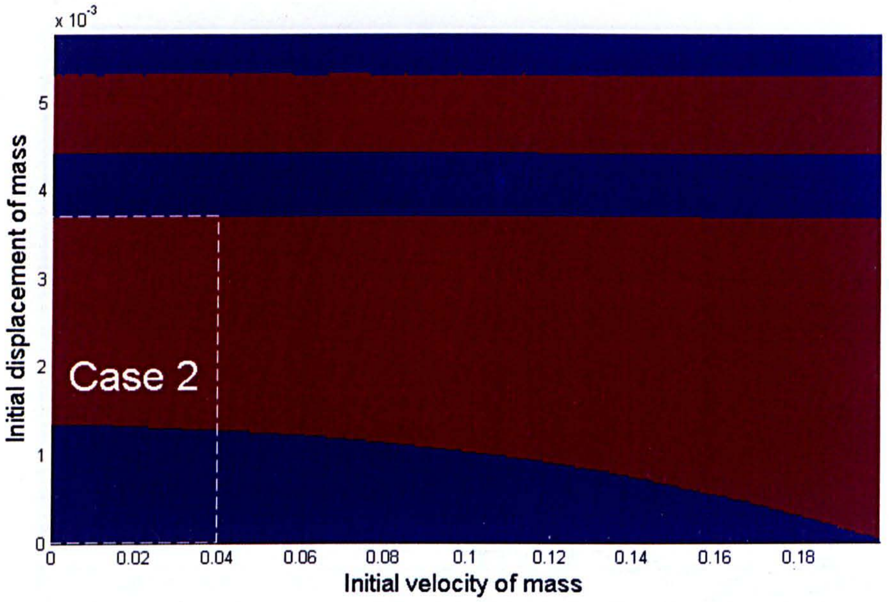


Figure 8.7: Case 3: multiple bifurcations; case 2 region illustrated. Red=high, blue=low.

predictive uncertainty is high. However, this serves to illustrate one of the advantages of GPs: despite a poor fit, the high uncertainty in prediction is recognised and can be used to suggest locations for further training data. The MAP tree is shown in Figure 8.9(b) with the six leaves (regions in 8.9(a)) as circled numbers and the variable splits shown at each branch. It is evident that a considerably more complex tree is required here than for case 2, and that ideally the tree should be even more complex than shown to more closely model the curved bifurcation.

Model uncertainty is also reflected in the estimates of main effects and sensitivity indices. Figure 8.10(a) shows the main effect plots of both models. The main effect lines agree to a large extent, though the tree-structured GP captures the large high-amplitude region more accurately. Figure 8.10(b) shows the 95% confidence intervals of the main effect plots. The uncertainty is markedly higher in the initial velocity line. This could be due to the region of high uncertainty discussed previously. Another effect of this is that although the y_0 main effect has very small error in most regions, for the lowest values of y_0 the uncertainty increases substantially.

Posterior estimates of sensitivity indices are shown in Figure 8.11 for both the standard and tree-structured GPs. The estimates are quite similar for the MEIs, though perhaps the most notable difference is that the TSI of initial velocity is somewhat higher in the tree-structured GP model, suggesting more interaction variance between the two parameters. This is again possibly a result of the poorly-fitted region in the tree-structured GP. Nevertheless the sensitivity indices are quite similar overall.

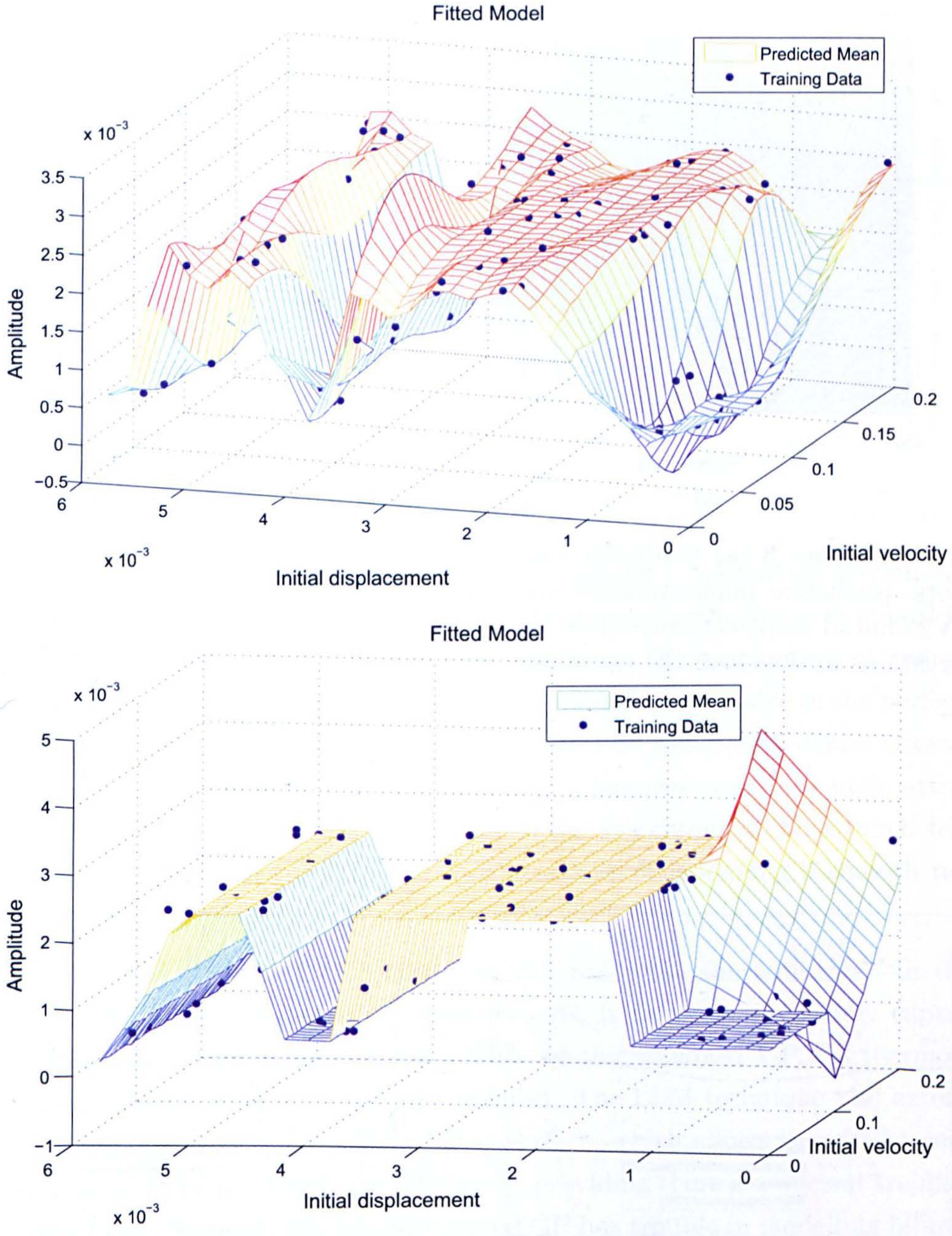
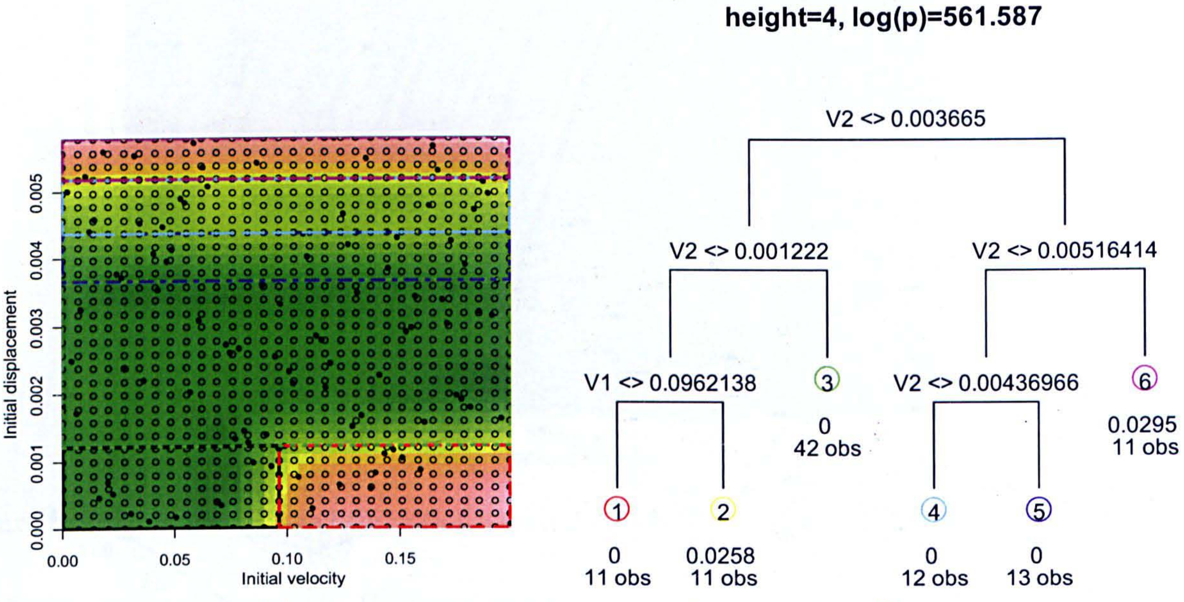
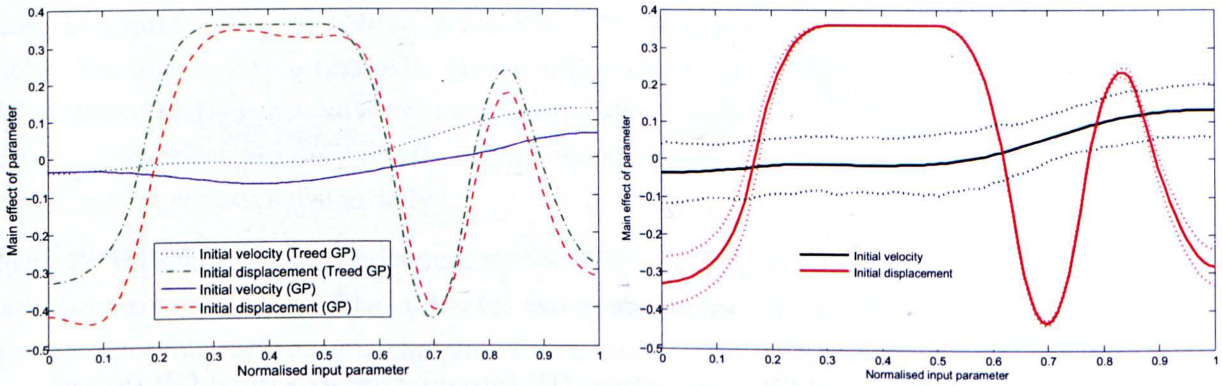


Figure 8.8: Case 3 fitted emulators: GP (above); tree-structured GP (below).



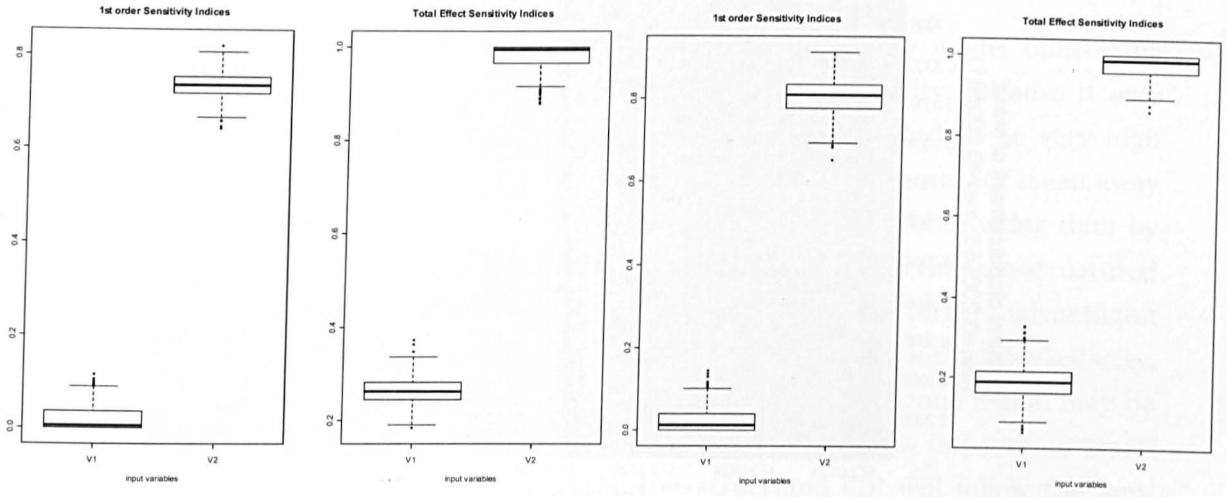
(a) (b)

Figure 8.9: Case 3 (a) predictive uncertainty and tree divisions, training data marked as dots, predictive points marked as circles); (b) corresponding tree, V1=initial velocity, V2=initial displacement; circled numbers are leaf regions; "obs" denotes number of observations in that leaf; the remaining number is the MAP estimate of σ^2.



(a) (b)

Figure 8.10: Case 3, posterior means of main effects: (a) both models; (b) tree-structured GP with 95% confidence intervals (dotted lines).



(a) (b)

Figure 8.11: Posterior estimates of sensitivity indices: (a) GP (V1=initial velocity, V2=initial displacement); (b) tree-structured GP

8.3.3 Discussion

The ability of standard and tree-structured GPs to model a bifurcating system has been examined for simple and more complex regions. It is no surprise that a standard GP has some trouble modelling a bifurcation, because this forces the roughness hyperparameters to be high, which are then used over the full model, and are unsuitable in the perfectly flat high- and low-amplitude regions. If the training data were sufficiently dense, however, the GP would become increasingly accurate, though this removes one of the main attractions of GP-based SA. Nevertheless, the Duffing oscillator was chosen to be a tough test, and in a great number of cases the GP is an excellent way of modelling a smooth response surface.

The tree-structured GP is rather better, but still has some shortcomings. In the case of a bifurcation that is parallel to a parameter axis, it can very accurately capture the bifurcation in a way that is beyond the ability of the standard GP. Furthermore, the computational cost of doing so is actually reduced. The LLM technique also extends the flexibility of the model by recognising a linear surface, which allows the almost constant-amplitude regions to be modelled very effectively, providing there is sufficient training data to invoke the LLM. However, the tree-structured GP has trouble in modelling bifurcations that are functions of more than one parameter (not parallel to any axis, or curved). In case 3 the tree-structured GP fails to capture the curved bifurcation, resulting in areas of high error. In order to closely model the curved bifurcation, a very complex tree would be required, since partitioning can only be performed parallel to axes. Kim et al. [135] have indeed addressed this issue with the use of Voronoi tessellations, although these such divisions of input space could be regarded as overkill for simple bifurcations.

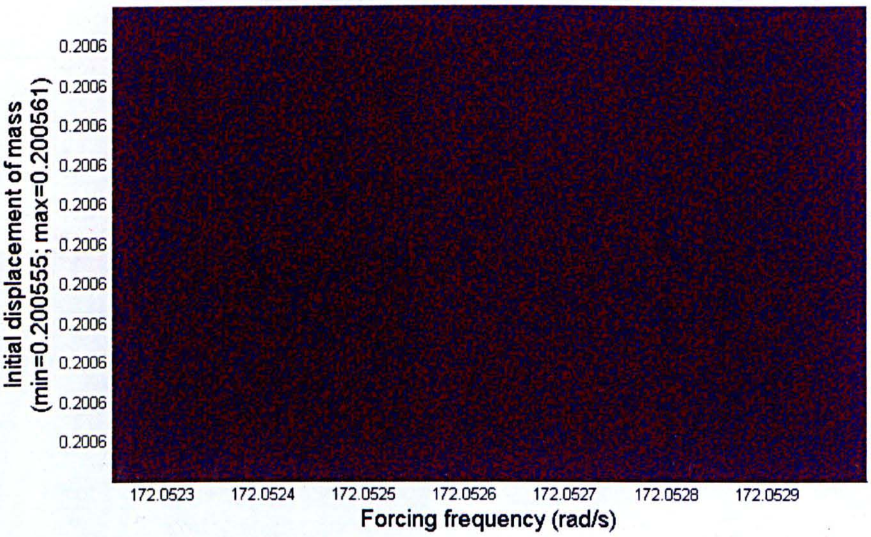


Figure 8.12: "Dust" region: initial conditions are displacement and forcing frequency. Red=high, blue=low.

Main effects and sensitivity indices can be readily gained from both models, which tend to agree to a large extent in the cases investigated. However, as a result of the over-fitting of the data by the standard GP from the presence of bifurcations, the sensitivity of the output to some parameters can be over-estimated slightly. In both cases however, there are very significant computational savings over Monte Carlo estimates. Both models provide a powerful framework from which to apply SA to complex models.

As a point of interest, some other areas of parameter space were investigated that reveal very complex responses that illustrate the potential difficulty of emulating certain systems. Figure 8.12 shows the amplitude response where the initial displacement is varied along with the forcing frequency for a chosen range of values. There is no discernable pattern, other than the appearance of scattered dust. Indeed, the amplitude response shows strong indications of being fractal [134]. It should be remembered though that the Duffing oscillator is an extreme example with numerous bifurcations, which would perhaps not be expected in many systems.

8.4 Conclusions

This chapter has outlined an extension to the GP emulator used in previous chapters, that uses a classification and regression tree (CART) to divide the input space into non-overlapping regions and fits separate GPs in each leaf (region) dictated by the tree. The construction of the tree has been presented, using a Bayesian approach that specifies a prior distribution over trees and parameters, and gives a posterior prediction based on training data. The use of Markov Chain Monte Carlo methods as a way of estimating trees and parameters has been discussed.

This "tree-structured GP" has been shown to be able to effectively model bifurcating data, whereas a single GP covering all input space has great difficulty, because it uses a single set of roughness parameters for all the data, which are forced to be very high due to the bifurcation. This introduces unwanted fluctuations in the posterior mean away from the bifurcation. In contrast, the tree-structured GP can model bifurcating data by positioning the leaf boundary at the point of bifurcation. The fact that the tree-structured GP can specify separate sets of hyperparameters for each region has further advantages: prior variance estimates between regions can be independent, allowing for heteroskedacity. Also the roughness coefficients can vary from region to region; indeed, one region may be modelled with a linear model and another with a GP. This also has impacts on small extrapolations outside of the data range — the tree-structured GP will follow the *local* trend outside of range of training data, whereas the single GP will necessarily snap back to a global trend, which is likely to be a poorer estimate. Finally, because the tree-structured GP inverts a number of smaller covariance matrices in training itself to the data, it actually comes with a reduction of computational expense over the single GP.

The main limitation of the tree-structured GP, however, is that splits in the input space are necessarily functions of a single variable and are therefore always parallel to axes. An example of a curved bifurcation in the Duffing oscillator case study here showed that the tree-structured GP has great trouble in modelling anything but axis-aligned bifurcations, requiring overly-complicated trees to describe such data. However, the tree-structured GP naturally includes variance estimates and techniques such as cross-validation can be used to identify poor model fits.

The following chapter investigates the use of tree-structured GPs on a bifurcating FE model.

Chapter 9

Emulating a Bifurcating System

Given the indications of possible model bifurcations in the aortic valve (AV) model in Chapter 5, a method for performing uncertainty and sensitivity analyses on bifurcating systems, based on tree-structured Gaussian processes (GPs), has been outlined in Chapter 8. The method has been shown to be more effective than using a single GP since it can handle discontinuities at leaf boundaries. Furthermore it actually reduces computational expense.

The tree-structured GP emulator will now be applied to a case study of a rigid-stent prosthetic AV. The AV model in Chapter 5 has been shown to have very nonlinear responses with respect to sinus/aorta material properties, which is to say that leaflet opening is highly dependent on the constraints imposed on the leaflet boundaries. Taking this to the extreme here, the leaflet boundaries are constrained in all degrees of freedom, which is the case of a conventional "rigid-stent" bioprosthetic valve. The response of this kind of model is expected to result in bifurcations since it is an example of a buckling or snap-through problem.

The rigid-stent model will be used to test the emulators outlined in Chapters 3 and 8. The effect of reducing training data is also briefly investigated in Section 9.2.3.

9.1 Dry Leaflet Model

The model used here simulates a rigid-stent prosthetic AV. The model is based on the "wet model" configuration from Chapter 4. It incorporates the soft tissue material model and simulates half of the leaflet, with assumptions of symmetry across the leaflet and cyclic symmetry around the valve. In contrast to the original wet model, the loading on the leaflet is due only to pressure (rather than Eulerian fluid elements) to simplify the analysis and the sinus has been removed since this is now treated as rigid. The points of the leaflet that were formerly connected to the sinus are now constrained in all degrees of

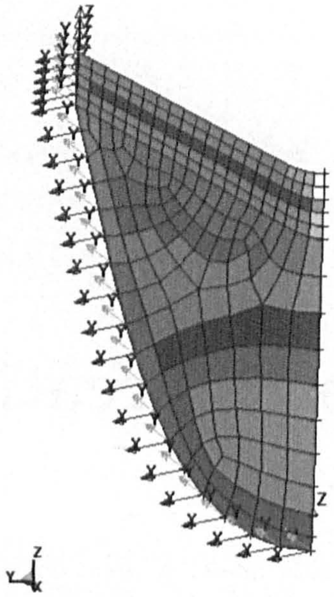


Figure 9.1: An illustration of the mesh of the dry leaflet model. Constrained nodes are marked with arrows.

freedom, simulating the attachment of the leaflet to a rigid stent, as used in many porcine bioprosthetic devices. Figure 9.1 illustrates the layout of the model.

Since the leaflet is an example of a buckling or snap-through problem, bifurcations were expected to occur as a result of varying the loading on the structure, and potentially the material properties of the leaflet. To keep the analysis relatively simple, two parameters were picked to vary - the pressure exerted on the leaflet, and the post-transition modulus of the soft tissue material model (see Section 4.3.2). The range of pressure was deliberately made wider than is necessarily realistic in the human body in the interests of capturing a bifurcation in the response; the realistic range can be considered as a subset of the range presented here.

100 points were used to train the emulators, using the usual maximin LHS strategy. This is a large amount of data for two dimensions, so the effect of reducing the number of training points is investigated in Section 9.2.3 as a matter of interest.

9.2 Results

9.2.1 Raw Data

Figure 9.2 shows the movement of the leaflet over 12ms for a mid-range pressure load. This represents the movement of the leaflet for any area of parameter space where it reverses its curvature (i.e. the valve opens). It is clear that the movement of the leaflet is nonlinear over time, since at the point of reversal-of-curvature (ROC) the leaflet accelerates towards

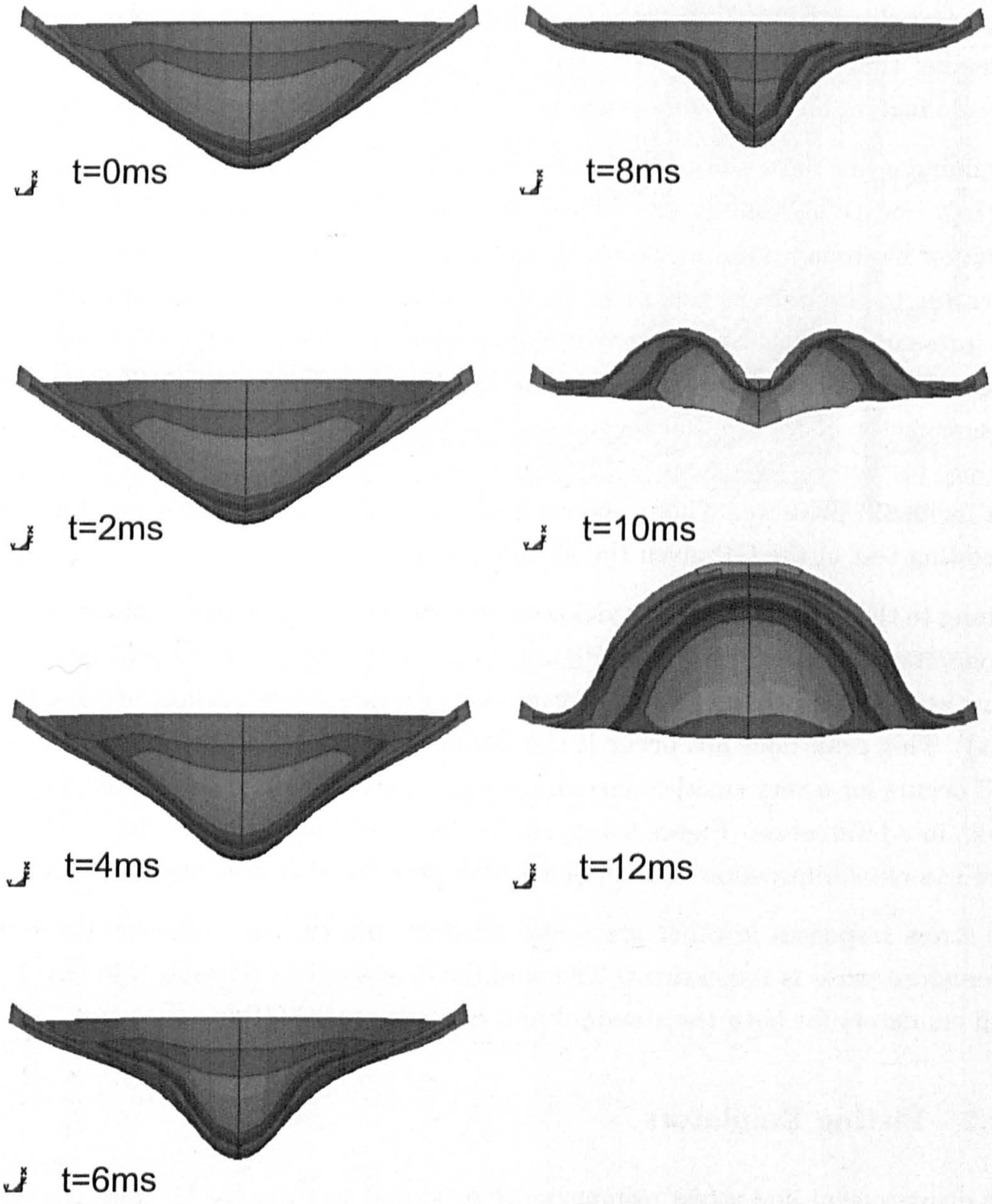


Figure 9.2: The opening of the "rigid-stent" leaflet over 12ms (top view).

its reversed state. This is illustrated in more detail in Figure 9.3 (a), which shows that the initial application of pressure is not sufficient to displace the leaflet by anything but a very small amount (i.e. in the first 5ms of the graph). However, when the leaflet reaches a critical pressure the displacement very quickly increases to the point where the leaflet has reversed its curvature. There are then oscillations due to the lack of damping in the model, although these do not affect the results to a significant degree. The critical pressure occurs at different times depending on the pressure scaling factor used for that run of the model (remember that the pressure is ramped over the 20ms from zero to the value dictated by the scale factor) For the lowest pressure values the valve does not open at all.

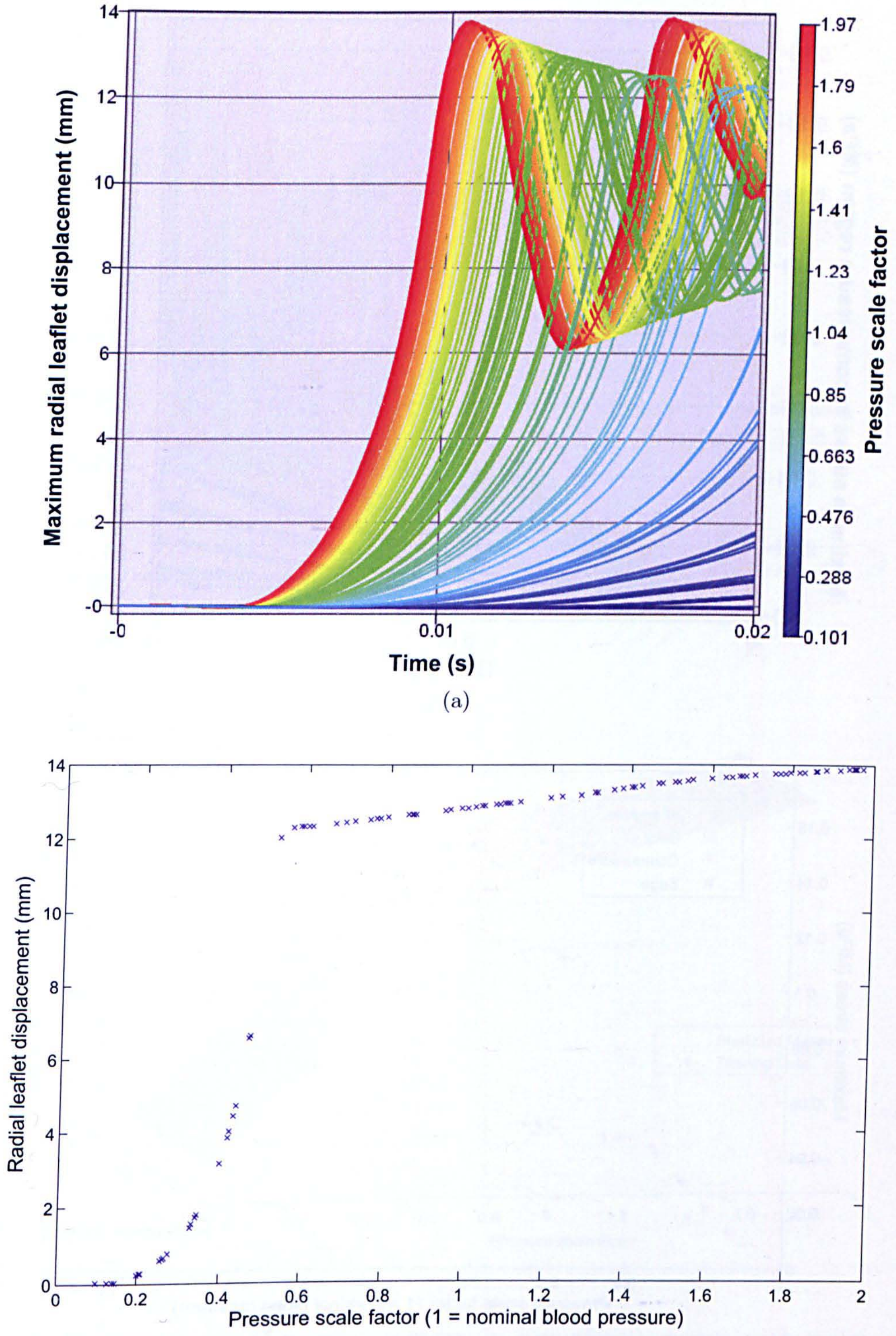
Examining a raw data plot of the maximum leaflet displacement against pressure (Figure 9.3 (b)), one thing that is immediately evident is that the material parameter (post-transition modulus) does not seem to be affecting the output in any way, since there is no scatter to the data in this plot. Instead, there is a very clear nonlinear relationship with pressure. Displacement increases in a kind of exponential fashion with pressure, representing the diminishing resistance of the leaflet as it reaches the ROC point. At a pressure factor of around 0.5 the pressure is sufficient to completely reverse the leaflet and any further displacement is due to stretching of the leaflet, which is roughly linear with increased pressure. This response does not constitute a bifurcation, but will be an interesting test of the GP given the abrupt changes in smoothness.

Turning to the variation of stress yields some very interesting response plots. History plots of commissure stress are shown in Figure 9.4(a). There is a very sharp peak in stress that occurs at the point where the leaflet reverses its curvature (the earliest of these is at about 0.01s). This peak does not occur if the leaflet does not reverse its curvature, and since ROC occurs for a very small change in pressure, the response of *maximum* stress actually results in a bifurcation. Figure 9.4(b) shows this trend: at a pressure factor of around 0.5 there is a clear bifurcation at the point where pressure is just enough to open the valve.

The stress responses in other areas also produce bifurcations, however, the response of commissure stress is the clearest. This and the displacement response will now be used to train emulators for both the standard and tree-structured GPs.

9.2.2 Fitting Emulators

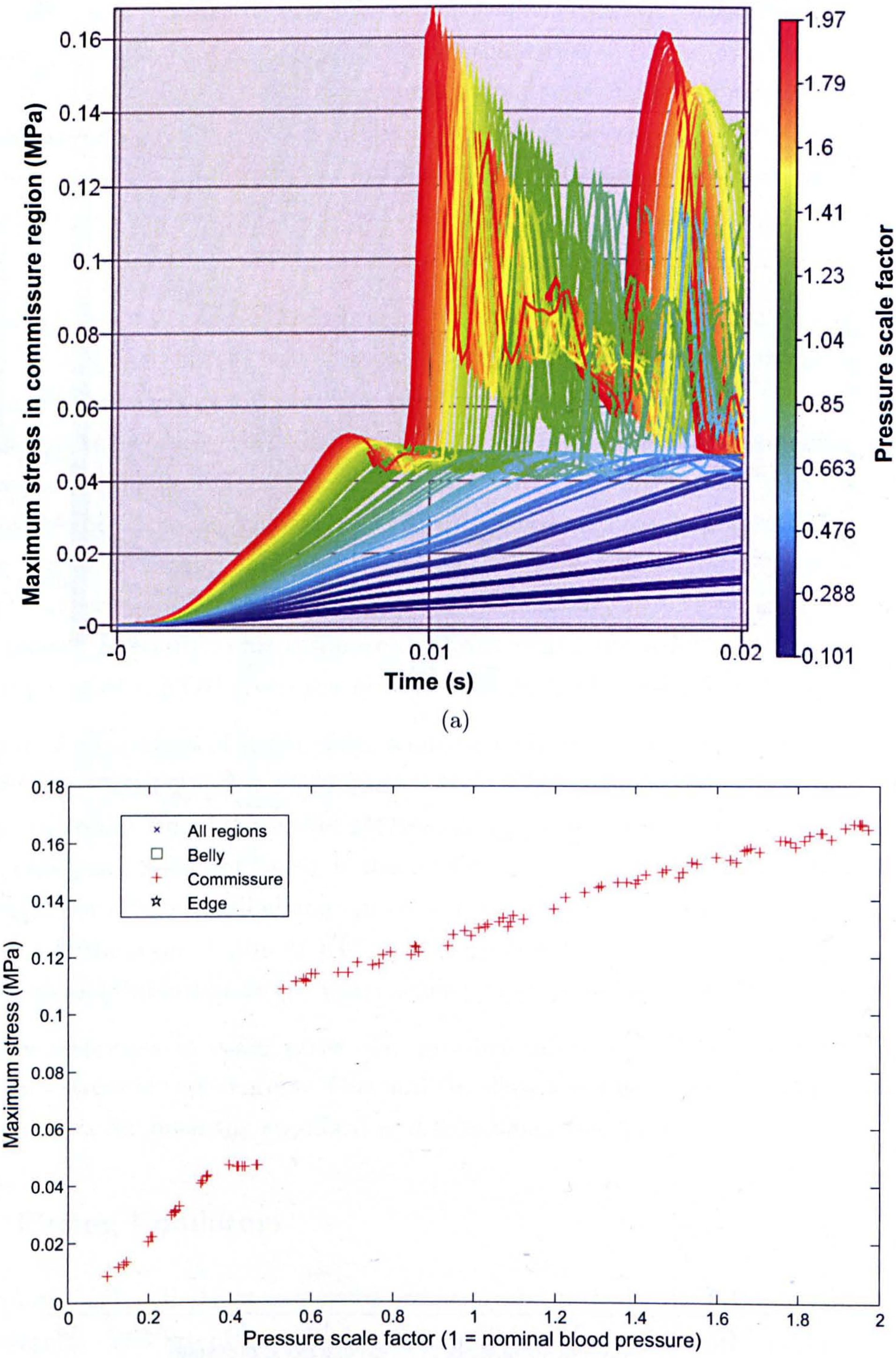
The displacement and stress responses are now used to train the GP and tree-structured GP emulators. The intention here is to examine the ability of both emulators to model bifurcating data from an FE model. Beginning with the clearest bifurcation, that of the commissure stress, Figure 9.5 shows the fitted emulator means from both the tree-structured and standard GPs with the training data overlayed. The difference in quality of fit is immediately evident: similar to the Duffing oscillator response in Chapter 8, the standard GP is constrained by the necessity of assigning the same length-scales (roughness



(a)

(b)

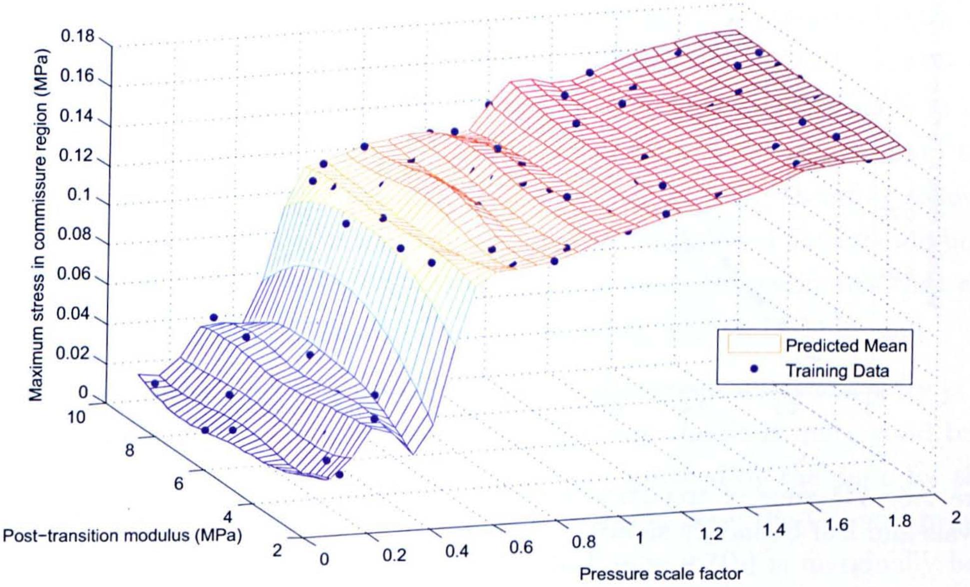
Figure 9.3: Radial leaflet displacement: (a) history plot for all runs (colours represent pressure values); (b) scatter plot of maximum leaflet displacement against pressure.



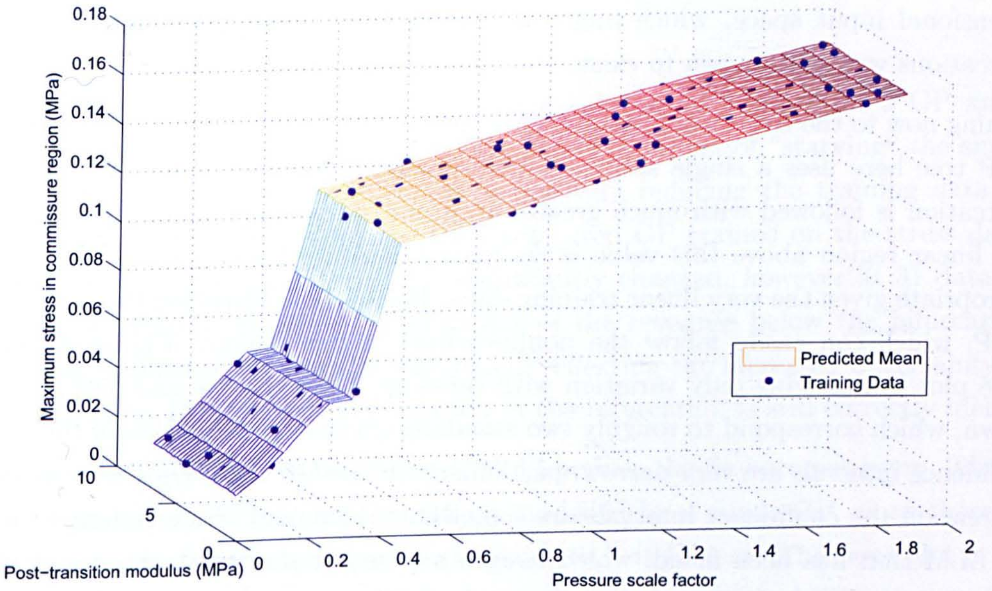
(a)

(b)

Figure 9.4: Stress in commissure region: (a) history plot for all runs (colours represent pressure values); (b) scatter plot of maximum stress against pressure.



(a)



(b)

Figure 9.5: Posterior mean of maximum stress in commissure region with training data overlayed (dots): (a) GP; (b) Tree-structured GP (split at scale factor = 0.467).

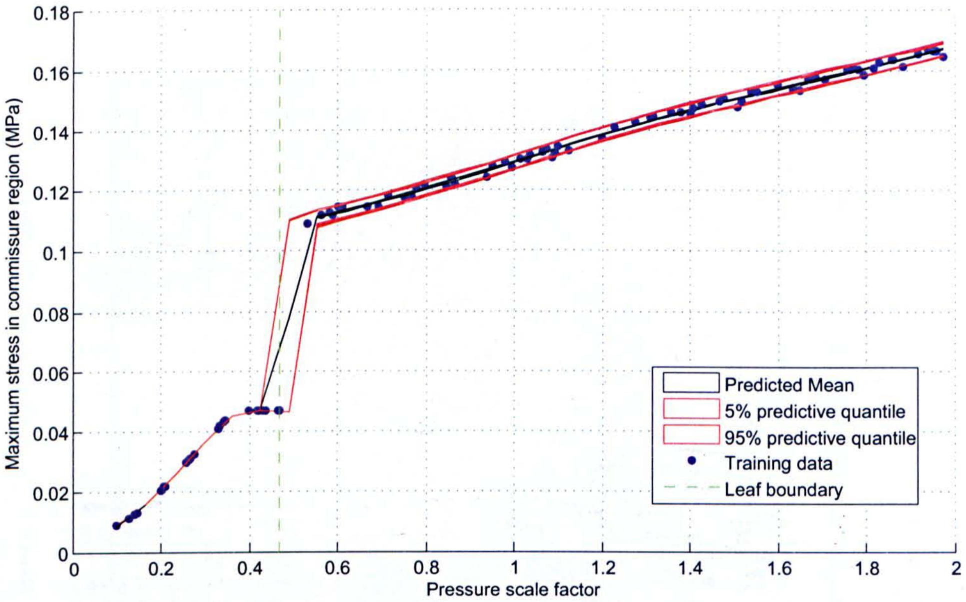


Figure 9.6: 2D view of tree-structured GP fit to commissure stress data. Confidence intervals and leaf boundary shown.

parameters) to all of the input space. As a result, it introduces oscillations close to the bifurcation that should not exist. Note that this is also with 100 data points for a two-dimensional input space, which might be considered a dense training set for a GP, if bifurcations were not known to exist.

Turning now to the tree-structured GP (Figure 9.5(b)), the fit is considerably better. The MAP tree here uses a single split at a pressure scale factor of 0.467, which means the bifurcation is followed with much greater accuracy, since it occurs at the leaf boundary. The linear region above this value is assigned a limiting linear model (LLM), which is appropriate given the very linear training data. Below the bifurcation the data is assigned a GP, which can easily follow the nonlinearities in this region. Figure 9.6 shows the same plot but viewing only variation with pressure. Here the 5% and 95% quantiles are shown, which correspond to roughly two standard deviations either side of the mean. The confidence intervals are very narrow, particularly in the GP area. On the right side of the bifurcation the confidence intervals are somewhat wider, and of a constant width due to the LLM that has been fitted, which assigns a constant posterior variance to all points. Overall it must be said that the tree-structured GP provides a very marked improvement over the standard GP, all the more given that the bifurcation is parallel to an axis, which is very easy for the tree to deal with.

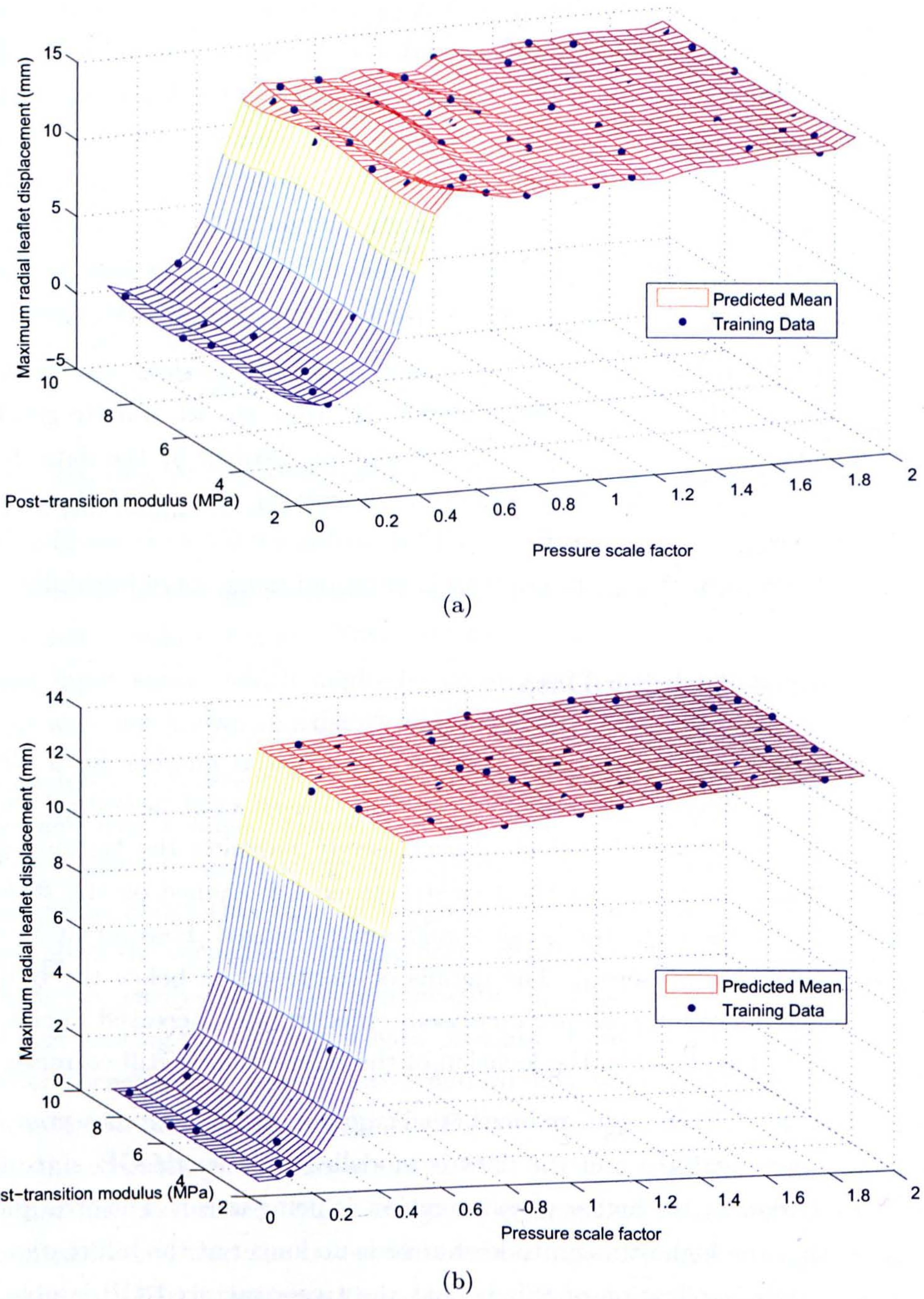
Figure 9.6 raises a further possible advantage of the tree-structured GP, which is in the context of extrapolation. For predictive points outside of the range of the training data, both the tree-structured and standard GPs will tend to veer back to their prior means, which are linear regression fits through the training data. In the case of the standard GP this regression fit is taken over the *whole* set of training data. Taking the data in Figure 9.5 as an example, it is clear that if a flat surface is fitted to all the data, it not going to be particularly close to *any* of the training points. Since the GP will necessarily revert to this prior mean outside of the range of training data, even for predictive points that are a small distance away from the training data it will provide predictive estimates that are not intuitively realistic. On the other hand, the tree-structured GP can take different prior means for different leaves of the tree. For example, in the linear region to the right of the bifurcation, predictions at higher pressures will follow the linear trend of that leaf, and not revert to a global prior mean (as in the case of the GP), therefore following the *local* trend rather than the global trend. Although extrapolation of data should be treated with caution, the tree-structured GP should produce predictive estimates that are more in line with intuition than the standard GP, particularly when a LLM is used.

Examining now the emulators' fits of the leaflet displacement data, the story is similar. Figure 9.7(a) shows the posterior mean of the GP. Here the fit is quite good but again there are undulations in the emulator fit that are not justified by the data, for the same reasons outlined before. Since there is no real bifurcation here, the GP still provides an acceptable fit however. The tree-structured GP in Figure 9.7(b) is marginally better — in particular the abrupt change in gradient is captured much more faithfully.

9.2.3 Reduced Training Data

Given that 100 training data points is quite a substantial number for a GP and tree-structured GP, it is interesting to briefly investigate the effect of "starving" the emulators of data. Figure 9.8 shows the effect of successively reducing the training data set by removing random data points for the tree-structured GP trained on the stress data. At 60 points the fit of the emulator is not significantly changed, however at 31 data points the fit is considerably worsened. The details of the response below the bifurcation are lost and the confidence intervals are increased, reflecting the increased uncertainty about predictive points. Despite this, the location of the bifurcation is still correctly identified.

When the training data is again reduced, to 17 points, the fit becomes poor. The bifurcation is now not identified and the data is modelled by a single GP, since there is not enough data to identify the higher-pressure region as being linear. The error plot in Figure 9.8(f) shows that the highest magnitude of error is no longer at the bifurcation, but in the linear region. The implication of this is that the tree-structured GP is able to emulate a bifurcating system, but only when there is sufficient training data to invoke a division in input space. This is unsurprising, but it is particularly crucial to have enough data *in the vicinity* of the bifurcation, since a bifurcation may look like a smooth increase when training data is scarce, which can be modelled more simply by a single GP. It is possible that adjusting parameters in the tree prior may have some improvement on the fit if a



(a)

(b)

Figure 9.7: Posterior mean of maximum leaflet displacement with training data overlayed (dots): (a) GP; (b) Tree-structured GP.

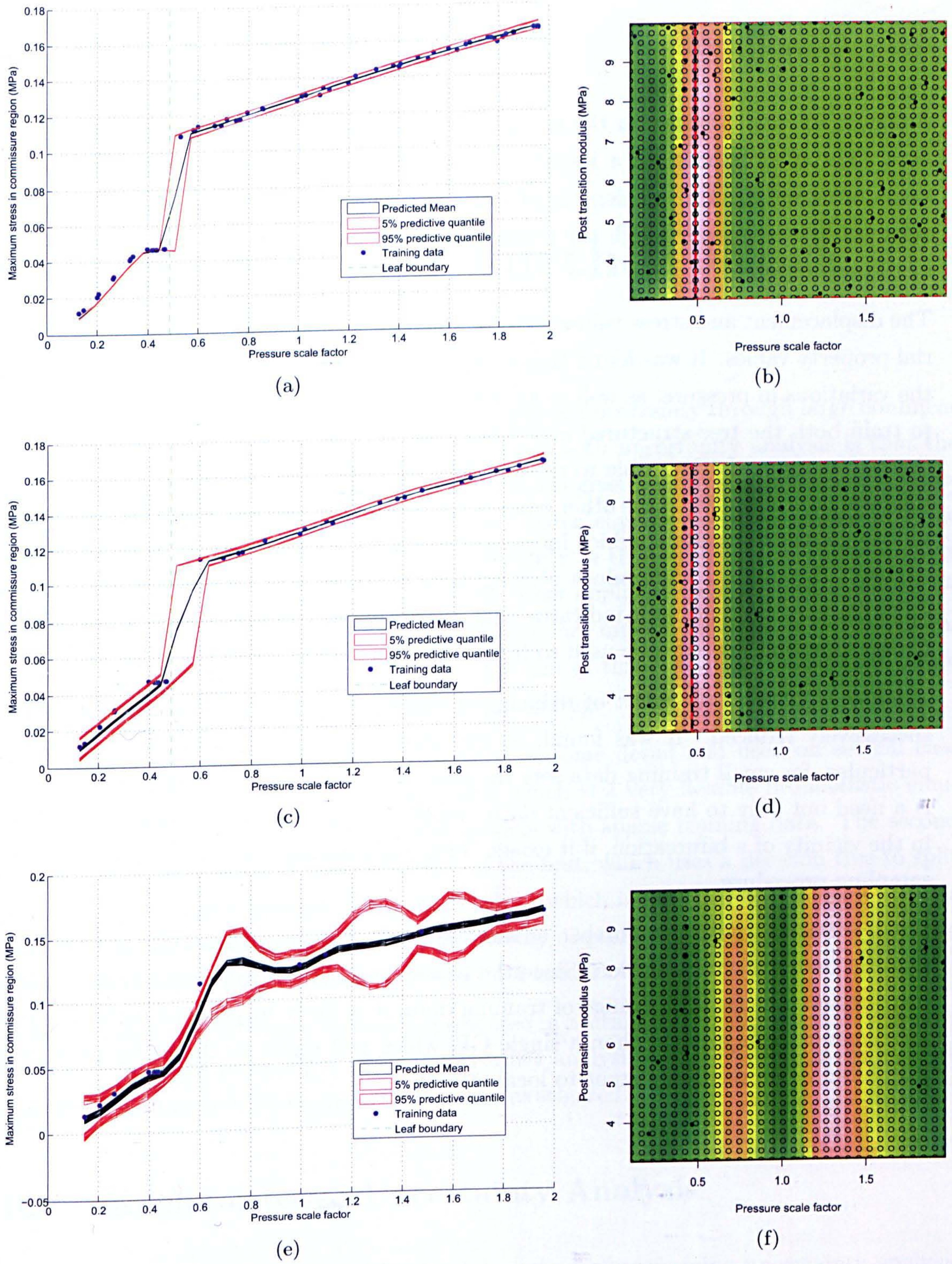


Figure 9.8: Tree-structured GP posterior mean of commissure stress with reduced training data: (a) 60 points; (b) 60 points error plot; (c) 31 points; (d) 31 points error plot; (e) 17 points; (f) 17 points error plot.

bifurcation is suspected, but more data is always preferable.

9.3 Conclusions

A simple FE model based on the aortic valve model from Chapter 4 has been presented here. The model simulates a leaflet in a rigid-stent prosthetic valve, where the leaflet boundaries are fixed in all degrees of freedom. The opening of the leaflet is therefore achieved by snapping through the reversal-of-curvature point, which creates bifurcations in certain responses.

The displacement and stress values were sampled for a range of pressure and leaflet material property values. It was found that clear bifurcations in maximum stress resulted from the variations in pressure, as well as abrupt changes in displacement. The data were used to train both the tree-structured and "standard" GPs — it was found that, as expected, the tree-structured GP is able to capture bifurcations and variations in smoothness over the response. The GP, on the other hand, is forced to use high values of roughness parameters due to the bifurcation, which introduces unwanted undulations in the much smoother regions. However, if high values of roughness parameters are noticed when training a GP, this may be used as a indicator that the GP is not correctly emulating the data. Cross-validation may also be used to further investigate the emulator fit.

In order to observe the effect of reducing the number of training data, the data set was successively reduced. It was found, as expected, that the quality of fit worsened. In particular, for small training data sets the bifurcation may not be identified at all. There is a need not only to have sufficient data overall, but particularly to have sufficient data in the vicinity of a bifurcation, if it exists. This could perhaps be achieved by an adaptive sampling procedure.

Finally it is noted that a further advantage of the tree-structured GP is that it may perform better in extrapolation. Since the tree-structured GP will revert to its *local* prior distribution outside of the range of training data, it is likely to produce better predictive estimates in extrapolation than a single GP, which will revert to the *global* prior (which may often be markedly different to local trends).

Chapter 10

Conclusions and Further Work

This thesis has focused on the problem of propagating uncertainty through large nonlinear models. It has been stated that the main problem with uncertainty analysis is that the computational expense is often prohibitive. This has given rise to a number of emulator-based approaches, where a much simpler and faster emulator is used to describe the response of the model in the presence of uncertainty in the inputs. Such emulators may also readily be used for sensitivity analysis and other applications in computer science. A great number of flexible emulators exist, but many are untested outside of mathematical journals and are unknown to engineering. One aim of this thesis has been to introduce new techniques to structural dynamics and engineering.

Two particular emulators have been described in some detail and used on several case studies. The first is a Gaussian process (GP), which is a very flexible probabilistic emulator, capable of emulating very nonlinear models with sparse training data. The second emulator is an extension, or generalisation, of the first, which uses a decision tree to split the input space into a number of regions, each of which is assigned a separate GP. This second emulator was investigated as a possible way of performing uncertainty analysis on bifurcating data, motivated by observations on a model of the aortic valve.

In concurrence with the thesis objectives presented in Section 1.4, conclusions are presented here: first from the perspective of the uncertainty analysis techniques used; and second, regarding findings specific to the case studies investigated.

10.1 Emulator-based Uncertainty Analysis

Emulators have been shown to be a powerful method of performing uncertainty analysis and sensitivity analysis for a reasonable computational cost. Some specific conclusions are:

- GPs have been shown to be an excellent way of emulating nonlinear computer models, since they are efficient, analytically tractable, and give an account of predictive uncertainty by variance estimates at unknown points. The GP can give analytical estimates of all the widely-used global sensitivity measures. For the majority of computer models therefore, GPs are a very good choice when compared to other emulators in the literature.

- A limitation of GPs is that they necessarily assume a smooth response. In the case of bifurcating data the GP has been shown to be insufficient, since it uses constant roughness parameters across the full range of input space, which results in a poor fit because the data is very "rough" at the bifurcation, whereas it is much smoother away from it. The GP cannot make this distinction.

- The tree-structured GP has been shown to be capable of estimating all the uncertainty and sensitivity measures that are estimated by the single GP, whilst offering several additional advantages, which are listed here:

  1. Bifurcations can be modelled by positioning divisions of input space at the point of bifurcation. The tree-structured GP has been shown to correctly identify points of bifurcation on two case studies, although it is noted that a sufficient density of training data is necessary to invoke a split.
  2. Different hyperparameters can be used for each region. This allows the modelling of data with a non-constant variance, as well as assigning smooth (or even linear) trends to a certain region, while other regions are allowed to vary rapidly. A further impact is that prior means are assigned that better reflect local trends, so small extrapolations are likely to be more accurate.
  3. Since the tree-structured GP inverts a number of smaller covariance matrices rather than a single large one (in the case of a single GP), the computational expense of training is actually reduced, although there is some trade-off due to the necessity of numerical integration to estimate hyperparameters.

- One main limitation of the tree-structured GP is that it only allows axis-aligned data splits. When a bifurcation exists that is not parallel to an axis, the GP has to use an overly-complicated tree to describe the data. The problem is compounded in the case of a curved bifurcation. Tree-structured GPs are therefore inherently unsuitable for modelling these types of bifurcations.

- Both emulators can however identify their own uncertainty about the data fit through estimations of variance. Areas of high predictive uncertainty can therefore be identified and more data added to these regions if necessary. Cross validation can also be used to measure the fit of the emulator.

10.2 Practical Observations

Even with the use of sophisticated emulators such as those mentioned, the main limitation on propagating uncertainty is computational expense. There are also a number of practical difficulties that arise. The following are observations and suggestions from the author, based on experience from practical implementation of uncertainty analysis:

1. The first difficulty likely to be encountered is the quantification of uncertainty. Quite often there is simply not enough data to reliably characterise probability distributions. The assertion of many mathematicians that probability distributions can be elicited from "expert knowledge" is often, in the experience of the author, somewhat optimistic. When determining hyperelastic coefficients or Prony series parameters, information about distributions is usually scarce and assumptions must be made. This could be tackled by investigating ways of propagating more vague uncertainty theories, such as evidence theory or possibility theory, through large systems. A formal investigation into the uncertainty induced by selection of probability distributions could be useful, or the use of hierarchical models for modelling parameter uncertainty.

2. A great deal of computational expense can be saved by simplifying the model itself. This may seem obvious, but is often overlooked. As an example, an early version of the "wet" heart valve model in Chapter 4 required over 24 hours of simulation time. After including assumptions of symmetry, cyclic symmetry, shaping the fluid mesh to only include elements where absolutely necessary, and running the model only over a necessary 20ms, the simulation time was reduced to around 1 hour. This should always be the the first port of call when looking to reduce computational expense in uncertainty analysis.

3. A practical difficulty is that building a complex model that is stable over the full range of parameter space can be quite difficult. While a nonlinear model may be easy to solve at nominal parameter values, it can fail to solve in extremes of parameter space, or (perhaps worse) return unrealistic results, which can severely taint measures of uncertainty and sensitivity. Improving the stability of models has in fact been one of the most time-consuming parts of the work here for the author, and should be accounted for.

4. In order to lower computational expense, the number of input variables should be kept to a minimum. Screening can help to identify parameters that are not influential — even a simple global sensitivity analysis could be a worthwhile investment before running a much more thorough analysis on the few important parameters.

5. A further difficulty is that, considering that all the emulators here only consider

univariate outputs, it can often be difficult to find a meaningful univariate output to a complex model. This problem was illustrated in both the heart valve and airship models, where oscillations in model outputs made the determination of a single value difficult. Section 10.4 suggests some possible further work based on this.

6. Sensitivity analysis can be used to identify errors in a model. Experience has shown that when variation in a model's output is not explained to much extent by main effects of parameters, or low-order interactions, this can suggest that the model is not functioning correctly. If high interaction variances are noted, it is worth manually checking that the model is behaving as expected before drawing any conclusions.

10.3 Heart Valves and Airships

Uncertainty and sensitivity analysis on the heart valve and airship models suggests several conclusions:

- The uncertainty in biomechanical models is very substantial, but the awareness of uncertainty analysis in biomechanical modelling is quite low. Through the heart valve model it was extrapolated that biomechanical models in general will benefit greatly from a formal quantification of uncertainty.
- In both models, sensitivity analysis was shown to be a powerful tool for investigating the interactions and mechanisms at work in a large FE model. Trends can noticed that might not be evident from a single run at nominal parameter values. Some instances of these were:
  - The stiffness of the sinus and aorta is instrumental in determining stress and displacement in the heart valve. In particular, it is noted that an optimum stiffness of aorta is desirable, that prevents buckling but does not transfer tensile load to the leaflet.
  - Although the material properties of the airship frame were not found to vary significantly with temperature, the stress in the frame did vary substantially with temperature, as a result of loads transferred to the frame by the temperature-dependent balloons.

10.4 Further Work

The work here has several obvious future directions. From the emulation point of view, the tree-structured GP could potentially be improved by using a data division method that allows more general types of splits that are a function of two variables or more. Voronoi tessellations have been used to this end [135] — this method could potentially be adapted for use in uncertainty analysis and sensitivity analysis using a numerical integration approach. In the case where Voronoi tessellations may be too complex to describe the data, it is even conceivable to use the RJ-MCMC algorithm to explore between tree-structured GPs and tessellated GPs, finally selecting the most probable one. Another approach might be to use Bayesian mixture models of multiple GPs.

Another issue that could be addressed is to implement adaptive sampling into the process of uncertainty analysis. The choice of the number and location of training data required to accurately emulate a model is usually far from obvious. It would be extremely useful to sample the model on a step-by-step basis, where each new data point is chosen to optimise some criteria, such as a reduction in predictive variance. It would not be difficult to program such an algorithm to alter input decks for FE software, so that the program can be left to run successive simulations until it has built an emulator to a sufficient accuracy. In the case of the tree-structured GP this would allow a clustering of data in the neighbourhood of bifurcations, ensuring that bifurcations are captured as accurately as possible.

A concern that has not been fully addressed is that of model validation, since it was decided that this aspect was outside the remit of this thesis. When the output of a model is uncertain, it is hard to validate it against test data. Possible ways of doing this could include comparison of output moments, such as means and variances. A more thorough way might even be to compare sensitivities to certain model parameters, if sufficient validation data are available.

Finally, since all the models investigated here have been time-dependent, and the emulators consider only univariate outputs, it has been noted that it can be very difficult to find a single value for each model run that meaningfully describes a time-varying system. Some very recent work has been performed to address this issue by Conti *et al.* using an iterative approach [136, 137], as well as multivariate emulation by Urban and Fricker [138] — it might be fruitful to investigate this work in the context of engineering.

References

[1] Elsevier B.V. Scopus. http://www.scopus.com, 2010.

[2] M. Anderson. *Presentation at Workshop on Model Validation and Uncertainty Quantification at the 19th International Modal Analysis Conference.* Uncertainty Quantification at Los Alamos National Laboratories, Orlando, Florida, 2002.

[3] G. J. Klir and R. M. Smith. On measuring uncertainty and uncertainty-based information: Recent developments. *Annals of Mathematics and Artificial Intelligence*, 32(1-4):5–33, 2001.

[4] L.A. Zadeh. Probability measures of fuzzy events. *Journal of Mathematical Analysis and Applications*, 23(2):421–427, 1968.

[5] Z. Pawlak. Rough set approach to knowledge-based decision support. *European journal of operational research*, 99(1):48–57, 1997.

[6] D. Dubois and H. Prade. Rough Fuzzy Sets and Fuzzy Rough Sets. *International Journal of general systems*, 17(2):191–209, 1990.

[7] Z. Pawlak. *Rough sets: Theoretical Aspects of Reasoning about Data.* Springer, 1991.

[8] G.J. Klir. *Uncertainty and Information: Foundations of Generalized Information Theory.* Wiley-IEEE Press, 2006.

[9] R.E. Moore. Interval analysis. *Englewood Cliffs, New Jersey*, 1966.

[10] Andrew Gelman. *Bayesian Data Analysis.* Chapman & Hall/CRC, London, 2nd ed. edition, 2004.

[11] Anthony O'Hagan. *Uncertain Judgements : Eliciting Experts' Probabilities.* Wiley, Chichester, 2006.

[12] LA Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 1(1):3–28, 1978.

[13] D. Dubois and H. Prade. *Possibility Theory: An Approach to Computerized Processing of Uncertainty.* Plenum Press, New York, 1988.

[14] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

[15] G.J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall PTR Upper Saddle River, NJ, USA, 1995.

[16] D. Dubois and H. Prade. Fuzzy sets and probability: misunderstandings, bridges and gaps. In *Fuzzy Systems, 1993., Second IEEE International Conference on*, pages 1059–1068. IEEE, 2002.

[17] L. Chen and SS Rao. Fuzzy finite-element approach for the vibration analysis of imprecisely-defined systems. *Finite Elements in Analysis and Design*, 27(1):69–83, 1997.

[18] CK Mechefske. Objective machinery fault diagnosis using fuzzy logic. *Mechanical Systems and Signal Processing*, 12(6):855–862, 1998.

[19] AP Dempster. A Generalization of Bayesian Inference. *Journal of the Royal Statistical Society. Series B (Methodological)*, 30(2):205–247, 1968.

[20] G. Shafer. *A Mathematical Theory of Evidence*. Princeton university press Princeton, NJ, 1976.

[21] K. Worden, G. Manson, and T. Denoeux. An evidence-based approach to damage location on an aircraft structure. *Mechanical Systems and Signal Processing*, 23(6):1792–1804, 2009.

[22] O. Basir and X. Yuan. Engine fault diagnosis based on multi-sensor information fusion using Dempster-Shafer evidence theory. *Information Fusion*, 8(4):379–386, 2007.

[23] B.S. Yang and K.J. Kim. Application of Dempster-Shafer theory in fault diagnosis of induction motors using vibration and current signals. *Mechanical Systems and Signal Processing*, 20(2):403–420, 2006.

[24] L.A. Zadeh. Toward a generalized theory of uncertainty (GTU)–an outline. *Information Sciences*, 172(1-2):1–40, 2005.

[25] J. Yen. Generalizing the Dempster-Shafer theory to fuzzy sets. *Classic Works of the Dempster-Shafer Theory of Belief Functions*, pages 529–554, 2008.

[26] T.J. Ross. Engineering decisions involving aleatoric and epistemic uncertainty. In *Second International Forum on Engineering Decision Making*, Lake Louise, Canada, 2006.

[27] D. Dubois. Possibility theory and statistical reasoning. *Computational Statistics & Data Analysis*, 51(1):47–69, 2006.

[28] G. Manson. Calculating frequency response functions for uncertain systems using complex affine analysis. *Journal of Sound and Vibration*, 288(3):487–521, 2005.

[29] L. Jaulin. *Applied Interval Analysis.* Springer Verlag, 2001.

[30] D. Moens and D. Vandepitte. A fuzzy finite element procedure for the calculation of uncertain frequency-response functions of damped structures: Part 1–Procedure. *Journal of sound and vibration*, 288(3):431–462, 2005.

[31] B. Lallemand, A. Cherki, T. Tison, and P. Level. Fuzzy modal finite element analysis of structures with imprecise material properties. *Journal of Sound and Vibration*, 220(2):353–365, 1999.

[32] Vilfredo Pareto. *Manual of Political Economy.* A.M. Kelley, New York, 1971.

[33] A. Saltelli, K. Chan, and E. M. Scott. *Sensitivity Analysis.* Wiley series in probability and statistics. Wiley, New York, 2000.

[34] C. Daniel. On varying one factor at a time. *Biometrics*, 14:430–431, 1958.

[35] G.E.P. Box, W.G. Hunter, and J.S. Hunter. *Statistics For Experimenters: An Introduction to Design, Data Analysis, and Model Building.* John Wiley & Sons New York, 1978.

[36] D. Miller and M. Frenklach. Sensitivity analysis and parameter estimation in dynamic modeling of chemical kinetics. *International Journal of Chemical Kinetics*, 15(7):677–696, 1983.

[37] J.T. Hwang. Sensitivity analysis in chemical kinetics by the method of polynomial approximations. *International Journal of Chemical Kinetics*, 15(10):959–987, 1983.

[38] A. Saltelli, M. Ratto, T. Andres, and Ebooks Corporation. *Global Sensitivity Analysis: The Primer.* Wiley Online Library, 2008.

[39] W.J. Conover. *Practical Nonparametric Statistics.* John Wiley & Sons New York, 1980.

[40] I.M. Sobol. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55(1-3):271–280, 2001.

[41] J. E. Oakley and A O'Hagan. Probabilistic sensitivity analysis of complex models: a Bayesian approach. *Journal Of The Royal Statistical Society Series B*, 66(3):751–769, 2004.

[42] I.M. Sobol'. Sensitivity estimates for nonlinear mathematical models. *Mathematical Modelling and Computational Experiments*, 1:407–414, 1993.

[43] T. Homma and A. Saltelli. Importance measures in global sensitivity analysis of model output. *Reliability Engineering and System Safety*, 52(1):1–17, 1996.

[44] A. Shreider, Y. *Method of Statistical Testing : Monte Carlo Method.* Elsevier, Amsterdam ; London, 1964.

[45] M.D. McKay, RJ Beckman, and WJ Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.

[46] G. Stefanou. The stochastic finite element method: past, present and future. *Computer Methods in Applied Mechanics and Engineering*, 198(9-12):1031–1051, 2009.

[47] G. Stefanou and M. Papadrakakis. Assessment of spectral representation and Karhunen-Loève expansion methods for the simulation of Gaussian stochastic fields. *Computer Methods in Applied Mechanics and Engineering*, 196(21-24):2465–2477, 2007.

[48] R.G. Ghanem and P.D. Spanos. *Stochastic Finite Elements: a Spectral Approach.* Dover Pubns, 2003.

[49] N. Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938.

[50] Szegö, G. *Orthogonal Polynomials.* American Mathematical Society, 1939.

[51] B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93(7):964–979, 2008.

[52] RV Field and M. Grigoriu. On the accuracy of the polynomial chaos approximation. *Probabilistic Engineering Mechanics*, 19(1):65–80, 2004.

[53] S. Acharjee and N. Zabaras. Uncertainty propagation in finite deformations–A spectral stochastic Lagrangian approach. *Computer Methods in Applied Mechanics and Engineering*, 195(19-22):2289–2312, 2006.

[54] C. Soize. A nonparametric model of random uncertainties for reduced matrix models in structural dynamics. *Probabilistic Engineering Mechanics*, 15(3):277–294, 2000.

[55] C. Soize. A comprehensive overview of a non-parametric probabilistic approach of model uncertainties for predictive models in structural dynamics. *Journal of Sound and Vibration*, 288(3):623–652, 2005.

[56] S. Adhikari and M. I. Friswell. Random matrix eigenvalue problems in structural dynamics. *International Journal for Numerical Methods in Engineering*, 69(3):562–591, 2007.

[57] S. Adhikari and R. Chowdhury. A reduced-order non-intrusive approach for stochastic structural dynamics. *Computers and Structures*, 88(21-22):1230–1238, 2010.

[58] C.M. Bishop. *Pattern Recognition and Machine Learning.* Springer, New York, 2006.

[59] R.H. Myers, D.C. Montgomery, and C.M. Anderson-Cook. *Response Surface Methodology: Process and Product Optimization using Designed Experiments.* John Wiley & Sons Inc, 2009.

[60] B. Yeten, A. Castellini, B. Guyaguler, and WH Chen. A comparison study on experimental design and response surface methodologies. In *SPE Reservoir Simulation Symposium*, 2005.

[61] M.F. Hussain, R.R. Barton, and S.B. Joshi. Metamodeling: Radial basis functions, versus polynomials. *European Journal of Operational Research*, 138(1):142–154, 2002.

[62] C.M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, USA, 1995.

[63] ME Ricotti and E. Zio. Neural network approach to sensitivity and uncertainty analysis. *Reliability Engineering & System Safety*, 64(1):59–71, 1999.

[64] M. Marseguerra, R. Masini, E. Zio, and G. Cojazzi. Variance decomposition-based sensitivity analysis via neural networks. *Reliability Engineering & System Safety*, 79(2):229–238, 2003.

[65] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, 1989.

[66] M. C. Kennedy, C. W. Anderson, S. Conti, and A. O'Hagan. Case studies in Gaussian process modelling of computer codes. *Reliability Engineering and System Safety*, 91(10-11):1301–1309, 2006.

[67] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning.* The MIT Press, Cambridge, Massachusetts, 2006.

[68] G.J. Kerns. *Introduction to Probability and Statistics Using R.* Kerns, G.J., 2010.

[69] W.M. Bolstad. *Introduction to Bayesian statistics.* Wiley-IEEE, 2004.

[70] R Haylock. *Bayesian Inference about Outputs of Computationally Expensive Algorithms with Uncertainty on the Inputs.* PhD thesis, University of Nottingham, 1997.

[71] Marc Kennedy. Gem-SA. `http://www.tonyohagan.co.uk/academic/GEM/index.html`, 2009.

[72] Philippe Pibarot and Jean G. Dumesnil. Prosthetic heart valves: Selection of the optimal prosthesis and long-term management. *Circulation*, 119(7):1034–1048, 2009.

[73] K. J. Grande, R. P. Cochran, P. G. Reinhall, and K. S. Kunzelma. Stress variations in the human aortic root and valve: The role of anatomic asymmetry. *Annals of Biomedical Engineering*, 26(4):534–545, 1998.

[74] S. Silberman, A. Oren, M. Dotan, O. Merin, D. Fink, M. Deeb, and D. Bitran. Aortic valve replacement: Choice between mechanical valves and bioprostheses. *Journal of Cardiac Surgery*, 23(4):299–306, 2008.

[75] Mano Thubrikar. *The Aortic Valve.* CRC Press, Boca Raton, Florida, 1990.

[76] Alaster. Yoxall. *Design Philosophy for Aortic Valve Prostheses.* PhD thesis, University of Sheffield, 1997.

[77] K. L. Billiar and M. S. Sacks. Biaxial mechanical properties of the natural and glutaraldehyde treated aortic valve cusp - part i: Experimental results. *Journal of Biomechanical Engineering*, 122(1):23–30, 2000.

[78] M. S. Sacks and A. P. Yoganathan. Heart valve function: A biomechanical perspective. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1484):1369–1391, 2007.

[79] A. Ranga, R. Mongrain, R. Mendes Galaz, Y. Biadillah, and R. Cartier. Large-displacement 3D structural analysis of an aortic valve model with nonlinear material properties. *Journal of Medical Engineering and Technology*, 28(3):95–103, 2004.

[80] J. A. Stella, J. Liao, and M. S. Sacks. Time-dependent biaxial mechanical behavior of the aortic heart valve leaflet. *Journal of Biomechanics*, 40(14):3169–3177, 2007.

[81] T. C. Doehring, E. O. Carew, and I. Vesely. The effect of strain rate on the viscoelastic response of aortic valve tissue: A direct-fit approach. *Annals of Biomedical Engineering*, 32(2):223–232, 2004.

[82] M.S. Hamid, H.N. Sabbah, and P.D. Stein. Vibrational analysis of bioprosthetic heart valve leaflets using numerical models: effects of leaflet stiffening, calcification, and perforation. *Circulation Research*, 61(5):687, 1987.

[83] I. C. Howard, E. A. Patterson, and A. Yoxall. On the opening mechanism of the aortic valve: Some observations from simulations. *Journal of Medical Engineering and Technology*, 27(6):259–266, 2003.

[84] C. J. Carmody, G. Burriesci, I. C. Howard, and E. A. Patterson. An approach to the simulation of fluid-structure interaction in the aortic valve. *Journal of Biomechanics*, 39(1):158–169, 2006.

[85] R. Haj-Ali, L. P. Dasi, H. S. Kim, J. Choi, H. W. Leo, and A. P. Yoganathan. Structural simulations of prosthetic tri-leaflet aortic heart valves. *Journal of Biomechanics*, 41(7):1510–1519, 2008.

[86] E. J. Weinberg and M. R. K. Mofrad. Three-dimensional, multiscale simulations of the human aortic valve. *Cardiovascular Engineering*, 7(4):140–155, 2007.

[87] R. Gnyaneshwar, R. K. Kumar, and K. R. Balakrishnan. Dynamic analysis of the aortic valve using a finite element model. *Annals of Thoracic Surgery*, 73(4):1122–1129, 2002.

[88] J. De Hart, G. W. M. Peters, P. J. G. Schreurs, and F. P. T. Baaijens. A two-dimensional fluid-structure interaction model of the aortic value. *Journal of Biomechanics*, 33(9):1079–1088, 2000.

[89] J. De Hart, G. W. M. Peters, P. J. G. Schreurs, and F. P. T. Baaijens. A three-dimensional computational analysis of fluid-structure interaction in the aortic valve. *Journal of Biomechanics*, 36(1):103–112, 2003.

[90] Y. S. Morsi, W. W. Yang, C. S. Wong, and S. Das. Transient fluid-structure coupling for simulation of a trileaflet heart valve using weak coupling. *Journal of Artificial Organs*, 10(2):96–103, 2007.

[91] Hugh Coni and Nicholas Coni. *Blood Pressure: All You Need to Know.* Royal Society of Medicine, London, 2002.

[92] T. Belytschko and J.I. Lin. Explicit algorithms for the nonlinear dynamics of shells. *Computer methods in applied mechanics and engineering*, 42(2):225–251, 1984.

[93] J. Li, X. Y. Luo, and Z. B. Kuang. A nonlinear anisotropic model for porcine aortic heart valves. *Journal of Biomechanics*, 34(10):1279–1289, 2001.

[94] J.O. Hallquist. LS-DYNA theory manual. *Livermore Software Technology Corporation*, 3, 2006.

[95] J. A. Weiss, B. N. Maker, and S. Govindjee. Finite element implementation of incompressible, transversely isotropic hyperelasticity. *Computer Methods in Applied Mechanics and Engineering*, 135(1-2):107–128, 1996.

[96] Allan F. Bower. *Applied Mechanics of Solids.* Taylor and Francis, 2009.

[97] M. Mooney. A theory of large elastic deformation. *Journal of Applied Physics*, 11:582, 1940.

[98] K. M. Quapp and J. A. Weiss. Material characterization of human medial collateral ligament. *Journal of Biomechanical Engineering*, 120(6):757–763, 1998.

[99] Z. Li, J. Yu, J. Li, and J. Chen. Stress field analysis of aortic valve under physiological condition. *Chinese Journal of Biomedical Engineering*, 18(2):187–193, 1999.

[100] DC Drucker. *On the postulate of stability of material in the mechanics of continua.* Division of Engineering, Brown University, RI, 1963.

[101] O.J. Dunn and V.A. Clark. *Basic Statistics: A Primer for the Biomedical Sciences.* Wiley, 2009.

[102] W. M. Swanson and R. E. Clark. Dimensions and geometric relationships of the human aortic valve as a function of pressure. *Circulation Research*, 35(6):871–882, 1974.

[103] D.S. Bach, N.D. Kon, J.G. Dumesnil, C.F. Sintek, and D.B. Doty. Ten-year outcome after aortic valve replacement with the Freestyle stentless bioprosthesis. *The Annals of thoracic surgery*, 80(2):480–487, 2005.

[104] P.K. Chow, T.H. Ng, D. Heng, and P.O. Mack. A simple method of blood pressure measurement in the pig using a neonatal cuff. *Annals of the Academy of Medicine, Singapore*, 28(1):15, 1999.

[105] L. Liao and I. Pasternak. A review of airship structural research and development. *Progress in Aerospace Sciences*, 45(4-5):83–96, 2009.

[106] M. Battipede, M. Lando, P. A. Gili, and P. Vercesi. Peculiar performance of a new lighter-than-air platform for monitoring. In *Collection of Technical Papers - AIAA 4th Aviation Technology, Integration, and Operations Forum, ATIO*, volume 2, pages 749–762, 2004.

[107] M. Battipede, PA Gili, and M. Lando. Control Allocation System for an Innovative Remotely-Piloted Airship. *Proceedings of the AIAA Atmospheric Flight Mechanics Conference.*

[108] A. Cappadona, R. Lecca, M. Vazzola, P. Gili, P. Farina, and C. Surace. Innovative unmanned airship structural analysis: Dual-hull and exoskeletal configurations. In *Journal of Physics: Conference Series*, volume 181, page 012097. IOP Publishing, 2009.

[109] R Lecca. Analisi strutturale su dirigible non convenzionale a controllo remoto: Configurazione bifuso. Master's thesis, Facoltà di Ingegneria, Politecnico di Torino, Turin, Italy, 2008.

[110] A Cappadona. Analisi strutturale su dirigible non convenzionale a controllo remoto: Configurazione esoscheletro. Master's thesis, Facolt di Ingegneria, Politecnico di Torino, Torino, 2008.

[111] J.N. Reddy. *Mechanics of Laminated Composite Plates and Shells: Theory and Analysis.* CRC, 2004.

[112] US Department of Defense. *Polymer Matrix Composites Materials Properties*, volume 2. United States Department of Defense, 2002.

[113] N. Bessert and O. Frederich. Nonlinear airship aeroelasticity. *Journal of fluids and structures*, 21(8):731–742, 2005.

[114] J.M. Liu, C.J. Lu, and L.P. Xue. Investigation of airship aeroelasticity using fluid-structure interaction*. *Journal of Hydrodynamics*, 20(2):164–171, 2008.

[115] Y. Li, M. Nahon, and I. Sharf. Dynamics Modeling of Flexible Airships. *AIAA*, 2212:23–26, 2007.

[116] EASA. Certification Specifications for Normal, Utility, Aerobatic and Commuter Category Aeroplanes CS23, Book 1, 2009.

[117] J. Donea, A. Huerta, J.P. Ponthot, and A. Rodríguez-Ferran. *Arbitrary Lagrangian-Eulerian methods.* Wiley Online Library.

[118] Robert B. Gramacy and Herbert K. H. Lee. Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119–1130, 2008.

[119] Leo Breiman. *Classification and Regression Trees.* Wadsworth, Belmont, California, 1984.

[120] D.J.C. MacKay. *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003.

[121] T.H. Cormen. *Introduction to Algorithms.* The MIT Press, 2001.

[122] H. A. Chipman, E. I. George, and R. E. McCulloch. Bayesian CART model search. *Journal of the American Statistical Association*, 93(443):935–948, 1998.

[123] H.A. Chipman, E.I. George, and R.E. McCulloch. Bayesian treed models. *Machine Learning*, 48(1):299–320, 2002.

[124] Robert B. Gramacy. *Bayesian Treed Gaussian Process Models.* PhD thesis, University of California, 2005.

[125] B.A. Cipra. The best of the 20th century: Editors name top 10 algorithms. *SIAM News*, 33(4):1–2, 2000.

[126] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, et al. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087, 1953.

[127] W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97, 1970.

[128] Robert B. Gramacy. tgp: An R package for Bayesian nonstationary, semiparametric nonlinear regression and design by treed Gaussian process models. *Journal of Statistical Software*, 19(9), 2007.

[129] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[130] A.E. Gelfand and A.F.M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409, 1990.

[131] P.J. Green and D.I. Hastie. Reversible jump MCMC, 2009.

[132] F. Pappenberger, I. Iorgulescu, and K.J. Beven. Sensitivity analysis based on regional splits and regression trees (SARS-RT). *Environmental Modelling & Software*, 21(7):976–990, 2006.

[133] Robert B. Gramacy and Matthew Alan Taddy. Categorical inputs, sensitivity analysis, optimization and importance tempering with tgp version 2, an R package for treed Gaussian process models. *Journal of Statistical Software*, 33(6), 2010.

[134] K. Worden, G. Manson, T. M. Lord, and M. I. Friswell. Some observations on uncertainty propagation through a simple nonlinear system. *Journal of Sound and Vibration*, 288(3):601–621, 2005.

[135] H. M. Kim, B. K. Mallick, and C. C. Holmes. Analyzing nonstationary spatial data using piecewise Gaussian processes. *Journal of the American Statistical Association*, 100(470):653–668, 2005.

[136] S. Conti, J.P. Gosling, JE Oakley, and A. O'hagan. Gaussian process emulation of dynamic computer codes. *Biometrika*, 2009.

[137] S. Conti and A. O'Hagan. Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, 140(3):640–651, 2010.

[138] N.M. Urban and T.E. Fricker. A comparison of Latin hypercube and grid ensemble designs for the multivariate emulation of an Earth system model. *Computers & Geosciences*, 36(6):746–755, 2010.

[139] J.C. Lagarias, J.A. Reeds, M.H. Wright, and P.E. Wright. Convergence properties of the Nelder-Mead simplex algorithm in low dimensions. In *SIAM Journal of Optimization*, 1996.

Appendix A

Publications

K. Worden, W. Becker, M. Battipede, and C. Surace, Aspects of uncertainty analysis for large nonlinear computer models. *Applied Mechanics and Materials*, 24-25:24-41, 2010.

W. Becker, K. Worden, M. Battipede, and C. Surace, Uncertainty analysis of a dynamic model of a novel remotely-piloted airship. *Journal of Aircraft*, 2011 (to appear)

W. Becker, J. Rowson, J. Oakley, A.Yoxall, G.Manson, K.Worden, Bayesian sensitivity analysis of a model of the aortic valve. *Journal of Biomechanics*, 2011 (under review)

W. Becker, J. Rowson, J. Oakley, A.Yoxall, G.Manson, K.Worden, Bayesian sensitivity analysis of a nonlinear finite element model. *Mechanical Systems and Signal Processing*, 2011 (awaiting submission)

W. Becker and K. Worden. Bayesian sensitivity analysis of a bifurcating Duffing oscillator. in *Third International Conference on Uncertainty in Structural Dynamics*. 2010. Leuven, Belgium

K. Worden and W. Becker. On the identification of hysteretic systems: Part II: Bayesian sensitivity analysis. in *IMAC-XXVIII*. 2010. Jacksonville, Florida: Society for Experimental Mechanics Inc.

P. Sartor, R.K. Schmidt, W. Becker, K. Worden, D.A. Bond, and W.J. Staszewski. Conceptual design of a hard landing indication system using a flight parameter sensor simulation model. in *27th International Congress on the Aeronautical Sciences*. 2010. Nice, France.

P. Sartor, W. Becker, K. Worden, D.A. Bond, and W.J. Staszewski. Bayesian sensitivity analysis of flight parameters in the development of a hard landing indication system. in *The 5th European Workshop on SHM*. 2010, Sorrento, Italy.

W. Becker, P. Gili, C. Surace, and K. Worden. Bayesian uncertainty analysis of a nonlinear finite element model of an unmanned airship, in *2nd International Conference on Uncertainty in Structural Dynamics*. 2009: Sheffield, UK.

W. Becker, J. Rowson, J. Oakley, A.Yoxall, G.Manson, K.Worden. Bayesian sensitivity analysis of a large nonlinear model. in *ISMA2008*. 2008. Leuven, Belgium.

Appendix B

Practical Aspects of Bayesian Uncertainty Analysis

The following demonstrates the practical steps involved in calculating uncertainty and sensitivity quantities with the Bayesian method outlined in Chapter 3. The first "dry" aortic valve (AV) model will be used throughout as an example — see Sections 4.3.1, 5.1.1 and 5.2.2 for the model description, input uncertainties and results respectively.

Model Preparation and Training Data

1. The first task is to select the d input parameters to vary — these must be univariate and independence must be assumed. They are denoted here as random variables X_i,

$$X = \{X_1, X_2, ..., X_d\} \tag{B.1}$$

   In the heart valve model example, 8 univariate inputs were selected, comprising material, loading and geometric parameters (see Table 5.1). From a practical point of view, these need to be parameters that can be varied easily in the input file to the solver used. For example, LS-Dyna uses an ASCII input file, where the parameters chosen are specified as numerical values. Therefore it is simple to use a controlling program (LS-Opt in this case) to edit these parameter values for each new run.

2. Next, distributions must be assigned to the input parameters. Elicitation of distributions will not be dealt with here, but uniform distributions may be appropriate if only upper and lower limits are available on parameters. If more data is available, or the parameter is thought to be normally-distributed, means and variances may be specified. In the case of the AV model, normal distributions were assigned, so that means and standard deviations can be expressed as d-length vectors $\boldsymbol{\mu} = (\mu_1, ..., \mu_d)^T$ and $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_d)^T$, indexed by the dimension or variable index i. In the case of the

AV model, these vectors will be,

$$\begin{aligned}\boldsymbol{\mu} &= (2.45, 1.3, 2.05, 0.35, 1, 12.7, 31, 6)^T \\ \boldsymbol{\sigma} &= (0.8, 0.325, 0.65, 0.0085, 0.25, 0.675, 3, 2)^T\end{aligned}$$

For example, X_2 here refers to the variable Ts, with mean $\mu_2 = 1.3$ and standard deviation $\sigma_2 = 0.325$ (see again Table 5.1).

3. A sampling strategy must now be used to collect a set of n training data points (note that the inputs $\boldsymbol{X}$ together form a matrix because each input point is a d-dimensional vector):

$$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n) \tag{B.2}$$

The LHS design for the AV model is printed at the end of this appendix chapter — in this case 250 points were investigated, although one run was discarded as an non-convergent, therefore $n = 249$. The maximin Latin Hypercube approach has been found to work well, but is by no means the only choice — see Section 2.1.3. Note that it is convenient to consider each sample $\boldsymbol{x}_j$ as a point in 8-dimensional input space. Running these input parameter values through the model n times results in n model realisations. It is necessary to pick an output to consider - here the maximum displacement will be considered. This yields an n-length vector of training data outputs (maximum displacements in this case),

$$\mathbf{y} = (y_1, y_2, ..., y_n)^T \tag{B.3}$$

Again, the AV output vector is printed at the end of this appendix chapter. Note that the univariate output means that the GP here is being constructed to predict displacement only. Separate emulators must be built for other outputs.

4. Finally, the roughness matrix B in the covariance function (Equation 3.5) must be estimated from the data. The likelihood function of B is found in Equation 3.43, and is restated here as,

$$f(B|\mathbf{y}) \propto \hat{\sigma}^{\frac{q-n}{2}} |A|^{\frac{-1}{2}} |\Sigma_w|^{\frac{-1}{2}} \tag{B.4}$$

In order to estimate B it is necessary to maximise this function with respect to the diagonal elements $\{b_i\}_{i=1}^{d}$ of B, according to the principle of maximum likelihood estimation. Note that the terms here ultimately *only involve the training data and the roughness parameters* (see below for a fuller description). This means that this can essentially be regarded as a function of $\mathbf{b} = \text{tr}b$, i.e. $f((b))$ where $\mathbf{b} = (b_1, ...b_d)^T$. This function has a maximum, which can be found by numerically searching through the space of $\mathbf{b}$ using, for example, a simplex search algorithm - see [139]. Typically it is more convenient to maximise the logarithm of this equation due to the very small numbers that can be involved. In the case of the AV model the vector of roughness parameters was as follows (i.e. these are the values of B which produce the maximum value of B.4),

$$\mathrm{tr}(B) = (2.66, 0.187, 2.17, 0.276, 0.087, 1.010, 2.67, 2.99)^T \tag{B.5}$$

Calculating Posterior Means and Variances

The equations for the posterior mean and variance are restated here from Equations 3.47 and 3.51 respectively:

$$E^* \left\{ E(Y|\boldsymbol{x}_r) \right\} = \boldsymbol{\phi}_r(\boldsymbol{x}_r)^T \hat{\mathbf{w}} + \mathbf{t}_r(\boldsymbol{x}_r)^T \mathbf{e} \tag{B.6}$$

$$\begin{aligned} \mathrm{cov}^* \left\{ E(Y|\boldsymbol{x}_r), E(Y|\boldsymbol{x}'_p) \right\} &= \hat{\sigma}^2 \int_{\mathcal{X}_{-r}} \int_{\mathcal{X}_{-p}} c^{**}(\boldsymbol{x}, \boldsymbol{x}') dP(\boldsymbol{x}'_{-p}) dP(\boldsymbol{x}_{-r}) \\ &= \hat{\sigma}^2 \begin{bmatrix} U_{r,p}(\boldsymbol{x}_r, \boldsymbol{x}'_p) - \boldsymbol{t}_r(\boldsymbol{x}_r) A^{-1} \boldsymbol{t}_p(\boldsymbol{x}_p)^T + \\ \left\{ \boldsymbol{r}_r(\boldsymbol{x}_r) - \boldsymbol{t}_r(\boldsymbol{x}_r) A^{-1} \Phi \right\} \Sigma_w \left\{ \boldsymbol{r}_p(\boldsymbol{x}_p) - \boldsymbol{t}_p(\boldsymbol{x}_p) A^{-1} \Phi \right\}^T \end{bmatrix} \end{aligned} \tag{B.7}$$

Starting with Equation B.6, the posterior mean, note that apart from the two integrals, this equation consists of the estimate of the vector of weights $\hat{\mathbf{w}}$, and a further term $\mathbf{e}$. These are restated here, from Equations 3.18 and 3.50 respectively,

$$\hat{\mathbf{w}}^T = \mathbf{y}^T A^{-1} \Phi (\Phi^T A^{-1} \Phi)^{-1} \tag{B.8}$$

$$\mathbf{e} = A^{-1}(\mathbf{y} - \Phi \hat{\mathbf{w}}) \tag{B.9}$$

Of the terms in these equations, $\mathbf{y}^T$ is the vector of training data outputs, which is available. Φ is restated as,

$$\Phi^T = (\phi(\boldsymbol{x}_1), ..., \phi(\boldsymbol{x}_n))$$

where, in the case that $\phi(\boldsymbol{x})^T = (1, \boldsymbol{x})$, Φ can be constructed from $\boldsymbol{X}$, i.e.

$$\Phi = \begin{pmatrix} 1 & x_{11} & x_{21} & \dots & x_{d1} \\ 1 & x_{12} & x_{22} & \dots & x_{d2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1n} & x_{2n} & \dots & x_{dn} \end{pmatrix} \tag{B.10}$$

where x_{ij} refers to the ith dimension (equivalently the ith uncertain input) of the jth training point. The specific Φ of the AV example is not provided here, since it is very large. The only remaining term is A, which is a matrix formed by calculating the covariances between all permutations of input points using the covariance function from Equation 3.5, giving the matrix,

$$A = \begin{pmatrix} 1 & c(\boldsymbol{x}_1, \boldsymbol{x}_2) & \dots & c(\boldsymbol{x}_1, \boldsymbol{x}_n) \\ c(\boldsymbol{x}_2, \boldsymbol{x}_1) & 1 & & \vdots \\ \vdots & & \ddots & \\ c(\boldsymbol{x}_n, \boldsymbol{x}_1) & \cdots & & 1 \end{pmatrix}$$

In the case of the AV model this is a 249 × 249 matrix, therefore it is not explicitly presented here. Finally, the integrals ϕ_r and $\mathbf{t}_r$ result in vectors which are derived in Section 3.2.1 (Equations 3.56 and 3.58 respectively). ϕ_r is a $(d+1)$-length (i.e. of length 9 for the example here) column vector which is dependent on $\boldsymbol{x}_r$. For example, to calculate the posterior expected value of Y, r is set to the empty set, leaving,

$$\{\phi_r(\boldsymbol{x}_r) : r = \varnothing\} = (1, \mu_1, \mu_2, ..., \mu_d)^T \tag{B.11}$$

where the μ_i are the means corresponding to the distributions of the uncertain inputs X_i. The result for $\mathbf{t}_r$ will not be restated here, since it is more complex, but a glance at Equation 3.58 reveals that it is only dependent on the means and variances of the X_i, as well as the roughness parameters b_i and the training input matrix $\boldsymbol{X}$. Since $\mathbf{t}_r$ is of length n, in the case of the AV model this resulted in a 249-length column vector. The posterior expected value may now be calculated from Equation B.6 using the terms stated here.

Considering now the variance, note that Equation B.7 has essentially all the same terms, with the addition of the integral U. Since to get the posterior variance we again set $r = p = \varnothing$, none of these terms are a function of $\boldsymbol{x}$; all of the terms in this expression have been explicitly stated here, with the exception of U. The result is however given in Equation 3.104, and it can be seen that it is only a function of roughness parameters and parameter variances, resulting in a single scalar quantity.

Calculating Sensitivities

In order to plot main effects, Equation B.6 is again used, but setting $\boldsymbol{x}_r = x_i$, where x_i is the input variable of interest. Note that this changes ϕ_r and $\mathbf{t}_r$ slightly — now, they are functions of x_i. Therefore, in order to plot the main effect of x_i, $E^*\{E(Y|\boldsymbol{x}_i)\}$ is calculated at a range of values over the support of x_i and plotted. The posterior pointwise variance is similarly calculated by setting $\boldsymbol{x}_r = \boldsymbol{x}_p = x_i$ in Equation B.7.

For the sensitivity indices S_r, the calculations are somewhat more complex (see Section 3.2.2). The estimates are calculated by taking Equation 3.63 and dividing by the posterior variance of Y (already presented above as a special case of Equation B.7),

$$E^*(S_r) = \frac{E^*[E_r\{E_{-r}(Y|\boldsymbol{x}_r)^2\}] - E^*(E_r(Y)^2)}{E^*\{\text{var}(Y)\}} \tag{B.12}$$

To consider the sensitivity index of a single input x_i, simply set $r = i$ as usual. Note that since the estimate of S_i is based on the GP, which has a probabilistic output, it is itself a random variable. The quantity presented here is the expected value of S_i. Calculating the variance of S_i is not possible without numerical integration. The terms in the numerator of Equation B.12 (the denominator has been discussed) are presented in Equations 3.64 and 3.67. Although they may seem a little daunting, they are simply complex combinations of the terms already presented here, as well as the matrices P, Q and S, the results of which are presented in Section 3.2.3 (these matrices are of sizes $n \times n$, $(d+1) \times (d+1)$ and $(d+1) \times n$ respectively). It is not proposed to construct the matrices here, since the process is lengthy and would take an enormous amount of space, but it should be noted that *all of the terms here ultimately only depend on the training data, and the means and variances specified for the inputs.* This is one of the strengths of the GP approach - although the construction of these terms is time-consuming, they can be reused for any model.

Input matrix X^T

The input matrix is presented here in order to illustrate the process. This is a 249 × 8 matrix, where one sample point has been removed as an outlier. Column headings represent the inputs $\{X_i\}_{i=1}^8$.

Es	Ts	El	Tl	Tc	Rb	Phi	Fltrad
2.2168675e+006	1.2860000e-003	1.6036145e+006	2.2200000e-004	1.1310000e-003	1.2688000e-002	2.5289157e+001	2.2570280e+000
9.8795181e+005	1.8080000e-003	3.9216867e+006	3.4500000e-004	6.6500000e-004	1.3349000e-002	2.8566265e+001	7.1726910e+000
4.4578313e+006	1.3220000e-003	2.0578313e+006	3.7300000e-004	6.6900000e-004	1.3967000e-002	2.9722892e+001	3.5100400e+000
4.5301205e+006	1.4730000e-003	3.6397590e+006	2.3900000e-004	1.3110000e-003	1.2786000e-002	2.9000000e+001	9.5823290e+000
1.9277108e+006	1.1500000e-003	2.8566265e+006	1.0000000e-004	1.4120000e-003	1.1430000e-002	3.0590361e+001	3.7349400e+000
3.4457831e+006	1.9070000e-003	2.6060241e+006	2.5500000e-004	1.1590000e-003	1.3230000e-002	3.2662651e+001	7.9437750e+000
1.7108434e+006	1.5260000e-003	1.0000000e+005	1.1400000e-004	1.1910000e-003	1.1419000e-002	3.5746988e+001	4.7951810e+000
2.4337349e+006	1.7240000e-003	1.3843373e+006	1.4200000e-004	7.3700000e-004	1.3436000e-002	2.8759036e+001	7.8152610e+000
1.6024096e+006	1.0980000e-003	3.9530120e+006	2.8300000e-004	1.2230000e-003	1.1723000e-002	3.5265060e+001	7.2690760e+000
2.5963855e+006	2.0950000e-003	2.4180723e+006	1.6200000e-004	7.7700000e-004	1.1441000e-002	3.2325301e+001	9.9036140e+000
9.3373494e+005	1.3690000e-003	2.8253012e+006	4.1900000e-004	1.2950000e-003	1.2590000e-002	3.5650602e+001	7.7831330e+000
4.2048193e+006	9.8300000e-004	1.8542169e+006	3.4700000e-004	1.2070000e-003	1.3501000e-002	2.7795181e+001	3.6064260e+000
4.4216867e+006	8.3100000e-004	1.3216867e+006	2.3300000e-004	1.4560000e-003	1.1452000e-002	3.3240964e+001	7.0441770e+000
6.8072289e+005	1.8020000e-003	8.3614458e+005	5.0800000e-004	1.3880000e-003	1.2113000e-002	3.4397590e+001	6.0160640e+000
3.3554217e+006	1.0240000e-003	3.2012048e+006	4.3100000e-004	7.0500000e-004	1.2764000e-002	3.3819277e+001	2.1606430e+000
4.3493976e+006	1.2020000e-003	3.3108434e+006	3.9300000e-004	1.0260000e-003	1.3404000e-002	3.0831325e+001	7.5582330e+000
1.8734940e+006	1.8910000e-003	6.1686747e+005	5.3000000e-004	1.3190000e-003	1.3696000e-002	3.0253012e+001	2.2891570e+000
3.9698795e+006	8.2600000e-004	4.4457831e+005	2.3700000e-004	6.5700000e-004	1.3642000e-002	2.9819277e+001	9.4538150e+000
3.0481928e+006	1.0820000e-003	2.5277108e+006	2.9700000e-004	1.1180000e-003	1.3447000e-002	3.5120482e+001	2.9317270e+000
4.8373494e+006	1.2490000e-003	1.1650602e+006	2.6500000e-004	9.5400000e-004	1.2612000e-002	2.6783133e+001	6.7871490e+000
2.2710843e+006	1.5880000e-003	3.0915663e+006	5.7400000e-004	5.4000000e-004	1.2070000e-002	2.5674699e+001	2.1285140e+000
4.8012048e+006	2.0790000e-003	9.3012048e+005	3.6700000e-004	1.0700000e-003	1.3295000e-002	2.7265060e+001	5.7269080e+000
4.7469880e+006	1.3850000e-003	1.3686747e+006	3.3900000e-004	6.0400000e-004	1.2363000e-002	3.5554217e+001	2.3534140e+000
4.2228916e+006	2.0530000e-003	1.8385542e+006	3.3700000e-004	8.1300000e-004	1.3816000e-002	3.1168675e+001	7.8795180e+000
4.4397590e+006	2.0580000e-003	2.1518072e+006	5.1200000e-004	1.3270000e-003	1.3425000e-002	3.0108434e+001	4.0562250e+000
5.0000000e+006	1.5470000e-003	1.8072289e+006	1.6400000e-004	1.1950000e-003	1.3089000e-002	2.8180723e+001	5.4056220e+000
1.8915663e+006	1.2750000e-003	3.3891566e+006	5.6600000e-004	1.3230000e-003	1.3881000e-002	3.4783133e+001	2.6746990e+000
4.4939759e+006	1.0400000e-003	3.9759036e+005	2.2900000e-004	1.2110000e-003	1.2027000e-002	2.8807229e+001	3.7670680e+000
4.0963855e+006	1.9120000e-003	3.1385542e+006	2.1000000e-004	8.5700000e-004	1.2135000e-002	3.1795181e+001	9.6787150e+000
1.0060241e+006	1.4210000e-003	2.7313253e+006	5.6200000e-004	5.8000000e-004	1.2601000e-002	3.1361446e+001	6.4979920e+000
2.7951807e+006	1.0300000e-003	3.6710843e+006	5.8600000e-004	1.0060000e-003	1.2308000e-002	3.3289157e+001	9.7751000e+000
1.2048193e+006	1.1340000e-003	3.2481928e+006	2.7300000e-004	9.0200000e-004	1.3371000e-002	3.3915663e+001	9.0682730e+000

```
3.8975904e+006  1.9850000e-003  1.9638554e+006  4.4100000e-004  9.7000000e-004  1.3718000e-002  2.7072289e+001  3.4779120e+000
4.3313253e+006  9.9800000e-004  3.5771084e+006  5.6800000e-004  1.2030000e-003  1.2872000e-002  3.1554217e+001  3.6706830e+000
8.7951807e+005  1.2590000e-003  3.2638554e+006  4.1700000e-004  8.0500000e-004  1.1517000e-002  3.2951807e+001  6.5943780e+000
3.7891566e+006  1.4530000e-003  2.6216867e+006  5.7000000e-004  8.5300000e-004  1.2959000e-002  3.6566265e+001  4.1847390e+000
4.5662651e+006  1.5360000e-003  2.3240964e+006  4.8200000e-004  1.1710000e-003  1.2059000e-002  2.7168675e+001  5.4377510e+000
3.5000000e+006  1.6720000e-003  1.7289157e+006  3.1700000e-004  8.3700000e-004  1.2092000e-002  3.4590361e+001  3.9919680e+000
2.1987952e+006  1.0920000e-003  6.4819277e+005  5.3800000e-004  6.2900000e-004  1.1690000e-002  3.4831325e+001  2.8353410e+000
4.8915663e+006  1.1030000e-003  1.9325301e+006  4.5500000e-004  1.0380000e-003  1.1886000e-002  3.5698795e+001  3.7028110e+000
2.7590361e+006  2.0060000e-003  2.4096386e+005  5.7800000e-004  1.0540000e-003  1.2981000e-002  3.4204819e+001  8.5863450e+000
1.5662651e+006  1.2650000e-003  1.5722892e+006  4.3700000e-004  8.2900000e-004  1.3328000e-002  3.1891566e+001  5.2128510e+000
1.3493976e+006  1.0610000e-003  1.0554217e+006  3.9700000e-004  1.4960000e-003  1.3339000e-002  3.6277108e+001  8.9397590e+000
3.8614458e+006  8.6300000e-004  1.4000000e+006  5.8000000e-004  1.4600000e-003  1.2211000e-002  2.7457831e+001  8.5542170e+000
8.4337349e+005  9.3600000e-004  3.1927711e+005  3.6300000e-004  1.1390000e-003  1.1398000e-002  3.0012048e+001  2.5140560e+000
3.0843373e+006  1.5780000e-003  3.3734940e+006  2.0400000e-004  1.2430000e-003  1.1604000e-002  3.2373494e+001  9.1967870e+000
3.9156627e+006  2.0270000e-003  1.2590361e+006  4.0100000e-004  1.4440000e-003  1.2645000e-002  2.7698795e+001  7.7188760e+000
2.1807229e+006  1.8180000e-003  2.9349398e+006  5.6400000e-004  5.0800000e-004  1.3035000e-002  2.8710843e+001  4.5060240e+000
1.3313253e+006  1.7030000e-003  3.8120482e+006  3.7500000e-004  8.9800000e-004  1.3794000e-002  2.8855422e+001  2.9638550e+000
4.8554217e+006  1.7710000e-003  1.8855422e+006  4.1300000e-004  1.4840000e-003  1.1994000e-002  3.1457831e+001  7.6546180e+000
1.0602410e+006  8.8900000e-004  2.1048193e+006  3.1900000e-004  9.2600000e-004  1.1842000e-002  3.5939759e+001  9.3253010e+000
2.5240964e+006  1.1130000e-003  3.7337349e+006  2.9500000e-004  1.4000000e-003  1.2634000e-002  2.6108434e+001  3.3815260e+000
2.4879518e+006  8.2100000e-004  2.1674699e+006  1.3000000e-004  1.1060000e-003  1.1300000e-002  3.2759036e+001  8.9718880e+000
1.1506024e+006  1.5620000e-003  1.9168675e+006  4.1500000e-004  1.0820000e-003  1.2547000e-002  3.2518072e+001  9.8393570e+000
1.8192771e+006  1.4000000e-003  2.2771084e+006  2.8700000e-004  8.1700000e-004  1.3827000e-002  3.5795181e+001  5.0200800e+000
6.4457831e+005  2.0900000e-003  3.9686747e+006  5.4800000e-004  1.1750000e-003  1.1376000e-002  3.5506024e+001  9.7108430e+000
3.3915663e+006  1.7400000e-003  6.7951807e+005  5.4400000e-004  8.3300000e-004  1.1777000e-002  3.5987952e+001  3.6385540e+000
3.3734940e+006  1.9330000e-003  2.7469880e+006  1.9800000e-004  1.0460000e-003  1.2102000e-002  3.3530120e+001  6.0803210e+000
2.3072289e+006  1.0040000e-003  1.3132530e+005  2.0800000e-004  9.2200000e-004  1.2558000e-002  2.5530120e+001  7.9116470e+000
3.0120482e+006  2.1000000e-003  3.8192771e+005  2.8100000e-004  1.1470000e-003  1.1972000e-002  3.4060241e+001  4.3453820e+000
3.9337349e+006  1.1650000e-003  1.4313253e+006  3.2300000e-004  6.7700000e-004  1.3317000e-002  3.5024096e+001  4.4096390e+000
7.7108434e+005  1.3270000e-003  3.3421687e+006  5.9400000e-004  1.3310000e-003  1.1647000e-002  2.8325301e+001  9.1646590e+000
1.3132530e+006  1.7970000e-003  2.9662651e+006  5.0200000e-004  1.2710000e-003  1.1560000e-002  3.0927711e+001  4.5381530e+000
4.1867470e+006  1.8230000e-003  1.1566265e+005  1.6800000e-004  9.3800000e-004  1.2916000e-002  3.3385542e+001  6.6265060e+000
5.7228916e+005  1.5310000e-003  1.5879518e+006  3.6500000e-004  6.3300000e-004  1.1896000e-002  3.2084337e+001  2.4497990e+000
2.6144578e+006  9.5700000e-004  3.1698795e+006  3.8500000e-004  1.3510000e-003  1.1506000e-002  3.4445783e+001  4.2811240e+000
1.7650602e+006  1.7500000e-003  1.0397590e+006  2.5300000e-004  1.2350000e-003  1.2287000e-002  2.6156627e+001  4.4738960e+000
2.0722892e+006  1.9750000e-003  2.3397590e+006  3.2100000e-004  1.4520000e-003  1.3859000e-002  2.8084337e+001  5.3092370e+000
7.3493976e+005  1.9490000e-003  1.9481928e+006  4.3300000e-004  1.1670000e-003  1.1322000e-002  3.6903614e+001  9.2610440e+000
2.2891566e+006  1.8290000e-003  2.6373494e+006  5.9800000e-004  6.4500000e-004  1.1907000e-002  3.6180723e+001  8.2329320e+000
9.6987952e+005  1.3740000e-003  2.8795181e+005  2.7900000e-004  7.7300000e-004  1.2970000e-002  3.3433735e+001  8.2008030e+000
2.8674699e+006  1.7350000e-003  2.9036145e+006  4.6300000e-004  8.4900000e-004  1.3165000e-002  3.1120482e+001  6.3052210e+000
2.6867470e+006  2.0740000e-003  1.9795181e+006  2.0200000e-004  5.9600000e-004  1.2167000e-002  3.1024096e+001  3.1244980e+000
1.6566265e+006  8.5700000e-004  2.2457831e+006  1.8000000e-004  1.0580000e-003  1.3252000e-002  2.6831325e+001  7.7510040e+000
1.4036145e+006  9.9300000e-004  7.5783133e+005  3.3100000e-004  5.0000000e-004  1.1799000e-002  2.7650602e+001  5.1164660e+000
3.4277108e+006  9.6700000e-004  4.2891566e+005  5.6000000e-004  1.0020000e-003  1.3653000e-002  2.5048193e+001  3.7991970e+000
3.1385542e+006  1.6560000e-003  1.5566265e+006  4.2100000e-004  6.2400000e-004  1.2580000e-002  2.6686747e+001  5.5020080e+000
4.9457831e+006  1.9640000e-003  3.2325301e+006  2.8500000e-004  7.4900000e-004  1.3566000e-002  3.5216867e+001  8.6827310e+000
2.9939759e+006  1.1080000e-003  2.0108434e+006  2.4300000e-004  1.0620000e-003  1.2352000e-002  3.6759036e+001  5.9196790e+000
4.6204819e+006  1.3640000e-003  2.3554217e+006  4.1100000e-004  5.2000000e-004  1.2037000e-002  3.3963855e+001  6.8192770e+000
4.9819277e+006  8.3700000e-004  3.8433735e+006  5.5800000e-004  1.2750000e-003  1.2265000e-002  2.6590361e+001  5.9518070e+000
2.8855422e+006  1.6610000e-003  9.9277108e+005  2.6100000e-004  9.9000000e-004  1.1408000e-002  3.3481928e+001  8.8112450e+000
3.2108434e+006  9.7800000e-004  3.6554217e+006  1.1800000e-004  1.2990000e-003  1.3480000e-002  3.3771084e+001  7.1084340e+000
2.7048193e+006  1.8700000e-003  1.2903614e+006  5.7600000e-004  9.9400000e-004  1.2775000e-002  2.7361446e+001  5.8554220e+000
2.9397590e+006  1.0870000e-003  7.8915663e+005  4.8800000e-004  1.3550000e-003  1.3946000e-002  2.7506024e+001  8.0722890e+000
3.5542169e+006  1.8810000e-003  2.5746988e+006  5.8800000e-004  7.8900000e-004  1.2005000e-002  2.9337349e+001  7.6224900e+000
4.8734940e+006  8.8400000e-004  1.2277108e+006  3.7900000e-004  7.6100000e-004  1.3046000e-002  2.9144578e+001  4.6666670e+000
3.1024096e+006  1.4470000e-003  2.4024096e+006  1.4400000e-004  1.3710000e-003  1.2829000e-002  3.4012048e+001  5.0843370e+000
4.9638554e+006  8.7300000e-004  2.7783133e+006  2.9300000e-004  1.0940000e-003  1.3263000e-002  3.5361446e+001  7.2048190e+000
1.5481928e+006  1.6300000e-003  1.7132530e+006  1.8200000e-004  8.7800000e-004  1.2395000e-002  2.5578313e+001  7.6867470e+000
1.9457831e+006  1.7660000e-003  1.1963855e+006  3.0700000e-004  8.4500000e-004  1.3870000e-002  3.2180723e+001  4.2168670e+000
9.1566265e+005  1.4110000e-003  3.4831325e+006  2.1200000e-004  1.1020000e-003  1.3078000e-002  2.9385542e+001  2.6104420e+000
2.4698795e+006  1.2700000e-003  9.4578313e+005  3.7700000e-004  6.2000000e-004  1.2298000e-002  3.3192771e+001  5.1485940e+000
4.0783133e+006  1.4320000e-003  5.0722892e+005  5.2400000e-004  7.2500000e-004  1.2883000e-002  3.1602410e+001  9.5502010e+000
2.5421687e+006  2.0480000e-003  3.0759036e+006  4.4900000e-004  8.9000000e-004  1.2243000e-002  3.6518072e+001  6.5301200e+000
4.1325301e+006  1.8960000e-003  1.6506024e+006  1.4800000e-004  5.3200000e-004  1.3306000e-002  3.0638554e+001  9.2931730e+000
2.1084337e+006  1.5200000e-003  7.7349398e+005  5.4600000e-004  7.9300000e-004  1.1593000e-002  2.5192771e+001  2.0963860e+000
3.2650602e+006  1.8130000e-003  6.9518072e+005  2.2000000e-004  8.7300000e-004  1.2807000e-002  2.7602410e+001  3.2530120e+000
4.6927711e+006  9.3100000e-004  2.8879518e+006  4.9000000e-004  1.0980000e-003  1.1311000e-002  2.8614458e+001  3.1566270e+000
1.4759036e+006  2.0160000e-003  3.6626506e+005  1.5800000e-004  6.1600000e-004  1.3707000e-002  3.2614458e+001  7.3975900e+000
9.5180723e+005  1.9540000e-003  1.4469880e+006  5.3400000e-004  9.5800000e-004  1.2753000e-002  2.5481928e+001  9.1004020e+000
2.0000000e+006  1.6820000e-003  4.1325301e+005  2.1600000e-004  5.9200000e-004  1.2905000e-002  3.3578313e+001  3.4136550e+000
2.9578313e+006  1.0560000e-003  3.0602410e+006  2.4900000e-004  7.1700000e-004  1.3013000e-002  2.9289157e+001  2.5461850e+000
4.6024096e+006  8.6800000e-004  1.7602410e+006  2.1400000e-004  7.0900000e-004  1.3902000e-002  3.1409639e+001  3.2851410e+000
3.2289157e+006  1.3430000e-003  3.1072289e+006  3.7100000e-004  1.3960000e-003  1.2840000e-002  3.0493976e+001  2.5783130e+000
4.2590361e+006  1.9220000e-003  3.6867470e+006  3.1100000e-004  1.0660000e-003  1.1755000e-002  3.6710843e+001  4.6345380e+000
1.8373494e+006  9.5100000e-004  1.6349398e+006  4.6900000e-004  1.2590000e-003  1.3620000e-002  3.6325301e+001  2.3212850e+000
4.5843373e+006  1.3590000e-003  1.6662651e+006  4.7800000e-004  9.6600000e-004  1.3241000e-002  3.2903614e+001  4.3775100e+000
6.6265060e+005  1.0350000e-003  2.4493976e+006  3.4900000e-004  7.1300000e-004  1.3924000e-002  3.0445783e+001  3.5421690e+000
4.7289157e+006  1.6350000e-003  1.7831325e+005  3.8900000e-004  7.5300000e-004  1.2699000e-002  2.6734940e+001  4.1526100e+000
4.7831325e+006  1.3170000e-003  2.2927711e+006  3.5700000e-004  1.3840000e-003  1.1343000e-002  2.5626506e+001  2.7710840e+000
3.1746988e+006  1.5100000e-003  2.0734940e+006  2.5700000e-004  1.0340000e-003  1.3414000e-002  2.7313253e+001  5.8875500e+000
3.0301205e+006  1.3060000e-003  5.6987952e+005  1.8800000e-004  6.9300000e-004  1.1940000e-002  2.6301205e+001  3.1887550e+000
3.6807229e+006  1.0450000e-003  2.5903614e+006  5.4200000e-004  1.1830000e-003  1.2710000e-002  2.5963855e+001  5.4698800e+000
3.7349398e+006  1.0660000e-003  6.3253012e+005  1.9400000e-004  1.2670000e-003  1.3382000e-002  2.7843373e+001  5.7911650e+000
4.3855422e+006  1.0190000e-003  1.2746988e+006  3.6900000e-004  8.0100000e-004  1.2124000e-002  2.9963855e+001  9.9678710e+000
2.8313253e+006  1.6410000e-003  2.7156627e+006  5.5600000e-004  1.3920000e-003  1.1625000e-002  2.5433735e+001  8.8755020e+000
2.8132530e+006  1.4580000e-003  3.3265060e+006  5.8400000e-004  6.0800000e-004  1.3512000e-002  3.4879518e+001  2.0000000e+000
3.3373494e+006  1.3900000e-003  3.1855422e+006  3.5500000e-004  6.3700000e-004  1.2449000e-002  2.5915663e+001  4.8915660e+000
1.6385542e+006  1.7760000e-003  2.5120482e+006  1.2600000e-004  1.4320000e-003  1.1929000e-002  3.0156627e+001  6.4016060e+000
3.5722892e+006  1.4420000e-003  2.7000000e+006  5.3600000e-004  1.4760000e-003  1.3545000e-002  2.7939759e+001  8.1044180e+000
4.4036145e+006  1.2330000e-003  2.8096386e+006  1.9600000e-004  8.4100000e-004  1.3848000e-002  2.6253012e+001  9.1325300e+000
1.2590361e+006  1.2910000e-003  3.7024096e+006  6.0000000e-004  1.5000000e-003  1.3360000e-002  2.7554217e+001  5.9839360e+000
2.9759036e+006  1.6930000e-003  1.6192771e+006  3.1500000e-004  5.6400000e-004  1.1333000e-002  3.0975904e+001  5.2449800e+000
5.5421687e+005  1.7550000e-003  3.8590361e+006  2.7500000e-004  1.3670000e-003  1.2796000e-002  3.0397590e+001  6.8835340e+000
```

```
4.7650602e+006  1.3010000e-003  2.1987952e+006  4.9600000e-004  8.9400000e-004  1.3957000e-002  2.6012048e+001  6.3373490e+000
3.6626506e+006  1.8390000e-003  2.7228916e+005  4.3900000e-004  9.1000000e-004  1.3675000e-002  2.9867470e+001  8.4899600e+000
3.7710843e+006  1.7820000e-003  6.0120482e+005  2.1800000e-004  1.1100000e-003  1.3772000e-002  3.3674699e+001  4.3132530e+000
1.1144578e+006  2.0110000e-003  7.1084337e+005  5.1600000e-004  1.0500000e-003  1.1745000e-002  3.1939759e+001  2.1927710e+000
1.4578313e+006  1.1920000e-003  8.8313253e+005  2.5900000e-004  1.2910000e-003  1.3002000e-002  3.0060241e+001  4.9879520e+000
6.9879518e+005  1.7610000e-003  3.4361446e+006  1.3400000e-004  1.0220000e-003  1.3805000e-002  3.2277108e+001  6.1767070e+000
1.3855422e+006  1.7290000e-003  1.7445783e+006  3.0300000e-004  5.4800000e-004  1.2439000e-002  2.9674699e+001  8.2971890e+000
1.7289157e+006  1.3480000e-003  3.6084337e+006  1.2000000e-004  1.2270000e-003  1.1614000e-002  2.7120482e+001  8.6184740e+000
4.0060241e+006  1.6770000e-003  5.3855422e+005  5.9600000e-004  1.3390000e-003  1.1788000e-002  3.3867470e+001  2.4819280e+000
1.0240964e+006  1.1970000e-003  3.5457831e+006  1.5400000e-004  9.7800000e-004  1.3393000e-002  3.4253012e+001  3.8313250e+000
1.7469880e+006  1.1180000e-003  2.0265060e+006  4.9400000e-004  6.4900000e-004  1.1669000e-002  2.8903614e+001  4.8273090e+000
1.9096386e+006  8.4200000e-004  1.3060241e+006  4.7100000e-004  1.3760000e-003  1.2319000e-002  3.4301205e+001  8.9076310e+000
2.6686747e+006  1.4060000e-003  1.0710843e+006  4.8000000e-004  7.0100000e-004  1.3523000e-002  2.6204819e+001  9.3574300e+000
1.7831325e+006  1.3330000e-003  1.0084337e+006  4.7300000e-004  8.2100000e-004  1.3198000e-002  2.6060241e+001  5.6305220e+000
1.2409639e+006  1.1860000e-003  2.5590361e+006  5.0000000e-004  6.8500000e-004  1.3111000e-002  3.4734940e+001  2.9959840e+000
3.8795181e+006  1.2800000e-003  1.6975904e+006  4.4500000e-004  1.4680000e-003  1.2200000e-002  3.4686747e+001  2.4176710e+000
2.0180723e+006  1.7870000e-003  5.5421687e+005  4.6700000e-004  1.4160000e-003  1.1387000e-002  3.4638554e+001  8.2650600e+000
2.2349398e+006  1.9590000e-003  2.0421687e+006  1.8600000e-004  1.3030000e-003  1.3555000e-002  3.6421687e+001  7.1405620e+000
1.0421687e+006  1.2540000e-003  1.7759036e+006  1.1600000e-004  9.1400000e-004  1.1734000e-002  2.7024096e+001  9.9357430e+000
3.6265060e+006  8.1600000e-004  1.2120482e+006  5.4000000e-004  6.5300000e-004  1.2189000e-002  3.0542169e+001  3.0923690e+000
3.5180723e+006  1.4840000e-003  2.3084337e+006  3.8300000e-004  6.8900000e-004  1.3837000e-002  3.4975904e+001  7.5261040e+000
2.3795181e+006  1.1240000e-003  1.6819277e+006  3.9500000e-004  1.0860000e-003  1.3913000e-002  2.5771084e+001  3.5742970e+000
2.1445783e+006  1.4160000e-003  3.7807229e+006  4.6500000e-004  6.1200000e-004  1.3122000e-002  2.7746988e+001  8.7148590e+000
4.2409639e+006  1.1600000e-003  3.9373494e+006  1.2400000e-004  1.1220000e-003  1.1831000e-002  3.0301205e+001  4.0883530e+000
3.4638554e+006  1.5830000e-003  4.6024096e+005  2.9100000e-004  1.1870000e-003  1.2016000e-002  2.9915663e+001  4.1204820e+000
4.6385542e+006  1.5050000e-003  3.3493976e+005  3.2700000e-004  1.1790000e-003  1.1528000e-002  3.5313253e+001  5.2771080e+000
1.2771084e+006  1.4940000e-003  1.5096386e+006  4.2300000e-004  1.3070000e-003  1.2655000e-002  2.5337349e+001  8.8433730e+000
4.7108434e+006  1.6090000e-003  1.4698795e+005  1.1000000e-004  5.7600000e-004  1.1820000e-002  3.1650602e+001  2.2249000e+000
2.7409639e+006  1.4370000e-003  3.2168675e+006  4.2900000e-004  5.3600000e-004  1.1582000e-002  2.7987952e+001  8.1365460e+000
7.1686747e+005  1.8600000e-003  1.1180723e+006  1.7000000e-004  1.1630000e-003  1.3978000e-002  3.6373494e+001  9.6465860e+000
2.4518072e+006  1.5570000e-003  8.2048193e+005  4.0700000e-004  1.4080000e-003  1.3154000e-002  2.5385542e+001  6.0481930e+000
2.6325301e+006  1.1290000e-003  2.4650602e+006  1.7800000e-004  7.2900000e-004  1.2178000e-002  2.8662651e+001  5.3413650e+000
1.0963855e+006  1.3380000e-003  3.5614458e+006  3.1300000e-004  1.2510000e-003  1.1463000e-002  3.3000000e+001  3.8634540e+000
1.4939759e+006  8.5200000e-004  1.8698795e+006  1.0800000e-004  1.2870000e-003  1.3664000e-002  3.5457831e+001  9.0361450e+000
2.1265060e+006  1.8340000e-003  3.2795181e+006  1.0200000e-004  8.2500000e-004  1.2742000e-002  3.0686747e+001  3.3172690e+000
1.8012048e+006  2.0370000e-003  2.2614458e+006  5.3200000e-004  1.0900000e-003  1.3989000e-002  3.2710843e+001  9.8714860e+000
3.7168675e+006  8.0000000e-004  2.5433735e+006  4.6100000e-004  5.8800000e-004  1.3588000e-002  3.2855422e+001  8.3614460e+000
3.6084337e+006  1.5730000e-003  2.3867470e+006  4.4300000e-004  1.2550000e-003  1.2569000e-002  2.9192771e+001  9.0040160e+000
4.0602410e+006  1.7140000e-003  2.8722892e+006  3.4300000e-004  1.1550000e-003  1.3631000e-002  3.6228916e+001  4.7630520e+000
3.5903614e+006  1.7920000e-003  3.9843373e+006  5.2600000e-004  7.4100000e-004  1.2927000e-002  3.4927711e+001  5.5662650e+000
2.2530120e+006  1.1390000e-003  2.6530120e+006  3.0500000e-004  8.8200000e-004  1.1365000e-002  2.5722892e+001  9.4859440e+000
8.9759036e+005  8.0500000e-004  2.0963855e+005  1.3800000e-004  1.0180000e-003  1.3761000e-002  3.4542169e+001  5.7590360e+000
2.3975904e+006  1.4270000e-003  2.4337349e+006  5.8200000e-004  1.3590000e-003  1.2373000e-002  3.2228916e+001  1.0000000e+001
3.4819277e+006  1.5520000e-003  3.7963855e+006  1.5600000e-004  8.6100000e-004  1.2525000e-002  3.6036145e+001  2.6425700e+000
4.3132530e+006  8.7800000e-004  2.9192771e+006  1.7400000e-004  5.4400000e-004  1.2666000e-002  3.1313253e+001  5.6947790e+000
3.8072289e+006  9.1500000e-004  5.8554217e+005  5.9000000e-004  5.2800000e-004  1.3219000e-002  3.2807229e+001  4.9236950e+000
2.3433735e+006  1.3950000e-003  4.9156627e+005  1.3200000e-004  1.2790000e-003  1.3686000e-002  3.5168675e+001  7.5903610e+000
7.5301205e+005  2.0320000e-003  1.1493976e+006  1.4000000e-004  1.3800000e-003  1.2428000e-002  3.2469880e+001  7.4939760e+000
5.3614458e+005  1.2390000e-003  4.7590361e+005  1.2800000e-004  1.4880000e-003  1.3783000e-002  2.9530120e+001  5.3734940e+000
3.5361446e+006  1.4890000e-003  6.6385542e+005  1.7600000e-004  1.4800000e-003  1.2623000e-002  3.1987952e+001  7.0120480e+000
1.1325301e+006  1.8550000e-003  2.9819277e+006  2.6300000e-004  1.0420000e-003  1.2677000e-002  3.4349398e+001  7.4297190e+000
1.5120482e+006  1.7450000e-003  2.5662651e+005  1.0600000e-004  7.6500000e-004  1.2048000e-002  3.4493976e+001  6.5622490e+000
4.2951807e+006  1.6140000e-003  2.9975904e+006  4.4700000e-004  5.5600000e-004  1.2471000e-002  3.4156627e+001  3.2208840e+000
1.6927711e+006  1.4680000e-003  2.2301205e+006  3.0100000e-004  1.0300000e-003  1.1810000e-002  3.5891566e+001  8.0080320e+000
2.0903614e+006  1.6250000e-003  1.0867470e+006  2.2700000e-004  1.4360000e-003  1.3892000e-002  2.8228916e+001  8.4578310e+000
2.5783133e+006  9.6200000e-004  1.8228916e+006  5.5000000e-004  9.8200000e-004  1.1983000e-002  3.0734940e+001  7.0763050e+000
3.4096386e+006  1.2960000e-003  2.4807229e+006  5.9200000e-004  1.0740000e-003  1.1961000e-002  3.0783133e+001  2.8032130e+000
1.5301205e+006  2.0220000e-003  2.3710843e+006  1.7200000e-004  6.6100000e-004  1.3534000e-002  3.5409639e+001  7.4618470e+000
1.9638554e+006  1.9170000e-003  1.9397590e+005  3.6100000e-004  1.4640000e-003  1.3751000e-002  3.2421687e+001  6.9799200e+000
1.1867470e+006  1.0710000e-003  9.6144578e+005  4.8400000e-004  5.6800000e-004  1.2731000e-002  2.9771084e+001  6.7550200e+000
1.1686747e+006  1.9020000e-003  3.5144578e+006  2.8900000e-004  6.8100000e-004  1.1766000e-002  3.1265060e+001  9.2289160e+000
4.3674699e+006  1.4630000e-003  3.4987952e+006  2.7700000e-004  9.9800000e-004  1.3610000e-002  3.6807229e+001  7.8473900e+000
3.3192771e+006  2.0630000e-003  2.6843373e+006  1.8400000e-004  7.2100000e-004  1.3740000e-002  2.6493976e+001  5.0522090e+000
1.4397590e+006  1.4790000e-003  3.1228916e+006  2.4500000e-004  5.7200000e-004  1.2276000e-002  3.1072289e+001  4.6024100e+000
6.0843373e+005  1.1710000e-003  1.4939759e+006  5.1400000e-004  1.3630000e-003  1.2330000e-002  3.6614458e+001  2.0642570e+000
3.1204819e+006  9.2500000e-004  3.2951807e+006  2.4100000e-004  9.1800000e-004  1.3176000e-002  3.6662651e+001  6.3694780e+000
3.9879518e+006  2.0690000e-003  1.4783133e+006  3.8100000e-004  1.1510000e-003  1.2720000e-002  3.3626506e+001  4.0240960e+000
4.9096386e+006  1.8490000e-003  1.3373494e+006  2.4700000e-004  1.1990000e-003  1.3208000e-002  2.6349398e+001  8.7791160e+000
2.5602410e+006  1.2280000e-003  3.8277108e+006  4.5100000e-004  1.1140000e-003  1.3599000e-002  2.6927711e+001  4.6987950e+000
1.0783133e+006  1.0770000e-003  3.7650602e+006  5.0400000e-004  1.2310000e-003  1.1918000e-002  2.8277108e+001  5.8232930e+000
2.8493976e+006  1.5410000e-003  8.9879518e+005  3.4100000e-004  7.4500000e-004  1.2254000e-002  3.6469880e+001  8.3293170e+000
3.7530120e+006  1.3530000e-003  2.1831325e+006  2.0600000e-004  1.4920000e-003  1.3729000e-002  2.6445783e+001  9.3895580e+000
4.1506024e+006  1.6510000e-003  8.6746988e+005  5.7200000e-004  1.3430000e-003  1.3284000e-002  3.2566265e+001  4.8594380e+000
4.8192771e+006  9.4100000e-004  3.5927711e+006  4.3500000e-004  6.0000000e-004  1.2493000e-002  2.8518072e+001  5.1807230e+000
2.1626506e+006  1.0510000e-003  3.7493976e+006  1.5000000e-004  1.0140000e-003  1.1680000e-002  3.3722892e+001  2.0321290e+000
1.9819277e+006  1.8860000e-003  2.7939759e+006  4.8600000e-004  8.6900000e-004  1.2406000e-002  2.8132530e+001  2.7068270e+000
4.2771084e+006  1.2440000e-003  1.4156627e+006  4.0900000e-004  9.4600000e-004  1.1712000e-002  3.0204819e+001  3.3493980e+000
3.2831325e+006  1.0140000e-003  1.1024096e+006  2.6700000e-004  1.4480000e-003  1.2384000e-002  3.6084337e+001  4.5702810e+000
4.1686747e+006  8.9900000e-004  1.2433735e+006  1.2200000e-004  1.3150000e-003  1.1484000e-002  3.3096386e+001  3.0281120e+000
4.1144578e+006  1.2070000e-003  3.6240964e+006  4.9200000e-004  6.7300000e-004  1.3469000e-002  2.6397590e+001  8.1686750e+000
7.8915663e+005  1.9380000e-003  3.8903614e+006  5.2000000e-004  6.9700000e-004  1.1875000e-002  2.9048193e+001  8.3935740e+000
8.2530120e+005  1.6460000e-003  3.4048193e+006  5.1800000e-004  6.4100000e-004  1.3057000e-002  2.5144578e+001  6.2088350e+000
8.0722892e+005  1.1810000e-003  3.4674699e+006  3.5900000e-004  7.3300000e-004  1.2460000e-002  2.7409639e+001  5.5983940e+000
1.2228916e+006  8.4700000e-004  1.5409639e+006  5.0600000e-004  1.2470000e-003  1.3133000e-002  2.9240964e+001  6.4658630e+000
6.2650602e+005  1.8760000e-003  3.9060241e+006  3.2900000e-004  5.5200000e-004  1.2861000e-002  3.3337349e+001  9.7429720e+000
4.5120482e+006  1.9690000e-003  3.0132530e+006  1.3600000e-004  7.6900000e-004  1.2948000e-002  2.9626506e+001  6.8514060e+000
3.2469880e+006  8.9400000e-004  3.3578313e+006  1.1200000e-004  1.4240000e-003  1.2157000e-002  3.5843373e+001  5.6626510e+000
1.4216867e+006  2.0430000e-003  1.4626506e+006  1.6000000e-004  5.0400000e-004  1.1853000e-002  2.5096386e+001  4.9558230e+000
2.3614458e+006  1.8650000e-003  9.7710843e+005  3.8700000e-004  1.4720000e-003  1.2504000e-002  3.2036145e+001  8.0401610e+000
3.1566265e+006  9.1000000e-004  3.0289157e+006  3.5100000e-004  1.4280000e-003  1.2233000e-002  3.0879518e+001  9.4216870e+000
2.0361446e+006  1.0090000e-003  2.2144578e+006  4.2500000e-004  9.3400000e-004  1.3100000e-002  3.6855422e+001  6.4337350e+000
3.1927711e+006  1.9430000e-003  3.4204819e+006  2.5100000e-004  9.0600000e-004  1.3458000e-002  3.3048193e+001  3.9277110e+000
4.0240964e+006  9.4600000e-004  2.1361446e+006  4.5900000e-004  9.4200000e-004  1.2341000e-002  3.6132530e+001  8.7469880e+000
```

```
1.8554217e+006  1.6670000e-003  1.3530120e+006  3.9900000e-004  5.6000000e-004  1.4000000e-002  2.8373494e+001  7.3333330e+000
1.3674699e+006  9.8800000e-004  3.1542169e+006  3.0900000e-004  1.2830000e-003  1.3067000e-002  2.6542169e+001  9.8072290e+000
3.8253012e+006  1.8440000e-003  2.4963855e+006  5.5200000e-004  1.4040000e-003  1.2818000e-002  3.2132530e+001  6.6586350e+000
4.5481928e+006  1.9900000e-003  2.1204819e+006  3.9100000e-004  9.7400000e-004  1.1571000e-002  2.8469880e+001  6.1124500e+000
3.3012048e+006  1.1760000e-003  9.1445783e+005  4.9800000e-004  1.1270000e-003  1.2222000e-002  3.4108434e+001  4.4417670e+000
2.0542169e+006  9.0400000e-004  1.1807229e+006  5.2200000e-004  9.8600000e-004  1.2514000e-002  2.5819277e+001  8.5220880e+000
4.6566265e+006  1.6980000e-003  2.9506024e+006  5.1000000e-004  5.8400000e-004  1.1636000e-002  2.9096386e+001  4.7309240e+000
2.9216867e+006  1.5940000e-003  2.8409639e+006  1.6600000e-004  1.1350000e-003  1.3577000e-002  2.8951807e+001  9.5180720e+000
2.9036145e+006  1.5990000e-003  7.2650602e+005  4.7600000e-004  8.6500000e-004  1.2851000e-002  2.9481928e+001  2.3855420e+000
2.7228916e+006  1.3120000e-003  2.2530120e+005  4.2700000e-004  1.3470000e-003  1.3024000e-002  2.9433735e+001  3.8955820e+000
2.6506024e+006  2.0010000e-003  2.6686747e+006  1.4600000e-004  1.2190000e-003  1.2417000e-002  2.6975904e+001  4.2489960e+000
3.6445783e+006  1.9280000e-003  3.0361446e+005  3.5300000e-004  7.8100000e-004  1.1951000e-002  2.6879518e+001  8.6506020e+000
1.2951807e+006  1.9800000e-003  3.4518072e+006  5.2800000e-004  1.2390000e-003  1.2482000e-002  3.5072289e+001  5.5341370e+000
1.6204819e+006  1.7190000e-003  8.5180723e+005  2.6900000e-004  9.5000000e-004  1.1539000e-002  3.3144578e+001  3.9598390e+000
2.3253012e+006  8.1000000e-004  3.5301205e+006  5.5400000e-004  7.8500000e-004  1.3273000e-002  3.1216867e+001  6.6907630e+000
5.9036145e+005  1.1450000e-003  5.2289157e+005  1.5200000e-004  1.2630000e-003  1.2081000e-002  2.5240964e+001  2.8995980e+000
8.6144578e+005  1.5000000e-003  4.0000000e+006  4.0500000e-004  1.0100000e-003  1.3187000e-002  3.1506024e+001  6.9156630e+000
3.0662651e+006  9.7200000e-004  1.7915663e+006  2.2400000e-004  5.2400000e-004  1.2937000e-002  2.5000000e+001  7.9759040e+000
4.6746988e+006  1.2120000e-003  3.7180723e+006  3.3500000e-004  1.4400000e-003  1.2146000e-002  3.7000000e+001  2.7389560e+000
5.1807229e+005  1.5150000e-003  1.9012048e+006  4.5700000e-004  5.1200000e-004  1.3490000e-002  2.8036145e+001  3.4457830e+000
3.6987952e+006  1.1550000e-003  1.1337349e+006  2.3500000e-004  1.0780000e-003  1.1473000e-002  3.1698795e+001  6.1445780e+000
2.4156627e+006  1.2230000e-003  1.5253012e+006  2.7100000e-004  1.2150000e-003  1.3935000e-002  3.1843373e+001  7.2369480e+000
3.8433735e+006  1.5670000e-003  3.0445783e+006  2.3100000e-004  7.5700000e-004  1.1658000e-002  2.8421687e+001  3.0602410e+000
4.0421687e+006  1.9960000e-003  8.0481928e+005  1.0400000e-004  5.1600000e-004  1.1864000e-002  2.5867470e+001  6.2409640e+000
2.7771084e+006  1.7080000e-003  3.5060241e+005  3.2500000e-004  7.9700000e-004  1.2894000e-002  3.1746988e+001  6.7228920e+000
3.9518072e+006  1.6040000e-003  3.8746988e+006  4.0300000e-004  1.1430000e-003  1.1549000e-002  2.7216867e+001  6.2730920e+000
4.4759036e+006  1.3800000e-003  7.4216867e+005  1.9000000e-004  9.3000000e-004  1.1701000e-002  3.6951807e+001  8.4257030e+000
5.0000000e+005  1.6200000e-003  2.7626506e+006  4.5300000e-004  9.6200000e-004  1.1354000e-002  2.7891566e+001  7.3012050e+000
2.5060241e+006  2.0840000e-003  2.0891566e+006  2.9900000e-004  8.0900000e-004  1.3143000e-002  2.6638554e+001  9.6144580e+000
1.5843373e+006  1.2180000e-003  1.9951807e+006  2.0000000e-004  1.3350000e-003  1.1495000e-002  2.9578313e+001  7.3654620e+000
4.9277108e+006  1.6880000e-003  1.6265060e+005  3.3300000e-004  1.4200000e-003  1.2536000e-002  3.5602410e+001  6.9477910e+000
```

Output vector y of *dispmax*

Note this is a 249-length column vector, but has been concatenated into 8 columns to save space.

```
1.0517368e-002  1.2149238e-002  1.2778797e-002  1.0438524e-002  1.1316178e-002  1.4058323e-002  1.2260369e-002  9.4692980e-003
1.2984094e-002  1.0463430e-002  1.0431446e-002  1.1874163e-002  1.4470492e-002  1.4266127e-002  1.1842597e-002  1.2528631e-002
1.3538514e-002  1.1400815e-002  1.2066969e-002  8.5284984e-003  1.4845266e-002  1.2550013e-002  1.0295700e-002  1.2080729e-002
1.1817125e-002  1.1452659e-002  1.2181603e-002  1.7571467e-002  1.2686501e-002  1.3103659e-002  1.3384693e-002  1.5193165e-002
9.5173074e-003  9.8429995e-003  1.1847140e-002  1.2247588e-002  1.0179620e-002  1.1897994e-002  1.3184062e-002  1.0168973e-002
1.2950058e-002  1.1430232e-002  1.0721275e-002  1.5768064e-002  1.0920897e-002  1.0322843e-002  1.3174658e-002  1.2585424e-002
1.4176343e-002  1.3343953e-002  1.8738842e-002  1.1619367e-002  1.4544714e-002  2.2815260e-002  1.1286737e-002  1.1489451e-002
1.4312770e-002  1.0734795e-002  1.2253579e-002  1.4253709e-002  1.0849587e-002  1.0993997e-002  9.1683633e-003  1.2332125e-002
1.1382438e-002  1.5718768e-002  1.1007987e-002  9.9540394e-003  1.3695232e-002  1.1328717e-002  1.0342494e-002  1.3245803e-002
1.0950584e-002  1.3783782e-002  1.3538435e-002  1.0740980e-002  1.4163483e-002  1.3301925e-002  1.0404659e-002  1.9394086e-002
1.3246846e-002  1.6437514e-002  1.4524558e-002  1.3548825e-002  1.4312856e-002  1.5296498e-002  1.3540757e-002  1.3057992e-002
1.3111978e-002  1.1395234e-002  1.4492392e-002  1.2977695e-002  1.2816510e-002  1.6301154e-002  1.0904888e-002  1.3573635e-002
1.1276532e-002  1.6937981e-002  1.1977768e-002  1.6491521e-002  9.9086959e-003  1.3480405e-002  1.2235371e-002  9.4491145e-003
1.3605745e-002  1.1220364e-002  1.3824311e-002  1.4469971e-002  1.1573350e-002  2.5628225e-002  1.0640890e-002  1.5160110e-002
1.0584482e-002  1.1633974e-002  1.3116613e-002  8.3422934e-003  1.4502272e-002  1.3221633e-002  1.1778715e-002  1.1457515e-002
1.2502775e-002  1.1430972e-002  1.1613076e-002  1.2651073e-002  1.2221624e-002  1.2814407e-002  1.2055724e-002  1.4939785e-002
1.3497259e-002  1.2062900e-002  9.8555221e-003  1.1811131e-002  1.2476672e-002  1.4766657e-002  1.4141094e-002  9.8998286e-003
1.7220549e-002  1.0982998e-002  1.2033878e-002  1.0758035e-002  1.4327171e-002  1.0825838e-002  1.3525684e-002  1.1386871e-002
1.2982548e-002  1.4334147e-002  1.3645004e-002  1.3897070e-002  1.3434405e-002  1.2064893e-002  1.2463517e-002  1.5935688e-002
1.2240906e-002  9.8760414e-003  1.2239858e-002  1.2369741e-002  1.2165621e-002  1.4663143e-002  1.1486694e-002  9.2087284e-003
9.3030566e-003  1.1543852e-002  1.4815091e-002  9.7703307e-003  9.5891232e-003  1.2069071e-002  1.1815600e-002  1.3234171e-002
1.2458410e-002  1.2835051e-002  1.0272673e-002  1.0873921e-002  1.1854619e-002  9.3232885e-003  1.2399439e-002  1.0612506e-002
1.1367155e-002  1.4954931e-002  1.2904876e-002  1.1164168e-002  1.2438636e-002  1.4966236e-002  1.1629284e-002  1.2704661e-002
1.3596663e-002  1.0405706e-002  1.1684026e-002  1.0432446e-002  1.2343723e-002  1.7574678e-002  1.4303284e-002  1.0692546e-002
1.1676529e-002  1.1834316e-002  1.3221815e-002  1.2805241e-002  1.3665318e-002  1.5060603e-002  1.2466436e-002  1.5795075e-002
1.1649431e-002  1.1375037e-002  1.2097828e-002  1.3145729e-002  1.0549099e-002  1.1414178e-002  1.2500886e-002
1.2199951e-002  1.6307321e-002  1.4928826e-002  1.1284063e-002  1.7740421e-002  1.4203514e-002  1.5247019e-002
1.2016550e-002  1.2946383e-002  1.1410772e-002  1.1011109e-002  1.2891382e-002  1.2565908e-002  1.2623096e-002
1.1867043e-002  1.3581425e-002  1.3584448e-002  1.2023216e-002  1.1666917e-002  1.1681601e-002  1.1044223e-002
1.2441021e-002  1.0237037e-002  1.4111106e-002  1.2848239e-002  1.0120491e-002  1.2301229e-002  1.0056120e-002
1.1187838e-002  9.8627232e-003  1.1830584e-002  1.5568879e-002  1.6531252e-002  1.3986419e-002  1.2306691e-002
1.4716009e-002  1.6330621e-002  1.3426282e-002  1.4479214e-002  1.1756031e-002  1.1398993e-002  1.3052148e-002
```

Appendix C

The Aortic Valve Model

The following is a reduced input deck for the AV model. This is included since the author has found example decks from other people to be very useful (and equally, their absence to be somewhat frustrating). It is common practice to give reduced input decks in LS-Dyna conference publications. Only the important cards are included here and node and element definitions are removed due to obvious space limitations.

```
$# LS-DYNA Keyword file created by LS-PREPOST 2.4 - 22Jun2009(14:31)
$# Created on May-07-2010 (18:19:31)
*KEYWORD
*TITLE
$# title
FSI Model of Aortic Valve (mm/tonne/s)
*PARAMETER
R es        2.500000R ts        1.100000R c51       6.500000R pdif     9.9010E-6
R c1s       0.200000R xlam      1.140000
*CONTROL_ALE
$Basically default values here.
$#     dct      nadv      meth      afac      bfac      cfac      dfac      efac
         4         0         1     0.000     0.000     0.000     0.000     0.000
$#   start       end     aafac     vfact      prit       ebc      pref   nsidebc
     0.0001.0000E+20  1.000000 1.0000E-6         0         0     0.000         0
*CONTROL_CONTACT
$
$ Some control cards are omitted here
$
*CONTROL_TERMINATION
$#  endtim    endcyc     dtmin    endeng    endmas
  0.020000         0     0.000     0.000     0.000
*CONTROL_TIMESTEP
$#  dtinit    tssfac      isdo    tslimt     dt2ms      lctm     erode     ms1st
     0.000  0.700000         0     0.000     0.000         0         0         0
$ ==============
$ DATABASE cards (OMITTED)
$ ==============
$ ==============
$ BOUNDARY/CONSTRAINT cards
$ ==============
*BOUNDARY_SPC_SET_ID
$#      id                                                               heading
         0Upper node ring constraint
*SET_NODE_LIST_TITLE
Aortic Lagrangian No
$#     sid       da1       da2       da3       da4
```

```
         2     0.000     0.000     0.000     0.000
*BOUNDARY_SPC_SET_ID
$#      id                                                               heading
         3ALE Ventricular constraints
$#    nsid       cid      dofx      dofy      dofz     dofrx     dofry     dofrz
         3         0         1         1         1         1         1         1
$
$ Further boundary cards contraining shell nodes and ALE nodes are omitted here
$
*BOUNDARY_AMBIENT_EOS
$This controls the pressure of the inflow region.
$#     pid       lc1       lc2
        12         3         4
$Controls pressure of outflow region
        11         6         4
*CONTACT_ENTITY
$ This is the sliding symmetry contact plane.
$#     pid    geotyp      ssid     sstyp        sf        df        cf    intord
       112         1        10         0  1.000000     0.000     0.000         0
$#      bt        dt        so        go
    0.0001.0000E+20         0         1
$#      xc        yc        zc        ax        ay        az
     0.000     0.000     0.000  1.000000     0.000     0.000
$#      bx        by        bz
     0.000  1.000000     0.000
$#   inout        g1        g2        g3        g4        g5        g6        g7
         0     0.000     0.000     0.000  1.732051 -1.000000     0.000 30.000000
*PART
$# title
Sinus
$#     pid     secid       mid     eosid      hgid      grav    adpopt      tmid
         3         3        11         0         0         0         0         0
*SECTION_SHELL_TITLE
Sinus
$#   secid    elform      shrf       nip     propt   qr/irid     icomp     setyp
         3         2  0.833300         2         1         0         0         1
$#      t1        t2        t3        t4      nloc     marea      idof    edgset
&ts       &ts       &ts       &ts           0.000     0.000     0.000         0
*MAT_MOONEY-RIVLIN_RUBBER_TITLE
Sinus and Aortic root
$#     mid        ro        pr         a         b       ref
        11 1.3000E-9  0.499000&c1s           0.000     0.000
$#     sgl        sw        st      lcid
     0.000     0.000     0.000         0
*PART
$# title
LPres
$#     pid     secid       mid     eosid      hgid      grav    adpopt      tmid
        11         8         7         6         1         0         0         0
*SECTION_SOLID_ALE_TITLE
Pressure outflow
$#   secid    elform       aet
         8        11         4
$#    afac      bfac      cfac      dfac     start       end     aafac
     0.000     0.000     0.000     0.000     0.000     0.000     0.000
*MAT_NULL_TITLE
Low pressure blood
$#     mid        ro        pc        mu     terod     cerod        ym        pr
         7 1.0600E-9     0.000 3.4500E-9     0.000     0.000     0.000     0.000
*EOS_GRUNEISEN_TITLE
EOS for low pressure outflow (diastole pressure)
$Simplified EOS for blood - coefficients for water used here,
$but properties of blood in mat_null card
$P=GAMAO*E0=11.3kPa
$#   eosid         c        s1        s2        s3     gamao         a        e0
         6 1.4840E+6  1.979000     0.000     0.000  0.110000     0.000  0.102700
$#      v0
     0.000
*HOURGLASS_TITLE
```

```
Hourglass control for ALE parts
$#    hgid       ihq        qm       ibq        q1        q2    qb/vdc        qw
         1         1 1.0000E-5         0  1.500000  0.060000  0.100000  0.100000
*PART
$# title
HPres
$#     pid     secid       mid     eosid      hgid      grav    adpopt      tmid
        12         9         6         5         1         0         0         0
*SECTION_SOLID_ALE_TITLE
Pressure Inflow
$#   secid    elform       aet
         9        11         4
$#    afac      bfac      cfac      dfac     start       end     aafac
     0.000     0.000     0.000     0.000     0.000     0.000     0.000
*MAT_NULL_TITLE
High pressure blood
$#     mid        ro        pc        mu     terod     cerod        ym        pr
         6 1.0600E-9     0.000 3.4500E-9     0.000     0.000     0.000     0.000
*EOS_GRUNEISEN_TITLE
EOS for high pressure inflow (systole pressure)
$Simplified EOS for blood - coefficients for water used here,
$but properties of blood in mat_null card
$P=GAMAO*EO=16kPa
$#   eosid         c        s1        s2        s3    gamao          a        e0
         5 1.4840E+6  1.979000     0.000     0.000  0.110000     0.000  0.145000
$#      v0
     0.000
*PART
$# title
The Void
$#     pid     secid       mid     eosid      hgid      grav    adpopt      tmid
        13        11         5         0         1         0         0         0
*SECTION_SOLID_ALE_TITLE
The Void
$#   secid    elform       aet
        11        11         1
$#    afac      bfac      cfac      dfac     start       end     aafac
     0.000     0.000     0.000     0.000     0.000     0.000     0.000
*MAT_VACUUM_TITLE
Void
$Just here to allow elform11 to be used.
$#     mid       den
         5 1.0000E-9
*PART
$# title
L-0.25
$#     pid     secid       mid     eosid      hgid      grav    adpopt      tmid
       101       101        10         0         0         0         0         0
*SECTION_SHELL_TITLE
L-0.25
$#   secid    elform      shrf       nip     propt   qr/irid     icomp     setyp
       101         2  1.000000         2         1         0         0         1
$#      t1        t2        t3        t4      nloc     marea      idof    edgset
  0.250000  0.250000  0.250000  0.250000     0.000     0.000     0.000         0
*MAT_SOFT_TISSUE_TITLE
Soft Tissue Model for leaflet
$#     mid        ro        c1        c2        c3        c4        c5
        10 1.3000E-9  0.040000     0.000  0.008000 18.830000&c51
$#      xk      xlam      fang     xlam0
 2200.0000&xlam        90.000000  1.000000
$#    aopt        ax        ay        az        bx        by        bz
     0.000     0.000     0.000     0.000     0.000     0.000     0.000
$#     la1       la2       la3      macf
     0.000     0.000     0.000         1
*PART
$# title
L-1.1
$#     pid     secid       mid     eosid      hgid      grav    adpopt      tmid
       102       102        10         0         0         0         0         0
```

```
*SECTION_SHELL_TITLE
L-1.1
$#   secid    elform      shrf       nip     propt   qr/irid     icomp     setyp
       102         2  1.000000         2         1         0         0         1
$#      t1        t2        t3        t4      nloc     marea      idof    edgset
  1.100000  1.100000  1.100000  1.100000     0.000     0.000     0.000         0
$
$ Further part definitions follow for different thickness parts of leaflet (omitted)
$
*PART
$# title
Ambient blood
$#     pid     secid       mid     eosid      hgid      grav    adpopt      tmid
       111        10         4         4         1         0         0         0
*SECTION_SOLID_ALE_TITLE
Ambient Blood
$#   secid    elform       aet
        10        11         1
$#    afac      bfac      cfac      dfac     start       end     aafac
     0.000     0.000     0.000     0.000     0.000     0.000     0.000
*MAT_NULL_TITLE
Ambient Blood
$#     mid        ro        pc        mu     terod     cerod        ym        pr
         4 1.0600E-9     0.000 3.4500E-9     0.000     0.000     0.000     0.000
*EOS_GRUNEISEN_TITLE
EOS for ambient blood
$Simplified EOS for blood - coefficients for water used here,
$but properties of blood in mat_null card
$P=GAMAO*E0=11.7kPa
$#   eosid         c        s1        s2        s3     gamao         a        e0
         4 1.4840E+6  1.979000     0.000     0.000  0.110000     0.000  0.102700
$#      v0
     0.000
*PART
$# title
Rigid Plane for contact
$#     pid     secid       mid     eosid      hgid      grav    adpopt      tmid
       112       112         8         0         0         0         0         0
*SECTION_SHELL_TITLE
Section for rigid plane for symmetry
$#   secid    elform      shrf       nip     propt   qr/irid     icomp     setyp
       112         2  0.833300         2         1         0         0         1
$#      t1        t2        t3        t4      nloc     marea      idof    edgset
     0.000     0.000     0.000     0.000     0.000     0.000     0.000         0
*MAT_RIGID_TITLE
Rigid material for contact_entity
$#     mid        ro         e        pr         n    couple         m     alias
         8 1.0000E-9  1.000000  0.300000     0.000     0.000     0.000
$#     cmo      con1      con2
  1.000000         7         7
$# lco or a1        a2        a3        v1        v2        v3
     0.000     0.000     0.000     0.000     0.000     0.000
*HOURGLASS_TITLE
Hourglass control for shell elements
$#    hgid       ihq        qm       ibq        q1        q2    qb/vdc        qw
         2         4  0.100000         0  1.500000  0.060000  0.100000  0.100000
*INITIAL_VOLUME_FRACTION_GEOMETRY
$Fills the outside of the HV with "void", leaving the inside filled with ALE fluid.
$#   fmsid   fmidtyp     bammg    ntrace
       111         1         0         0
$# conttyp   fillopt     fammg        vx        vy        yz    unused    unused
         1         1         3     0.000     0.000     0.000         0         0
$#     sid     stype   normdir   xoffset    unused    unused    unused    unused
         3         1         0     0.000         0         0         0         0
*DEFINE_COORDINATE_SYSTEM_TITLE
Csys
$#     cid        xo        yo        zo        xl        yl        zl
         2     0.000     0.000 -3.600000  0.500000  0.866026 -3.600000
$#      xp        yp        zp
```

```
 -4.430000 -2.558000 -3.600000
*DEFINE_CURVE_TITLE
Ventricular pressure scaled to E0
$ This curve defines ventricular pressure as a function of E0
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
         1         0  1.000000 9.0910E-6     0.000     0.000         0
$#                a1                  o1
               0.000        11300.000000
           0.0100000        12500.000000
           0.0200000        13600.000000
           0.0500000        15300.000000
           0.0800000        16000.000000
           0.1000000        15500.000000
$
$ Further pressure curves follow (omitted)
$
*SET_MULTI-MATERIAL_GROUP_LIST_TITLE
Fluids that interact with HV
$This is all the fluid MMGIDs, without the void and the LP region,
$which shouln't interact with the structure.
$#  ammsid
         1
$# ammgid1   ammgid2   ammgid3   ammgid4   ammgid5   ammgid6   ammgid7   ammgid8
         1         2         0         0         0         0         0         0
*SET_MULTI-MATERIAL_GROUP_LIST_TITLE
Void
$#  ammsid
         2
$# ammgid1   ammgid2   ammgid3   ammgid4   ammgid5   ammgid6   ammgid7   ammgid8
         3         0         0         0         0         0         0         0
*ALE_MULTI-MATERIAL_GROUP_PART
$MMID for high pressure inlet
$#     pid
        12
$
$ Further MMGIDs follow (omitted)
$
*CONSTRAINED_LAGRANGE_IN_SOLID_TITLE
$#  coupid                                                                 title
         1Coupling sinus
$#   slave    master     sstyp     mstyp     nquad     ctype     direc     mcoup
         3         3         1         0         3         4         2        -1
$#   start       end      pfac      fric    frcmin      norm   normtyp      damp
    0.0001.0000E+10  0.100000     0.000  0.500000         0         0         0
$#      cq      hmin      hmax     ileak     pleak   lcidpor     nvent  blockage
     0.000     0.000     0.000         0  0.010000         0         0         0
$#  iboxid   ipenchk   intforc   ialesof    lagmul    pfacmm      thkf
         0         0         0         0     0.000         0     0.000
$
$ Further CLIS cards follow (omitted)
$
*ELEMENT_SOLID
$
$ (Omitted) - these cards define the ALE mesh
*ELEMENT_SHELL_BETA
$
$ (Omitted) - Angles are specified in each element for fibre alignment. These
$ cards are for leaflet elements only
*ELEMENT_SHELL
$
$ (Omitted) - These cards are for the sinus and aorta
*NODE
$
$ (Omitted) - Node definitions
*END
```