Surrogate modelling of a patient-specific mathematical model of the left ventricle in diastole

Alan Lazarus

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Mathematics and Statistics
College of Science and Engineering
University of Glasgow

March 2022

Abstract

Personalised medicine is a relatively new area of healthcare that uses patient-specific data at multiple scales, and different scientific models, to inform disease prognosis and treatment planning. Recently, there has been particular interest in the translation of mathematical models to the clinical setting. These models are usually implemented in the form of a computer code that relates a set of model parameters with a set of observable quantities. Often these parameters have a physiological meaning, and their estimation can provide information about the level of function or dysfunction of a particular physiological process. An important example is in modelling the behaviour of the left ventricle (LV) in diastole. This model relates cardiac tissue properties (the parameters) with the kinematic behaviour of the LV that can be observed from cardiac magnetic resonance images. The personalisation of this model to different patients depends not only on the parameters, but also on the geometry of the LV, which varies from patient to patient. Improved representation of the LV geometry, combined with improved modelling capabilities, has led to increasingly accurate and personalisable models that can better replicate the real world process. This increased model fidelity is accompanied by increased computational costs, which hinders the application of these models in the clinical setting.

A natural solution to the problem posed by computational cost is to use statistical emulation. In emulation, we build a model that efficiently replicates the behaviour of the expensive simulator. Although conceptually a simple idea, the application of this methodology to mathematical models can be complicated. In the context of the LV model, this complexity is largely tied to the LV geometry. By its very principle, personalised medicine relies on the ability of the emulator to generalise to different LV geometries, meaning that the LV geometry itself must be treated as an input to the model. However, the high dimension of the LV geometry representation makes it incompatible with the statistical emulation framework. To resolve this issue, the work in this thesis uses a low-dimensional representation of the LV geometry to reduce the dimension of the input space of the model and construct a *generalisable emulator* of the LV model.

Of primary interest is the efficient estimation of the parameters of the LV model, in a time frame compatible with the clinical setting. For this purpose, the generalisable emulator allows for the efficient use of Markov chain Monte Carlo, providing a measure

of uncertainty in the parameters. A common problem in complex models, as is the case in the LV model, is the presence of weak practical identifiability. This manifests as large uncertainty in the posterior distributions of the parameters. In a Bayesian framework, this issue can be tackled using a more informative prior distribution. For the LV model, an informative prior that includes information from ex vivo studies is proposed, improving the estimation of the model parameters. Also motivated by the weak identifiability of the model, a new parameterisation of the model is considered. This involves a comprehensive sensitivity and inverse uncertainty quantification study that sheds extra light on the identifiability—both practical and structural—of the LV model. Finally, the problems posed by the measurement of clinical data, and the discrepancy between the model and reality, is considered and methods are proposed that account for this in the inference framework. Critically, the culmination of the work in this thesis highlights the problems that need to be resolved before the LV model can be applied in the clinical setting.

Contents

List of Tables

List of Figures

Acknowledgements

Firstly, I would like to thank my supervisors, Professor Dirk Husmeier and Dr Hao Gao, for their guidance and support. They were always there to offer me encouragement when work wasn't going to plan and to answer my questions on statistics and cardio-mechanics no matter how silly, and repetitive, they might have been (I feel like Hao might have suffered more in this regard!).

I was fortunate to be part of the Softmech group at Glasgow, which offered me the perfect environment to develop as a researcher and gave me the chance to collaborate on various different projects over the course of my PhD. While there are lots of people I could thank in this regard, a special thank you should go to Professor Xiaoyu Luo, Dr Umberto Noè, Dr Vinny Davies, Dr Agnieszka Borowska, Dr Lukasz Romaszko and David Dalton.

I would like to thank my examiners, Dr Vladislav Vyshemirsky and Professor Peter Challenor, for their suggestions on improvements that could be made to this thesis.

Life in Glasgow over the four years of my PhD was made much more enjoyable by the people I met and the friends I made, especially Craig, Marnie, Michael, Sam and Vinny. Thank you for making these four years more memorable.

My family have offered me their support over the course of my academic years and for this I will always be grateful. A special thank you must go to my sister, Heather, and brother-in-law, Marcus, who provided the best possible support in the weeks surrounding my viva. While the same could be said of my friends and the rest of my family, I really am not sure how I would have coped with everything had it not been for the two of them.

Lastly, I wish to thank GlaxoSmithKline and The University of Glasgow for funding my PhD project.

Declaration

With the exception of chapters 1, 2 and 3, which contain introductory material, all work in this thesis was carried out by the author unless otherwise explicitly stated.

Chapter 1

Introduction

The heart is often referred to as the most important organ in the body, as it is responsible for the pumping of blood that supplies oxygen and nutrients to the rest of the body's organs. Understanding of the heart's pump function dates back centuries and is regarded as the greatest medical discovery of all time (Friedland, 2009). In its early stages, cardiovascular research was primarily the work of physicians who strived, through ex vivo experiments, to understand the physiological behaviour of the heart. Over the last century, cardiovascular research has become more inter-disciplinary, the result of which has been a more quantitative understanding of the heart's function (Peirlinck et al., 2021). This is epitomised by the growing interest in descriptions of the mechanical behaviour of the heart (Niederer et al., 2019), which is essential for mathematical models of cardiac function. Mathematical models provide a causal link between the material behaviour of the heart and its kinematic behaviour, the latter of which is also observable—and measurable—given the cardiac imaging methods that are now routinely available. The combination of these two aspects can provide deeper insight into the heart's level of function. Importantly, they provide a better understanding of the tissue properties in vivo, with the potential to increase our understanding of the heart's physiology and pathophysiology.

Mathematical models of cardiac function are often constructed as partial differential equations that replicate the spatio-temporal evolution of the heart. These models can be validated against measurable data, with the validated models implemented in the form of a computer code or simulator that requires anatomical specifications of the heart at different scales. Accurate measurement of these features allows us to tailor cardiac models to a particular patient and provide patient-specific forward modelling of cardiac behaviour. Assuming the kinematic behaviour that is predicted by these models can also be measured in the clinical setting, the patient-specific forward model can be used for patient-specific inverse modelling, through which we recover the map from the observable to unobservable behaviour. A natural solution to the inverse problem is to use brute-force gradient-based optimisation routines: define a loss function that quantifies the squared difference between

model predictions and observed data, and follow the gradient of this loss through the input space to the minimum. Computation of this gradient (which is performed numerically) requires running the simulator, and this must be performed at each iteration of the optimisation. Despite the huge computational costs associated with such a procedure, these are still the most widely used methods in the literature today.

As scientific research becomes more quantitative, statistical and machine learning methods naturally become increasingly relevant for the synthesis of all the available information (Alber et al., 2019), with recent work showing their applicability in the field of cardiology (Quer et al., 2021; Hannun et al., 2019; Al'Aref et al., 2020). While we might think that the fields of mechanistic and data-driven modelling, which are fundamentally built upon contrasting views of model building, should be in competition, their uses in cardiovascular research are becoming more complementary. Recent implementations involve data-driven replication of mechanistic cardiac models (Maso Talou et al., 2020) and neural network modelling frameworks that explicitly incorporate the mechanistic descriptions in the model (Sahli Costabal et al., 2020). Meanwhile, classification has been used to show the informativeness of the mechanistic model's parameters (Gao et al., 2017).

Statistical emulation, which has been applied to other scientific disciplines ranging from epidemiology (Andrianakis et al., 2015) to climate modelling (Tran et al., 2016), is a method that allows for efficient approximation of an expensive simulator. Treating the mathematical descriptions as a black box, a statistical emulator learns to replicate the input to output map given a set of input-output pairs obtained from the model. This relies on data from the process-driven model, with which an emulator is trained in a data-driven manner. As such, it relies on the mechanistic description when obtaining the input-output pairs and can be seen to incorporate the mechanistic description into the model. This allows for efficient model calibration, where we learn the model settings of the simulator from a set of measured data (Kennedy and O'Hagan, 2001). Moreover, the improvement in computational costs allow for the consideration of *inverse uncertainty quantification* (I-UQ), where we learn a distribution on the parameters instead of a single point estimate. This is important in the context of cardio-mechanical models, where uncertainties exist in both the model and the measured data. A common approach taken to I-UQ involves quantifying our belief about the parameters both before and after observing the data. This *Bayesian* approach to inference allows us to incorporate information in our inference scheme that has been gained from previous experiments, which can be of particular use when dealing with mathematical models of cardiac processes since these have often been studied before in the lab. This idea of prior integration will be explored in this thesis.

The general idea in statistical emulation is the following: obtain a set of training data from a simulator and train a regression model to replicate the function, using information gained from these forward simulations. While this would appear simple, its application

in more complicated models can be difficult. Issues such as dimensionality of inputs and smoothness of the simulator function can make the application of emulation far from straightforward. This thesis will explore this extensively in the context of a mathematical model of cardiac function. This exploration will involve various different studies, first of all concerned with the task of representing the anatomical geometry in the model, which allows for generalisation to different patients, and then considering how we can improve the estimation of the model parameters, first through the introduction of a new prior distribution and then by learning a parameterisation based on a combined SA and I-UQ study. The layout of this thesis is as follows:

Chapter 2 provides background on the mathematical model of the left ventricle (LV) in diastole, which is the model for which statistical emulation and parameter inference will be considered.

Chapter 3 presents background on statistical methods that will be used in this thesis. This primarily considers the methods required for parameter inference, with a particular focus on Bayesian methods, as well as different regression models that can be used for statistical emulation.

Chapter 4 considers emulation of a simplified simulator, where the left ventricle geometry is fixed. This acts as a proof of concept study of statistical emulation for the mathematical model of the LV in diastole.

Chapter 5 considers dimension reduction of the LV geometry, which is necessary for generalisation of the statistical emulator to different LV geometries.

Chapter 6 uses the results from Chapter 5 in a first attempt at generalising the emulator to new LV geometries. This consists of extending the input space of the emulator from Chapter 3 with a low-dimensional LV geometry representation.

Chapter 7 attempts to tackle the issue of weakly identifiable model parameters. This chapter presents a prior model that takes account of results from ex vivo experiments to include information not available from clinical measurements.

Chapter 8 considers the parameterisation of the mathematical model. Using a combined sensitivity analysis and inverse uncertainty quantification study, a new parameterisation of the model is proposed.

Chapter 9 considers the impact that physiological conditions, such as LV geometry, have on the inverse estimation.

Chapter 10 provides a summary of the main results and discusses some future research directions.

Chapter 2

Cardiac mechanics model of the left ventricle

This chapter introduces the mathematical model that describes the left ventricle (LV) in diastole, which is the focus of the statistical analysis in the remainder of this thesis. This starts with an overview of the heart's physiology, before considering the LV in more detail. The mathematical description of the filling process is then introduced, before putting the work in this thesis in the context of patient specific modelling from the literature.

2.1 Left ventricle dynamics

2.1.1 Background on heart physiology

The heart consists of four chambers—the right and left atria and the right and left ventricles (see Figure 2.1)—that act in unison to drive the flow of blood in the circulatory system. This occurs in two phases: the systemic circulation maintains the supply of oxygen-rich blood to the distal organs via the systemic arteries, while the pulmonary circulation controls the provision of oxygen depleted blood to the lungs. As the main driving force behind these systems, the heart acts as a pump in alternating periods of relaxation (diastole), which allows for refilling, and contraction (systole), which pushes blood out of the heart. More specifically, during *atrial diastole* the right atrium (RA) collects deoxygenated blood once it has passed through the systemic circulation, and the left atrium (LA) collects oxygenated blood from the pulmonary circulation. The blood then passes from the LA into the LV and the RA into the right ventricle (RV) during *ventricular diastole*. Once the LV cavity has filled with blood passing in from the left atrium, the ventricular myocardium starts to contract, triggered by the action potentials propagated from the sinoatrial node. The pressure of the blood within the LV increases and once this surpasses the aortic pressure, the aortic valve opens and the blood is pumped into the

systemic circulation, where it passes round the body to all the distal organs. A similar process occurs in the RV to push blood out of the heart towards the lungs. In a healthy heart, the right and left sides act synchronously, meaning that left and right ventricular diastole and systole happen almost simultaneously. More details on the physiology of the entire heart can be found in the literature (Quarteroni et al., 2017; Voorhees and Han, 2015)

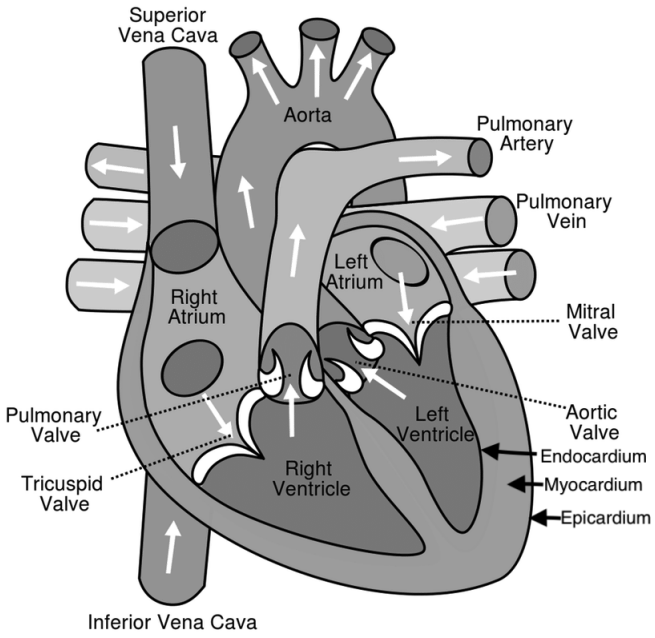


Figure 2.1: Diagram of the heart. The left and right atria and ventricles work together to drive blood flow around the body. Image taken from: `https://commons.wikimedia.org/wiki/File:Diagram_of_the_human_heart.svg`

2.1.2 More details on the left ventricle in diastole

This thesis is concerned with the behaviour of the LV in diastole, when the LV is passively filled with blood from the LA. The dynamics of the LV are largely driven by a muscular tissue—called the *myocardium*—that lies between two thin layers called the endocardium (the inner layer) and epicardium (the outer layer), as shown in Figure 2.1. In systole, when the LV is said to be *active*, the myocardium generates a contractive force that pushes blood out of the heart. During diastole, we refer to the myocardium as *passive*; having contracted to overcome the aortic pressure in systole, the muscle relaxes and permits the flow of blood from the LA. As the LV fills with blood, the blood within the cavity exerts a force on the myocardium, which responds via a resisting force (*stress*) to the deformation (*strain*) induced by the force of the blood. In general, this resistance force is nonlinear and prevents excessive expansion of the LV cavity. This can be understood in a similar manner to the stiffness properties in nonlinear springs: with increasing deformation, the resisting

force that pulls the spring back to its resting shape varies. In the LV, the resting state is referred to as the *unloaded state* and the properties of the myocardium that determine the level of resistance are referred to as the *myocardial properties*.

An understanding of the underlying myocardial properties is synonymous with an understanding of the stress and strain behaviour of the tissue. While the deformation (strain) can be measured in vivo, no methods currently exist for measuring myocardial stress. This stress is what is fundamentally of interest, since it has been linked to various physiological and pathological processes (Avazmohammadi et al., 2019; Palit et al., 2018a). While the stress is not measurable in vivo, the relationship between strain and stress can be described mathematically, in a form that is fundamentally the same as Hooke's law. However, the nonlinear response of the myocardium means that we require a more complex mathematical description, in what is referred to as a *strain-energy density function*. This describes the change in mechanical energy of the tissue during deformation (Voorhees and Han, 2015). Essentially, this calibrated strain energy function can be used to provide a measure of the stiffness of the myocardium material and is an important component of the mathematical model of the LV (Avazmohammadi et al., 2019).

2.2 A mathematical model of the left ventricle in diastole

The mathematical model of interest simulates the LV filling in diastole. A basic visualisation of the model is provided in Figure 2.2. Given an LV geometry, end diastolic pressure of the blood within the LV cavity (EDP) and a set of parameter values describing the myocardial properties (also referred to as the *material parameters*, *material properties* or *constitutive parameters*), the model predicts the deformed state of the LV, which is summarised using several quantities measured from the deformed mesh. In particular, three different types of strain can be extracted (see Figure 2.3), along with the volume of the cavity at end of diastole (LVV). The strain predictions are obtained at well-defined locations on the LV wall, and these same quantities, measured at the same location, can be obtained from cardiac magnetic resonance (CMR) scans of the LV. Matching the model predictions of the strains and LVV with the corresponding measured data formulates the inverse problem that will be explored in this thesis.

A technical description of the mathematical model requires a background in soft tissue mechanics, which is not within the scope of this thesis. Instead, a high level summary is provided, and the reader is referred to the extensive literature on these models for more mathematical details. In particular, Nordsletten et al. (2011) provides general background on mechanics before specifying this to the case of cardio mechanics. Voorhees and Han (2015) give a general and accessible introduction to cardiac biomechanics.

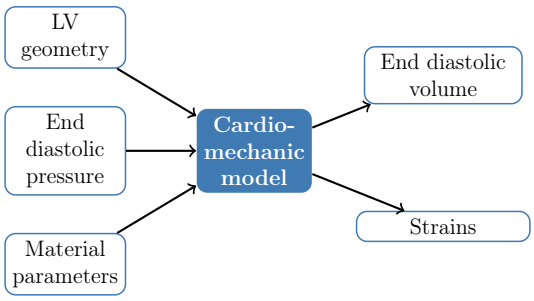


Figure 2.2: The LV simulator can be considered as a function from the joint material parameter-LV geometry-pressure space. Given a set of material parameters, EDP and LV geometry, the simulator inflates the LV geometry, simulating the passive filling process. From the deformed mesh, the strains and cavity volume at end of diastole can be extracted (see Figure 2.3 for details on the strain extraction).

2.2.1 Geometry representation

Simulation for a particular subject requires the LV geometry G, over which the mathematical model will be solved. The LV geometry can be extracted from CMR scans using a procedure that will be discussed in Section 5.2.1. This is discretised using hexahedral elements, forming a dense finite element mesh as shown in Figure 2.4. The LV geometry and its representation for cardio-mechanical modelling will be discussed in Chapter 5.

The myocardium is made up of a series of fibres known as myofibres, which consist of myocytes. Their orientation varies as we move from endocardium to epicardium (see the right of Figure 2.4, and correctly accounting for this feature is an important part of accurate cardio mechanic modelling (Wang et al., 2013a). Ideally, a patient-specific myofibre structure would be included in the model, but although methods are progressing in this direction (Toussaint et al., 2013; Khalique et al., 2020), it is not currently possible to acquire the myofibre structure in routine patient care. Instead, a rule based method (RBM) is used for describing the myofibre structure (Wang et al., 2013a). To do so, a local material coordinate system was defined, the so-called fibre ($\mathbf{m}$)–sheet ($\mathbf{s}$)– sheet-normal ($\mathbf{n}$) system, which is visualised in Figure 2.4. Unless otherwise specified, the simulations in this thesis assume the fibre angle varies linearly from $\alpha_{\text{endo}} = -60°$ at the endocardium to $\alpha_{\text{epi}} = 60°$ at the epicardium[1], with the sheet orientation along the transmural direction.

2.2.2 Constitutive Law of the Myocardium

The properties of the myocardium are governed by the intra- and extra-cellular structure of the tissue. The primary component of this tissue is the myofibres, which provide the intra-

[1]Note that the fibre angle in this study has the opposite sign compared to the literature convention in which α_{endo} is positive and α_{epi} is negative (Genet et al., 2014a). Using the opposite signs will not affect the generated fibre fields since we use the opposite circumferential direction for defining the fibre angle that is the angle between the fibre vector and the circumferential direction. The generated fibre vectors have also been validated with literature reported fields in previous studies (Gao et al., 2014).

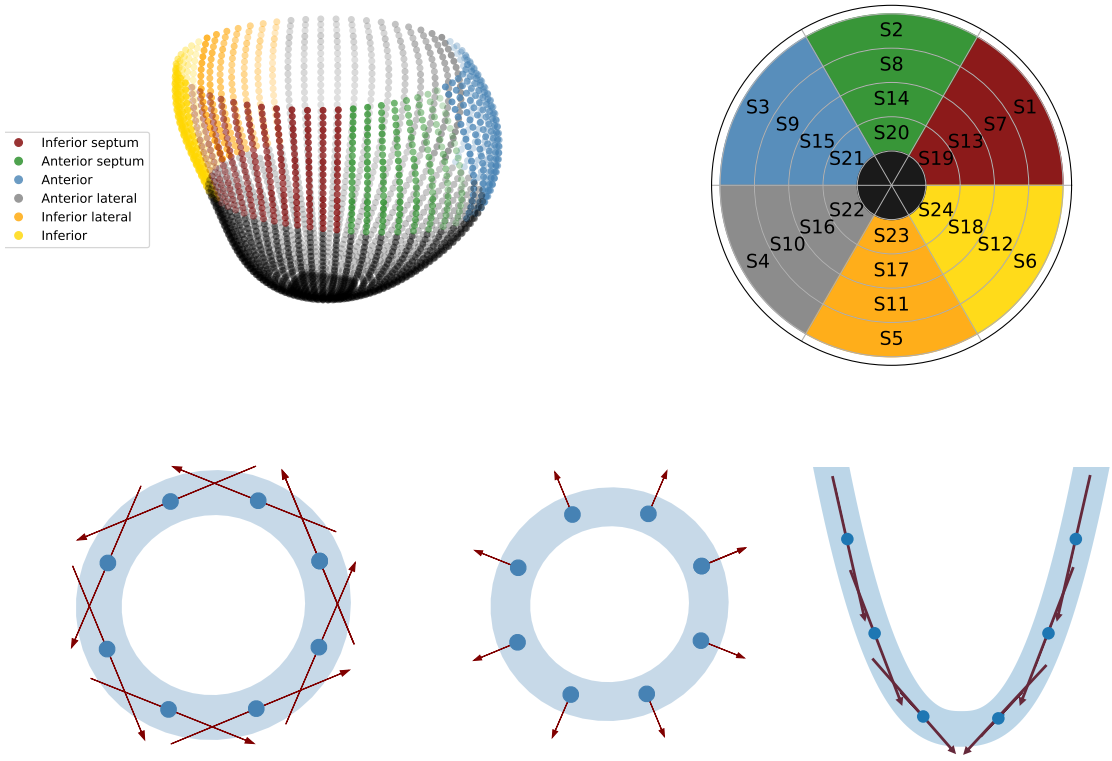


Figure 2.3: The simulator inflates the LV from early diastole, simulating the LV passive filling process to produce a deformed mesh. To extract the simulated strains from the deformed LV mesh, the clinical convention is used following the AHA division. This involves dividing the LV circumferentially into 6 segments in the circumferential direction (top left) and 4 segments in the longitudinal direction based on the CMR images from base to mid-ventricle. In total, this gives 24 strains regions/locations, indicated by the plot in the top right. Since these strains are at well defined locations, corresponding strains can also be measured from CMR scans. Three different strains can be extracted at each strain region, as shown in the bottom row. From left to right, these plots indicate how each type of strain is measured along the circumferential, radial and longitudinal directions. In simple terms, strain represents the ratio of length change of a line segment after deformation.

cellular structure and are surrounded by the extracellular matrix (ECM) (Avazmohammadi et al., 2019). A component of the ECM that is of particular relevance is collagen, which is a fibrous material. The myofibres and collagen play complementary roles in governing the response of the myocardium. In particular, small-stretch behaviour is governed by the myofibres, with collagen having very little effect. At higher stretch the collagen kicks in, interacting with the myofibres and governing the high stretch mechanical behaviour of the muscle (Voorhees and Han, 2015). These two features, among many others, result in a complex stress-strain relationship that can be captured by a strain energy density function (Holzapfel and Ogden, 2009).

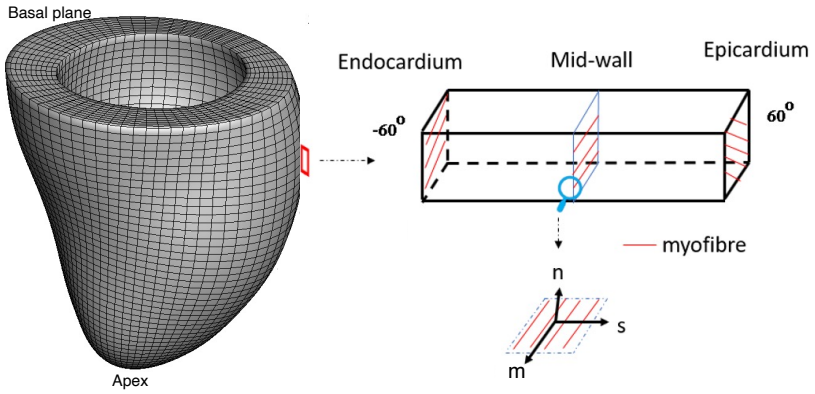


Figure 2.4: An example hexahedral mesh representation of the LV geometry that is used in the simulator. The diagram on the right shows the rotation of the myofibres from epicardium to endocardium, with the axis in the bottom right showing the myofibre (**m**), sheet (**s**) and sheet normal (**n**) directions.

In this study, the passive myocardium is considered as an incompressible (no change in volume), anisotropic (stress response is direction dependent) and hyperelastic (no energy dissipation and the stress–strain relationship can be described by a strain energy density function) material, described by the constitutive law introduced by Holzapfel and Odgen (Holzapfel and Ogden, 2009). The so-called H-O model is given by:

$$\begin{aligned}\boldsymbol{\Psi} = &\frac{a}{2b}\left[\exp\left\{b\left(I_1-3\right)\right\}-1\right] \\ &+\sum_{i\in\{\mathrm{f,s}\}}\frac{a_i}{2b_i}\left[\exp\left\{b_i\left(\max(I_{4i},1)-1\right)^2\right\}-1\right] \\ &+\frac{a_{\mathrm{fs}}}{2b_{\mathrm{fs}}}\left\{\exp\left(b_{\mathrm{fs}}I_{8\mathrm{fs}}^2\right)-1\right\}\end{aligned} \tag{2.1}$$

in which a, b, a_{s}, b_{s}, a_{f}, b_{f}, a_{fs}, b_{fs} are the material parameters. The term depending on a and b relates to the isotropic response of the myocardium, which we can think of as a global basis material property. The remaining terms relate to the anisotropic response of the material, which is best understood with reference to Figure 2.4: a_{f} and b_{f} characterise the reinforced stiffness along the myofibres (in the direction of **m**); a_{s} and b_{s} characterise the reinforced stiffness along the sheet direction (in the direction of **s**) and a_{fs}, b_{fs} describe the shear response between the fibre and sheet directions. The max() function in (8.1) ensures the fibres can only support extension but not compression. The remaining terms in the H-O model, denoted I_j, are called invariants. I_1 is the first invariant of the right Cauchy-Green deformation tensor $\mathbf{C} = \mathbf{F}^{\top}\mathbf{F}$, with $\mathbf{F}$ the deformation gradient, and I_{4i}

and $I_{8\text{fs}}$ are structure-based invariants that account for myocardial anisotropy and shear properties. Specifically:

$$I_1 = \text{tr}(\mathbf{C}), \qquad I_{4\text{f}} = \mathbf{m}_0 \cdot (\mathbf{C}\mathbf{m}_0)\,,$$
$$I_{4\text{s}} = \mathbf{s}_0 \cdot (\mathbf{C}\mathbf{s}_0)\,, \quad I_{8\text{fs}} = \mathbf{m}_0 \cdot (\mathbf{C}\mathbf{s}_0)\,,$$

where $\mathbf{m}_0$ and $\mathbf{s}_0$ are the unit fibre and sheet directions in the reference configuration. Interested readers are referred to Holzapfel and Ogden (2009) for a detailed description of the H-O model. A general introduction to soft tissue mechanics can be found in Holzapfel (2000).

2.2.3 Simulation of LV Passive Filling

The LV diastolic filling process is described by a quasi-static pressure-loaded boundary-value problem over the computational domain (Ω) occupied by the LV geometry. A linearly ramped pressure from 0 to P mmHg is applied to the endocardial surface, where P is the end diastolic blood pressure inside the LV cavity. The basal plane (see Figure 2.4) is fixed along the longitudinal and circumferential directions, with only expansion in the radial direction allowed. The equilibrium equation at the current configuration (Ω_t) is given by:

$$\begin{cases} \nabla \cdot (\boldsymbol{\tau}) = 0 & \text{in } \Omega_t, \\ \boldsymbol{\tau} \cdot \mathbf{n} = \mathbf{t} & \text{in } \Gamma^{\text{endo}}, \\ \mathbf{u} = \mathbf{u}_0 & \text{in } \Gamma^{\text{base}}, \end{cases} \tag{2.2}$$

where $\mathbf{n}$ is the normal direction of the endocardial surface Γ^{endo}, $\mathbf{t}$ is the traction force resulting from the loaded pressure at Γ^{endo}, and $\mathbf{u}_0$ is the prescribed displacements at the basal surface Γ^{base}. The base (otherwise known as the basal plane) is identified in Figure 2.4 while the endocardium can be understood via Figure 2.1. The myocardial Cauchy stress ($\boldsymbol{\tau}$) is derived from the H-O model as

$$\boldsymbol{\tau} = \mathbf{F} \sum_{i=1,4\text{f},4\text{s},8\text{fs}} \frac{\partial \boldsymbol{\Psi}}{\partial I_i} \frac{\partial I_i}{\partial \mathbf{F}} - \lambda \mathbf{I}, \tag{2.3}$$

in which $\mathbf{I}$ is the identity matrix, and λ is the Lagrange multiplier to enforce the incompressibility. This expression makes explicit the statement made earlier, namely that the strain energy function governs the stress-strain relationship of the material.

For the work in this thesis, equation (2.2) was solved using the general-purpose finite-element package ABAQUS (Simulia, Providence, RI, USA).

2.3 The need for patient-specific inverse estimation

Currently, there is a drive towards the translation of cardiac modelling to the clinical setting, particularly for the purpose of personalised healthcare (Chabiniok et al., 2016; Peirlinck et al., 2021). For this purpose, calibrated cardio-mechanic models have been shown to provide information (Krishnamurthy et al., 2013a; Wang et al., 2013a; Asner et al., 2016; Wang et al., 2020), as well as clinical biomarkers (Hadjicharalambous et al., 2015; Gao et al., 2017), of cardiac diseases. Successful implementation of these models in healthcare is dependent on two aspects of ongoing research. Firstly, we require that the models are increasingly personalised to account for increasing scales of patient-specific information. This has led to the study of patient-specific parameter estimation (Asner et al., 2016; Gao et al., 2015), fibre angle representation (Guan et al., 2021) and load-free geometry approximation (Krishnamurthy et al., 2013a). A second aspect, which partially intersects the idea of patient-specific modelling, is in the use of statistics or machine learning techniques that allow for efficient, effective, use of these increasingly accurate mathematical models. For instance, Wang et al. (2020) used PCA in a procedure that inferred the unloaded state of the LV geometry, while Cai et al. (2021); Dabiri et al. (2019) used different machine learning and statistical models for replication of an expensive cardio-mechanic model. Critically, we rely on progress being made in the intersection of personalisation and modelling efficiency before cardio-mechanic modelling can be considered as a decision support mechanism in patient-specific healthcare.

This thesis will consider two areas of patient-specific modelling. The first is in the representation of the LV geometry, and the second is the use of statistical inference schemes for estimation of the material parameters of the H-O model. Both of these ideas have been considered in the literature in isolation, but no studies have tackled a combination of them both. In all chapters, the LV geometry at early diastole is assumed to be the load-free state. This is a very commonly adopted assumption in the literature, due to the difficulty in jointly identifying both the passive parameters and the ground truth load-free state (Asner et al., 2016). A prevailing interest throughout the thesis will be the representation of the LV geometry for efficient patient-specific modelling. Generally, there are two ways we can approach this problem: low order LV geometry representation or dimension reduction. The former method was adopted by Hong et al. (2019), where they represented the LV with a series of deformation modes, reducing the computational cost of the finite element method. On the other hand, dimension reduction has been quite widely considered for the task of disease analysis (Suinesiaputra et al., 2018; Farrar et al., 2016), while Maso Talou et al. (2020) introduced PCA to reduce the dimension of the LV geometry and allow for patient-specific prediction of the displacement field of the LV in diastole. Although this second approach does not reduce the computational cost of the cardio-mechanical simulations, the simplification of the LV geometry representation allows

for the use of surrogate models to replicate the expensive simulator. This idea will be the basis of the methods used in this thesis.

Chapter 3

Statistical background

The main aim of this thesis is to explore the use of surrogate modelling for a mathematical model of the left ventricle (LV) in diastole. This involves the inclusion of LV geometry information in the input space, and the construction of a regression model that allows for efficient approximation of the expensive mathematical model. The purpose of the present chapter is to outline different methodologies from the literature that will be required in the work carried out in this thesis.

3.1 Statistical emulation

3.1.1 Simulators and emulators

The aim of modelling is to replicate the behaviour of a real world system. *Mechanistic* models, which are often associated with the field of mathematical modelling, depend on an understanding, or assumption, of how the real-world system behaves. The modeller constructs a proxy for reality by expressing, mathematically, the different causal mechanisms within the system and how they impact the observed behaviour (Baker et al., 2018). The causal nature of these models makes them perfect for analysis of the real world: if we validate the model on several real world datasets then we can be confident of the model's ability to generalise to unseen cases. Moreover, the parameters of these models often have a physical interpretation, meaning their estimation provides us with further insights. Of interest in this thesis is a mechanistic model expressed in the form of a computer code, or *simulator*. Given a set of inputs, the simulator, ζ, provides a prediction, $\boldsymbol{y}$, of how the system will behave:

$$\boldsymbol{y} = \boldsymbol{\zeta}(\boldsymbol{\theta}, \mathbf{z}), \tag{3.1}$$

where $\boldsymbol{\theta}$ is a set of parameters of the model and $\mathbf{z}$ is a set of additional model conditions that are required to run the simulator. An example of such a model is the Lotka-Volterra

model of predator and prey populations where $\boldsymbol{y}$ would be the populations of the predator and prey, $\boldsymbol{\theta}$ would be the four parameters of the model and $\mathbf{z}$ would be the time over which the system is solved. Associated with prediction from the simulator is a computational cost, which can be large for complex systems.

An alternative approach to modelling a real world system, built on methods from statistics and machine learning, is *data-driven* modelling. At its core, data-driven modelling can completely ignore the underlying behaviour of the system. Rather than explicitly encoding the laws of nature into the model, relationships are captured by learning to replicate the patterns in the observed data. In the Lotka-Volterra example, we could train a regression model on measured lynx-hare populations observed over time t. Such a model would allow us to predict, at an unseen time point, the population of the lynx and hare, but it would not allow us to predict the population for a separate system of animals. Moreover, the parameters of the model, which are just standard regression parameters, would have little physical meaning. This means that little information about the real world—aside from prediction—can be elicited from the trained model. One benefit, however, is that these models are generally more computationally efficient than their mechanistic counterparts.

The ideas of mechanistic and data-driven modelling can be combined to efficiently link the underlying behaviour of a system with its observable behaviour. In *emulation* (Gramacy, 2020; Currin et al., 1988; Conti et al., 2009; Conti and O'Hagan, 2010; Currin et al., 1991), we train a data-driven model to replicate a simulator. For the simulator in (3.1), we would generate training data by evaluating the simulator over a range of different parameter values for a fixed set of model conditions and train a regression model (the emulator), $\eta(\cdot)$, to emulate the input to output map. While the mechanisms of the regression model do not capture the causal structure of the real world system, the emulator can be used to predict in the same way as the simulator. Moreover, given a new dataset we can efficiently learn the inputs, otherwise known as *model calibration* (Kennedy and O'Hagan, 2001), and gain information about the system in question. The black-box nature of data-driven modelling does introduce extrapolative limitations, both in the space of parameters and model conditions. Essentially, this means that prediction at locations of parameters or model conditions outwith those present in the training set can be less accurate. For this reason, we have to be careful when generating the training data, introducing us to the area of statistical research known as design of computer experiments.

3.1.2 Design of computer experiments

When we attempt to approximate continuous functions with only a finite set of data points, uncertainties and errors are inevitable. As a result, one of our aims during emulator construction is to minimise these uncertainties by encoding in the emulator all our knowledge about the simulator function. This should involve asking questions like: do all the inputs

have equal effect? (see Section 3.1.3) and does the complexity of the function change as we vary the inputs? (ie is the function nonstationary?). One of the obvious tools to deal with such issues is in the surrogate model itself, by building into our model the features that we know are present in the simulator function. An equally important—and related— method lies in the design of computer experiments, which is concerned with the positioning of our training locations in input space. If we want to make efficient use of a more specialised model, then we should also account for the simulator behaviour in the design. The reverse is also true: to maximise the information gained by a design of computer experiments that accounts for function variation, it must be paired with a statistical model that has the capacity to capture the behaviours in the data.

To concern ourselves with design of computer experiments is to concern ourselves with the efficiency of emulator construction. If efficiency is not a concern, and we can afford to densely fill the configuration space, then the choice of input locations will have little effect on the fitted function. However, if we want to maximise the efficiency of model training, then this can be expressed as an optimal design problem where the objective is to maximise the information gained by the simulator runs.

Generally speaking, two strategies for design of computer experiments exist: sequential and space filling designs. Sequential designs, using methods like Active Learning Mackay (MacKay, 1992b) or mutual information based approaches (Beck and Guillas, 2016), allow us to gradually select the location of our training points based on information we have gained about the simulator function. An overview of sequential design strategies, which is outwith the scope of this thesis, can be found in the literature (Gramacy, 2020, Chapter 6; Santner et al., 2018, Chapter 6). The work in this thesis will consider space filling designs (Gramacy, 2020, Chapter 4; Santner et al., 2018, Chapter 5) and their adaptations to the context of the LV simulator from Section 2.2.

Quasi-random sequences

Intuitively, a space filing design corresponds to uniform coverage of the space. Naturally, one might assume a pseudo random generator, using a function like *runif* in R, would be appropriate for this task. However, this can lead to clusters of points as shown in the right-hand side of Figure 3.1, which will certainly not maximise the information gained from our simulator runs. The problem here is exactly the point of pseudo-random number generators: given the first n points, the $(n+1)$th is selected randomly, with no conditioning on the existing design. To overcome this problem, quasi-random sequences (also referred to as low-discrepancy sequences due to their ability to efficiently fill space) (McClarren, 2018, Section 7.3), generate sequences of seemingly random numbers, subject to some property of point dispersion. We can consider this as introducing a bias: rather than sampling the $(n+1)$th point independently, we sample point $n+1$ conditional on the first

n points subject to some space filling criterion. The result, as shown in the left-hand side of Figure 3.1, is a sequence that more efficiently fills the configuration space. In Figure 3.1, a Sobol sequence—which was originally proposed for solving high dimensional integrals (Sobol, 1967)— was used to generate the sequence, but many other methods such as a Halton sequence or Hammersley sequence could be used. For more details see McClarren (2018, Section 7.3) and Santner et al. (2018, Section 5.6.1). One of the advantages of quasi-random sequences is that they can be extended: if we obtain a sequence of 100 points, train an emulator and realise more simulations are required for a desired level of accuracy then we can continue the sequence to obtain the 101st point, subject to the same space filling criteria.

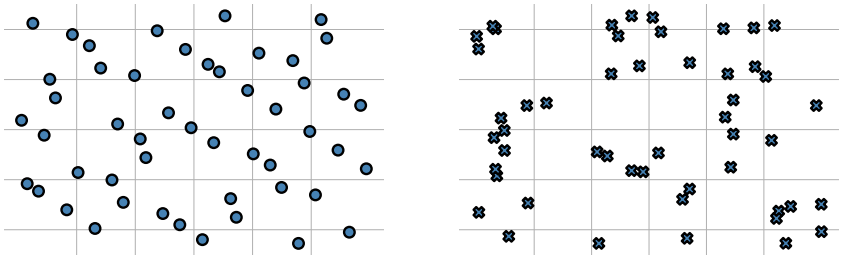


Figure 3.1: Comparison of quasi random (left) and pseudorandom (right) sequences. Both are generated with 100 points.

Latin hypercube sampling designs

Quasi-random sequences are not a perfect solution to the design of experiments problem. It is well documented that the space filling criterion they use leads to strange behaviour in higher dimensions, as shown in Figure 3.2. This is commonly referred to as the poor projection properties of the sequence (Santner et al., 2018).

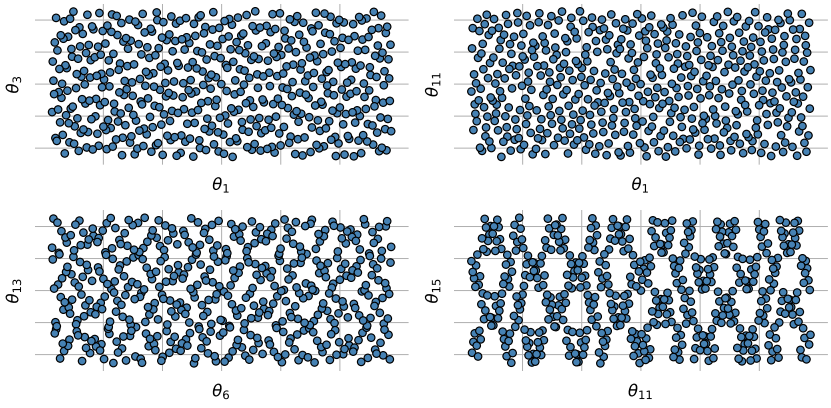


Figure 3.2: In low dimensions, the Sobol sequence fills the space uniformly but patterns begin to appear in higher dimensions.

An alternative to quasi-random sequences are Latin hypercube designs (LHD), which

use a combination of stratified sampling and pseudo random number generation in order to fill a space efficiently (Dette and Pepelyshev, 2010;Gramacy, 2020, Chapter 4; Santner et al., 2018, Section 5.2.2). The samples obtained using LHD tend to have better space filling properties than Sobol sequences, but they have less variation in interpoint distances. In the literature, it has been shown that this variation can be beneficial in some modelling applications (Zhang et al., 2019). Generally, the LHD provides better designs in higher dimensions and we see in Figure 3.3 that the projections no longer show any patterns. One issue with this approach is that we lose the natural ability to continue the design if we decide we need more points and have to rely on methods like maximin sampling instead.

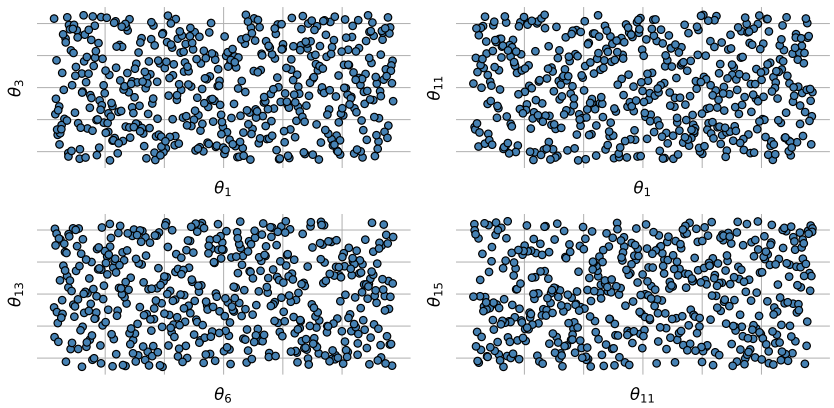


Figure 3.3: The projection properties of a LHD are far better than those of a Sobol sequence.

3.1.3 Sensitivity Analysis

Often, computer codes can be complex and heavily parameterised. For the purpose of statistical emulation, these high dimensional input spaces can be problematic for reasons of design and calibration. In Figure 3.2, we saw that patterns begin to appear in a Sobol sequence in higher dimensions. Therefore, in high dimensional problems it would make sense to order the inputs by importance in the Sobol sequence input space, so that important inputs exhibit good projection properties and the joint space is filled densely. SA allows for this by providing a quantitative measure of an input's importance in the function.

In what follows, *sensitivity analysis* (SA) is referred to in the sense of *global* sensitivity analysis. For detail on local sensitivity analysis, see McClarren (2018, Part II). Let us consider an arbitrary function f that maps a set of D input parameter values $\theta_1, \theta_2, ...\theta_D$ to associated output quantities of interest, $y_1, y_2, ...y_M$:

$$f(\theta_1, \theta_2, ...\theta_D) = (y_1, y_2, ...y_M) \tag{3.2}$$

By performing SA on this forward model, we can quantify the influence that the uncertainty in each random input parameter θ_i has on the observed variation in any chosen output quantity y_j. Using the approach of Sobol (2001), SA of the forward model f can be performed by computing the *first-order* and *total-effect* Sobol sensitivity indices of the random input parameters. Below, a brief overview of this approach to SA is given. For a comprehensive overview, the reader is directed to Saltelli et al. (2010) and Gramacy (2020, Chapter 8).

SA begins with a selection of the inputs and outputs of interest, along with an assumed distribution for the input quantities. This distribution is often referred to as an *uncertainty distribution* (Gramacy, 2020, Chapter 8). The first-order sensitivity index, denoted S_{ji}, of input parameter θ_i for output y_j is defined as:

$$S_{ji} = \frac{\mathbb{V}_{\theta_i}\left[\mathbb{E}_{\boldsymbol{\theta}_{\sim i}}\left[y_j \mid \theta_i\right]\right]}{\mathbb{V}[y_j]} \tag{3.3}$$

where $\mathbb{E}$ and $\mathbb{V}$ are the expectation and variance operators respectively. The expectation term in the numerator of (8.4) is called the main-effect function. It returns the expected value of the output y_j given a value of the i^{th} input θ_i, after all other inputs have been integrated out. The variance of the main-effect function is then taken, providing a scalar summary of its variation with respect to θ_i. This result is then standardised with respect to the unconditional output variance, allowing the index to be interpreted as the fraction of the total observed variance in y_j attributable to varying θ_i alone.

The first-order sensitivity index does not account for interaction effects between different input parameters, which can potentially be significant for nonlinear models. For this purpose, we can also consider the total-effect index, T_{ji}, for each input θ_i, which is defined with respect to output y_j as:

$$\begin{aligned} T_{ji} &= \frac{\mathbb{V}[y_j] - \mathbb{V}_{\boldsymbol{\theta}_{\sim i}}\left[\mathbb{E}_{\theta_i}\left[y_j \mid \boldsymbol{\theta}_{\sim i}\right]\right]}{\mathbb{V}[y_j]} \\ &= \frac{\mathbb{E}_{\boldsymbol{\theta}_{\sim i}}\left[\mathbb{V}_{\theta_i}\left[Y_j \mid \boldsymbol{\theta}_{\sim i}\right]\right]}{\mathbb{V}[y_j]} \end{aligned} \tag{3.4}$$

where $\boldsymbol{\theta}_{\sim i}$ is the vector of all random inputs except for θ_i. The numerator of the total-effect index gives the difference between the unconditional output variance, and the variance observed in the output once θ_i has been accounted for. This is again standardised with respect to the unconditional output variance, meaning that the total-effect index represents the proportion of observed variance in the output that is due to θ_i, including all possible interaction effects with other input parameters.

The first-order and total-effect indices must usually be calculated numerically, using evaluations from the forward model. While efficient low-discrepancy Monte-Carlo sampling

schemes have been proposed for performing these calculations (Saltelli et al., 2010), the computational expense of each forward evaluation of the cardiac-mechanic model makes this approach impractical for expensive simulators. For this reason, the use of surrogate models has been proposed in the literature (see Melis et al. (2017); Rodriguez-Cantano et al. (2018); Gramacy (2020, Chapter 8)).

The results of an SA can be used to influence our modelling procedure. Strongly influential parameters can be prioritised for measurement or estimation from clinical data, while weakly influential parameters can be set to fixed values within their range of uncertainty. It might seem strange that we use an emulator to perform a sensitivity study to then build another emulator. For starters, if we can build an emulator accurate enough for the sensitivity study then can we not also build an emulator accurate enough for calibration and prediction purposes? The answer to this is that, while SA is based on expected values and is more concerned with the global shape of the function, calibration and prediction are more reliant on accurate approximations at a point level. In other words, accuracy is more critical for calibration than SA, and SA can be seen more as a coarse-grained prescreening step. In high dimensional cases with expensive simulators, it will be more computationally efficient to build a less accurate emulator in high dimensions, perform an SA and then reduce the input dimension to construct an accurate emulator over the inputs that matter for function prediction, or to inform the design of the emulator in the ways discussed in this section.

3.2 Uncertainty quantification

In *model calibration*, we are interested in inferring the parameters of a model for a new set of data. Usually, we are interested in model calibration based on real data. That is, given a set of measurements, $\boldsymbol{y}$, we wish to gain some interpretation of the underlying behaviour of the system. The way in which we arrive at our inference can differ philosophically, but the common goal of learning from the data requires a relationship with the parameters $\boldsymbol{\theta}$. This relationship can be expressed through a noise model that tells us how the data were generated from the parameters. The noise model is expressed as a probability distribution, associated with which is a probability density function, $p(\boldsymbol{y}|\boldsymbol{\theta})$, that assigns a probability to different possible values of $\boldsymbol{y}$ for fixed $\boldsymbol{\theta}$. The task in statistical inference is to invert the parameter→data relationship and recover, based on the data, the underlying parameters (Robert, 1994). This goal is epitomised in the *likelihood function*, which is a reframing of the noise model as a function of the parameters, expressing the evidence about $\boldsymbol{\theta}$ obtained from the data. Although itself not a proper probability density function, the likelihood function is calculated using the density function of the associated noise model (but now considered as a function of $\boldsymbol{\theta}$). This is usually denoted as $L(\boldsymbol{\theta}|\boldsymbol{y})$ or $p(\boldsymbol{y}|\boldsymbol{\theta})$, the latter of

which will be adopted in this thesis.

3.2.1 Frequentist methods

In a frequentist inference framework, we consider the data themselves as random variables, realisations of which can be drawn from an underlying population, while the parameters are considered fixed without any probabilistic treatment. From these realisations, we wish to obtain our approximation of the population parameters, which is done by forming an *estimator*. This estimator is itself a function of the data, and it could be evaluated for any hypothetical realisations of the data that we were to obtain, giving an *estimate* of the parameters of interest corresponding to that particular set of realisations: $\hat{\theta} = \delta(\boldsymbol{y})$. Uncertainty in our estimator naturally follows from these conceptualised data realisations: if we repeatedly sample from the data distribution and re-evaluate our estimator, a *sampling distribution* is induced. Thus, uncertainty in the estimator is a result of the randomness in the data.

While the estimator used for an inference task is entirely optional, different estimators can be shown to be more optimal (in some sense) than others. From a decision theoretic viewpoint, this optimality can be quantified via the risk, $R(\theta^*, \delta) = \mathbb{E}_{\mathcal{D}}[L(\theta^*, \delta(\mathcal{D}))]$, where L is some loss function. An example is the mean squared error loss function, where the risk naturally decomposes into the bias-variance trade off:

$$\mathbb{E}_{\boldsymbol{y}}[(\hat{\theta} - \theta^*)^2] = \mathbb{E}[(\hat{\theta} - \bar{\theta}) - (\bar{\theta} - \theta^*)^2] = \mathbb{E}[(\hat{\theta} - \bar{\theta})^2] + (\bar{\theta} - \theta^*)^2 \tag{3.5}$$

where $\bar{\theta} = \mathbb{E}(\hat{\theta})$. This motivates a minimum variance unbiased estimator (MVUE). Conceptually, estimators should be close to the ground truth value of the parameters we are estimating, which is formalised in a notion of *unbiasedness*. This is an intuitive property, but we should also consider the variance of the estimator, which for unbiased estimators is lower bounded by the Cramer-Rao lower bound. The MLE can be shown to asymptotically achieve the CRLB, and so has the smallest asymptotic variance among unbiased estimators. Although this thesis will generally take a Bayesian approach to inference, optimisation of the mean squared error will be used at various times. Under an assumption of iid Gaussian noise, optimisation of the MSE is equivalent to maximum likelihood estimation, which is a commonly used estimator.

Conceptually, frequentist UQ involves repeated sampling from the underlying data distribution and re-evaluation of the estimator. If this procedure is repeated *ad infinitum*, the induced distribution of the estimates approaches the sampling distribution. In reality, we never know the true underlying data distribution and we instead have to come up with alternative approaches that effectively replicate this process. One procedure to do so is called bootstrapping, where we form some approximation to the underlying data distribu-

tion in order to repeatedly sample synthetic datasets. The way in which the resampling is performed is dictated by the method adopted. In the non-parametric bootstrap this is based on the empirical distribution of the data (we draw with replacement from the dataset), and in the parametric case we assume a parametric form for the underlying data distribution and, having determined our estimates based on a current sample, use these to generate residuals that can be shuffled and added to the model prediction to generate a new dataset. A more detailed look at the bootstrap can be found in the literature (Wakefield, 2013, Section 2.7; Murphy, 2012, Section 6.2.1).

3.2.2 Bayesian inference

Bayesian inference (Gelman et al., 2013; Robert, 1994) differs from frequentist statistics in its treatment of the parameters. Treating a fixed but unknown parameter value probabilistically, the aim is to quantify our level of belief of the underlying parameter. Initially, this is expressed through a *prior distribution*, $\pi(\boldsymbol{\theta})$. Similar to frequentist inference, the probabilistic nature of the data is represented by the distribution, $p(\boldsymbol{y}|\boldsymbol{\theta})$. The product of the prior and the data distribution gives the joint probability distribution of the data and the parameters before seeing the data. The aim in Bayesian inference, however, is to represent our inferences *a posteriori* when we no longer think of the data as random. To do so, we wish to *condition* our inferences about $\boldsymbol{\theta}$ on the observed $\boldsymbol{y}$, which is done by normalising the joint distribution according to Bayes theorem to obtain the posterior distribution:

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\int p(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}}. \tag{3.6}$$

In special cases, where the prior is said to be *conjugate*, the posterior has the same functional form as the prior and the update can be performed in closed form.

Based on the posterior we can perform other inferences. Due to the probabilistic treatment of the parameters, these resulting calculations are typically expressed as integrals:

$$\mathbb{E}_{\boldsymbol{\theta}}[f(\boldsymbol{\theta})] = \int f(\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{y})d\boldsymbol{\theta}. \tag{3.7}$$

for some suitable function f. For instance, if we are interested in the posterior predictive distribution, $p(\tilde{\boldsymbol{y}}|\boldsymbol{y})$, then we use $f(\boldsymbol{\theta}) = p(\tilde{\boldsymbol{y}}|\boldsymbol{\theta})$. In all but the simplest of cases, these integrals are not available in closed form and we can replace integration with summation through Monte Carlo estimation. Here, we sample from $p(\boldsymbol{\theta}|\boldsymbol{y})$ (see Robert and Casella (2005, Chapter 2) for some standard techniques) and approximate (3.7) with the following

sum:

$$\mathbb{E}_{\boldsymbol{\theta}}[f(\boldsymbol{\theta})] \approx \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{\theta}_i), \quad \boldsymbol{\theta}_i \sim p(\boldsymbol{\theta}|\boldsymbol{y}) \tag{3.8}$$

which converges almost surely to (3.7) by the strong law of large numbers (Robert and Casella, 2005, Section 3.2). Essentially, we can summarise a posterior distribution and perform any proceeding inferences using samples obtained from that distribution. Several methods exist for obtaining these samples in situations where the posterior density can be evaluated. For more details see (Robert and Casella, 2005, Chapter 3).

In many practical uses of Bayesian inference it is the calculation of (3.6) itself that is the obstacle, meaning that we cannot calculate the posterior in closed form never mind sample from it. The problem is the calculation of the marginal likelihood in the denominator of (3.6). Fortunately, several methods exist that allow us to bypass the direct calculation of this integral. The one that is most common and will be considered in this thesis is Markov chain Monte Carlo (Brooks et al., 2011).

3.2.3 Markov chain Monte Carlo

MCMC methods perform a series of moves through the support of a target distribution using a tractable proposal distribution that depends only on the previously seen point. The collection of random variables $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n)}$ produced using such a procedure is called a *Markov chain* because the conditional probability distribution of $\theta^{(n+1)}$ depends only on $\theta^{(n)}$ (the Markov property). This conditional probability distribution is governed by a Markov kernel, $K(\theta^{(n)}, \theta^{(n+1)})$ and we say that a Markov chain (on a continuous state space) is *stationary*, with stationary distribution π, if the following relationship holds:

$$\pi(\theta^{(n+1)}) = \int p(\theta^{(n+1)}|\theta^{(n)})\pi(\theta^{(n)})d\theta^{(n)} = \int K(\theta^{(n)}, \theta^{(n+1)})\pi(\theta^{(n)})d\theta^{(n)}. \tag{3.9}$$

Intuitively, this means that were we to sample an initial point from the stationary distribution and update this point using the Markov kernel associated with the Markov chain, the updated point is guaranteed to come from the stationary distribution. Assurance of convergence to the stationary distribution is provided by a property known as as ergodicity, which ensures that the stationary distribution is also the chain's unique *limiting distribution*. More details on the theory of Markov chains can be found in the literature (Murphy, 2012, Section 17.2.3).

From the definitions of stationarity and ergodicity, we can see the obvious potential of Markov chains for sampling from posterior distributions. By constructing a Markov chain with a specified target distribution as its limiting distribution, we could simulate from the posterior distribution. This is the idea underpinning MCMC methods, which were

originally developed by Metropolis et al. (1953) and remain the most widely used approach for sampling from intractable posterior distributions. The two classical approaches are the Metropolis (Hastings) algorithm (Metropolis et al., 1953; Metropolis and Ulam, 1949; Hastings, 1970) and the Gibbs sampler (Geman and Geman, 1984), both of which will now be discussed in greater detail.

Metropolis Hastings algorithm

The Metropolis Hastings algorithm (MH) replaces the task of direct sampling from the posterior distribution with the task of direct sampling from a proposal distribution, $q(\cdot|\theta^{(n)})$. Considering the case where we wish to sample from the posterior distribution from (3.6), the algorithm works as follows. Starting from point $\theta^{(n)}$ a new point, ϕ, is proposed from $q(\phi|\theta^{(n)})$. Rather than accepting every move that is proposed, the sampler decides whether or not to move to ϕ based on the following criterion:

$$\theta^{(n+1)} = \begin{cases} \phi & \text{if } \alpha(\theta^{(n)}, \phi) > u \\ \theta^{(n)} & \text{else} \end{cases} \tag{3.10}$$

where $u \sim \mathcal{U}(0,1)$ and $\alpha(\theta^{(n)}, \phi) = \min\left(\frac{p(\phi|\boldsymbol{y})q(\theta^{(n)}|\phi)}{p(\theta^{(n)}|\boldsymbol{y})q(\phi|\theta^{(n)})}, 1\right)$. In other words, we accept the move from $\theta^{(n)}$ to ϕ with probability $\alpha(\theta^{(n)}, \phi)$. Repeatedly accepting and rejecting moves based on this acceptance criterion generates a Markov chain that is guaranteed to converge to the stationary distribution. This is the result of a property known as detailed balance (Chib and Greenberg, 1995; Murphy, 2012, Section 24.3.6). One of the biggest benefits of the MH algorithm is that the intractable denominator from (3.6) cancels out in the acceptance criterion:

$$\alpha(\theta^{(n)}, \phi) = \min\left(\frac{p(\boldsymbol{y}|\phi)\pi(\phi)q(\theta^{(n)}|\phi)}{p(\boldsymbol{y}|\theta^{(n)})\pi(\theta^{(n)})q(\phi|\theta^{(n)})}, 1\right), \tag{3.11}$$

meaning that MH allows us to sample from the posterior distribution without requiring the calculation of the marginal $p(\boldsymbol{y})$. In the case where the proposal distribution is symmetric then this also cancels out in the acceptance criterion, leaving a ratio of the likelihood times the prior. This simpler case, which was formed before the MH algorithm, is called the Metropolis algorithm (Metropolis et al., 1953).

Gibbs sampler

The MH algorithm provides a method for sampling from distributions whether univariate or multivariate, but it can become inefficient in higher dimensional spaces. In this setting, a Gibbs sampler, which is in fact a special case of the MH algorithm (Gelman et al., 2013), can be more efficient. Suppose that we wish to sample from the joint distribution,

$p(\theta_1, .., \theta_m)$, with initial values $(\theta_1^{(0)}, .., \theta_m^{(0)})$. In its simplest form, where each variable is sampled separately, the algorithm cycles through the vector of variables at each iteration, sampling the ith variable from the conditional distribution $p(\theta_i|\boldsymbol{\theta}_{-i})$, where the variables in $\boldsymbol{\theta}_{-i}$ are set to their current values. This means that at the $(t+1)$th iteration, the ith variable is sampled as:

$$\theta_i^{(t+1)} \sim p(\theta_i|\boldsymbol{\theta}_{-i}), \quad \boldsymbol{\theta}_{-i} = (\theta_1^{t+1}, .., \theta_{i-1}^{t+1}, \theta_{i+1}^{t}, ..., \theta_m^t) \tag{3.12}$$

There are a few points worth mentioning. Firstly, the joint distribution of n variables does not have to be broken up into n univariate conditional distributions. Instead, we can sample from multiple lower dimensional joint distributions in a procedure called a *blocked Gibbs sampler*. Secondly, in some cases the lower dimensional conditional distributions may not be available in closed form. Although the conditioning of the Gibbs sampler can be useful, MH will be needed to sample from some of the lower dimensional distributions. Such a setting is often referred to as a Metropolis-within-Gibbs sampler. For more details on the Gibbs sampler see Robert and Casella (2009, Chapter 7); Gelman et al. (2013, Section 11.1)

3.2.4 Efficiency, convergence and simple changes to the Metropolis Hastings algorithm

Convergence of the reversible Markov chain generated by MCMC is guaranteed by ergodicity. However, this asymptotic guarantee can have little meaning in practice where we are interested in the behaviour of the sampler in finite time. Convergence diagnostics exist that can be used to provide evidence of convergence to the limiting distribution, given the trajectory of the sampler. Importantly, these different diagnostics are necessary conditions for convergence but none are sufficient conditions. One of the most commonly used diagnostics is the Gelman-Rubin statistic (Gelman and Rubin, 1992), which is also frequently referred to as the potential scale reduction factor (PSRF). Evaluation of the PSRF requires a set of M chains initiated from overly-dispersed initial points. Over-dispersion should be relative to the posterior distribution, but since this will rarely (if ever) be known, a space filling design can be used to select the initial locations of the chains. The between-chain variance (B) can be used to define the variance of the averages of the chains while the within-chain variance (W) provides the average of the variances:

$$B = \frac{N}{M-1}\sum_{j=1}^{M}(\bar{\theta}_j - \bar{\theta})^2 \quad W = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{N-1}\sum_{i=1}^{N}(\theta_j^{(i)} - \bar{\theta}_j)^2 \tag{3.13}$$

where $\bar{\theta}_j = \frac{1}{N}\sum_{i=1}^{N}\theta_i^{(i)}$ and $\bar{\theta} = \frac{1}{M}\sum_{j=1}^{M}\bar{\theta}_j$. An estimator of the posterior variance can be found as follows:

$$\hat{\text{var}}(\theta|\boldsymbol{y}) = \frac{N-1}{N}W + \frac{1}{N}B \tag{3.14}$$

where N is the number of samples in each parallel chain. The PSRF is obtained as:

$$\hat{R} = \sqrt{\frac{\hat{\text{var}}(\theta|\boldsymbol{y})}{W}} \tag{3.15}$$

which $\to 1$ as $n \to \infty$. Usually, we can take a value of less than 1.01 as being indicative of convergence of the Markov chain. We can imagine the situations where the PSRF will be high. If B is large, for instance in the case where several chains converge to different modes of the posterior, then the between chain variance dwarfs the within chain variance. Meanwhile, in situations where the proposal makes very small moves through the posterior support, W will be small (unless the sampler is run for a very long time) and again we will have a PSRF far greater than 1. The consequences of a poor proposal mechanism can also be assessed using the effective sample size.

By the definition of a Markov chain, the samples obtained using MCMC are suboptimal relative to the ideal scenario of iid samples from the posterior distribution. The Markov property introduces autocorrelation to the chain, meaning that even upon convergence, the samples will be identically distribution, but not independently distributed. We can measure the optimality of MCMC samples, relative to the iid scenario, using the effective sample size (ESS):

$$n_{\text{eff}} = \frac{N}{1 + 2\sum_{t=1}^{\infty}\rho_t} \tag{3.16}$$

Roughly speaking, the ESS measures how many independent draws contain the same level of information as the correlated samples obtained using MCMC. Typically, low values of the ESS indicates poor mixing, which suggests a problem with the sampler. On the other hand, large values suggest efficient sampling, but this can be misleading. For instance, if we are sampling from a multi-modal distribution, a chain stuck in one mode could have a very high ESS despite the fact that the mixing properties are objectively poor. Checking both the ESS and PSRF should help avoid this issue.

Unsurprisingly, methods have been proposed that aim to tackle the problem of inefficient sampling in MCMC. In particular, adaptive Metropolis (Haario et al., 2001), learns a proposal scale from past samples in the MCMC trajectory. This is effective at moving along ridges because the proposal mechanism can adjust to the shape of the ridge. An alternative mechanism, aimed at improving mixing, was proposed by Mira (2001). In delayed rejection, the sampler can propose alternative moves if a step is rejected by

the acceptance criterion. Haario et al. (2006) combined adaptive metropolis with delayed rejection in delayed rejection adaptive Metropolis (DRAM). Although providing efficient exploration of local regions of the posterior, such as ridges of high posterior density, DRAM cannot overcome the issues caused by multi-modality or heavy tailed posterior densities. Instead, population MCMC, also called evolutionary Monte Carlo (Liang and Wong, 2001), can be used. Population MCMC uses a series of Markov chains that explore the posterior at different "temperatures", similar to the idea of power posteriors (Friel and Pettitt, 2008). The chains at low temperatures are able to explore the entire posterior support while chains at high temperatures focus on local regions. As well as proposing new points in each parallel chain, the algorithm also allows for swap moves between chains, allowing for information to move up and down the temperature ladder.

3.2.5 Hamiltonian Monte Carlo

The original algorithm

No matter the choice of proposal mechanism from Section 3.2.4, MH can be inefficient in high dimensions or when the geometry of the posterior becomes complicated. In these scenarios, Hamiltonian Monte Carlo (Betancourt, 2017; Monnahan et al., 2017; Neal, 1996; Brooks et al., 2011, Chapter 5) (HMC) can be used. HMC suppresses the random walk behaviour of MH by introducing a momentum variable, $\boldsymbol{\phi}$ that guides the sampler towards regions of high posterior mass, allowing for more efficient moves through the sample space. The algorithm borrows ideas from Hamiltonian dynamics, updating the parameters and momentum variables at each iteration, with the momentum variables, $\boldsymbol{\phi}$, used in the update rule for the parameters, $\boldsymbol{\theta}$.

In Hamiltonian mechanics, systems are defined in phase space using their position $\mathbf{q}$ and momentum $\mathbf{p}$, and evolve according to Hamilton's equations:

$$\frac{dp_i}{dt} = \frac{\partial H}{\partial q_i}, \quad \frac{dq_i}{dt} = \frac{\partial H}{\partial p_i} \tag{3.17}$$

where H is the Hamiltonian. For the purpose of HMC simulations, we are interested in systems with a Hamiltonian of the form:

$$H(\boldsymbol{q}, \boldsymbol{p}) = \textit{POTENTIAL ENERGY} + \textit{KINETIC ENERGY} = V(\boldsymbol{q}) + K(\boldsymbol{p}) \tag{3.18}$$

In practice, Hamilton's equations are approximated by discretising time based on a small step size, ϵ. This can be done according to the leapfrog method, a numerical integrator that will be discussed later in the context of HMC. The evolution of a system according to Hamiltonian dynamics satisfies several convenient properties: reversibility, conservation of energy and preservation of volume. The first of these corresponds to a one-to-one

relationship between points $(\mathbf{q}, \mathbf{p})$ and $(\mathbf{q}_*, \mathbf{p}_*)$, where the former have been updated to the latter according to Hamilton's equations. The second property tells us that the total energy (as quantified by the Hamiltonian) is preserved over time as the system evolves. Lastly, preservation of volume, which is known as Liouville's theorem, tells us that a region of phase space has constant volume when evolving under Hamiltonian dynamics. Each of these properties is important for the implementation in HMC.

To consider the use of Hamiltonian dynamics in MCMC, we must consider a probabilistic treatment of a Hamiltonian system. The Hamiltonian defines the energy of the system, which leads to a probability distribution on phase space using the canonical distribution (Brooks et al., 2011, Chapter 5):

$$p(\mathbf{q}, \mathbf{p}) = \exp(-H(\mathbf{q}, \mathbf{p})) = \exp(-V(\mathbf{q}))\exp(-K(\mathbf{p}))) \tag{3.19}$$

By the conservation of total energy of the system and preservation of volume, the probability distribution is unchanged when evolving a system under Hamiltonian dynamics (Bishop, 2006, Section 11.5). This makes it particularly efficient for exploring the space in an MCMC simulation, as will now be discussed.

In HMC, Hamiltonian dynamics is used to evolve a system, allowing for more efficient proposals that better explore the target space. Treating $\boldsymbol{\theta}$ as the position of the system, auxiliary momentum variables $\boldsymbol{\phi}$ are introduced to reframe the system in phase space. Their joint distribution is expressed (and factorises) as follows:

$$p(\boldsymbol{\theta}, \boldsymbol{\phi}) = p(\boldsymbol{\theta})p(\boldsymbol{\phi}|\boldsymbol{\theta}) \tag{3.20}$$

and the system is evolved in phase space according to Hamilton's equations, with the factorisation above ensuring that simulation from the typical set of the joint distribution in phase space leads to simulation from the typical set of the target distribution (Betancourt, 2017). Comparing with the canonical distribution from (3.19), we see the correspondence between the potential energy and the target distribution:

$$V(\boldsymbol{\theta}) = -\log \pi(\boldsymbol{\theta}) - \log p(\boldsymbol{y}|\boldsymbol{\theta}) \tag{3.21}$$

This means the potential energy of the system is prescribed by the target distribution of interest $p(\boldsymbol{\theta}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$. The selection of the distribution on the momentum variables (and therefore the kinetic energy of the system) is up to the user. Typically, quadratic kinetic energy is used, corresponding to a multivariate Gaussian:

$$\boldsymbol{\phi} \sim \mathcal{MVN}(\mathbf{0}, \mathbf{M}) \tag{3.22}$$

where $\mathbf{M}$ is referred to as the *mass matrix*. Usually this is assumed to be a diagonal

matrix (Gelman et al., 2013, Section 12.4), traditionally taken to be a unit diagonal. The choice of this distribution affects the efficiency of the algorithm and ideally $\mathbf{M}$ should scale with the inverse covariance matrix of the target distribution (Gelman et al., 2013, Section 12.4).

Starting from a location $(\boldsymbol{\theta}, \boldsymbol{\phi})$, HMC updates the points in phase space using the leapfrog method, where space is discretised according to a step size, ϵ. Each iteration of the algorithm begins by updating the momentum variables, drawing them from (3.22). Following this, the leapfrog method is used to update the system in phase space, with each trajectory running for L leapfrog steps (Brooks et al., 2011, Chapter 5). A single leapfrog step consists of two updates of the momentum variables, once before updating the parameters and once afterwards in so-called half updates of the momentum, that sandwich an update of the position of the system:

$$\begin{aligned}\boldsymbol{\phi} &= \boldsymbol{\phi} + \frac{\epsilon}{2}\nabla p(\boldsymbol{\theta}|\boldsymbol{y}) \\ \boldsymbol{\theta} &= \boldsymbol{\theta} + \epsilon \mathbf{M}^{-1}\boldsymbol{\phi} \\ \boldsymbol{\phi} &= \boldsymbol{\phi} + \frac{\epsilon}{2}\nabla p(\boldsymbol{\theta}|\boldsymbol{y})\end{aligned}$$

Of the advantageous properties of Hamiltonian dynamics listed earlier, only conservation of the Hamiltonian is not satisfied by the leapfrog scheme. By updating the system according to the leapfrog integrator, we ensure that volume is preserved because the update applied to a variable in phase space is dependent only on the other variable. Such an integration method, that satisfies preservation of volume, is referred to as symplectic.

Under true Hamiltonian dynamics, total energy, and therefore the Hamiltonian, is conserved. This means that the system should move around with constant probability density and acceptance rate of 1. In practice, the discretisation introduces a small deviation to the joint density in phase space. Although a large error will affect the efficiency of the algorithm, the discretisation error is corrected with a typical MH acceptance step based on the following acceptance criterion:

$$\alpha = \min\left(1, \exp(-H(\boldsymbol{\theta}_*, \boldsymbol{\phi}_*) + H(\boldsymbol{\theta}, \boldsymbol{\phi}))\right) \tag{3.23}$$

The smaller the error, the higher the acceptance rate of the algorithm. Endowed with the above acceptance criterion, it can be proven that HMC satisfies detailed balance (which uses the reversibility of Hamiltonian dynamics) and that it leaves the target distribution invariant, which relies on preservation of volume (Brooks et al., 2011, Chapter 5).

No-U-Turn Sampler

The use of HMC requires the selection of a step size, ϵ and a number of leapfrog steps. One option is to use Bayesian optimisation to tune the number of leapfrog steps and step size based on some criterion of efficiency (Wang et al., 2013b). An alternative is to update them on the fly to achieve some target acceptance rate. Hoffman and Gelman (2014) proposed an automatic tuning of the number of leapfrog steps in a method referred to as the No-U-Turn Sampler (NUTS). In NUTS, the leapfrog trajectory continues until it turns back on itself or a maximum number of leapfrog steps is reached. Conceptually, a u-turn in the trajectory leads to a reduction in the distance between the current position in the trajectory and the starting position. Formally, this is assessed using the derivative of the squared distance between the initial and current trajectory position with respect to simulation time. This derivative is the dot product between the current momentum and the vector from the initial position to the current position. If the derivative is negative, then the angle between these vectors is greater than 90 degrees, which implies that the momentum is pointing into the wrong direction and the trajectory is moving back to the initial position. This is the moment when the trajectory starts to turn back on itself. The tuning of the leapfrog steps can be combined with an automatic tuning of the step size, such as the dual averaging step size adaptation used by Hoffman and Gelman (2014), allowing for more efficient sampling than standard HMC.

3.2.6 Bayesian calibration for complex simulators

As was mentioned above, the aim in model calibration is to recover, for some new dataset $\boldsymbol{y}_*$, the parameters of the model that led to the data. In the absence of any evidence otherwise, the following additive noise model can be assumed, where $\zeta(\boldsymbol{\theta})$ is the output of the simulator at $\boldsymbol{\theta}$:

$$\boldsymbol{y}^* = \boldsymbol{\zeta}(\boldsymbol{\theta}) + \boldsymbol{\xi}, \quad \boldsymbol{\xi} \sim \mathcal{MVN}(0, \sigma^2\mathbf{I}) \tag{3.24}$$

Giving rise to the likelihood function:

$$p(\boldsymbol{y}^*|\boldsymbol{\theta}, \sigma^2) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\|\boldsymbol{y}^* - \boldsymbol{\zeta}(\boldsymbol{\theta})\|_2^2 \tag{3.25}$$

which is evaluated by running the simulator at a set of parameters and then plugging the simulator output, along with a value of the noise variance, into the likelihood function. Assigning a prior to the parameters $\boldsymbol{\theta}, \sigma^2$, we can sample from the posterior distributions using MCMC. Each iteration would rely on at least one call of the simulator, which can be replaced with an emulator in situations where the simulator is expensive to evaluate (Kennedy and O'Hagan, 2001).

An alternative technique commonly used for model calibration in the literature is Bayesian History Matching (BHM), which uses an implausibility criterion to gradually rule out regions of parameter space as implausible (Gardner et al., 2020; Volodina and Challenor, 2021). Due to computational costs this criterion is usually evaluated using an emulator, taking into account both the predictive mean and the predictive variance. The selection of non-implausible points occurs in waves, with simulations from a shrinking non-implausible set being added to the emulator training set, constructing an emulator that is increasingly precise in the region of space where the predictions from the model best match the observed data. While history matching increasingly excludes areas of low posterior probability in the parameter space, it does not provide a sample from the posterior distribution. For this purpose, we could combine the idea of history matching with HMC, as proposed by Rasmussen (2003). Their algorithm used an exploratory phase akin to history matching to rule out areas in parameter space with low posterior probability and build an emulator for the areas with high posterior probability. This exploratory phase is followed by the sampling phase, which converges to a proper sample from the posterior distribution. Since BHM usually requires online evaluation of the simulator during each wave of the algorithm, the computational costs of forward simulation are incurred *a posteriori*. For the work carried out in this thesis, the aim is to incur the computational costs of simulation before the measured data are available.

3.2.7 Bayesian calibration in the presence of model discrepancy

While (3.25) is a commonly used noise model, it makes the assumption that the input-output map of the simulator perfectly replicates the underlying process. Were this the case, and $\boldsymbol{y}^*$ were subject to iid Gaussian noise, we could use the inferred parameters to draw conclusions about the behaviour of the system in line with the intuition of the mathematical modeller. Unfortunately, the relationship assumed in (3.25) often misrepresents the true relationship between the measurements and the simulator, resulting in *model mismatch*. Model mismatch encompasses the error between the simulator and the true process—often referred to as *model discrepancy*—and the misspecification of the measurement noise. Acknowledging the presence of model mismatch, a more accurate representation of the relationship between the simulator and the measured data could be of the following form:

$$\boldsymbol{y}^* = \boldsymbol{\zeta}(\boldsymbol{\theta}) + \boldsymbol{\beta} + \boldsymbol{\alpha} + \boldsymbol{\xi}, \tag{3.26}$$

with the extra term $\boldsymbol{\beta}$ representing the structure from the real world system that is missing from the simulator, and $\boldsymbol{\alpha}$ providing the structure from the measurement noise that is missing from the iid Gaussian assumption.

Intuitively, inference in the presence of model mismatch impacts our parameter estimation, leading to biased, over-confident, parameter estimates if the missing structure is within the capacity of the model, or inflated uncertainty if this is not the case. Lei et al. (2020) referred to the parameter estimates in such a scenario as *pseudo-true* values, whose values no longer provide the same insight about the state of the physical system. One solution, proposed by Kennedy and O'Hagan (2001), is to consider the following noise model:

$$\boldsymbol{y}^* = \boldsymbol{\zeta}(\boldsymbol{\theta}) + \boldsymbol{\phi}(\mathbf{x}) + \boldsymbol{\xi}, \tag{3.27}$$

which introduces an extra term $\boldsymbol{\phi}(\mathbf{x})$ to account for $\boldsymbol{\beta} + \boldsymbol{\alpha}$ from (3.26). Importantly, $\boldsymbol{\phi}(\mathbf{x})$ is independent of the model parameters $\boldsymbol{\theta}$.

The use of (3.27) is dependent on our ability to restrict the flexibility of $\boldsymbol{\phi}(\mathbf{x})$. For this reason, it is worth distinguishing between two different situations. Often, structure is deliberately left out of a mathematical model for the sake of computational efficiency. If we know the effect that these *known unknowns* have on the system, then a more informative prior structure can be used for $\boldsymbol{\phi}(\mathbf{x})$ and we may disentangle the two sources of model mismatch during parameter inference. However, in many cases the differences between a mathematical model and reality occur as a result of the modeller's ignorance. These *unknown unknowns* are more difficult to account for in (3.27) as we have no information about the mismatch term. Although information is omitted from the model without the modeller's knowledge, the modeller nevertheless makes some assumptions about the smoothness of these unknown effects (often in the form of a Gaussian process prior, see Section 3.4), which affects the predicted uncertainty. While the discrepancy function can be quite flexible, correcting for changes made in the parameters and leading to inflated uncertainty as shown by Brynjarsdóttir and O'Hagan (2014), some of these corrections will be in the tail of the distribution defined by the Gaussian process and will hence effectively be ruled out. Recently, model discrepancy has been discussed in the context of cardiovascular modelling (Plumlee, 2017; Lei et al., 2020), using methods similar to the one of Kennedy and O'Hagan (2001).

3.3 Neural networks

3.3.1 Overview

Neural network research started with the conception of artificial neurons, intended to replicate the behaviour of the brain (Mcculloch and Pitts, 1943; Rosenblatt, 1958). An example is shown in Figure 3.4, where a series of inputs, denoted x_i, pass along weighted connections to the neuron, indicated by a blue circle. Note that the arrows here correspond to the direction of travel of information through the network and do not correspond to any

probabilistic dependence. The weights on the connections tell us the weight with which the inputs contribute to the *activation* of the neuron, which is neural network parlance for a weighted linear combination of the inputs. The neuron applies a transformation to this activation through a function referred to as an *activation function*, indicated by ϕ in the diagram. In the original work of Mcculloch and Pitts (1943), this activation function was assumed to be a threshold function, motivated by brain activity where neurons are triggered by the magnitude of a signal. In later years, more flexible activation functions were considered, examples of which are found in Figure 3.5. The operation performed by the neuron in Figure 3.4 draws obvious parallels with the basis functions used in basis function regression models (see chapters 3 and 5 of Bishop (2006)), only now the weights into the neuron (the parameters of the basis function) are adaptive. The real expressiveness of artificial neurons comes from combining multiple neurons in a neural network.

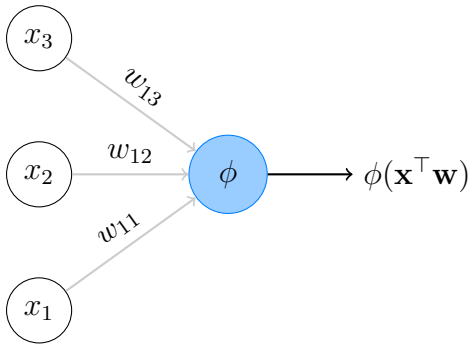


Figure 3.4: Artificial neuron example. The neuron takes a weighted sum of its three inputs and outputs a transformation of this linear combination.

Feedforward neural networks are constructed by stacking series of artificial neurons on top of each other in a collection of layers. These layers are then connected by weighted connections that specify how information will pass through the network during the *forward pass*, which is the process of feeding data into the model and producing an output. A typical neural network is visualised in Figure 3.6, where the hidden layers are represented by blue circles and the nodes containing ones represent the bias parameters of each layer. If we consider all the activation functions in Figure 3.6 as equal and denoted by ϕ, then the forward pass of this network, for input $\boldsymbol{\theta}$, could be summarised by scaling up the calculations from Figure 3.4 in the following set of matrix equations:

$$\begin{aligned}
\mathbf{Z}_1 &= \phi(\boldsymbol{\theta}\mathbf{W}^{(1)} + \mathbf{b}^{(1)}) = \phi(\tilde{\boldsymbol{\theta}}\tilde{\mathbf{W}}^{(1)}) \quad \text{the output of H1} \\
\mathbf{Z}_2 &= \phi(\mathbf{Z}_1\mathbf{W}^{(2)} + \mathbf{b}^{(2)}) = \phi(\tilde{\mathbf{Z}}_1\tilde{\mathbf{W}}^{(2)}) \quad \text{the output of H2} \\
\mathbf{Z}_f &= \mathbf{Z}_2\mathbf{W}^{(3)} + \mathbf{b}_3 = \tilde{\mathbf{Z}}_2\tilde{\mathbf{W}}^{(3)} \quad \text{the output of the final layer,}
\end{aligned} \tag{3.28}$$

where the parameters $\mathbf{b}^{(i)}$ represent the contribution of the bias parameters to the activation of a node. These were absorbed into the matrix multiplication by appending a row

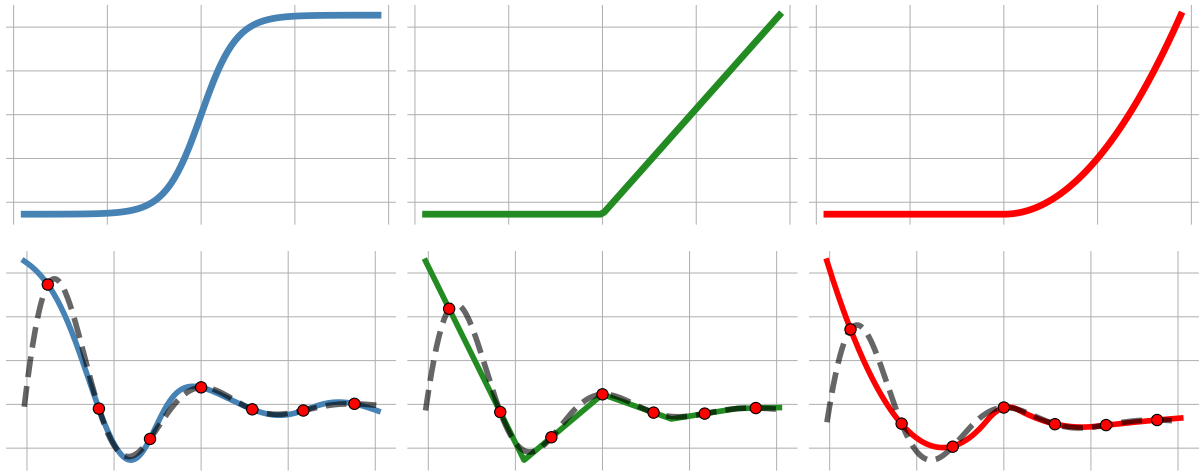


Figure 3.5: Demonstration of different activation functions: left=tanh, middle=ReLU and right=quadratic ReLU. The tanh is a standard sigmoidal shaped activation function. The ReLU is piecewise linear, with the kink making the overall function nonlinear. The quadratic ReLU adjusts the linear portion of the ReLU to be a quadratic, allowing for smooth interpolation. The bottom row shows regression functions fitted using neural networks with the different activation functions.

containing the bias parameters to the matrices $\mathbf{W}^{(i)}$ and a column of ones to the inputs of the layer. Both updates were denoted by $\tilde{\cdot}$ notation.

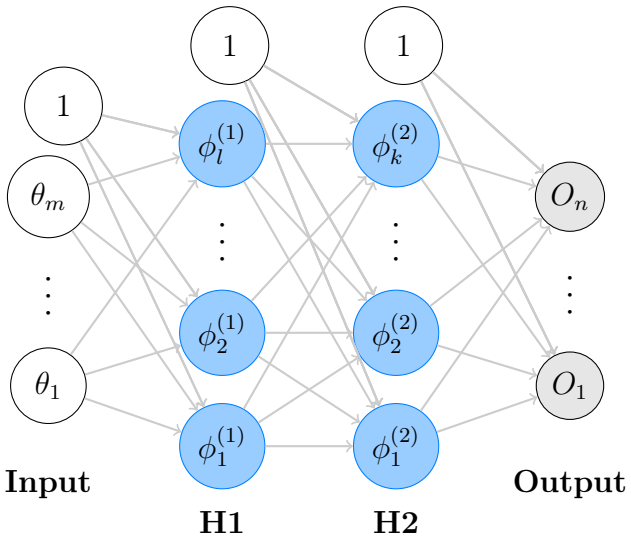


Figure 3.6: An example of a neural network with two hidden layers.

3.3.2 The general regression problem

Let us consider a typical regression problem where we have a set of input and output training data, $\{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_n, \mathbf{y}_n)\}$. Assuming Gaussian noise on the data, the following

negative log-likelihood quantifies the fit of the neural network model to the data:

$$L(\mathbf{w}, \mathbf{x}_i, \mathbf{y}_i) = \sum_{i=1}^{n} ||(\boldsymbol{y}_i - f(\mathbf{x}_i, \Omega)||^2, \tag{3.29}$$

where Ω contains all the weights of the model. This function can be minimised with respect to Ω to obtain our estimate of the parameters that fit the data. Typically, the minimum does not exist in closed form and we rely on iterative optimisation. If we assume that the activation functions are themselves differentiable, then the loss function in (3.29) is also differentiable, meaning we can use gradient based methods to minimise (3.29). Given the repeated composition of functions within a network, differentiation is non-trivial and requires the use of back-propagation. Back-propagation relies on a forward pass of the model to propagate information through to the loss function from (3.29). The error that is produced can then be propagated back through the model (hence the name back-propagation) using repeated invocations of the chain rule of calculus. This allows for the use of exact gradient calculations in the optimisation and can also be used to obtain the gradient of neural networks with respect to the inputs themselves. A detailed discussion on the back-propagation algorithm can be found in the literature (see section 16.5.4 of Murphy (2012) for an overview or chapter 7 of Rojas (1996) or section 14.8 of Bishop (1995) for more detailed discussions). Back-propagation can be implemented using automatic differentiation (Baydin et al., 2017) in software such as Tensorflow.

3.3.3 Training

Optimisation

The duration of neural network training is usually specified by the number of training epochs. One training epoch is defined as the forward pass of the entire dataset through the network. In the simplest training routine, the parameters of the model are updated once per epoch in a procedure known as batch gradient descent. The update equation relies on a learning rate, η, that specifies the size of step taken and an expected value of the gradient, $\Delta\mathbf{w}_t$:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \eta\Delta\mathbf{w}_t \quad \text{with } \Delta\mathbf{w}_t = \frac{1}{n}\sum_{i=1}^{n} \nabla L(\mathbf{w}_t, \{\mathbf{x}_i, \mathbf{y}_i\}). \tag{3.30}$$

The learning rate affects the efficiency of the optimisation procedure. If this is too high we might overshoot the minimum while too small a learning rate results in slow convergence. Usually, neural networks are trained on very large datasets and the calculation of the expected value of the gradient, which relies on the forward pass of each training example, is expensive. To overcome this, we can approximate $\Delta\mathbf{w}_k$ using a subset of the data in

stochastic optimisation (see Chapter 8 of Goodfellow et al. (2016) for more details). Wilson and Martinez (2003) present the benefits of purely stochastic (so called on-line training) gradient descent, where approximations are made with only one data example, over batch gradient descent. In recent times, it has become more common to perform stochastic optimisation by passing the data through the network in a series of mini-batches. In doing so, we approximate the expected value of the gradient using a subset of the data cases (Masters and Luschi, 2018), which is a closer approximation to the quantity in (3.30) than the approximation made during purely stochastic optimisation. Unlike batch gradient descent, the number of parameter updates no longer matches the number of epochs. Instead, when training a model to fit a training set of size n using mini-batch gradient descent with batch size m, the number of parameter updates per epoch is equal to $\frac{n}{m}$. The kth update per epoch is as follows:

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \eta\Delta\mathbf{w}_k \quad \text{with } \Delta\mathbf{w}_k = \frac{1}{m}\sum_{i=km}^{(k+1)m} \nabla L(\mathbf{w}_k, \{\mathbf{x}_i, \mathbf{y}_i\}), \tag{3.31}$$

One advantage of stochasticity is that it helps us to avoid local optima in the likelihood landscape (Bottou, 1991). Keskar et al. (2017) show that introducing stochasticity to the optimiser encourages the convergence to flatter local optima and these are widely considered to achieve improved generalisation properties, which also makes sense from a Bayesian evidence viewpoint (Smith and Le, 2018). Mini-batch gradient descent will be used in the training of most neural networks in this thesis, which requires the tuning of the mini-batch size and learning rate.

The behaviour of the loss surfaces of neural networks, and their impact on optimisation of the parameters, is still a source of great unknowns. While local optima are theoretically guaranteed as a result of non-identifiabilities such as weight space symmetries (see section 4.4 of Bishop (1995)), these are often inconsequential for model training since they do not affect the value of the loss at the local optima. A well-documented issue is the presence of saddle points in the loss surface, which Dauphin et al. (2014) argued are more common than local optima in high dimensional non-convex loss problems. Recent optimisation routines like the Adam optimiser (Kingma and Ba, 2015) overcome the saddle point problem by adapting the learning rate during optimisation (Staib et al., 2019). These optimisers also help to efficiently traverse valleys in the loss function by using momentum, which prevents the optimiser from zig-zagging along the sides of the valley. Consider the update performed in the tth step of gradient descent with momentum to be denoted by u_t, then the $(t+1)$th update of the parameters follows the following rule:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \eta\Delta\mathbf{w}_t + \gamma\mathbf{u}_t \tag{3.32}$$

where γ is the momentum variable. As the optimiser progresses, these momentum updates accumulate and encourage the sampler to move in their direction. Ruder (2016) provides a useful overview of many commonly used optimisation algorithms and their different benefits.

Model selection and generalisation of NN emulators

In the case of deterministic training data, as is often the case in statistical emulation, the aim of neural network training is to replicate the process that produced the data. In general, we are interested in this process over the continuous input space, which can be measured using the following:

$$\mathbb{E}_{\mathcal{D}\sim p(\mathcal{D})}(L(\mathcal{D};\Omega)) \tag{3.33}$$

Unfortunately, this integral is intractable so we approximate it using the training data, as was done in (3.29), and the integral over a probability distribution becomes a sum over the training examples. It s well documented that this approximation of an integral with a sum introduces the risk of overfitting, which Mitchell (1997) defined as follows:

Given a hypothesis space $\mathcal{H}$, a hypothesis $h \in \mathcal{H}$ is said to overfit the training data if there exists some alternative hypothesis $h' \in \mathcal{H}$, such that h has smaller error than h' over the training examples, but h' has a smaller error than h over the entire distribution of instances.

In neural network training, our hypothesis space contains the infinite configurations of the neural network weights. The idea of overfitting a neural network to noisy data is quite natural and is visualised in Figure 3.7. Here we see that, having been initialised with small weights, the network initially underfits the data (the solid red line is the network function), and provides a poor fit to the true underlying function (the black dashed line). As training progresses, the function gradually becomes more complex before eventually overfitting to the data. This plot displays another phenomenon: the inductive bias resulting from the weight initialisation. Small weights correspond to an inductive bias favouring simpler models whereas larger weights correspond to an inductive bias for more complicated models.

The idea that small weights provide simpler models motivates different approaches to regularisation in neural network training, the main ones being early stopping (Caruana et al., 2000) and weight decay (Krogh and Hertz, 1992). In early stopping, we monitor the ability of the network to generalise to unseen cases using a held out validation set. As training progresses and the network begins to overfit, the loss evaluation on the validation set begins to increase, as indicated by the orange dashed line in the top row of Figure 3.7. Rather than continuing training until the minimum of the training error, we should stop at

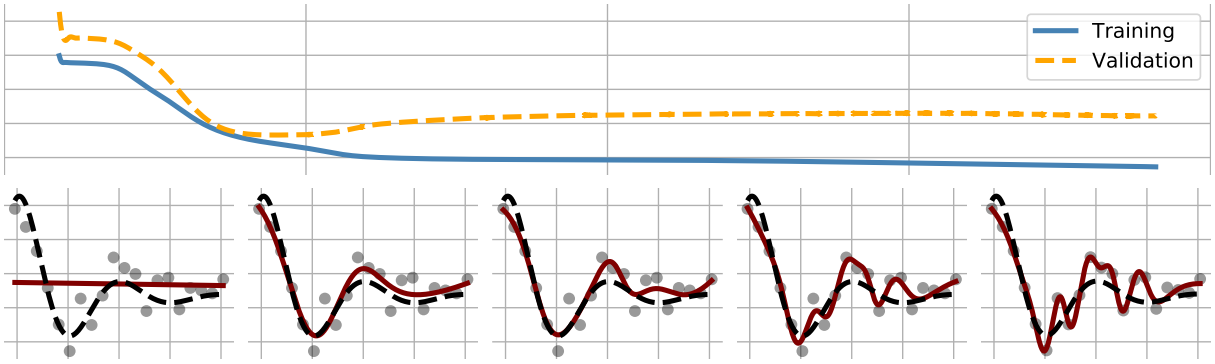


Figure 3.7: Showing the idea of early stopping. The top row shows the training error as a solid blue line and the validation error as a dashed orange line. The validation error begins to increase when training progresses beyond simple models. We can visualise this in the bottom row where the trained models are displayed at different stages of training. The training data are shown in these plots as grey scatter points.

the minimum of the validation loss, which would roughly correspond to the second fitted function (from the left) in the bottom row of Figure 3.7. Weight decay is an alternative method that uses an L2 penalty to penalise large weights. The effect is identical to that of the ridge penalty in ridge regression: reduce variance in the final estimator at the expense of introducing a small bias. The size of this penalty is usually tuned on a validation set.

Unlike the noisy data case, training a neural network as a statistical emulator of a deterministic simulator means training on deterministic data. In this scenario, we concern ourselves with the generalisability of the model. The generalisation properties of network parameters can be considered similarly to the case of over-fitting: small parameters generally elicit good generalisation properties in the model and large values provide models that are more complex. In Figure 3.7, we saw how easy it was with noisy data to move from the region of simple models to the region of complex ones. With deterministic data, the optimiser is no longer as vulnerable to this phenomenon because the data do not provide a gradient to drive the model into this regime. This is visualised in Figure 3.9, where a very heavily parameterised model containing 200 hidden nodes is fitted to a noise-free dataset without any convexity appearing in the validation loss curve. Figure 3.9 shows what is typically observed in the case of deterministic data. Rather than a convex validation loss curve, the validation loss converges to a larger loss value, indicating the poor generalisation of the function. However, the fitted function itself is actually overly simple and the poor generalisation is not a result of overfitting, but rather a small dataset size. For this reason, as well as selection of network architecture (the model selection), it is still important that we track the performance of the neural network on a validation set during training, but regularisation techniques will not necessarily be useful.

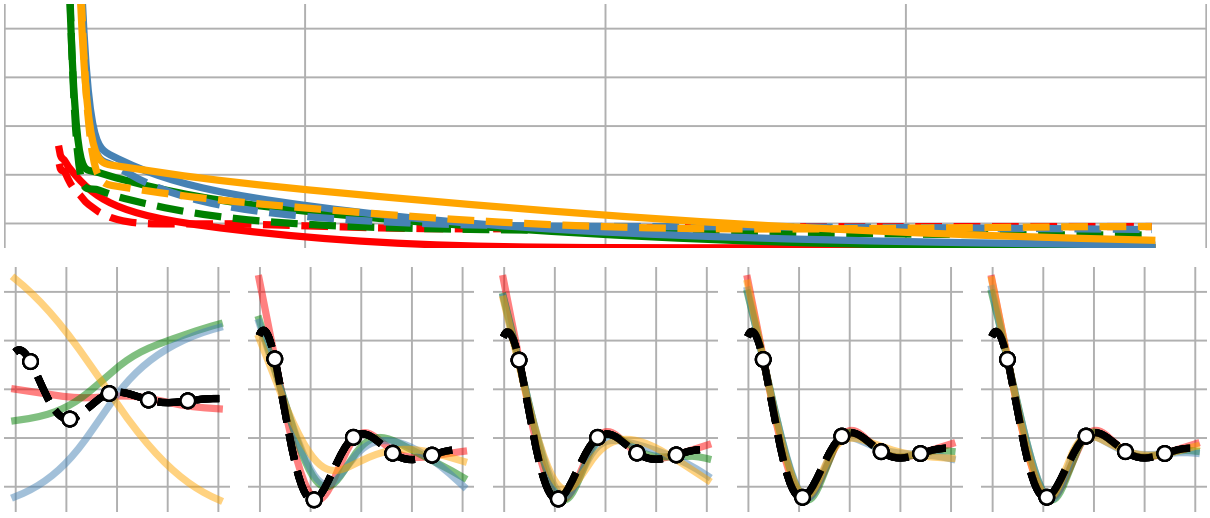


Figure 3.8: A heavily over-parameterised neural network trained from four random initialisations at small weights. All optimisations converge to similar models and the generalisation is good.

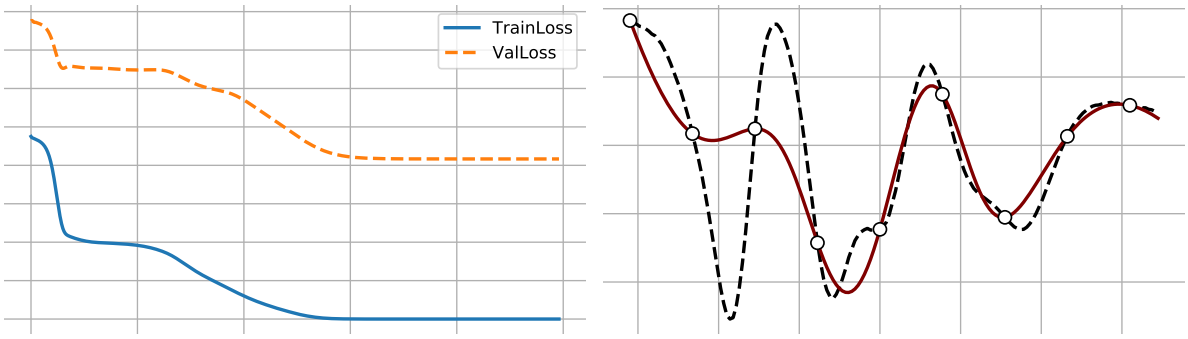


Figure 3.9: A heavily over-parameterised neural network trained on a deterministic function. Rather than an increasing validation loss, the poor generalisation appears simply as a validation loss significantly higher than the training loss. In the plot on the right, the dashed line shows the actual function while the solid red curve shows the function learnt by the neural network.

3.3.4 Bayesian neural networks

Optimisation of neural network models provides us with a single best estimate of the simulator function. As we saw in Figure 3.8 if the simulator is smooth, this can give a good fit to the function. However, if we are concerned with a complex simulator function for which we have limited training data, quantification of our epistemic uncertainty is important. The parametric nature of neural networks necessitates the specification of prior information in weight space:

$$\mathbf{w} \sim p(\mathbf{w}|\alpha) \tag{3.34}$$

where α is the hyperparameter of the prior (in general, we can have more than one hyperparameter). A common choice for this prior is a zero-mean Normal distribution. This is motivated in a similar manner to L2 regularisation since it encourages smaller parameters. We can alternatively use a Laplace prior on the weights, which induces sparsity in the network similar to the LASSO. Having placed a prior on the weights, we wish to update to a posterior distribution in light of the data using Bayes theorem (3.6):

$$p(\mathbf{w}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\mathbf{w})p(\mathbf{w}) \tag{3.35}$$

The update of the weight distribution is typically not available in closed form and computational approaches must be used to approximate it. One option is to assume the posterior is peaked and obtain a MAP estimate of the parameters. Essentially, this is equivalent to optimisation with weight decay and gives us no quantification of our uncertainty. To allow for a posterior distribution instead of a point estimate, MacKay (1992a) proposed the use of the Laplace approximation, approximating the posterior with a Gaussian distribution centred on the MAP estimate. The hyperparameters of the Gaussian prior can then be found by optimising the evidence $\int p(\boldsymbol{y}|\mathbf{w}, \alpha, \beta)p(\mathbf{w}|\alpha, \beta)d\mathbf{w}$, which is analytically tractable given Gaussian assumptions on the prior and data model. See MacKay (1992a); Bishop (1995, Section 10.4) for further details on this approach, as well as some of its disadvantages. Neal (1996) proposed the use of HMC to learn the sample from the posterior distribution from (3.35) and this is widely accepted as the state of the art for fully Bayesian inference in neural networks. Neal (1996) includes a detailed discussion on the choice of priors, for which he uses a hierarchical form. One such choice is to use an Inverse Gamma hyperprior on the weight variances:

$$\mathbf{w} \sim \mathcal{N}(0, \sigma_w^2), \sigma_w^2 \sim \mathcal{IG}(\alpha, \beta) \tag{3.36}$$

In the simplest case, the same variance hyperparameter is shared within a layer of the network, but more intricate relationships can be modelled. For instance, ARD is introduced by sharing a unique common weight variance hyperparameter between all of the connections out of a particular input to the network. Neal (1996) advocated making the model as flexible as possible since the marginalisation provided by the Bayesian approach helps to guard against overfitting One drawback of this fully Bayesian approach is the large computational cost. Since HMC relies on gradients, which in turn relies on back-propagation, obtaining one sample using HMC can require thousands of forward passes of the entire dataset through the model.

Several solutions have been proposed to the computational costs of fully Bayesian HMC. These include variational inference (Graves, 2011), which attempts to approximate a single mode of the posterior using a parameterised distribution, and ensemble approaches

(Lakshminarayanan et al., 2017). The latter of these has received particular attention in recent years and has even been suggested to outperform HMC. One argument for this is as follows: the expressiveness of neural network models leads to a diverse set of equally likely hypotheses and these are often spread throughout the parameter space in a series of modes. Using HMC, we can only move between modes if these are connected by a region of high posterior density. On the other hand, ensemble approaches specifically target the multi modality of the likelihood landscape. Fundamentally, the idea is to initialise optimisers from different initial points in weight space that will then converge to different modes (Fort et al., 2019), therefore capturing the variety of solutions contained within the different modes. Recently, ensembles have been improved by Pearce et al. (2020) who used randomised prior distributions to regularise the optimisation of each network in an ensemble. This has been shown to outperform standard ensembles and an example for a toy function is shown in Figure 3.10, showing the good agreement with Gaussian processes (see Section 3.4), at least for a simple toy problem in 1D.

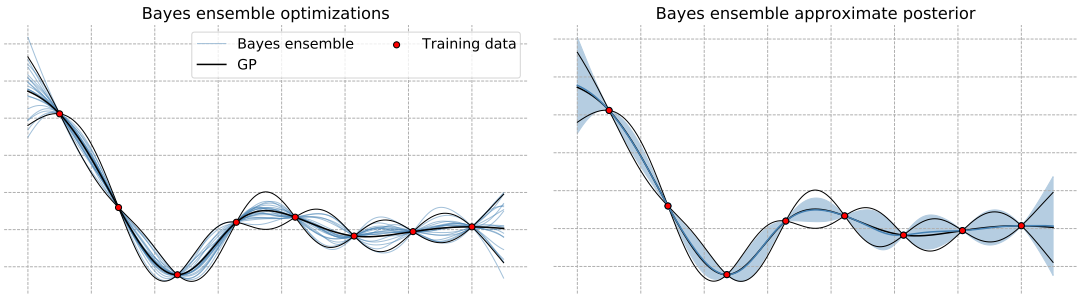


Figure 3.10: Demonstration of the Bayesian ensemble. 20 optimised models are used to obtain the posterior predictive distribution.

3.4 Gaussian processes

3.4.1 Gaussian process regression

Given data $\boldsymbol{y}$ measured at inputs $\boldsymbol{\theta}$, the task in regression is to recover the underlying process that produced the data. Typically, $\boldsymbol{y}$ will be subject to some noise, which we often assume is additive: $\boldsymbol{y} = f(\boldsymbol{\theta}) + \boldsymbol{\xi}$. In the absence of any evidence otherwise, this noise can also be assumed to be iid Gaussian. Recovery of the underlying process requires a mathematical description of the function. In some cases knowledge will be available that allows us to specify a parametric representation for f and inference for f then amounts to the estimation of a set of parameters. In reality, we often do not have information that allows for a well specified parametric form and might instead consider the more flexible, but still parametric, neural network. If we wished to take a Bayesian approach to regression, this requires prior specification in weight space, as discussed in Section 3.3.4.

However, beyond fairly simple knowledge (smooth functions, for instance, are a natural consequence of the Gaussian prior on neural network weights), prior elicitation in weight space can be challenging.

Instead of relying on a parametric model to encode all the information available about a function, we can express this information directly in function space. If we wish to take a probabilistic approach and quantify our uncertainty about the function, then we require a probabilistic representation of a (possibly infinite) set of random variables, for which a Gaussian process (GP) (Rasmussen and Williams, 2006; Roberts et al., 2013) can be used. A GP can be considered as the infinite dimensional extension of the multivariate Gaussian distribution. Assigning a GP prior to a function f, we assume a distribution that is fully defined by a mean $m(\boldsymbol{\theta})$ and kernel (covariance) function $k(\boldsymbol{\theta}, \boldsymbol{\theta})$:

$$f \sim \mathcal{GP}(m(\boldsymbol{\theta}), k(\boldsymbol{\theta}, \boldsymbol{\theta}')). \tag{3.37}$$

The mean and kernel functions depend on sets of tunable parameters, which are referred to as the hyperparameters of the GP. Generally, the form of the kernel k is as follows:

$$k(\theta, \theta') = \sigma_f^2 \tilde{k}(\theta, \theta; \boldsymbol{\beta}) \tag{3.38}$$

where σ_f^2 is the signal variance, controlling the amplitude of the signal and $\boldsymbol{\beta}$ is the vector of kernel hyperparameters. We can gather all of these hyperparameters into a single vector of hyperparameters, $\boldsymbol{\psi} = (\sigma_f^2, \boldsymbol{\beta})$. The expression $\tilde{k}$ can take various forms, as will be discussed in Section 3.4.2.

In light of the observed data, we wish to update the prior on functions to a posterior. Although the infinite dimensionality of the prior would appear intractable, the marginalisation property of multivariate Gaussian distributions similarly applies in the case of GPs. This means that any finite subset of realisations of the function, $\{f(\boldsymbol{\theta}_i)\}_{i=1}^n$, has a multivariate Gaussian distribution. A consequence of this property is that, despite specifying our prior in the infinite dimensional space of functions, all proceeding inferences can be performed while ignoring those locations with no data. In particular, the posterior predictive distribution at new points θ_*, depends only on the function at the locations of the training data. Starting from the joint distribution:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} k(\boldsymbol{\theta}, \boldsymbol{\theta}) & k(\boldsymbol{\theta}, \theta_*) \\ k(\theta_*, \boldsymbol{\theta}) & k(\theta_*, \theta_*) \end{bmatrix}\right) \tag{3.39}$$

the predictive distribution $p(\mathbf{f}_*|\mathbf{f})$ is obtained using standard Gaussian conditional expres-

sions (Bishop, 2006, equations 2.81 and 2.82):

$$p(\mathbf{f}_*|\mathbf{f}) = \mathcal{N}(k(\theta_*, \boldsymbol{\theta})K(\boldsymbol{\theta}, \boldsymbol{\theta})^{-1}\mathbf{f},\ k(\theta_*, \theta_*) - k(\theta_*, \boldsymbol{\theta})K(\boldsymbol{\theta}, \boldsymbol{\theta})^{-1}k(\boldsymbol{\theta}, \theta_*)). \tag{3.40}$$

In the context of emulation, by conditioning on the function at the observed locations, we limit the distribution to those functions that are consistent with the observed simulations, as is demonstrated in Figure 3.11.

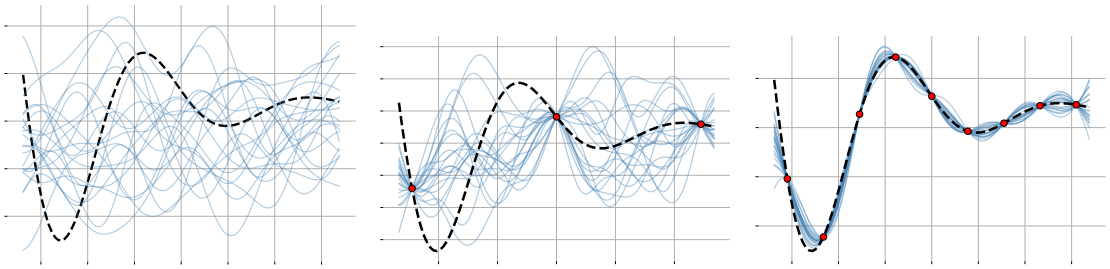


Figure 3.11: Starting from a prior in function space on the left, samples from the GP prior are very different from the ground truth function (the simulator). When we condition on the training simulations (red points) we limit the distribution of functions to those that are consistent with the simulations. As the number of simulations increases, we gradually begin to limit the distribution of functions to those that are consistent with the simulator.

Complete specification to the data requires tuning the hyperparameters, $\boldsymbol{\psi}$, of the kernel function. The typical approach is to first integrate the function out of the model, obtaining the marginal likelihood $p(\boldsymbol{y}|\boldsymbol{\psi})$:

$$\begin{aligned}\int p(\boldsymbol{y}, f|\boldsymbol{\psi})d\mathbf{f} &= \int p(\boldsymbol{y}|\mathbf{f})p(\mathbf{f}|\boldsymbol{\psi})d\mathbf{f} \\ &= \int \mathcal{N}(\boldsymbol{y}|\mathbf{f}, \sigma^2)\mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K})d\mathbf{f} \\ &= \mathcal{N}(\boldsymbol{y}|\mathbf{0}, \mathbf{K} + \sigma^2\mathbf{I})\end{aligned}$$

which can be optimised to obtain the hyperparameters in a type II maximum likelihood manner. Due to the interpolative properties of GPs, the value of these hyperparameters has less effect on the fit as the number of training data increases.

3.4.2 Kernel functions

The knowledge encoded in a GP prior is largely prescribed by our choice of kernel function, k. This function quantifies the covariance between each unique pair of functions points, specifying the properties we expect to see in the function. A posteriori, the choice of k will generally only affect the function inference between the deterministic training data since GPs are perfect interpolators. Several types of kernels exist that can be used for

this purpose, and some that are relevant to the work in this thesis will now be discussed in more detail.

Stationary kernel functions

A stationary GP kernel (see Rasmussen and Williams (2006, Section 4.2.1) is a function of $\boldsymbol{\theta} - \boldsymbol{\theta}'$, meaning that it is invariant under translation of the input space. The notion of invariance can be extended if k is a function of $|\boldsymbol{\theta} - \boldsymbol{\theta}'|$, in which case the kernel function is called isotropic. Such a kernel does not depend on the direction of change between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, only the distance between them. A common example of an isotropic kernel is the squared exponential, which was used for the plots in Figure 3.11:

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sigma_f^2 \prod_{i=1}^{p} \exp\left\{ \frac{(\theta_i - \theta_i')^2}{\lambda_i} \right\} \tag{3.41}$$

where σ_f^2 (signal variance) and λ_i (lengthscale), are the parameters of the kernel function and the hyperparameters of the resulting GP. The lengthscale dictates the wiggliness of the function (often referred to as the number of zero crossings of the GP) while the signal variance controls the amplitude. Making the lengthscale a vector in the multidimensional setting, as was done in (3.41), introduces ARD to the kernel function. The SE kernel assumes that the function in question is infinitely differentiable, which can sometimes be a restrictive assumption. For this reason, we may also consider the Matérn class of kernels. This class of kernels contain a (nonoptimisable) order parameter that determines the order of differentiability of the kernel function. Generally, this allows us to produce functions that are less smooth than those of the squared exponential. For more details on the Matérn, as well as other stationary kernel functions, see Rasmussen and Williams (2006, Section 4.2.1).

Kernel functions from neural networks

Neal (1996) explored the behaviour of neural networks in the limit of an infinite width hidden layer and showed that these are equivalent to GP priors when assuming Gaussian priors on the weights. Recently, this idea has been extended to deep neural networks (Lee et al., 2018). The realisation that neural networks in the infinite width limit are equivalent to GPs has motivated a range of work on kernel functions derived from infinite width neural networks with a given activation function. Williams (1997) derived the sigmoidal kernel from a neural network with one hidden layer of erf activation functions (the erf is shaped similarly to the tanh function). More recently, Cho and Saul (2009) derived the ArcCosine kernel based on the ReLU activation function. This kernel function contains an order parameter that specifies the order of the positive portion (up to quadratic). The

order 1 and order 2 ReLU were shown in Figure 3.5. This kernel is expressed as follows, where n is the order of the ReLU:

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \frac{\sigma_f^2}{2\pi}\sqrt{(b + w\boldsymbol{\theta}\cdot\boldsymbol{\theta})(b + w\boldsymbol{\theta}'\cdot\boldsymbol{\theta}')}J_n(\phi) \tag{3.42}$$

where

$$\phi = \cos^{-1}\left(\frac{b + w\boldsymbol{\theta}\cdot\boldsymbol{\theta}'}{\sqrt{b + w\boldsymbol{\theta}\cdot\boldsymbol{\theta}}\sqrt{b + w\boldsymbol{\theta}'\cdot\boldsymbol{\theta}'}}\right)$$

$$J_n(\phi) = \begin{cases} \pi - \phi & n = 0 \\ \sin\phi + (\pi - \phi)\cos\phi & n = 1 \\ 3\sin\phi\cos\phi + (\pi - \phi)(1 + 2\cos^2\phi) & n = 2 \end{cases}$$

where w and b are the variances of the weights and bias parameters in the first layer and σ_f^2, which can be interpreted the same way as the signal variance in stationary GPs, stems from the variance on the weights in the second layer of the network. Making w a vector allows for different variance parameters for each input, introducing ARD to the model. More recently, Tsuchida et al. (2018) derived the kernel equivalent to the leaky ReLU activation function. All of the kernels derived from neural networks are said to be non-stationary. This means that instead of depending only on the distance between two points, the kernel depends on the value of the inputs themselves.

Kernel on a circular domain

The kernel functions considered up to this point apply only in Euclidean space. If we wish to consider inputs distributed on a manifold embedded in Euclidean space, we must define an appropriate kernel function. One such example is the definition of a GP on a circular domain. This kernel function requires the consideration of the geodesic distance between two regions, which is given by $d(\gamma, \gamma') = \text{acos}(\cos(\gamma - \gamma'))$ where γ and γ' are the circumferential positioning of the two points of interest. The first thought would be to apply a standard kernel, replacing the measure of Euclidean distance with this geodesic distance. However, as studied by Wood (1995), valid kernels in Euclidean space can violate the constraint of positive-definiteness when we replace the measure of Euclidean distance with that of the geodesic distance. For this reason, Padonou and Roustant (2016) considered the use of compact support kernel functions for GPs on circular domains. They proposed the use of the C^2-Wendland function to account for correlation in the circumferential direction:

$$W_c(t) = \sigma_f^2\left(1 + \tau\frac{t}{c}\right)\left(1 - \frac{t}{c}\right)_+^{\tau}, \quad c \in (0, \pi]; \tau \geq 4 \tag{3.43}$$

where c, τ and σ_f^2 are the kernel hyperparameters and t is the geodesic distance between two strain regions such that $0 \leq t \leq \pi$. The effect of c and τ, can be found in Figure 3.12, showing samples from the GP prior assuming different values of the hyperparameters. From this figure we see that increasing c, which defines the compact support of the kernel, or decreasing τ, which can be considered as a standard smoothness parameter, increases the function smoothness.

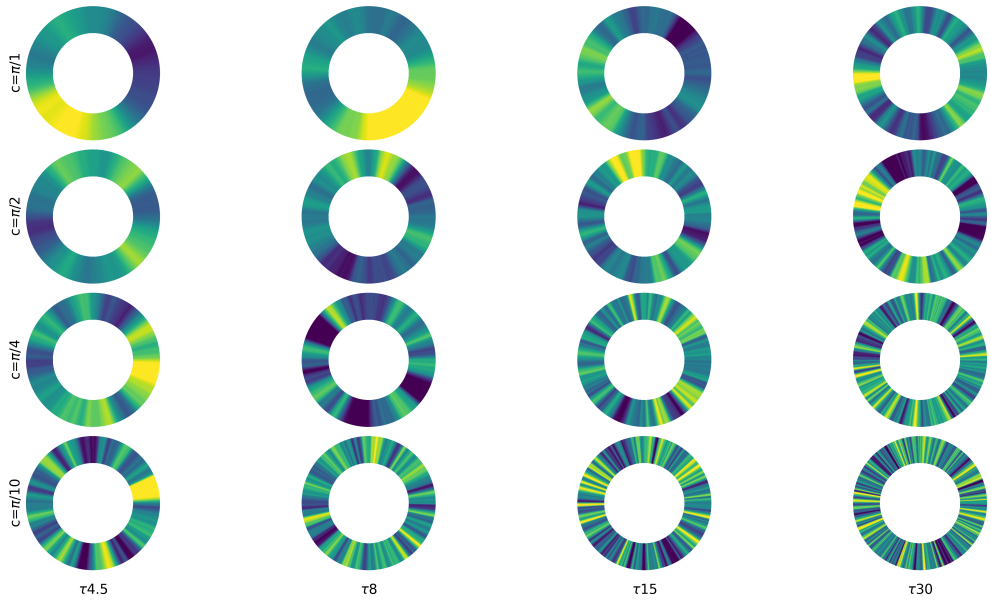


Figure 3.12: Samples from a GP with polar kernel for different values of the hyperparameters.

Nonstationary GP kernel

While the kernels derived from neural networks allow for nonstationarity in the function, we can choose to model this nonstationarity explicitly by introducing a lengthscale function. Gibbs (1997) proposed the following kernel and proved its positive definiteness:

$$k(\theta, \theta') = \sigma(\theta)\sigma(\theta')\sqrt{\frac{2l(\theta)l(\theta')}{l(\theta)^2 + (\theta')^2}} \exp\left(- \frac{(\theta - \theta')^2}{l(\theta)^2 + l(\theta')^2} \right) \tag{3.44}$$

This kernel has more recently been used by Heinonen et al. (2016), where they modelled the signal variance, lengthscale and noise variance with GP functions and sampled the parameters using HMC. In the supplementary material of their paper, Heinonen et al. (2016) proposed the following multi-dimensional extension of the kernel:

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sigma(\boldsymbol{\theta})\sigma(\boldsymbol{\theta}')\sqrt{\frac{2l(\boldsymbol{\theta})l(\boldsymbol{\theta}')}{l(\boldsymbol{\theta})^2 + (\boldsymbol{\theta}')^2}} \exp\left(- \frac{||\boldsymbol{\theta} - \boldsymbol{\theta}'||^2}{l(\boldsymbol{\theta})^2 + l(\boldsymbol{\theta}')^2} \right) \tag{3.45}$$

While a slightly different form was proposed by Gibbs (1997):

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sigma_f^2 \prod_i \sqrt{\frac{2 l_i(\boldsymbol{\theta}; \boldsymbol{\phi}) l_i(\boldsymbol{\theta}'; \boldsymbol{\phi})}{l_i(\boldsymbol{\theta}; \boldsymbol{\phi})^2 + l_i(\boldsymbol{\theta}'; \boldsymbol{\phi})^2}} \exp\left(- \sum_i \frac{(\theta_i - \theta_i')^2}{l_i(\boldsymbol{\theta}; \boldsymbol{\phi})^2 + l_i(\boldsymbol{\theta}'; \boldsymbol{\phi})^2} \right) \tag{3.46}$$

Similar to the nonstationary kernel, Gibbs (1997) proposed using a nonlinear map to transform the input points into a space where the function is stationary. For this purpose, they proposed the use of radial basis functions, and a similar approach has recently been used by Zammit-Mangion et al. (2021). The idea to transform the input space was further explored by Calandra et al. (2016) who suggested the use of a neural network to transform the inputs. The benefit of the approaches used in the former two studies is their use of one-to-one mappings, which is generally not a property of neural network models. The one-to-one property would be particularly useful in design problems if we wished to generate our design in the warped space and recover the corresponding values in the original space. This will be further discussed in Chapter 8.

3.4.3 Multivariate Gaussian processes

Often, we are interested in modelling multiple functions from related processes. For instance, the simulator of the LV model produces 24 circumferential strains and an end diastolic volume, all of which we expect to be highly correlated. Multivariate (also called multi-output) Gaussian processes attempt to model the correlation between these processes, introducing a dependence of function $f_i(\cdot)$ on y_l for $l \neq i$. One approach is to assume an underlying latent process, $\mathbf{x}_l(\cdot)$, through which all the signals are related as a weighted sum. Under this framework, which is called the linear model of coregionalisation, modelling of this latent process as a GP leads to a GP prior over the outputs of the system since GP priors are closed under linear maps. Moreover, cross-covariance between the outputs is induced by the autocovariance of this underlying latent process, providing a dependency between the outputs of the simulator. More details on this approach, and extensions to convolutional processes, can be found in Alvarez and Lawrence (2009). Requeima et al. (2019) proposed an autoregressive approach, where the outputs are regressed on each other. This idea is similar to the autoregressive approach proposed for multi-fidelity modelling by Perdikaris et al. (2017). However, determining an optimal ordering of the outputs for training the autoregressive models could be troublesome with high dimensional outputs.

Conti and O'Hagan (2010) limit the multioutput approach to situations where one assumes a single latent process underlying all outputs of the system such that lengthscales are the same for each output. This corresponds to a GP prior over the outputs of the

simulator $\boldsymbol{y} = \boldsymbol{\zeta}(\cdot)$ as follows:

$$\boldsymbol{\zeta}(\cdot)|\mathbf{B}, \Sigma, \mathbf{R} \sim \mathcal{N}(\mathbf{m}(\cdot), k(\cdot,\cdot)\boldsymbol{\Sigma}) \tag{3.47}$$

where $k(\cdot,\cdot)$ is a kernel function providing the spatial correlation, $\boldsymbol{\Sigma}$ provides the covariance between the outputs of the simulator and $\mathbf{m}(\cdot)$ is a mean function. The diagonal matrix $\mathbf{R}$ stores the lengthscale parameters of k in its diagonal $\mathbf{r}$. The covariance structure leads to the following:

$$\text{cov}(\zeta_1(\boldsymbol{\theta}_3), \zeta_2(\boldsymbol{\theta}_4)) = k(\boldsymbol{\theta}_3, \boldsymbol{\theta}_4)\boldsymbol{\Sigma}_{12} \tag{3.48}$$

where $\zeta_l(\cdot)$ is the lth simulator output. Considering a simulator $\boldsymbol{\zeta}$ with q outputs, we obtain the emulator training set by evaluating the simulator at a set of n inputs $\boldsymbol{\Theta}$ and storing the simulations in a matrix $\mathbf{Y}$. The covariance is expressed in the following way:

$$\text{cov}(\text{vec}(\mathbf{Y}), \text{vec}(\mathbf{Y})) = \boldsymbol{\Sigma} \otimes \mathbf{K}_{\boldsymbol{\Theta}\boldsymbol{\Theta}} \tag{3.49}$$

where $\text{vec}(\cdot)$ is the vectorised version of the matrix, $\mathbf{K}_{\boldsymbol{\Theta}\boldsymbol{\Theta}}$ is the $n \times n$ matrix containing the kernel evaluations over $\boldsymbol{\Theta} \times \boldsymbol{\Theta}$ and $\otimes$ is the Kronecker product. This makes explicit the constant lengthscale within each of the different outputs—only the cross-covariance changes. The posterior distribution for this model can be found in Appendix B.1.

3.4.4 Gaussian process approximations

The flexibility of Gaussian process models stems from their ability to grow in complexity as new data points are added to the training set. Although this property makes them perfect interpolators, the downside is that the computational complexity also grows with the training set, both for training and prediction. For a training set of size n, the training costs are $\mathcal{O}(n^3)$ and for prediction, costs are $\mathcal{O}(n^2)$[1]. For this reason, various approximations to full GPs have been developed, with two of the most popular methods being sparse and local GPs.

When using local GPs (Gramacy and Apley, 2015), a small subset of the available data is used to train a model. If we are interested in prediction at point θ, a popular choice is to use the nearest neighbours from the training set to train a model and predict at θ. Noting the suboptimality of this approach, Gramacy and Apley (2015) proposed taking distant points into account to minimise mean squared predictive error. Although these methods do reduce the costs of GP prediction, they only provide a locally emulated surface. If we wish to use the emulator in Bayesian model calibration, we need a global model for which

[1] Assuming precomputation of the inverse covariance matrix. Under the same assumption, the computational cost of the predictive mean is $\mathcal{O}(n)$, as shown in Appendix A.1.

sparse GPs (SGP) are more appropriate. The idea in sparse GP regression is to funnel the information from n data points through a set of m inducing points (Quiñonero Candela and Rasmussen, 2005; Titsias, 2009; Bauer et al., 2016). An important innovation in the development of SGPs was the introduction of inducing points that can be freely optimised along with the hyperparameters of the GP kernel (Snelson and Ghahramani, 2006). This idea was used by Titsias (2009), who treated the inducing points as variational parameters that can be optimised using variational inference (Blei et al., 2017). In variational inference, we introduce a family of parametric distributions on some variables of interest, $\mathbf{z}$, and optimise an objective referred to as the evidence lower bound (ELBO), with respect to the parameters of the family of distributions (Blei et al., 2017). As the name suggests, the ELBO lower bounds the marginal likelihood. To introduce this to the sparse GP, let $\mathbf{u}$ be a vector of inducing variables. A variational distribution, $q(\mathbf{u})$, is introduced with parameter to be optimised using variational inference. Letting $\boldsymbol{y}$ be observed values and $\mathbf{f}$ the underlying function values, the ELBO is derived by application of Jensen's inequality:

$$\begin{aligned}
\log p(\boldsymbol{y}) &= \log \int\int p(\boldsymbol{y}, \mathbf{f}, \mathbf{u}) d\mathbf{f} d\mathbf{u} \\
&= \log \int\int \frac{q(\mathbf{f}, \mathbf{u})}{q(\mathbf{f}, \mathbf{u})} p(\boldsymbol{y}, \mathbf{f}, \mathbf{u}) d\mathbf{f} d\mathbf{u} \\
&\geq \int\int q(\mathbf{f}, \mathbf{u}) \log \frac{p(\boldsymbol{y}, \mathbf{f}, \mathbf{u})}{q(\mathbf{f}, \mathbf{u})} d\mathbf{f} d\mathbf{u} \\
&= \int\int p(\mathbf{f}|\mathbf{u}) q(\mathbf{u}) \log \frac{p(\boldsymbol{y}|\mathbf{f}) \cancel{p(\mathbf{f}|\mathbf{u})} p(\mathbf{u})}{\cancel{p(\mathbf{f}|\mathbf{u})} q(\mathbf{u})} d\mathbf{f} d\mathbf{u} = \mathcal{F}(q)
\end{aligned}$$

Maximising this lower bound with respect to the variational distribution, q, gives:

$$\mathcal{F} = \log \mathcal{N}(\boldsymbol{y}|\mathbf{0}, \sigma^2 \mathcal{I} + \mathbf{Q}_{nn}) - \frac{1}{2\sigma^2} \mathrm{tr}(\mathbf{K}_{\boldsymbol{\theta}\boldsymbol{\theta}} - \mathbf{Q}_{nn}) \tag{3.50}$$

where $Q_{nn} = \mathbf{K}_{\boldsymbol{\theta}u} \mathbf{K}_{uu}^{-1} \mathbf{K}_{uu}$ and a matrix $\mathbf{K}_{xx}$ corresponds to the matrix containing kernel evaluations over $x \times x$. We can then learn the inducing inputs and kernel hyperparameters by optimising (3.50). Sparse GP training costs are $\mathcal{O}(nm^2)$ where m is the number of inducement points and n is the number of training points. With precomputation (see Appendix A.2, the predictive mean is $\mathcal{O}(m)$ and predictive variance is $\mathcal{O}(m^2)$

3.5 Dimension reduction

Consider a dataset used to train an emulator, consisting of a set of input-output pairs. Focusing on the matrix in which the inputs of this dataset are stored $\mathbf{X}$, this is two dimensional (say $n \times p$) with the number of rows n representing the number of training

locations and the number of columns p corresponding to the dimension of the input space. The methods discussed in Section 3.4.4 target the problem of large n in the dataset, by making more efficient use of the data. One approach considered was the sparse GP, which was said to funnel the information in the n data points through a set of $m < n$ inducement points. In the current section, the interest shifts to the issue of large p problems, and how we can funnel the information of the p dimensions through some lower dimensional representation.

In the simplest case, we can assume a map from the original p dimensional space through the following linear transformation:

$$\mathbf{r} = \mathbf{X}\mathbf{V} \tag{3.51}$$

where $\boldsymbol{X}$ has dimension $n \times p$ and $\mathbf{r}$ has dimension $n \times q$, where $q < p$. As a result of the transformation, we recover a projection of the data, $\boldsymbol{X}$, in a lower dimensional space of dimension q. How we recover the map into this lower dimensional space depends on how we measure levels of redundancy in the data, which is the information in the original dataset that will be removed when we funnel the data through the transformation.

Principal component analysis (PCA) assumes that any redundant structure in the data can be measured as correlation between the different dimensions. With PCA we find a set of vectors, $\mathbf{V}$, onto which the data can be projected to remove these correlations between the dimensions. To do so, a low dimensional projection is obtained such that the variance of the data along this projection is maximal, which is met by optimising the following criterion:

$$\max_{\mathbf{v}} \operatorname{cov}(\boldsymbol{X}\mathbf{v}) = \max_{\mathbf{v}} \mathbf{v}\boldsymbol{\Sigma}\mathbf{v}^T \tag{3.52}$$

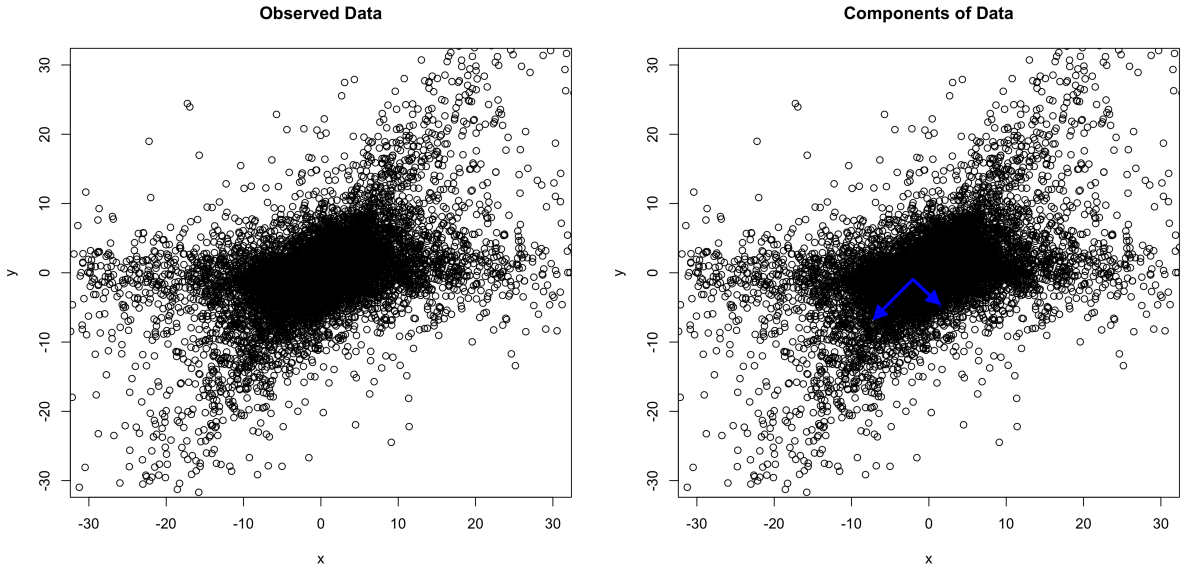
where $\mathbf{v}$ is one of the columns of $\mathbf{V}$ and $\boldsymbol{\Sigma}$ is the covariance matrix of the data, obtained as:

$$\boldsymbol{\Sigma} = \frac{1}{n}((\mathbf{X} - \bar{\mathbf{x}})^\top (\mathbf{X} - \bar{\mathbf{x}}))$$

We find, by solving (3.52) subject to a constraint that the vector $\mathbf{v}$ has unit norm, that the first principal component is provided by the eigenvector of $\boldsymbol{\Sigma}$ with the largest corresponding eigenvalue. Generalising to the $j\,th$ principal component, we solve (3.52) subject to the added constraint of the principal component being uncorrelated with the previous $j - 1$ components. For further details on the derivation of PCA, see section 6.1.1 of Hyvarinen et al. (2002).

The previous description of PCA alludes to one of its more implicit limitations. Namely, the mutual orthogonality of the principal components. Once we have found the first m

components, the constraint that the next has zero correlation leads to an orthogonal projection relative to the previous components. Obviously, this solution could be suboptimal as observed in the often concocted example displayed in Figure 3.13 where two signals are mixed to obtain two non-orthogonal clouds of points. The projection obtained using PCA fails to find the two obvious directions of principal variation since the second component must be found to be orthogonal relative to the first. This orthogonality constraint is rooted in the assumption that any dependence between dimensions is captured by, at the highest order, the correlation, which requires also an implicit assumption of Gaussianity, for which stochastic dependence and correlation are equivalent because the Gaussian distribution does not depend on higher order moments. The solution to this problem is then to consider higher order measures of dependence than correlations. Independent component analysis (ICA), which was formulated for the mathematically similar task of signal decomposition (Hyvarinen et al., 2002), accounts for these higher order dependencies through a measure of kurtosis. However, this comes at the cost of added computational complexity since the recovery of the basis now requires iterative optimisation rather than closed form matrix calculations. Moreover, the recovered components do not have the natural ranking that is provided by PCA.



(a) (b)

Figure 3.13: A mixture of two t-distributions is frequently used as a counter example to PCA where we have two signals that clearly do not lie orthogonal to one another.

All of the restrictions of PCA can be overcome using a neural network model referred to as an autoencoder. A key uniqueness of these models, compared with standard neural networks, is that the same quantities are used as the inputs and targets of the model. A simple autoencoder, with two layers of weights and a single bottleneck layer, is visualised

on the left of Figure 3.14. The hidden layer is referred to as a bottleneck layer, with the portion of the network between the inputs and bottleneck the encoder and the proceeding part of the network referred to as the decoder.. These names are intuitive: the encoder finds a projection into a lower dimensional space that optimally represents the data while the decoder reconstructs (an approximation of) the original shape. Training this model as one neural network, optimised using a mean squared error objective function, means that the encoder and decoder work together as the encoder learns a projection into a new space that can then be optimally reconstructed by the decoder. This leads to an important contrast between autoencoders and standard feedforward neural networks. In the latter, we generally wish to learn a map from inputs to outputs that is nonlinear. In an autoencoder, while the optimal transformation into the low dimensional space (the encoder) may be nonlinear, the decoder acts to linearise any nonlinearities in order to reconstruct the original data as well as possible.

With linear activation functions in the hidden layer, the transformation performed by an autoencoder has the following form:

$$\mathbf{h} = \mathbf{x}\mathbf{W}_1$$
$$\hat{\mathbf{x}} = \mathbf{h}\mathbf{W}_2$$

where the transformation onto $\mathbf{h}$ is of the same form as the projection onto the principal components in PCA. While the autoencoder is trained without any assumption of orthogonality on the weight matrix, it can be shown that the resulting transformation is equivalent to the one obtained using PCA. A more surprising equivalence is the following: even with the introduction of nonlinear activation functions to the bottleneck layer of the network on the left of Figure 3.14, the optimal solution is still no better than PCA (Bishop, 2006, Section 12.4.2). To improve upon linear methods we must employ a hidden layer in both the encoder and decoder, as shown in Figure 3.14. This makes both the encoder and decoder universal approximators, which ensures that they can model any suitably smooth function (Hornik, 1991).

For visualisation purposes, t-Distributed Stochastic Neighbour Embedding (t-SNE) van der Maaten and Hinton (2008) provides a nonlinear projection of the data into a lower dimensional space by minimising a KL divergence between conditional probabilities of nearest neighbours in both spaces. However, since an explicit transformation is never learned, the method is limited to only data visualisation.

The final dimensionality reduction approach is one that is specific to the shape of the left ventricle. In standard dimension reduction routines, we begin by subtracting the mean mesh shape from the dataset. The methods then learn to adjust the shape around

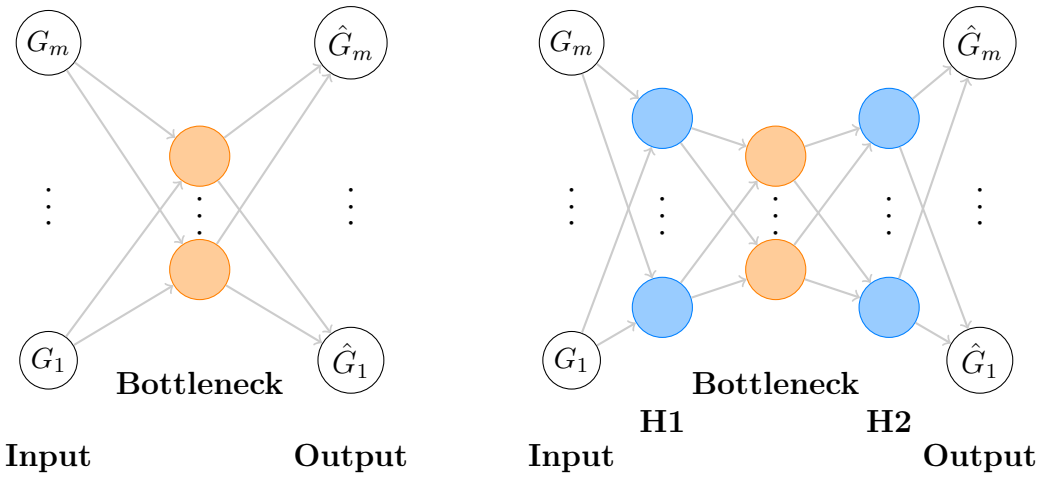


Figure 3.14: Visualisations of a shallow autoencoder (left) and deep autoencoder (right). The bottleneck layer provides the low-dimensional representation of G.

this mean mesh, making the learning process easier while also restricting the models to the relevant shape. An alternative approach to this is to consider a fixed parametric representation of the LV geometry, which can be tuned to fit a new subject specific LV geometry. This allows us to introduce domain specific knowledge to the learning problem at the cost of restricting ourselves to a pre-defined class of geometries. This idea was used to represent LV geometries by Di Achille et al. (2018) where they described the LV geometry using a symmetric 6 parameter model given by[2]:

$$\begin{aligned}
\rho_{\text{epi}} &= R_b(e\cos(\psi) + (1-e)(1-\sin(\psi))), \\
\eta_{\text{epi}} &= -Z\sin(\psi), \\
\rho_{\text{endo}} &= (R_b - L)(e\cos(\psi_{\text{endo}}) + (1-e)(1-\sin(\psi_{\text{endo}}))), \\
\eta_{\text{endo}} &= -(Z-H)\sin(\psi_{\text{endo}}),
\end{aligned} \tag{3.53}$$

where ρ denotes the radial coordinate, η the axial coordinate and the parameters of interest, which we collect in the vector $\boldsymbol{\kappa} = (R_b, Z, L, H, e, \psi_0)$, where ψ_0 is the lower bound on ψ. These parameter are optimised to fit the model to a specific LV geometry, where volumes can be calculated using a discretisation of space and assigning wall and cavity indicators to each point.

[2]This is a slightly adjusted version of the original formulation in Di Achille et al. (2018), which may have been subject to some typos.

Chapter 4

Emulation of a fixed left ventricle geometry simulator

The work in this chapter acts as a first step into the use of statistical emulation for efficient prediction of the left ventricle behaviour in diastole. Considering emulation of the LV model for a fixed LV geometry, this is best framed as a proof of concept study that will aim to show that for a fixed LV geometry, statistical emulation provides a good approximation to the left ventricle simulator. Firstly, this work will assess the use of output emulation versus loss emulation for parameter estimation before comparing the use of three competing methods for emulation: univariate Gaussian processes, a multivariate Gaussian process and a neural network. With the neural network approach, the lower computational costs associated with neural networks allows efficient optimisation of a global model in the order of seconds. For the two GP approaches, local approximations are required, allowing more efficient prediction at a new test point but with the absence of any global function approximation. The work in this chapter is largely based on work published in Noè et al. (2019). The local GP results presented in this Chapter were taken from Noè et al. (2019), where Umberto Noè implemented the local univariate approach and I implemented the local multivariate approach. Although this publication contained some results on sparse GPs, these have been updated for use in this thesis. The neural network is a more recent update that was not included in the original publication, and was implemented by myself.

4.1 Background on the problem

4.1.1 The fixed geometry simulator

Recall the mathematical model that was introduced in Section 2.2. The multi-scale nature of this model, mapping properties at the tissue level to properties at the observable level, allows us to relate the stiffness of the passive myocardium, represented by the Holzapfel-

Ogden law, with quantities measured in the clinic. As was shown in Figure 2.2, these parameters are combined with the subject-specific left ventricle (LV) geometry and end diastolic pressure (EDP) to allow for patient-specific simulation of LV dynamics in diastole, paving the way for personalised non-invasive measurement of myocardial tissue health. Clinical application of these procedures introduces time constraints to the inference problem, which poses a problem in the context of cardiac modelling where computational costs are typically quite high. Requiring computation times of around ten minutes per forward simulation, implementation of this model in an iterative optimisation procedure for parameter inference can be expected to require around 2 days to run. More sophisticated methods from machine learning, such as Bayesian optimisation (Shahriari et al., 2016; Romaszko et al., 2021), can expedite this procedure but the costs are still more applicable to research purposes as opposed to clinical applications.

The problem with the optimisation procedures discussed above is the manner in which they use the simulator, with all the expense incurred during inference. Statistical emulation (Conti et al., 2009; Conti and O'Hagan, 2010) offers an alternative paradigm, with the burden of the forward simulations incurred prior to the stage of model calibration (Kennedy and O'Hagan, 2000). Running a large batch of forward simulations, we can train an approximation to the simulator that can then be used in the optimisation routine. In theory, this approximation has unbounded accuracy: adding extra training simulations for training the emulator will lead to increasingly accurate approximations. For the LV simulator, selection of training points is itself a problem because there is no method for sampling realistic LV geometries in their original space. One solution to this problem is to consider a low dimensional representation of the LV geometry. This will be considered in future chapters of this thesis, but in the current chapter the LV geometry is considered fixed, allowing us to focus on the map from material parameters to model outputs.

Personalisation of the model to new subjects does not only rely on the LV geometry. EDP of the blood within the LV cavity will likely be different for each individual patient. Inclusion of this variable in the emulator would allow us to measure this parameter in the clinic and feed it into the model during parameter estimation to allow for more personalised estimation. Although methods do exist for measuring the EDP, the invasiveness of the procedure ensures that in all but the most severe of cases (where the invasive measurement can be justified), this data will not be available. In the literature, a common approach is to fix the EDP to a population based average pressure during inference (Palit et al., 2018a). Inevitably, this will introduce a bias to the parameter estimation when considering inference conditional on measured data, but the task of inferring EDP along with material properties is ill-posed. On top of this, for healthy volunteers we can be confident that a population average provides a close approximation to the actual pressure in the cavity and Palit et al. (2018a) showed that only a small bias is incurred when dealing with pressure

around the healthy in vivo range (8-12 mmHg). For this reason, the EDP will not be included in the emulator's domain, instead fixing it to 8 mmHg (Bouchard et al., 1971) during training and any subsequent parameter estimation.

4.1.2 A reduced parameterisation

Fixing the LV geometry and EDP simplifies the task of statistical emulation of the LV simulator. Allowing us to isolate the map between material parameters and outputs of the model, this *fixed geometry LV simulator* provides the perfect setting for a proof of concept study, which can then be followed by the task of generalising the inference problem to new subjects. The resultant model can be summarised by the following map:

$$\zeta : \boldsymbol{\theta} \in \mathbb{R}^d \mapsto \boldsymbol{y} \in \mathbb{R}^{25} \tag{4.1}$$

where $\boldsymbol{y} \in \mathbb{R}^{25}$ consists of the LV cavity volume at end diastole (LVV) and 24 circumferential strains (see Figure 2.3 for a description), which can be measured from CMR scans in the clinic. In its original form, which is found in (8.1), the H-O model relies on $d = 8$ material parameters. Such a parameterisation exhibits poor identifiability (Gao et al., 2015) and often in the literature a reparameterisation is adopted (see Krishnamurthy et al. (2013b); Hadjicharalambous et al. (2015); Palit et al. (2018a) for some examples). The work in this chapter will adopt a similar approach by introducing four scaling constants that relate the Holzapfel-Ogden parameters with a set of population values from healthy volunteers:

$$\begin{aligned} a &= \theta_1\, a_0 & b &= \theta_1\, b_0 \\ a_{\mathrm{f}} &= \theta_2\, a_{\mathrm{f0}} & a_{\mathrm{s}} &= \theta_2\, a_{\mathrm{s0}} \\ b_{\mathrm{f}} &= \theta_3\, b_{\mathrm{f0}} & b_{\mathrm{s}} &= \theta_3\, b_{\mathrm{s0}} \\ a_{\mathrm{fs}} &= \theta_4\, a_{\mathrm{fs0}} & b_{\mathrm{fs}} &= \theta_4\, b_{\mathrm{fs0}} \end{aligned} \tag{4.2}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_4) \in [0.1, 5]^4$ are now the four parameters to be inferred from in vivo data, while a_0, b_0, a_{f0}, a_{s0}, b_{f0}, b_{s0}, a_{fs0}, b_{fs0} are reference values from the published literature (Gao et al., 2017)[1]. This parameterisation was motivated by the results of Gao et al. (2015).

4.2 Emulation methodology

Given experimental data $\mathbf{y}^*$, consisting of $J - 1$ circumferential strains and the LVV, the goal is to find the parameters $\hat{\boldsymbol{\theta}}$ at which the simulator prediction $\boldsymbol{\zeta}(\hat{\boldsymbol{\theta}})$ best approximates

[1]The reference values are, up to 2 decimal places: $a_0 = 0.22$, $b_0 = 1.62$, $a_{\mathrm{f0}} = 2.43$, $a_{\mathrm{s0}} = 0.56$, $b_{\mathrm{f0}} = 1.83$, $b_{\mathrm{s0}} = 0.77$, $a_{\mathrm{fs0}} = 0.39$, and $b_{\mathrm{fs0}} = 1.70$.

the data, according to some objective function. Assuming that the data are standardised, we can make the following assumption of iid Gaussian noise:

$$p(\boldsymbol{y}^*|\boldsymbol{\zeta}(\boldsymbol{\theta}), \sigma^2) = \prod_{j=1}^{J} \mathcal{N}(y_j^*|\zeta_j(\boldsymbol{\theta}), \sigma^2), \tag{4.3}$$

where σ^2 is the noise variance. This leads to a log-likelihood function of the following form:

$$l(\boldsymbol{y}^*|\boldsymbol{\theta}) = -\frac{J}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{j=1}^{J}(y_j^* - \zeta_j(\boldsymbol{\theta}))^2. \tag{4.4}$$

The noise variance has no impact on our point estimate of the parameters, so optimisation of the log-likelihood can be replaced by optimisation of the mean squared error or residual sum of squares. In the proceeding work, an RSS objective function will be used:

$$L(\boldsymbol{\theta}) = \frac{1}{J}\sum_{j=1}^{J}(y_j^* - \zeta_j(\boldsymbol{\theta}))^2. \tag{4.5}$$

For the LV simulator, optimisation of (4.5) is computationally intensive, requiring days of computation time. Given a set of input-output pairs from the simulator:

$$\mathcal{D} = \{\boldsymbol{\theta}_i, \boldsymbol{y}_i\}_{i=1}^{n}, \quad \boldsymbol{y}_i = \boldsymbol{\zeta}(\boldsymbol{\theta}_i) \tag{4.6}$$

we can approximate the loss evaluation in (4.5) using a statistical emulator. Several different emulation approaches will now be discussed, before considering in detail how these can be used to expedite the model calibration procedure.

4.2.1 Univariate Gaussian process

A general background on Gaussian processes (GPs) was provided in Section 3.4. This involved a discussion of the computational costs of GP modelling, and the fact that the non-parametric nature of GPs ensures that these costs grow as the dataset size increases. As a result, for large training sets we require approximations to the full GP and one such approach is a local GP. An approach based on M-nearest-neighbours in input space was proposed in Gramacy and Apley (2015). This method uses the standard GP methodology described in Section 3.4.1, but subsetting the training data. Whenever we require a prediction at a given input, we find the training inputs representing the M nearest neighbours in the input domain, which will form the local set of training inputs, and the corresponding outputs will represent the local training outputs. Note that every time we query a prediction at a different input, the training sets need to be re-computed and the GP needs

to be re-trained. However, because of the small number of neighbours, which is usually much smaller than 1000, this method is computationally fast; see Gramacy and Apley (2015) for further details.

Gramacy and Apley (2015) further discussed adding a fixed number of distant points in order to help in the estimation of the lengthscale parameters, but this comes with extra computational costs required by the iterative choice of which point to add to the set of neighbours. Given the time limitations required by the goal of real-time clinical decision support systems, this approach will not be pursued. Furthermore, this is mostly relevant when the interest lies in building predictive models able to make good predictions when the training data are distant from each other. Since we are working on a compact set that is very densely covered by the Sobol sequence, this is not necessary. For training data consisting of intput-output pairs $\mathcal{D} = \{(\boldsymbol{\theta}_1, y_1), \ldots, (\boldsymbol{\theta}_n, y_n)\} = \{\boldsymbol{\Theta}, \boldsymbol{y}\}$, we can summarise the algorithm as follows:

Algorithm 1 Predicting output j from an M nearest neighbour local Gaussian process at $\boldsymbol{\theta}^*$

1: Find the indices $\mathcal{I}(\boldsymbol{\theta}^*)$ of the points in $\boldsymbol{\Theta}$ having the M smallest Euclidean distances from $\boldsymbol{\theta}^*$;
2: Training inputs: $\tilde{\boldsymbol{\Theta}} = \{\boldsymbol{\theta}'_1, ..., \boldsymbol{\theta}'_{\mathrm{M}}\} = \{\boldsymbol{\theta}_i : i \in \mathcal{I}(\boldsymbol{\theta}^*)\}$;
3: Training outputs: $\tilde{\boldsymbol{y}} = \{y_i : i \in \mathcal{I}(\boldsymbol{\theta}^*)\}$;
4: Train a GP using the data $\mathcal{D}_{\mathrm{M}}(\boldsymbol{\theta}^*) = \{\tilde{\boldsymbol{\Theta}}, \tilde{\boldsymbol{y}}\}$;
5: Predictive mean: $\mu(\boldsymbol{\theta}^*) = m(\boldsymbol{\theta}^*) + \boldsymbol{k}(\boldsymbol{\theta}^*)^\top [\boldsymbol{K} + \sigma^2 \boldsymbol{I}]^{-1} (\tilde{\boldsymbol{y}} - \boldsymbol{m})$;
6: Predictive variance: $\nu(\boldsymbol{\theta}^*) = k(\boldsymbol{\theta}^*, \boldsymbol{\theta}^*) - \boldsymbol{k}(\boldsymbol{\theta}^*)^\top [\boldsymbol{K} + \sigma^2 \boldsymbol{I}]^{-1} \boldsymbol{k}(\boldsymbol{\theta}^*)$.

In the algorithm above, $\boldsymbol{K} = [k(\boldsymbol{\theta}'_i, \boldsymbol{\theta}'_j)]_{i,j=1}^{\mathrm{M}}$ is the $\mathrm{M} \times \mathrm{M}$ training covariance matrix, the $\mathrm{M} \times 1$ vector of covariances between the training points and the test point is $\boldsymbol{k}(\boldsymbol{\theta}^*) = (k(\boldsymbol{\theta}'_1, \boldsymbol{\theta}^*), \ldots, k(\boldsymbol{\theta}'_{\mathrm{M}}, \boldsymbol{\theta}^*))$ and $\boldsymbol{m} = (m(\boldsymbol{\theta}'_1), \ldots, m(\boldsymbol{\theta}'_{\mathrm{M}}))$ is the $\mathrm{M} \times 1$ prior mean vector. Note that whenever the target parameters change, e.g. during an iterative optimisation, the algorithm has to be repeated. A constant mean function $m(\boldsymbol{\theta}) = c$ and the squared exponential kernel will be considered, as widely used in the emulation of computer codes literature; see e.g. Fang et al. (2006); Santner et al. (2018). The model hyperparameters are estimated by maximising the log marginal likelihood using the Quasi-Newton method, with σ initialised at a small value, $\sigma = 10^{-2}$ (since the simulator function is deterministic). This method was found to be the best in the comprehensive comparison presented in Davies et al. (2019) and represents a benchmark for the current study, along with the expensive optimisation problem solved in Gao et al. (2015).

4.2.2 Multivariate GP

Given the multi-output nature of the simulator, a multivariate model could provide a better approximation. For this purpose, the method of Conti and O'Hagan (2010) will be

used. As discussed in Section 3.4.3, this approach assumes that the lengthscales of the different input dimensions are the same for each of the outputs in the model. For the LV simulator, this is a reasonable assumption to make because the shape of the output function is largely dictated by the H-O model, which is of course the same for each output.

Similar to the univariate case, a local approximation will be made. Instead of using the parameter space nearest neighbours, function space nearest neighbours will be used, so that only one emulator has to be trained. If we are interested in a new test case $\boldsymbol{y}^*$, we must find the K closest training outputs to $\boldsymbol{y}^*$. These training data are then used to train a MOGP, which can be used in place of the simulator to learn the material parameters. Supposing test data $\boldsymbol{y}^*$ and training data $\mathcal{D} = \{(\boldsymbol{\theta}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{\theta}_n, \boldsymbol{y}_n)\} = \{\boldsymbol{\Theta}, \mathbf{Y}\}$, the procedure for predicting at a set of parameters $\boldsymbol{\theta}^*$ is outlined in Algorithm 2. A squared

Algorithm 2 Predicting from a multivariate GP at $(\boldsymbol{\theta}^*, \boldsymbol{y}^*)$

1: Standardise the data, $\boldsymbol{y}^*$
2: Find the indices $\mathcal{I}(\boldsymbol{y}^*)$ of the points in $\mathbf{Y}$ having the M smallest Euclidean distances from $\boldsymbol{y}^*$;
3: Training inputs: $\tilde{\boldsymbol{\Theta}} = \{\boldsymbol{\theta}'_1, \ldots, \boldsymbol{\theta}'_{\mathrm{M}}\} = \{\boldsymbol{\theta}_i : i \in \mathcal{I}(\boldsymbol{y}^*))\}$;
4: Training outputs: $\tilde{\mathbf{Y}} = \{\boldsymbol{y}'_1, \ldots, \boldsymbol{y}'_K\} = \{\boldsymbol{y}_i : i \in \mathcal{I}(\boldsymbol{y}^*)\}$;
5: Train a multivariate output GP using the data $\mathcal{D}_K(\boldsymbol{y}^*) = \{\tilde{\boldsymbol{\Theta}}, \tilde{\mathbf{Y}}\}$;
6: Predictive mean at test point $\boldsymbol{\theta}_*$, see Appendix B.1;

exponential kernel was used for the MOGP and, while optimisation could have been used, the hyperparameters were sampled using Metropolis Hastings, assuming a prior of the form $\pi_R(\mathbf{r}) \propto \prod_{i=1}^{p}(1 + r_i^2)^{-1}$.

4.2.3 Global approaches

Sparse GP

Section 4.2.1 and Section 4.2.2 outlined methods that use local subsets of the data to overcome the computational complexities of Gaussian processes. Rather than fitting to the full global set of data, subsets of data were selected for training a Gaussian process emulator in a particular region of interest. For optimisation, this approach is appropriate because we never move far away from our particular point of interest. However, for some MCMC procedures, where proposals could be far away from the current point, we cannot afford to misrepresent parts of the configuration space. This is especially true if we wish to use HMC, where the method's efficiency stems from its ability to make these well-informed distant proposals. This sampler would inevitably step out of a locally emulated region to one where the likelihood cannot be evaluated accurately. This could affect our final posterior densities and introduce bias to any future decision making based on these inferences.

Sparse GPs (Titsias, 2009; Bauer et al., 2016), which were discussed in Section 3.4.4, offer a global alternative to local GPs. These methods tend to provide good approximation to the mean of the full GP, but we lose the pinching of the variance at the training data that is associated with standard GPs and this leads to less reliable functional uncertainty estimates. In the simulation study of this chapter, the variational sparse GP proposed by Titsias (2009) and discussed in Section 3.4.4 will be used.

Neural networks

In Section 3.4.2, the parallels between neural networks and GPs were discussed. Much of this was based on the work of Neal (1996) and later work by Williams (1997) who showed a correspondence between neural networks with the erf activation function and GPs with the sigmoid kernel. Despite their similarities, neural networks are rarely considered for the purpose of statistical emulation. Part of this neglect inevitably stems from the training process, which typically involves optimisation of the model parameters and no uncertainty quantification. As mentioned in Section 3.3.4, various Bayesian inference schemes have been proposed, but these are computationally expensive with large training sets. For this reason, only an optimised neural network will be considered in the current chapter. The parametric nature of neural networks is such that the full training set can be used to fit the model without any local approximation. The network parameters will be optimised using mini-batch gradient descent with the Adam optimiser and an MSE loss function (see Section 3.3.3 for more details), and the network architecture will be selected using cross-validation, as will be detailed later in this chapter.

4.2.4 Parameter estimation with a statistical emulator

Now that the general problem has been outlined at the beginning of this section, and several emulation strategies have been discussed, we can consider how these will be combined for efficient model calibration. Broadly speaking, the introduction of an emulator to (4.5) falls into two paradigms. The first of these is referred to as *emulation of the loss*. Given the set of data $\mathcal{D}$ and a vector of test observations $\boldsymbol{y}^*$ of length J, we can form a dataset of input-loss pairs:

$$\tilde{\mathcal{D}} = \{\boldsymbol{\theta}_i, L(\boldsymbol{\theta}_i)\}_{i=1}^n, \quad L(\boldsymbol{\theta}_i) = \frac{1}{J}||\boldsymbol{y}^* - \boldsymbol{y}_i||^2$$

Training an emulator on these data gives a surrogate model for the loss function. Assuming that the emulator is deterministic, which is the case with an optimised neural network,

the surrogate loss $\tilde{L}(\boldsymbol{\theta})$ can be optimised to obtain the parameter estimates $\hat{\boldsymbol{\theta}}$:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \tilde{L}(\boldsymbol{\theta}) \tag{4.7}$$

Taking a univariate GP approach to emulating the loss, using for instance Algorithm 1, we treat the unknown loss function as random. A trained GP provides a distribution on functions that is consistent with the observed loss evaluations (see Figure 3.11 for a visualisation). The distribution is defined by a posterior mean and variance, which were expressed in the case of the local GP in steps 5 and 6 of Algorithm 1. This posterior mean gives the best estimate for the unknown true function values under quadratic loss, providing a deterministic approximation to the loss function that can be optimised to estimate the parameters.

An alternative approach is to *emulate the outputs* by training a surrogate model directly on $\mathcal{D}$ from (4.6). Such an approach provides a direct approximation to the simulator that allows a surrogate-based loss function to be used in place of the simulator-based loss from (4.5). If we consider our emulator η to be deterministic, then the surrogate-based loss is formed by replacing the simulator in (4.5) with the emulator:

$$\hat{L}(\boldsymbol{\theta}) = \frac{1}{J}||\boldsymbol{y}^* - \boldsymbol{\eta}(\boldsymbol{\theta})||^2. \tag{4.8}$$

As before, training a GP on the training data gives a distribution of functions that is consistent with the training data (this time the outputs of the simulator). In the current analysis, the posterior mean of these GPs will be used to replace the simulator in the loss function. Appendix B.2 outlines an alternative loss function that accounts for the GP uncertainty, along with a simple simulation study. The issue with such an approach is that we incur the extra computational complexity of the posterior variance of the GP, which is $\mathcal{O}(n^2)$ for each of the J outputs where n is the number of training points. This can slow the parameter estimation substantially if we consider large training sets. For the purpose of in-clinic parameter estimation, it therefore seems justified to consider only the posterior mean.

4.3 Building and applying the emulator

Building and applying the emulator is a process in three phases. In the first phase, a set of constitutive parameter vectors $\boldsymbol{\theta}$ is generated from a Sobol sequence. In general, these are the 8D vectors from the Holzapfel-Ogden model defined in Section 2.2.2. However, based on the discussion in Section 4.1.2, the dimension has been reduced to 4D, as defined in (4.2). For each parameter vector $\boldsymbol{\theta}$, the dynamical equations from Section 2.2.3 are solved numerically with finite element discretisation, ideally by massive parallelisation, to

obtain the corresponding data vectors $\boldsymbol{y}$: these are 25D vectors of 24 peak circumferential strains and the LVV; see the beginning of Section 2.2 for details. In the second phase, a regression model is fitted to the training set obtained in the first phase. Depending on the mode of operation, this training set can either consist of a set of independent scalar outputs, output vectors, or loss functions. In the third phase, an iterative optimisation algorithm is applied to the emulated loss function, to obtain the parameter estimate for new test data. In a clinical context, Phase 1 is carried out before using the emulator in the clinic. Phase 3 is the way the emulator is used as a decision support system in the clinic, with the test data corresponding to data obtained from a patient. Phase 2 depends on the mode of operation. When emulating the output with a global model, this phase can be completed before using the emulator in the clinic. For the other approaches, the emulator has to be created "on demand" after obtaining the patient data.

4.3.1 Phase 1: Parallel simulations from the model

In order to build the emulator we need a set of training runs $\mathcal{D} = \{(\boldsymbol{\theta}_i, \boldsymbol{y}_i)\}_{i=1}^n$, for parameter vectors $\boldsymbol{\theta}_i$ and associated model outputs $\mathbf{y}_i$. In this study, the training inputs $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n]^\top$ represent $n = 10,000$ points from a Sobol sequence in $[0.1, 5]^4$, where the domain has been chosen to represent typical parameter ranges from the literature (Gao et al., 2017). The corresponding outputs $\boldsymbol{y}_i$ are obtained using the model detailed in Section 2.2.

The size of training set used here is typical of all the work in this thesis, where the number of simulations will be in the order of tens of thousands. While this incurs a large computational cost at the time of simulation, this is all performed before the parameter estimation. Importantly, this means that the added computational burden of the forward simulations does not directly contribute to the cost of parameter estimation in the clinic. When using a GP emulator, a direct impact of the training set size on parameter estimation stems from the predictive cost of the emulator, which grows with the training set size. In particular, with training sets containing 10,000 points we rely on approximations to the full GP. While this appears to provide evidence in favour of more sophisticated designs, there is more to be considered here. If we are set on using a GP model for emulation, efficient parameter estimation with a global GP requires training set sizes in the order of hundreds. In preliminary studies that looked at more effective designs for a global GP model, a training set in the order of 1,000 was required for accurate emulation. At this point, we begin to push the computational costs beyond those that would be appropriate for in clinic inference and must consider approximations. For this reason, taking a dense coverage that ensures accurate GP approximations seemed like the best choice for the simulator considered in this work, which requires around ten minutes of computation time. While the same arguments apply in principle for more complicated cardiac models

such as the bi-ventricle model, generation of a training set of similar size to the ones used in this thesis would be infeasible. In this circumstance, sequential methods can be used to more densely fill regions of space where the function is more complex.

4.3.2 Phase 2: Training the emulator

As discussed previously, three different GP emulation strategies will be considered along with a neural network. The parametric nature of the neural network means it can be trained on the entire $n = 10,000$ training points. The GPs require the construction of an M $\times$ M training covariance matrix $\mathbf{K} = [k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)]_{i,j=1}^{\mathrm{M}}$. The local GPs also require an M $\times$ 1 vector of covariances between the training points and the test point $\boldsymbol{k}(\boldsymbol{\theta}^*) = (k(\boldsymbol{\theta}_1, \boldsymbol{\theta}^*), \ldots, k(\boldsymbol{\theta}_{\mathrm{M}}, \boldsymbol{\theta}^*))$, while the sparse GP with M inducement points requires the M $\times$ 1 vector of covariances between the inducement points and the test point. Note that this covariance vector must be evaluated at each new point during optimisation while the covariance matrix and its inverse can be evaluated beforehand (see Appendix A.1).

For the univariate output GPs, a constant mean function $m(\boldsymbol{\theta}) = c$ is adopted. For the multi-output GP, the approach of Conti et al. (2009) and Conti and O'Hagan (2010) is followed, using a low-order polynomial $m(\boldsymbol{\theta}) = \boldsymbol{b}^\top \boldsymbol{h}(\boldsymbol{\theta})$ with $\mathbf{h}(\boldsymbol{\theta}) = (1, \boldsymbol{\theta})$. The choice of kernel function depends on the GP. For the local GP implementations (both univariate and multivariate) the stationary squared exponential kernel with ARD is used. This kernel is widely used in the emulation of computer codes literature, see e.g. Fang et al. (2006) and Santner et al. (2018). Although there is no evidence of stationarity, emulation is performed using a local training set of M nearest neighbours, selected from a dense set of n training data where M $<<$ n, and even a highly non-stationary function can be considered stationary over a small enough input range. For the sparse GP, the nonstationary ArcCosine kernel of order 2 (Cho and Saul, 2009) with ARD will be used, which was discussed in Section 3.4.2. For the univariate output GPs, the hyperparameters were estimated by maximising the log marginal likelihood using the Quasi-Newton method with multiple restarts (for avoiding local optima). For the multivariate output GPs, the Metropolis-Hastings algorithm was used to sample the hyperparameters. To optimise the weights and biases of the neural network, mini-batch gradient descent was implemented using the Adam optimiser (Kingma and Ba, 2015).

4.3.3 Phase 3: Using the emulator for parameter estimation

The final step is the minimisation of the loss function. When the outputs are emulated, then the loss function (4.5) is approximated by the surrogate loss (4.8), which can be minimised to obtain a parameter estimate. If the loss function is emulated directly, then the emulator itself is minimised with respect to $\boldsymbol{\theta}$. In either case, we face a high-dimensional

optimisation problem, for which a variety of algorithms have been proposed in the literature. In the present work, the optimisation routine changes depending on the emulation method. For the GP approaches from Algorithm 1 and Algorithm 2, the surrogate-based loss and the emulated loss are optimised using the Global Search algorithm by Ugray et al. (2007), implemented in MATLAB's Global Optimisation toolbox[2]. For the neural network, gradient descent with learning rate decay was used. This local optimisation is repeated 10 times from 10 different start points with the final parameter estimate selected as the one out of ten with the lowest loss evaluation. The relatively low number of start points was selected because we do not anticipate many (if any at all) local optima in the loss function.

4.4 Results

4.4.1 Comparing the neural network with a sparse GP

As outlined in Section 4.2.3, two global approaches to consider are sparse GPs and neural networks. The first study compares the predictive performance of both of these methods. The number of inducement points used for the sparse GP was 500. For in clinic inference, we would rely on a smaller number of inducement points to allow for efficient inference. However, allowing for a better prediction from the sparse GP than would be possible in practice will provide stronger evidence in favour of the neural network emulator. For the sparse GP emulator, 25 univariate sparse GP models were fitted to the entire training set of 10,000 training points. For the neural network a single 25 output model was trained, with the number of hidden nodes and batch size tuned using 10-fold cross-validation. The final network had two hidden layers of size 300 and a batch size of size 125 was used in optimisation.

Comparing the two methods visually in Figure 4.1, they appear almost identical. If we look closely at the extreme test points (those with the largest volume and smallest strain), there is some deviation of the sparse GP predictions from the lines at $y = x$. Quantitatively, the neural network outperforms the sparse GP by almost a full order of magnitude (6×10^{-6} versus 6×10^{-5}). Despite only being a small difference, this combined with the extra computational costs of the sparse GP are enough to justify adopting a neural network as a global emulator to compare with the local GP approaches.

[2] https://uk.mathworks.com/products/global-optimization.html

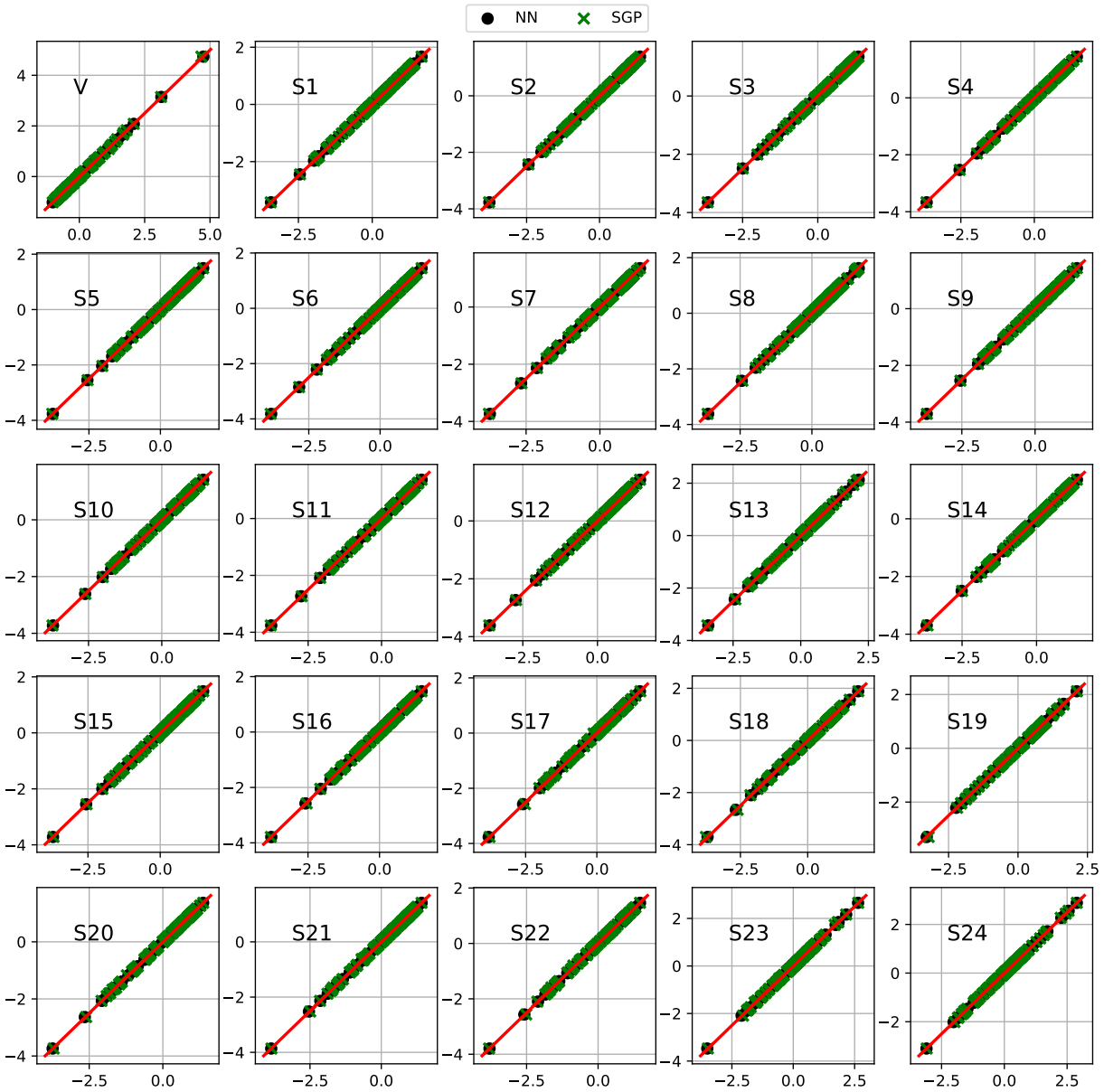


Figure 4.1: Comparing the predictive performance of a neural network and sparse GP on a held out dataset. Visually, the predictions (y-axis) appear identical to the ground truth value (x-axis), both for the sparse GP and the neural network. With the sparse GP there is a small deviation from the ground truth for the outlying prediction (largest volume) and overall, a small deterioration is indicated by a deterioration in predictive performance by a whole order of magnitude.

4.4.2 Comparing loss and output emulation for parameter estimation

The work in this section compares the paradigms of loss and output emulation, as discussed in Section 4.2.4. Of interest will be (univariate) GP emulation of the loss, univariate GP emulation of the outputs and multivariate GP emulation of the outputs. All GP simulations used local GPs with M = 100 nearest neighbours, with Algorithm 1 used

for the univariate output GP, and Algorithm 2 used for the multivariate output GP. A comparison requires test data, for which 100 test cases were generated by continuing the Sobol sequence used to generate the training data. These 100 parameter sets were then fed into the simulator, giving us a test set of parameter-output pairs: $\mathcal{D}_{\text{test}} = \{(\boldsymbol{\theta}_j, \boldsymbol{y}_j^*)\}_{j=1}^{100}$. For each test simulation, we can estimate the parameters using a particular emulation method and then compare the jth optimised parameter vector, $\hat{\boldsymbol{\theta}}_j$ and the jth ground truth parameter vector, $\boldsymbol{\theta}_j^*$, using the mean squared error:

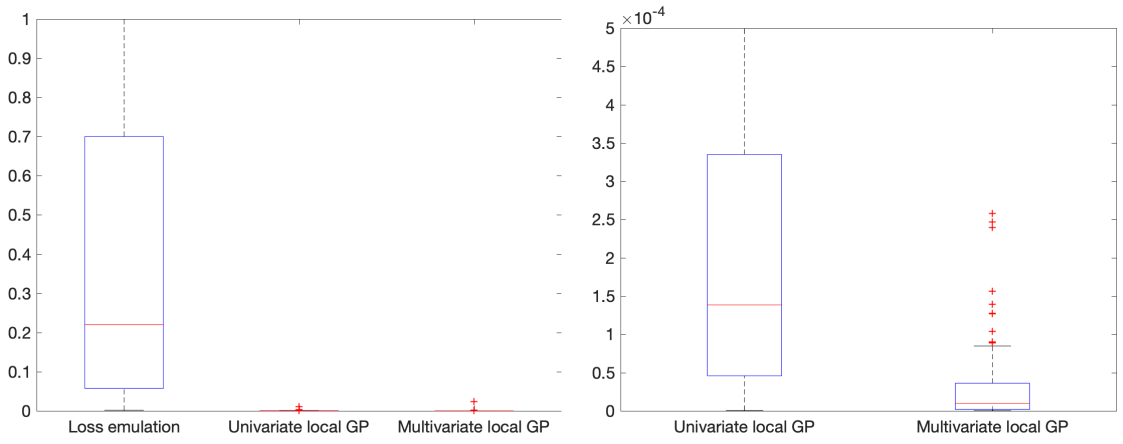
$$\text{MSE}_j = \frac{1}{4}||\hat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^*||^2$$

obtaining a sample of 100 MSE scores for each of the three emulation methods. In Figure 4.2, the distributions of the 100 MSE scores for each method are summarised using boxplots. Panel (a) shows the original vertical axis scale, while Panel (b) shows a reduced vertical axis scale to focus on the lowest MSE scores.

Figure 4.2 shows that emulating the output and then minimising the surrogate-based loss (4.8) leads to a substantially lower error than emulating the loss function directly and then minimising it. The explanation for this result is presumably related to the information loss inherent in mapping 25 separate outputs (the LVV and 24 circumferential strains) into a single scalar quantity. This suggests that the higher computational costs of the more detailed emulation are rewarded by higher accuracy. This is also a result that suits the task of in-clinic inference, since it is possible for output emulation to be performed in advance while loss emulation requires the approximation of a function that depends on the observed data. We can now compare the global emulation approach based on a neural network—which allows for pretraining of the emulator—with the local emulation approaches that use GPs.

4.4.3 Evaluating different output emulation approaches

For the test data outlined in Section 4.4.2, the parameters were also estimated using a neural network as the surrogate model in (4.8). This can be compared with the use of GPs (both univariate and multivariate), which will be implemented with 100 nearest neighbours as before. Table 4.1 provides the numerical results for the different output emulation approaches, along with the results of loss emulation. The parameter estimation error achieved with the multivariate output GP is about two orders of magnitude lower than the error obtained with the separate univariate output GPs and five orders of magnitude better than the results obtained using loss emulation. This suggests that additional accuracy can be achieved by explicitly modelling the correlations between the various outputs. Comparing the results of the MOGP and neural network, the GP is roughly an order of magnitude better. However, the computation times are very different:



(a) (b)

Figure 4.2: Distribution of the 100 out-of-sample MSE scores for each method. (a) Boxplots of the mean squared error for all 3 methods in the original y-axis scale and (b) with a reduced y-axis scale. The methods from left to right on the left plot are as follows: local GP emulation of the loss, local GP emulation of the outputs, and local multiple output GP emulation. The plot on the right includes only the output emulation methods.

the GP required around 12 minutes of computation time while the neural network required seconds. This is partly due to the different optimisation routines: the GP implementation uses Globalsearch while the NN approach uses gradient descent, but generally speaking the parametric nature of neural networks makes them more efficient predictors than GPs, once we exclude training time. The neural network results are encouraging, suggesting that we can use a neural network as a global emulator in situations where we wish to use MCMC to quantify our uncertainty in the parameters. It is important to note that neural networks will tend to require larger datasets so more forward simulations will be required. This is especially true when we do not adopt a Bayesian approach to network training.

Table 4.1: Comparison of the different emulation strategies. Median (1st, 3rd) quartiles of the mean squared error distribution for the out-of-sample parameter vectors. The method with the lowest median MSE is highlighted in bold: Output emulation using local multivariate output GPs.

		MSE
Statistical approximation	Emulation target	Median (1st, 3rd) quartiles
Single output GP	Output	1×10^{-4} $(4 \times 10^{-5},\ 3 \times 10^{-4})$
Single output GP	Loss	2×10^{-1} $(6 \times 10^{-2},\ 7 \times 10^{-1})$
Multivariate output GP	Output	5×10^{-6} $(1 \times 10^{-6},\ 3 \times 10^{-5})$
Neural network	Output	2×10^{-5} $(1 \times 10^{-5},\ 5 \times 10^{-5})$

A fixed value of M = 100 nearest neighbours was used to obtain the results presented so

far with Algorithm 2, to mimic the value used for the univariate output GPs in Algorithm 1. The dependence of the parameter estimation accuracy (in terms of MSE) on M, the number of nearest neighbours used in the local GP algorithm, will now be investigated by repeating the estimation of the GP for different values of M. The results are shown in Figure 4.3. The figure shows that when decreasing the value of M to smaller values, $M < 100$, the performance deteriorates, as expected. Interestingly, the performance also degrades when increasing the value of M beyond $M > 200$. At first, this appears counter-intuitive but a possible explanation is as follows. Including additional parameters that are far away from the target parameter have a negligible direct influence on the GP output, which follows from the exponential decay implied by the functional form of the kernel. However, if the characteristic length scales, expressed by the hyperparameters λ_k of the squared exponential kernel, are different in different areas of the parameter space, then the increase of M can lead to a suboptimal estimate of the length scale hyperparameters λ_k, and hence a deterioration of the performance overall. Varying length scales can be modelled using a nonstationary kernel, as used for the sparse GP in Section 4.4.1. However, in a small enough neighbourhood, as used by the local GPs, we can approximate a globally non-stationary function as stationary.

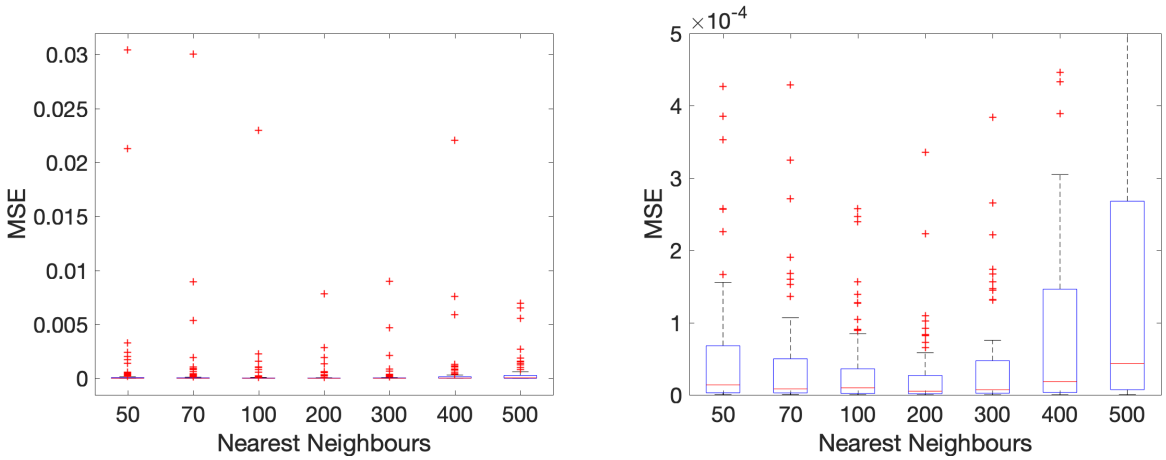


Figure 4.3: Validating the performance of the local multivariate output Gaussian process. The numerical values correspond to median out-of-sample MSE values in the predicted parameters using a local multivariate emulator as outlined in Section 2 with different numbers of nearest neighbours in function space. The plot on the right constrains the y-axis to exclude outliers, allowing us to see the minimum value of MSE occurring around $K = 100$.

Exploring this nonstationarity in more detail sheds some light on the outlying MSE values for the multivariate GP in Figure 4.2. In Figure 4.4, the input space nearest neighbours of three different points are visualised in output space. The leftmost plot highlights the nearest neighbours of the test point where the multivariate GP performs

worst. As we see, input space nearest neighbours are far apart in output space, suggesting a region of space where the function is varying more quickly. In such a scenario, the difference between output space and input space nearest neighbours is affected and we observe a deterioration in performance as a result of sparse coverage with the local training set. Importantly, this suggests the suboptimality of a standard space-filling design, since we should be aiming to more densely fill the region where the function varies more rapidly. This motivates the use of a different design scheme in Chapter 8.

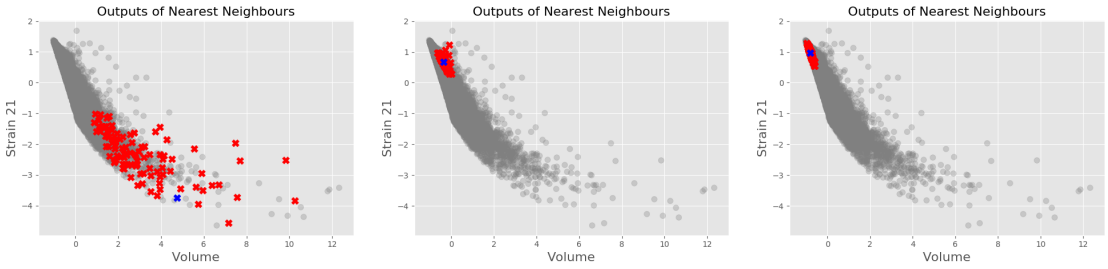


Figure 4.4: The nearest neighbours (in input space) from the training set for several different test points, visualised in output space. In each plot, the target point is in blue and the nearest neighbours are in red. The plot on the left highlights the nearest neighbours of the test point for which the multivariate method performs worst. For this point, nearest neighbours in input space are quite far apart in output space. This will lead to poorer input space coverage when we find nearest neighbours in output space.

4.4.4 Performance evaluation on MRI cine data

The results in Table 4.1 showed that the best strategy is output emulation using a local multivariate output GP. In this section, this method is used to estimate the constitutive parameter vector $\boldsymbol{\theta}$ of the H-O model from cine MRI data of a healthy volunteer. The data extracted from the MRI scans were the LV geometry, LVV and 24 circumferential strains at clearly defined positions on the LV wall. The LV geometry was the one that was used to generate the training data for the emulator. The LVV and 24 circumferential strains were standardised to provide the data $\boldsymbol{y}^*$ that were to be matched by the corresponding predictions from the biomechanical model $\boldsymbol{\zeta}(\boldsymbol{\theta})$. The emulation was carried out in the reduced 4D parameter space, which was mapped back into the 8D space via (4.2). For real data, the true parameters are unknown so the results were compared with estimates from an already published optimisation routine (Gao et al., 2015, 2017). This optimisation routine relies on an iterative optimisation scheme, solving the soft-tissue mechanical equations by (computationally expensive) brute-force numerical integration using finite element discretisation. Convergence time for the algorithm is in the order of days. A comparison between the benchmark parameters and the parameters obtained with the emulator is shown in Table 4.2. The values are very similar, with a mean square error

of only MSE = 0.00003. Four parameters lie outside the 95% confidence intervals obtained with the parametric bootstrap procedure described below, which reflects a very small bias resulting from the emulator. In practice, this bias would be too small to cause any difference in terms of clinical decision making.

One important aspect of the constitutive behaviour of the myocardium can be reflected in the Cauchy stress-stretch curves. Figure 4.5 shows the myofibre stress-stretch relationship for the healthy volunteer, obtained using the Holzapfel-Ogden constitutive law with the parameters shown in Table 4.2. The two panels refer to different directions: stretch along the sheet direction, panels (a,c), and along the myocyte, panels (b,d). The black dashed line shows the curve obtained from the literature gold standard method of Gao et al. (2017). The blue solid curve shows the stress-stretch relationship corresponding to the parameter vector $\hat{\boldsymbol{\theta}}$ estimated by optimising the surrogate based loss function (4.8) with MOGP as the emulator. To obtain an indication of the uncertainty in the parameters, a bootstrap approach can be used (see Section 3.2.1). First we obtain a point estimate of the parameters, $\hat{\boldsymbol{\theta}}$, from the noisy data, $\boldsymbol{y}^*$, and use it to compute the corresponding output in data space, $\hat{\boldsymbol{y}}$. We then obtain the residuals, $\hat{\boldsymbol{\xi}} = \boldsymbol{y}^* - \hat{\boldsymbol{y}}$ or $\hat{\xi}_i = y_i^* - \hat{y}_i$, for $1 \leq i \leq 25$, randomly sample with replacement from the set of residuals $\mathcal{R} = \{\hat{\xi}_i\}_{1 \leq i \leq 25}$, and generate surrogate data $\tilde{y}_i = \hat{y}_i + \tilde{\xi}_i$, where $\tilde{\xi}_i$ is the ith draw from $\mathcal{R}$. We then repeat the parameter estimation on the surrogate data $\tilde{\boldsymbol{y}}$ to obtain new parameter estimates $\tilde{\boldsymbol{\theta}}$, and repeat the procedure 100 times, to obtain a distribution of $\tilde{\boldsymbol{\theta}}$; this is the bootstrap distribution used for uncertainty quantification. For every sampled parameter vector $\tilde{\boldsymbol{\theta}}$ we compute the Cauchy stress-stretch curve. The results are shown in Figure 4.5. The figure shows the stress-stretch curve for the estimated parameter vector $\hat{\boldsymbol{\theta}}$ (in blue) and 95% confidence intervals for the estimated curves obtained from the sample of 100 Cauchy stress-stretch curves corresponding to the bootstrap sample $\{\tilde{\boldsymbol{\theta}}\}_{i=1}^{100}$. The literature curve lies mostly within this confidence interval, with a slight bias for large stretches resulting from the dimension reduction in (4.2). The estimate $\hat{\boldsymbol{\theta}}$ giving rise to Figure 4.5 was obtained with a reduction in computational time of 3 orders of magnitude (several days to 15 minutes[3]).

4.5 Discussion and conclusion

The work in this chapter has shown that statistical emulation is a viable option for efficient parameter estimation in the biomechanical model of the LV. By fixing the LV geometry and end diastolic pressure, the emulator was built to replicate the map from material parameters to outputs, which at the time of this study had not been considered in the literature. This provides the perfect building block for the work in the remainder of this

[3]Intel Xeon CPU, 2.9GHz, 32 cores and 32GB memory.

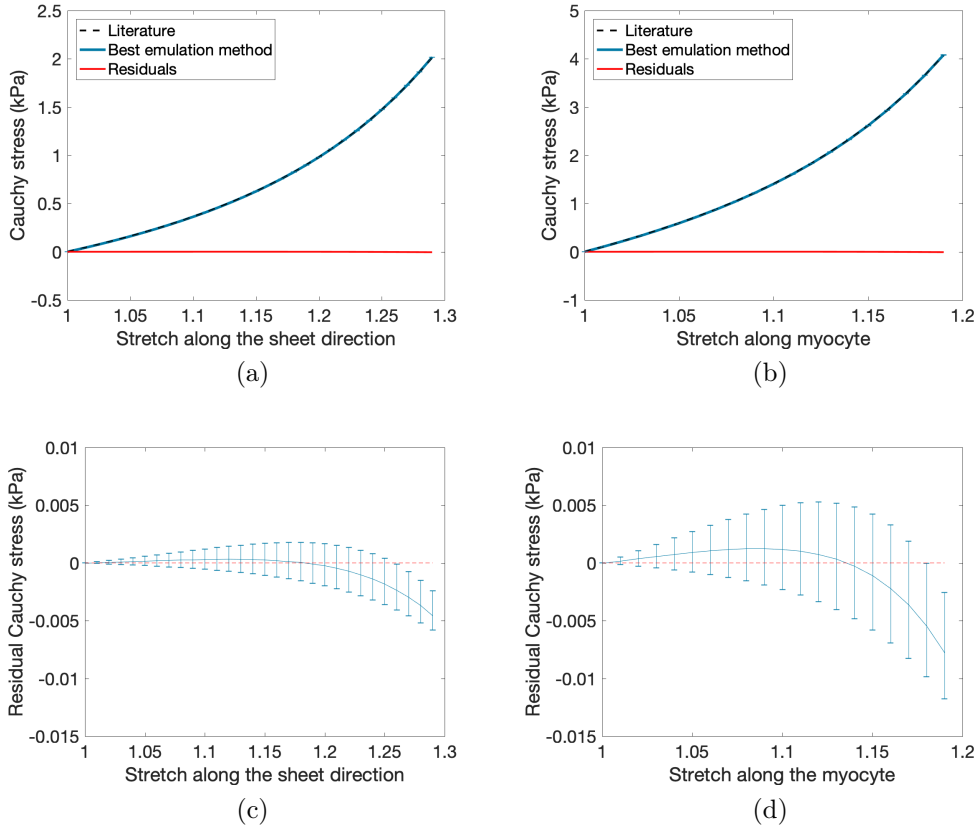


Figure 4.5: Plots of the Cauchy stress against the stretch along (a) the sheet direction and (b) the myocyte. The current best estimate (i.e. the literature gold standard) from Gao et al. (2017) is reported as a dashed black line. Estimates of the curves using the best emulation approach (M3) are given as a blue solid line. The error bars show a 95% confidence interval, obtained by using the bootstrap method described in Section 4.4.4. Each plot contains a residual curve, providing the difference between the true and estimated curves. These residual curves are plotted with 95% confidence intervals in (c) for the sheet direction and (d) along the myocyte.

thesis, aiming to make statistical emulation generalisable to different LV geometries. Several different emulation approaches were considered and the best performing method was found to be the multivariate output GP from Conti et al. (2009). This approach was compared with a brute force optimisation of the material parameters, considered as the literature state-of-the-art. The results using the emulator were very similar to the literature method, with a reduction in computational costs of three orders of magnitude. While the results in stress–stretch space showed a very small bias at high stretch in Figure 4.5, this would be unlikely to impact the decision making process in the clinic.

One of the disadvantages of the GP approach was the necessity for local approximations to the full model. For this reason, neural networks were proposed as an alternative to GPs for statistical emulation, with optimisation was used to train the network. The benefit of a neural network approach is that it allows for a global model, which will be important

Table 4.2: Estimates for real MRI data. The literature gold standard and the estimated parameters $\hat{\boldsymbol{\theta}}$ from emulation method M3. The confidence intervals (CIs) are obtained using a parametric bootstrap procedure.

	a	b	a_{f}	b_{f}	a_{s}	b_{s}	a_{fs}	b_{fs}
Literature	0.2245	1.6215	2.4267	1.8269	0.5562	0.7747	0.3905	1.6950
Point estimate (95% CI)	0.2249 (0.2246, 0.2252)	1.6247 (1.6223, 1.6264)	2.4218 (2.4128, 2.4685)	1.8232 (1.8165, 1.8283)	0.5633 (0.5586, 0.5673)	0.7845 (0.7795, 0.7901)	0.3916 (0.3885, 0.3943)	1.6994 (1.6862, 1.7114)

when considering the use of MCMC in future chapters. Moreover, parameter estimation with a surrogate loss relying on evaluation of a neural network emulator required only seconds of computation time, which is an order of magnitude faster than the local GP approaches.

Fundamentally, the work of this chapter can be considered as no more than a proof-of-concept study. For clinical translation, the surrogate model would need to be able to generalise to new LV geometries to allow for parameter estimation in a wider cohort of patients. To this end we need to find a low-dimensional representation of the LV geometry, allowing for it to be included in the emulator input space. There are various approaches that one can pursue for LV geometry representation, including the 6D parametric representation proposed in Di Achille et al. (2018). Alternatively, a standard dimension reduction method like principal component analysis (PCA; see e.g. Chapter 12 in Murphy (2012)) can be used to project the high dimensional LV geometry vector into a low dimensional coordinate system that contains maximum information about typical variations in the patient population. This task will be considered in the next chapter of this thesis.

Chapter 5

Analysing the left ventricle geometry representation

The generalisation of the emulator from Chapter 4 requires the inclusion of LV geometry variation in the model. For this purpose, dimension reduction will be considered, allowing us to more efficiently represent the LV geometry. The present chapter will consider a comparison of different dimension reduction techniques for the task of LV geometry reconstruction. It is shown that, despite the seemingly restrictive assumptions required, PCA performs similarly to more complicated approaches that allow for nonlinearities and fewer assumptions in the model. Lastly, the use of the LV geometry in identifying MI patients is considered in order to show the level of information that can be encoded in a low-dimensional space obtained using PCA, and the practical uses of dimension reduction methods beyond cardiac modelling. This work is loosely based by work published in Romaszko et al. (2019), but all autoencoder and PCA models have been retrained.

5.1 Background

Mathematical modelling of cardiac function relies on an anatomical representation of the heart. In the early stages of LV modelling, researchers developed models based on idealised LV geometries. Initially, a cylinder was used (Dumesnil and Shoucri, 1982), before advancing towards elliptical models of the LV shape (Huyghe et al., 1992; Di Achille et al., 2018). Peirlinck et al. (2021) and Chabiniok et al. (2016) provide summaries of the progression of shape representation, from the idealised cases through to the anatomical approaches used today. An anatomical reconstruction of the LV, which incorporates local variations in the shape, has allowed for higher fidelity simulations of cardiac mechanics. As a result, we can recreate patient specific dynamics to a level of accuracy that would be useful in the clinical setting. The benefits of accurate LV geometry reconstruction stretch beyond the cardiomechanical field to the analysis of LV geometry changes during disease. For

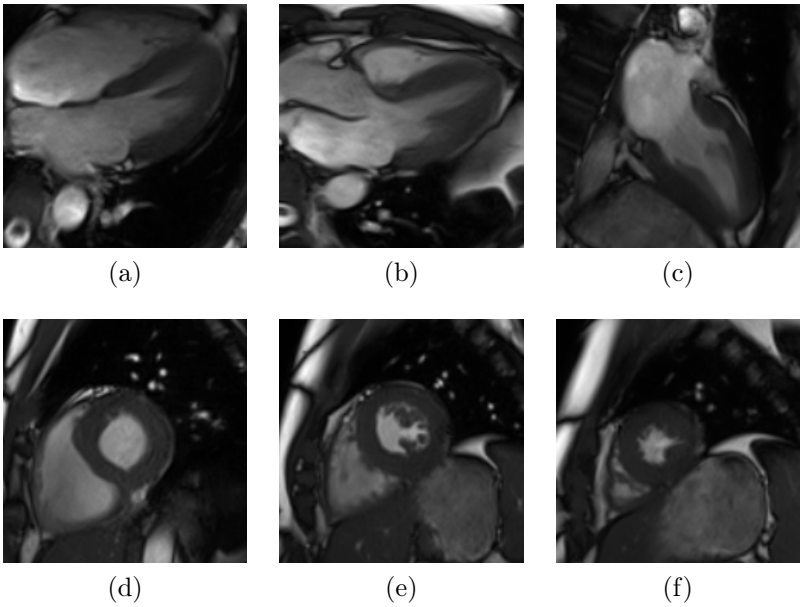


Figure 5.1: Top row: the three long axis views of the LV. (a) Four chamber view, (b) left ventricular outflow tract and (c) one chamber view. Bottom row: three example of short axis views. From left to right, these are taken at increasing distances from the basal plane.

instance, as a result of myocardial infarction (MI) the LV remodels to allow for normal cardiac output (Suinesiaputra et al., 2018). This remodelling manifests as a shape change in the LV (Azevedo et al., 2016), and quantitative measures of this shape change could be helpful in understanding disease and planning treatments.

For the use of mathematical models in the clinic, we require methods of extracting the LV geometry in vivo. This can be done using images of the heart, and the dataset in this chapter makes use of cardiac magnetic resonance (CMR) scans, examples of which are found in Figure 5.1. The extracted shape is represented as a series of nodes on the endocardial and epicardial walls, as shown in Figure 5.2. This representation of the LV geometry can then be used to generate a dense finite element mesh for simulating the diastolic filling process. Although the accuracy of the LV geometry representation allows for accurate simulation of the dynamics, the computational costs are prohibitive. Therefore, efficient use of these models relies on approximations such as lower order representations of the LV shape (Hong et al., 2019) or data-driven models like the one proposed by Maso Talou et al. (2020). Approaches like the latter, which allow us to generalise a surrogate model to different LV geometries, rely on some encoded form of the LV shape. The principal aim of the current chapter is to find a method for efficient representation of the LV geometry in a low-dimensional space. One approach to achieving this low-dimensional representation is through an idealised model of the LV geometry, such as the one used by Di Achille et al. (2018). Alternatively, we can allow for a more general shape representation through the use of standard dimension reduction methods.

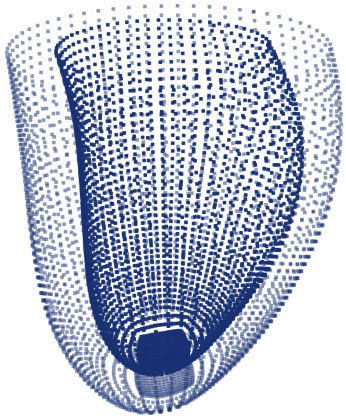


Figure 5.2: The LV is represented by a series of finite element nodes. There are 2896 nodes on the epicardial and endocardial walls providing a $2896 \times 2 \times 3 = 17376$ dimensional representation in Cartesian coordinates.

For the purpose of disease analysis, quantitative measures beyond the typical clinical indicators of shape change are difficult to obtain in the high-dimensional space in which the LV geometry resides. Although these standard indicators have obvious utility in every day medicine, they fail to acknowledge the more subtle changes in the LV geometry. It is widely believed that these changes could provide information in the analysis of cardiac disease (Farrar et al., 2016). The extraction of these shape properties relies on a more objective approach to the summary of the LV geometry, for which dimension reduction can be used. This idea has been considered in the literature (Suinesiaputra et al., 2018; Farrar et al., 2016) and also used to assess the effectiveness of a disease treatment for cardiac amyloidosis (Li et al., 2020b).

5.2 Methods

5.2.1 Extracting the LV geometry from CMR scans

Images of the LV can be obtained in vivo using cardiac magnetic resonance (CMR) scanning. From this procedure, we obtain multiple long axis views (usually 3 as shown in the top row of Figure 5.1) and short axis (SA) views (usually around 9, three examples of which are in the bottom row of Figure 5.1) of the LV. The SA images are obtained at equally spaced locations along the long axis (the z-axis) of the LV, with the meta-data of the CMR scan providing their exact location. Generally, we obtain images over the entire cardiac cycle starting from end of diastole, but in the current work only the early diastole images are used. Extracting information from these images in the form of segmentations, we then reconstruct the LV geometry at early diastole and represent it using a series of

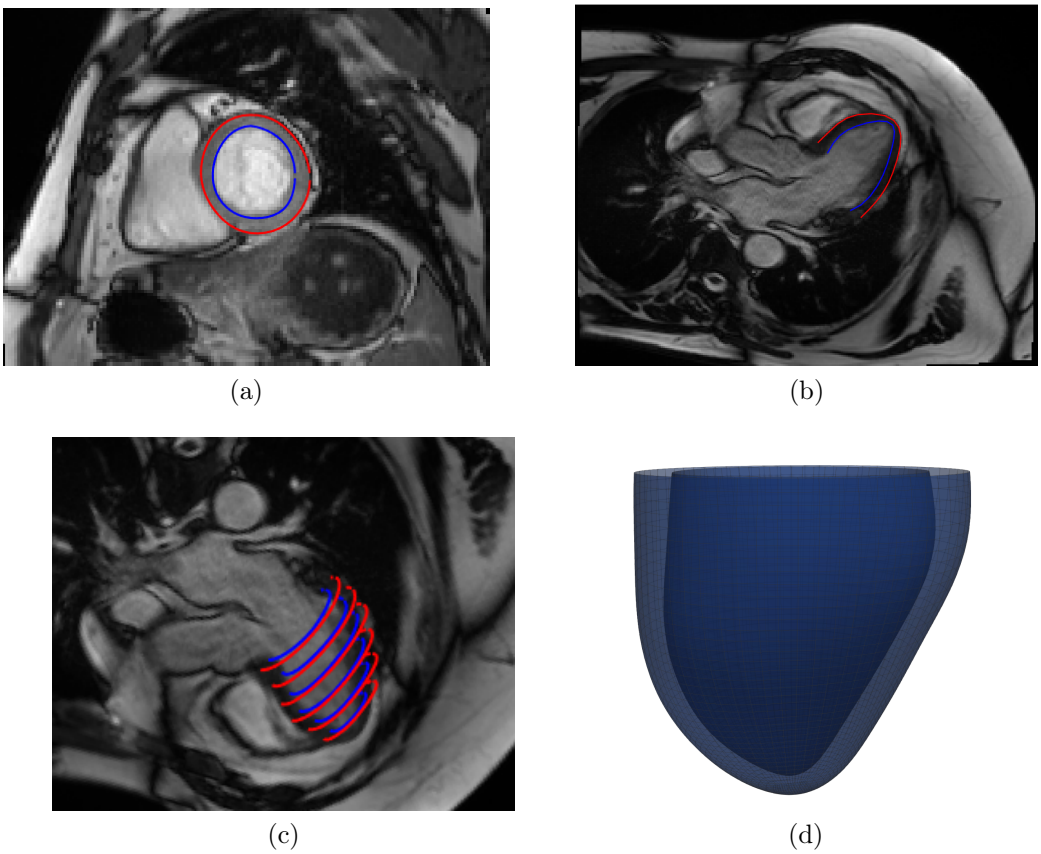
nodes on the endocardial and epicardial surfaces, as shown in Figure 5.2

State of the art segmentations of the different short axis and long axis views of the LV were obtained manually using in-house Matlab code, taking between 30 and 60 minutes per LV geometry. This involves tagging the boundaries of the short axis and long axis images. A cubic spline interpolant fitted to these landmark points provides boundaries of the endocardium and epicardium as shown in the top row of Figure 5.3. Superimposing, at the appropriate z-position, the short axis segmentations on top of the long axis views, we would expect that the boundaries roughly align. However, movement of the patient during the CMR scan usually affects this alignment. After segmenting the short and long axis boundaries, these artefacts are manually corrected by adjusting the x and y coordinates of the short axis segmentations in a process referred to as motion correction. For all LV geometries, the final representation is formed by deforming a template mesh in prolate coordinates[1] to match the segmented boundaries, subject to a smoothness constraint. The angular coordinates of this template are held fixed during fitting, meaning that only the radial coordinate is adjusted. In addition, the basal plane of the LV geometry is always aligned horizontally at $z = 0$ (in Cartesian coordinates). This is beneficial for shape analysis because it means that the shapes are all aligned in the same plane, around which we can find the principal sources of variation. An example LV geometry resulting from the reconstruction procedure is shown in panel (d) of Figure 5.3. A thorough overview of the reconstruction procedure, which is outwith the scope of this thesis, can be found in the literature (see the supplementary material of Gao et al. (2017) for a discussion of motion correction and Liu et al. (2009) for an overview of the general method).

5.2.2 Dimension reduction

The LV representation obtained using the procedure from Section 5.2.1 consists of a series of 2896 nodes on the endocardial and epicardial surfaces, giving a $2896 \times 2 \times 3 = 17376$ dimensional representation in 3d Cartesian coordinates. For the sake of brevity, this full representation of the LV geometry will be referred to as a 17k representation. Repeating the reconstruction procedure multiple times, for different LV geometries, provides a set of aligned shapes, each a 17k representation of a unique LV geometry. An example LV geometry, represented as a series of nodes, was given in Figure 5.2. Although this consists of two surfaces, the entire 17k dimensional representation is processed within one vector of coordinates, $\mathbf{g}^* = (g_1^*, ..., g_n^*)$. This allows for any dimension reduction approach to account for inter-layer effects like the thickness of the myocardium that can be important both in disease analysis and cardio-mechanical modelling.

[1] In prolate coordinates each coordinate is represented by two angles and a radius. The angles are the circumferential angle and the angle between the $z = 0$ plane and the point of interest while the radius is the distance from the centre of the LV to the coordinate.



(a) (b) (c) (d)

Figure 5.3: Visualising the geometry extraction procedure. (a) first we segment the different short axis slices. (b) next we segment the long axis images before performing motion correction (c) where we align the short axis segmentations with the long axis view. The final LV geometry, displayed in (d) is then obtained using a B-spline fitting procedure. This procedure was proposed by Liu et al. (2009).

In general, the aim of dimension reduction is to reduce the full LV geometry representation, $\mathbf{g}^*$, to a lower dimensional form, $\mathbf{h}^*$, while preserving the important structure in the original shape. For this methodology to be of use in disease analysis and cardio mechanical modelling, we require that we can recover the explicit mapping into the lower dimensional space:

$$f : \mathbf{g}^* \mapsto \mathbf{h}^*. \tag{5.1}$$

In addition to this, and of particular relevance to cardio-mechanical modelling, we require that the mapping, f, is invertible. Invertibility ensures that given a new point in the lower dimensional space we can then recover the corresponding LV geometry reconstruction. Of the methods discussed in Section 3.5, PCA, autoencoding neural networks and the idealised LV geometry all satisfy these properties and these will be considered in this chapter for the task of dimension reduction of the LV geometry[2]. The fundamental goal of

[2]Kernel PCA (Scholkopf et al., 1998) was also considered but preliminary results suggested that this

the dimension reduction, with regards to the outline of this thesis, is to efficiently represent the LV geometry in a cardio mechanical model. With this in mind, attempts were made to adapt these standard approaches to the specific context of biomechanical modelling. This approach is summarised in Appendix C, but it did not allow for any improvement in the LV geometry representation.

With the exception of the idealised LV approach, the application of dimension reduction requires a set of LV training geometries from which we can extract the relevant features of the data. From here, PCA is a simple application of the methodology outlined in Section 3.5, only requiring selection of the dimension of the low-dimensional space. For the autoencoder, we must specify a network architecture in terms of the number of nodes and type of activation functions at each layer of the network. The general structure to be considered can be found in Figure 5.4. This network has one input and output node for each dimension of the LV geometry representation. The blue nodes represent two hidden layers, where nonlinear activation functions can be used to make the model nonlinear. As was mentioned in Section 3.5, the hidden layers are essential if we hope to build a model that can outperform PCA.

As discussed above, the LV geometry lies in a 17k dimensional space. Building a neural network with one input and output node per dimension creates a very high dimensional parameter vector to be optimised during model training. For example, a network with two hidden layers of 100 nodes and a bottleneck of 5 nodes requires training of over 3 million parameters. The obvious concern with such a model is overfitting, which we can guard against using weight decay. An alternative approach to help avoid overfitting is to first reduce the dimension of the LV geometry using PCA. If we choose a large enough number of principal components then the reconstruction of the LV geometry is almost perfect. This allows us to reduce the number of parameters that must be optimised while incurring negligible loss in reconstruction accuracy. Romaszko et al. (2019) found that 60 PCs is enough to represent the LV geometry accurately so this will be used in the current study.

5.2.3 Model training

Autoencoder

Autoencoders, like all neural networks, require that we learn the weights of the model to fit the training set of data. Letting $\hat{\mathbf{g}}^{(i)} = A(\mathbf{g}^{(i)}, W, B)$ be the reconstruction of the ith LV geometry obtained by evaluating the autoencoder parameterised by $W = \{\mathbf{W}_1, ..., \mathbf{W}_k\}$

provided no improvement over PCA

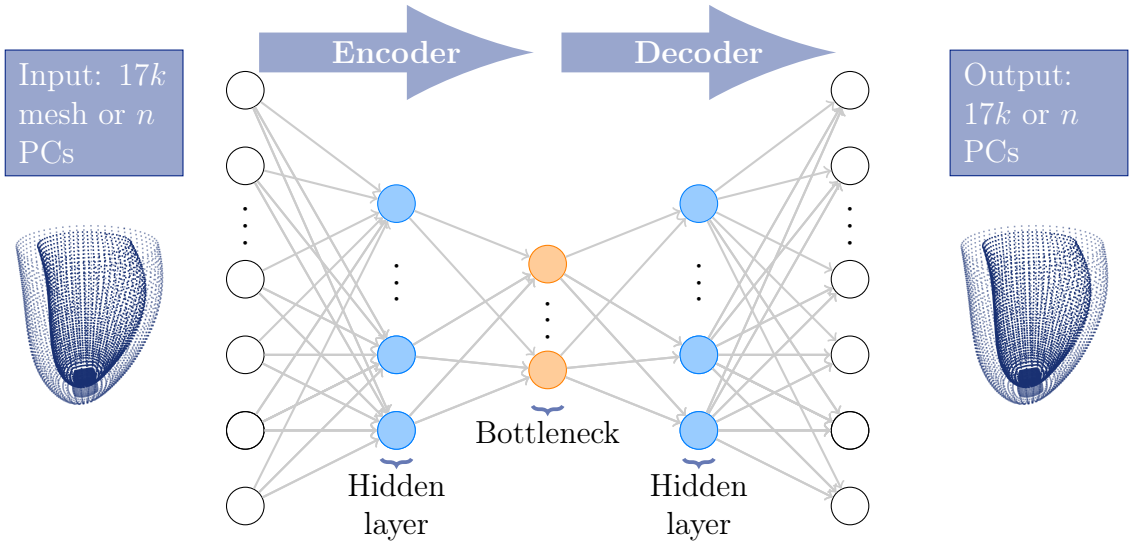


Figure 5.4: The typical autoencoder architecture. The input and output layers are the GT and estimated LV geometry, respectively. The intermediate layers (in blue) consist of a series of nonlinear operations that provide a reduction into a K dimensional space. This transformation could be performed using PCA or learnt as part of the entire network. The latter case would relax the orthogonality assumption of PCA. The hidden layer in orange provides our low-dimensional representation of the LV geometry in m dimensions. This diagram is adapted from Figure 1 in Romaszko et al. (2019).

and $\{\mathbf{b}_1, ..., \mathbf{b}_k\}$, we can define the following mean squared error (MSE) objective function:

$$L(W, B) = \sum_{i=1}^{m} \frac{1}{n} ||\mathbf{g}^{(i)} - \hat{\mathbf{g}}^{(i)}||^2, \tag{5.2}$$

which will be optimised with respect to W and B. Often, optimisation is performed using mini-batch gradient descent, as discussed in Section 3.3.3. Due to the small size of the training set (198 LV geometries), one large batch of all LV geometries will be passed through the model at each iteration of the optimisation of the autoencoder. The optimisation is performed in a leave-one-out cross-validation (LOOCV) scheme, with L2 regularisation and learning rate selected as the best performing combination from a grid of possible values. That is to say that the LOOCV is repeated a number of times for different learning rates and L2 penalties and the same value of these hyperparameters[3] is selected for the entire population based on the overall performance. The grid of L2 penalties was $[1, 3] \times 10^Z$, $Z \in [-8, ..., -3]$ and for the learning rate this was $[1, 3] \times 10^Z$, $Z \in [-2, -3]$, giving a grid of size 12×4. Two different activation functions—the tanh and leaky ReLU—will also be considered. Note that the results will not be given for each setting of hyperparameter separately. Instead, just the optimal results and the corresponding

[3]Strictly speaking, the learning rate is not a hyperparameter, but a tuning parameter affecting convergence. However, in the training scheme where the epochs are fixed for varying learning rates, we can think of this is a regularisation hyperparameter having the same effect as early stopping.

hyperparameters will be provided. Optimisation was carried out using the Adam optimiser (Kingma and Ba, 2015). The number of training epochs is less of a concern than usual because the use of L2 regularisation allows us to train until convergence on the training set.

Idealised LV method

For each of the 198 LVs in the dataset, the model from (3.53) is fitted twice by minimising two loss functions with respect to the parameter vector $\boldsymbol{\kappa}$ (see the parameter vector below (3.53)). These loss functions belong to the intersection-over-union (IOU) class and are based on a comparison of the volume of the wall $W_{\boldsymbol{\kappa}}$ and cavity $C_{\boldsymbol{\kappa}}$ of the LV prediction at parameters $\boldsymbol{\kappa}$ with the corresponding volumes in the true LV, denoted W_T and C_T, respectively. The two objective functions are given by:

$$E_{WC} = 1 - \frac{1}{2}\left\{\frac{W_{\boldsymbol{\kappa}} \cap W_T}{W_{\boldsymbol{\kappa}} \cup W_T} + \frac{C_{\boldsymbol{\kappa}} \cap C_T}{C_{\boldsymbol{\kappa}} \cup C_T}\right\}, \quad E_W = 1 - \frac{W_{\boldsymbol{\kappa}} \cap W_T}{W_{\boldsymbol{\kappa}} \cup W_T}, \tag{5.3}$$

where E=0 gives a perfect LV alignment in either case. The objective function on the left hand side in (5.3) is the original objective used by Di Achille et al. (2018) and takes into account both cavity and wall volumes. The function on the right is an alternative loss function based on the wall volume only, which was found to perform well in simulations.

Optimisations were performed in *MATLAB* using *fminsearch* for local optimisation and *particleswarm* for global optimisations. The calculation of the quantities $W_{\boldsymbol{\kappa}}$, $C_{\boldsymbol{\kappa}}$, W_T and C_T was done by set allocation of a grid of points. For each point on a three-dimensional grid, an indication of wall allocation and cavity allocation for the idealised (fitted) and actual (true) geometries was assigned. Using these set-assignments we obtain a discrete approximate to the loss functions in (5.3) that can be optimised to learn the 6 parameters. Suitable positivity constraints were placed on the values of these parameters and, where possible, each parameter was initialised at an estimate taken from the LV geometry.

5.3 Implementation

5.3.1 Data

The dimension reduction methods will be fitted to a population of 198 LVs, obtained from a mix of 70 healthy volunteers and 128 MI patients. The MI patients were reconstructed in an identical way to the HVs, without a separate treatment of the infarct region, as outlined in Section 5.2.1. The MI patients were identified from the British Heart Foundation MR-MI study population (ClinicalTrials.gov identifier: NCT02072850) from a population of patients with acute STEMI, who had been enrolled into a CMR cohort study between

Table 5.1: Autoencoder nomenclature.

Name	Description
17-y:FUNh-z:FUNb	Autoencoder with the full 17k LV geometry representation as input, y hidden nodes with FUNh activation functions and z nodes in the bottleneck layer with FUNb activation functions.
60-y:FUNh-z:FUNb	Autoencoder with the 60 PC reduced LV geometry representation as input, y hidden nodes with FUNh activation functions and z nodes in the bottleneck layer with FUNb activation functions.

July 14, 2011 and November 22, 2012. CMR imaging was performed at 1.5 Tesla (Siemens Avanto, Siemens Healthcare, Erlangen, Germany). For MI patients the scans were taken within 2 days after hospital admission. MI patients and HVs underwent the same imaging protocol except that HVs below the age of 45 did not receive intravenous gadolinium contrast. The CMR protocol involved steady-state free precession cine scans with a short-axis left ventricular stack from the base to the apex. The slice thickness was 7 mm with a 3mm gap, typical cine imaging parameters were matrix = 180×256, flip angle = 80^o, TR: 3.3 ms, TE: 1.2 ms, bandwidth: 930 Hz/pixel. The voxel size was $1.32 \times 1.32 \times 7\,\text{mm}^3$. Cine images were acquired in the four chamber view, left ventricular outflow tract and one chamber view, as was shown in the top row of Figure 5.1. In total 4 different operators—referred to as operators 1,2,3 and 4— segmented the data. The healthy volunteers were mostly segmented by operator 1, but operator 2 segmented a subset of ten. The MI patients were segmented by all operators, with short axis segmented by operators 1 and 2 and the long axis images segmented by operators 3 and 4. All four operators performed the motion correction. Operator 1, who segmented most of the HVs, checked all MI segmentations and updated many of the long axis segmentations in line with their own judgement. Although the use of different operators is not an ideal experimental setup, it is an inevitability in situations where data collection is so time-consuming.

5.3.2 Comparisons

The autoencoder shown in Figure 5.4 has three hidden layers, one of which is the bottleneck layer that provides the low-dimensional representation of the LV geometry. For reasons that were previously discussed (namely the universal approximation theorem) only a neural network with two hidden layers and a bottleneck layer will considered in the comparison of methods. Recall that, as a way of reducing the number of parameters to be optimised, PCA will be used to preprocess the data for the autoencoder.

In the following results, each different autoencoder model is referred to by the size of the layers of the network as follows: x-y:FUNh-z:FUNb where x is the size of the input layer (either 60 for PCA reduced inputs or 17 for the full 17k LV geometry representation),

y is the size of the hidden layer with activation function FUNh and z is the size of the bottleneck layer, which has activation function FUNb. For the activation function in the hidden layer, the tanh and leaky ReLU activation functions will be considered. The latter function is similar to the ReLU, which was visualised in Figure 3.5, but rather than being zero for negative values, the activation function takes value $0.3x$. This was proposed as a solution to the dying ReLU problem[4]. In the bottleneck layer, only linear activation functions will be used. The metric used for comparison of PCA and the autoencoder is the MSE between the ground truth test LV geometry, $\mathbf{g}^* = (g_1^*, ..., g_n^*)$, and its reconstruction, $\hat{\mathbf{g}}^* = (\hat{g}_1^*, ..., \hat{g}_n^*)$:

$$MSE^* = \frac{1}{n}||\mathbf{g}^* - \hat{\mathbf{g}}^*||^2, \tag{5.4}$$

5.4 Results

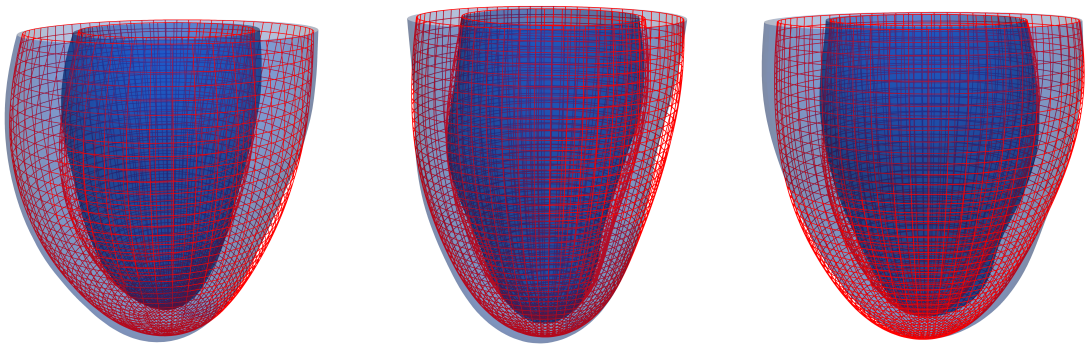
5.4.1 PCA

First, let us consider the results obtained using PCA. To correctly assess the performance of PCA, LOOCV was used, learning the principal components on 197 LV geometries and testing on the 198th. In Table 5.2 the quantification of the accuracy, in terms of the MSE between the ground truth and the PCA reconstruction, is provided for different numbers of principal components. The median value, obtained from the 198 MSE evaluations, is provided in the table. Inevitably, as we increase the number of principal components, the reconstruction error decreases. In most practical applications, it is beneficial to sacrifice some of this accuracy for the sake of a lower dimensional encoding of the geometry. For comparisons with other methods, it is also easier to have a fixed size of low-dimensional space that we aim for. Based on the results in Table 5.2, an argument could be made for using either 4 or 5 principal components, but since we observe a slight drop-off in proportion change beyond 5 PCs, this size of encoding will be assumed. In particular, notice that the improvement we observe in terms of median MSE when increasing from 4 to 5 PCs is matched by increasing the dimension from 5 to 7 PCs. Some intuition for the quantitative measures of reconstruction accuracy is provided by Figure 5.5. In this figure, the reconstructions of three different LV geometries using 5 principal components are plotted. The ground truth LV geometries and reconstructed LV geometries are represented by blue and red surfaces, respectively, and the MSE value is provided below each reconstruction.

[4]The dying ReLU problem occurs when the parameters into a unit lead to a negative activation to the node for all examples in the training set. The result of this is a gradient of zero in the direction of these parameters and the "death" of the unit because the parameters will never be updated in gradient descent.

Table 5.2: MSE in reconstructions of the LV geometry obtained wuth different numbers of principal components. Proportion change is the proportion reduction in MSE.

NPC	0	1	2	3	4	5	6	7
Median MSE	0.066	0.021	0.017	0.012	0.009	0.007	0.006	0.005
Proportion change	—	0.68	0.22	0.28	0.25	0.24	0.11	0.18



(a) MSE=0.004 (b) MSE=0.007 (c) MSE=0.009

Figure 5.5: Some example LVs reconstructed from 5 principal components. The solid blue surfaces show the endocardium and epicardium of the ground truth LV while the red meshes show the endocardium and epicardium predictions from 5 principal components. Also provided is the mean squared error between the two reconstructions.

5.4.2 Autoencoders

We can now consider the autoencoder for dimension reduction. Firstly, different architectures will be compared, before comparing the best performing model and PCA.

Comparing architectures

Based on the PCA results, an LV geometry encoding in 5 dimensions has been chosen. This fixes the size of the bottleneck layer of the autoencoder, meaning only the input layer size, hidden layer size and activation functions have to be selected. Based on results in Section C.2, a hidden layer of size 100 was used. Different activation functions are compared for both types of input (60 PCs and original 17k) in Figure 5.6. For the network that encodes the full LV geometry, the leaky-ReLU noticeably outperforms the tanh function. In the second model, where we consider first reducing the LV geometry to 60 dimensions with PCA before applying the autoencoder, the performance is similar in both cases. Indeed, comparing the 0.25,0.5 and 0.75 quantiles in the distribution of MSEs, the values were identical up to the 3rd digit. We can now compare both of these architectures (the 60 dimensional and 17k dimensional inputs), in both cases using leaky ReLU activation functions. This comparison is made in Table 5.3, where the methods are also compared with the mean mesh. Both outperform the mean mesh by roughly a whole order of magnitude, and both autoencoders performed identically according to the

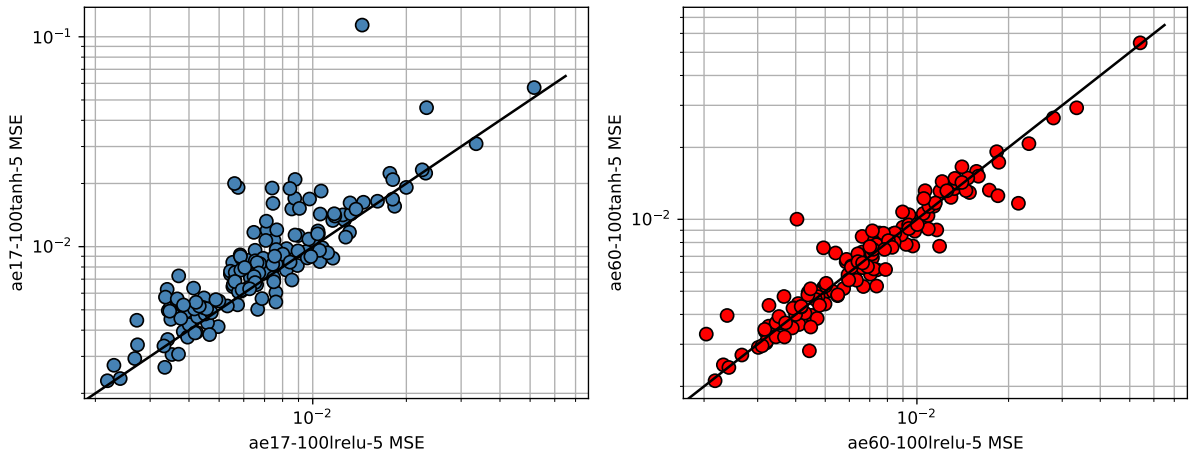


Figure 5.6: A comparison of activation functions for the two autoencoder architectures, as indicated by the axis titles of the plots. On the left is the autoencoder with 17k inputs and on the right the LV geometry has first been reduced to 60 dimensions using PCA. The data are plotted on the log scale for each of the visualisations. See Table 5.1 for a description of the models and labels of the axes.

quantiles considered in the table. Notice that different learning rates and L2 penalty were selected during training. This is not surprising, since the network with 17k dimensional inputs has a much larger number of weights, meaning a smaller learning rate and larger weight decay are required.

Table 5.3: Summary of the autoencoder performance, along with the optimal L2 penalty (L2) and learning rate (LR) for training each network. The mean mesh results are given as reference values. See Table 5.1 for a description of the final 2 methods.

Method	0.5 (0.25,0.75) quantiles of MSE	L2	LR
MM	0.051 (0.030,0.081)	—	—
60-100:lrelu-5:linear	0.007 (0.005,0.009)	3×10^{-7}	0.005
17-100:lrelu-5:linear	0.007 (0.005 , 0.009)	1×10^{-5}	0.001

Comparison with PCA

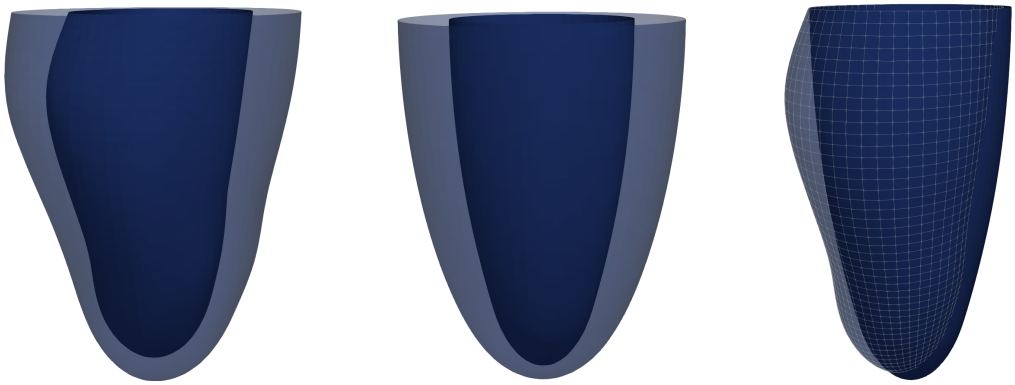
We can now compare the optimal autoencoder performance with that of PCA. Table 5.4 provides a summary of the overall performance of both methods, comparing a large number of quantiles of the MSE distribution to show the similarity in overall results of both methods. The similarity between the results is surprising. Although the exact error on each individual test case is different, the performance across the entire population is nearly identical. This suggests that, for the given training set of data (which is smaller than would be ideal), the extra complexities introduced by the autoencoder do not allow for any extra encoding of LV geometry information compared with PCA.

Table 5.4: Quantiles of MSE distributions for an autoencoder and PCA. For both methods, we obtain separate MSE values for each LV geometry. The table provides the quantiles of this distribution of errors. See Table 5.1 for a description of the model in the top row.

	Quantile				
	0.1	0.25	0.5	0.75	0.9
17-100:lrelu-5:linear	0.0034	0.0046	0.0068	0.0090	0.0131
PCA	0.0036	0.0045	0.0066	0.0091	0.0134

5.4.3 Parametric method

As a final comparison we can compare the parametric description of the LV geometry from Di Achille et al. (2018) with PCA. An example of this model fitted to an actual LV geometry is found in Figure 5.7. In panel (a) the original LV geometry is shown and in panel (b) is the fit of the idealised geometry. This is compared with the original mesh in panel (c) where we see the issues caused by the symmetry of the model. This model was originally devised for LV geometries at end of diastole, when the LV geometry is inflated and more elliptical. The limitation of this assumption is likely the reason for the poor performance over the entire population found in Figure 5.8, where we see that similar reconstruction accuracy is obtained when using only 1 principal component to encode the LV geometry.



(a) (b) (c)

Figure 5.7: Estimation of the LV geometry using the parametric approach. Panel (a) shows the original LV geometry, (b) is the fitted LV geometry and (c) shows the overlap of the endocardiums of both. The model assumes that the LV geometry is symmetric, which limits its flexibility in fitting LV geometries at early diastole where the LV tends to be more deformed.

5.4.4 Classifying MI cases based on LV geometry

The study population was made up of 70 healthy volunteers and 128 MI patients. As a test of the informativeness of the encoded LV geometry, we can attempt to classify MI

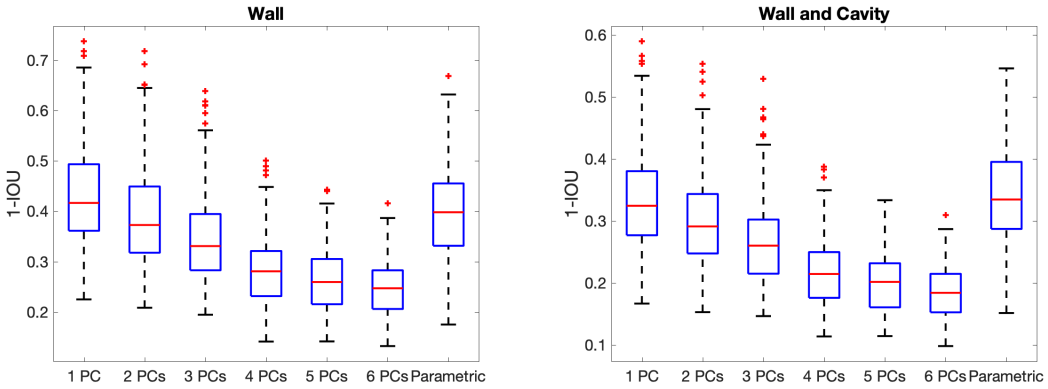


Figure 5.8: Comparison of the parametric approach and PCA for reconstructing the LV geometry. PCA easily outperforms the parametric approach, partially due to the symmetry assumption of the parametric method.

from healthy volunteers. It is well known that cardiac remodelling leads to a change in the volume of the LV at end of systole (Suinesiaputra et al., 2018). Of interest here is whether other aspects of the LV geometry allow for accurate classification of MI, so the data were scaled to be the same size as the mean mesh in order to discount the volume effect from the classification.

Firstly, we can consider 2D visualisations of the data. In Figure 5.9 two projections of the data are plotted, one using PCA (linear) and one using t-SNE (nonlinear). We see that there is no perfect separation of the two groups, but we could obtain a reasonable classification using a linear classification method. In particular, there appears to be separation of the two classes along the second PC. The physical interpretation of this mode, provided by the second mode of variation, is found in Figure 5.10. This plot suggests that MI and HV classes are separated based on elongation of the LV.

We can take a more flexible approach and consider the accuracy of a Gaussian process classifier for MI, allowing for a nonlinear boundary between the two groups. This is visualised in two dimensions in panel (a) of Figure 5.11, which shows the projection of the data onto 2 PCs along with the probabilities of MI assigned to different regions of the space by a fitted GP classifier. Note that this diagram is purely illustrative since the PCA and classification were not performed with LOOCV. In panel (b) of Figure 5.11 a ROC curve of the classifier performance with 2 PCs is shown, with an area under the curve (AUC) of 0.79 suggesting good classification of MI. For this study, full LOOCV was used so that each subject was left out of the PCA and GP classifier training, and prediction required first a projection into the trained PCA space and then classification with the GP classifier. The GP classifier was implemented in Scikit-learn (Pedregosa et al., 2011) with a logistic link function (see Chapter 3 of Rasmussen and Williams (2006) for more details on classification with GPs) and a Matérn 52 kernel, which is a standard stationary kernel function (see Section 3.4.2).

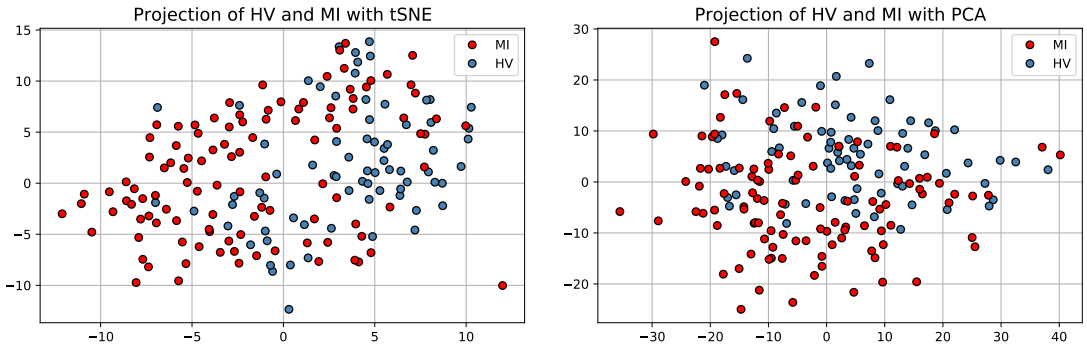


Figure 5.9: Projection of healthy volunteers and myocardial infarction patients with t-SNE on the left and PCA on the right. We see that there is some separation of the two classes but it probably is not linear and that the separation is possibly more pronounced in the case of t-SNE than with PCA.

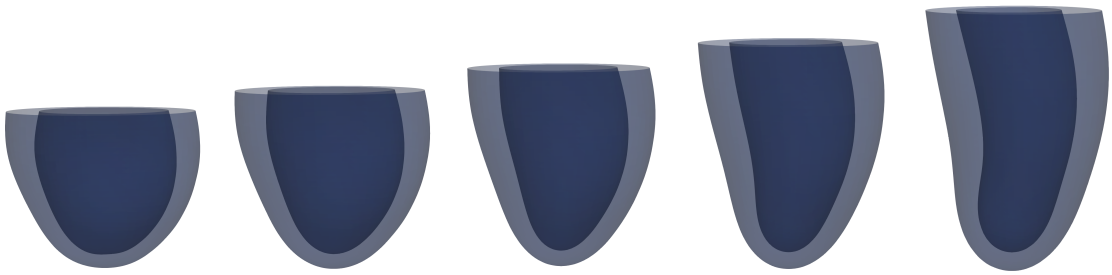
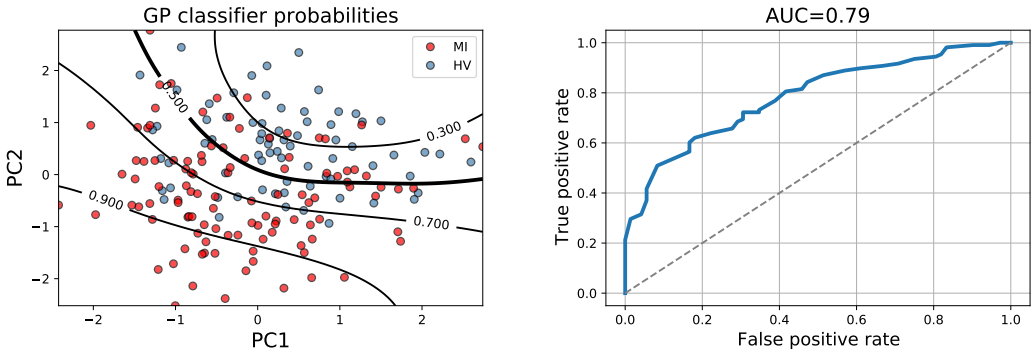


Figure 5.10: Visualising the second mode of variation from PCA. From left to right, we move in a positive direction along this mode (towards the MI subjects).



(a) (b)

Figure 5.11: A GP classifier is fitted to the 2 PC projections of the MI and HV patients, assigning probabilities as shown in (a). The ROC curve for the 2 PC classifier is provided in (b).

5.5 Discussion and conclusion

This chapter has explored the ability of different dimension reduction methods to encode the LV geometry in a low-dimensional space. Each of the standard methods was assessed in terms of MSE between the reconstruction and the ground truth LV geometry, with PCA

proving to be as good as an autoencoder when compared using this metric. A context specific parameterised geometry representation was also considered, but this was shown to be inferior to PCA. Some example reconstructions from 5PCs were plotted in Figure 5.5. Qualitatively, the performance at the 0.5 quantile (MSE=0.007) does not look very good. However, the performance of the algorithm can only really be considered with reference to operator errors. This will be explored extensively in Chapter 9.

Building on this dimension reduction work, a low-dimensional representation of the LV was used to classify myocardial infarction from a set of healthy volunteers. An AUC of 0.79 for a 2 PC classifier showed the high level of classification performance. The task of MI classification would be made easier by considering multiple time points in the cardiac cycle, allowing the classifier to identify MI using the stroke volume, which is also used by clinicians. In a project carried out with a collaborator[5] it was shown that with scans from early systole and end of diastole, accurate classification is possible directly from the CMR images using a deep convolutional neural network. The disadvantage of such an approach, as is frequently a criticism aimed at machine learning approaches, is a lack of interpretability of the results. On the other hand, by classifying based on principal component projections, we can use the modes of variation to analyse the shape changes between the two populations. In the current study, elongation of the LV appears to allow for fairly accurate classification of MI from healthy volunteers. The presence of multiple operators in the generation of the training data was a confounding factor in this study and records should have been kept that specify the operator in each example.

In Section 5.2.1 the procedures for extracting the LV geometry from CMR scans were discussed. This is a time consuming process that would ideally be automated for in clinic use. Recently, there have been important steps taken towards this goal (Bai et al., 2018) while Romaszko et al. (2021) used a convolutional neural network, regularised by constraining it to learn a PC representation, to accurately predict the LV geometry from CMR scans. Essentially, the deep neural network model learns to encode the LV geometry, directly from the CMR scan, in the low-dimensional space obtained with PCA. The upside of the PCA layer is that it acts as a regulariser that can only produce LV geometries consistent with the mapping from PCA, while also reducing the number of weights that must be learnt during network training. The downside is that the model can never be more accurate than PCA for generating LV geometries. An interesting future direction would be to combine automatic and manual segmentations to create a more accurate reconstruction of the LV geometry with only a small increase in computation time.

The final goal of the dimension reduction is to find a method that can be used to provide a lower-dimensional representation of the LV geometric for use in an emulator of the simulator. In this chapter, comparison in geometry space has suggested that PCA

[5]This was done by Lukasz Romaszko in a feasibility study.

should be used for this purpose. This will be put to the test in the next chapter where a first attempt will be made to generalise the statistical emulator to multiple LV geometries by including 5PCs in the emulator input space.

Chapter 6

Generalising the left ventricle emulator to varying LV geometries

The work in this chapter makes a first attempt at generalising the statistical emulator to multiple LV geometries. Based on the results from Chapter 5, PCA will be used to provide a low-dimensional LV geometry representation. For the purpose of cardio-mechanical modelling, this PCA representation will be benchmarked against a mean mesh to assess how much information is gained when we increase the quantity of LV geometry information. Using the PCA representation as a baseline, an attempt will be made to improve the results using a small number of forward simulations. Finally, the emulators will be used to explore the inference task for the LV model, showing the changing complexity of the problem over the input space and the poor identifiability of some of the parameters in the model.

6.1 Generalising the emulator to new LV geometries

In the mathematical model outlined in Section 2.2, the left ventricle is represented in the form of a discretised mesh with 2896 nodes on the endocardial and epicardial walls (see Figure 5.2). As discussed in Section 5.2.1, this LV geometry can be manually extracted from CMR scans, requiring around 40 minutes of manual work. Forming a dense finite element mesh from the extracted LV geometry and solving the set of momentum balance equations, the simulator provides us with an accurate prediction of the LV behaviour in diastole for a given set of material parameters. In Figure 6.1, a randomly selected strain from the LV simulator outputs is plotted for two different LV geometries in blue and orange, over a range of different material parameter configurations. Although the change in the output over parameter space for different LV geometries follows the same general shape, as governed by the H-O model, local perturbations ensure that at any one location, simulations can differ by as much as 0.02 for the circumferential strain. If we consider the fact that these strains generally vary between 0.1 and -0.3, the error obtained is around

10% of this interval and could lead to a bias in the model calibration. This suggests the importance of accounting for the LV geometry in the simulator.

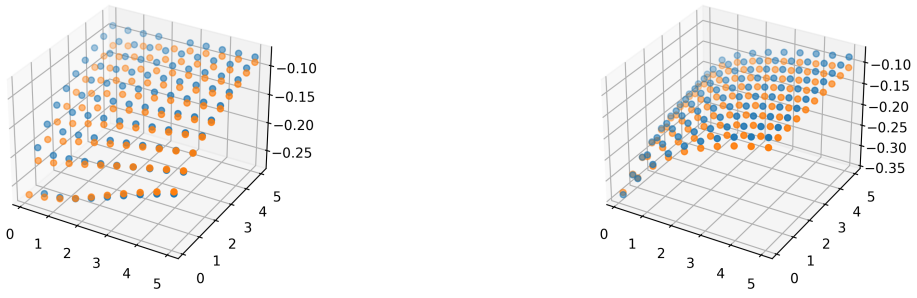


Figure 6.1: Left: the simulator function for varying θ_1 and θ_2 with $\theta_3 = \theta_4 = 1$. Right: the simulator function for varying θ_3 and θ_4 with $\theta_1 = \theta_2 = 1$. The two different colours indicate simulations from different LV geometries.

Given the dependence of the simulator on the LV geometry, application of these models in the clinical setting relies on a generalisation of a statistical emulator, similar to the one from Chapter 4, to a whole population of LV geometries. An immediate step towards generalisation would be to include the LV geometry as an input to the emulator. However, this would require extending the emulator input space by around 17k dimensions, by which point emulator design is impossible. As such, we require either a different approach to statistical emulation or a reparameterisation of the LV geometry component of the model input space.

First, we can consider the practicality of a different emulation approach. Making use of the segmented LV geometries from Chapter 5, a logical step towards generalising the statistical emulator would be to consider building surrogate models from multiple batches of high fidelity forward simulations over the sample of training left ventricle geometries. This would amount to the training of a series of emulators akin to the fixed geometry emulator considered in Chapter 4. Relying on a discrete space of LV geometry representation, we could simply assign a patient to their nearest neighbour from the training set and use this nearest neighbour emulator to learn the material parameters. Unfortunately, our sample of left ventricles is quite small (fewer than 200) and, although it is inevitable that our 17k dimensional LV geometries lie on a far lower dimensional manifold, we cannot be sure that the set of training geometries provides dense enough coverage of this manifold. With this in mind, we could consider a weighted ensemble of the different training emulators, with the weighting determined based on the closeness of fit of the LV geometries to the new test point of interest. However, the method for measuring this closeness of fit is unclear since we do not know which aspects of the LV geometry are important for accurate prediction from the LV simulator. Given these obstacles, the ensemble of emulators will

not be explored further in this chapter, but Appendix F preliminary results for a method that uses simulations from multiple LV geometries to inform a set of basis functions for patient-specific emulation are provided.

An alternative modelling paradigm considers an explicit LV geometry component in the input space of the emulator, allowing us to encapsulate the changing left ventricle dynamics with varying left ventricle shapes as a continuous function. This framework provides the opportunity to introduce a patient-specific component to the model, alleviating some of the concerns associated with the ensemble approach. In Figure 6.2, the emulator with a reduced LV geometry representation is visualised. Starting from a sample of LV geometries, we can learn a transformation that provides a low-dimensional representation. Given a new point in the low-dimensional space, we can invert this transformation and recover a synthetic LV geometry. Repeating this procedure multiple times—sampling a new point in low-dimensional space and inverting the transformation—provides a set of synthetic LV geometries that can be paired with a sample of material parameters. Evaluating these different geometry-parameter combinations in the simulator provides a training set of simulations for a statistical emulator that allows us to predict the LV dynamics in diastole for a new LV geometry at any material parameter configuration, subject to some error that is introduced by the LV geometry approximation. The aim of the current chapter is to investigate the parameter inference task in the presence of this LV geometry approximation.

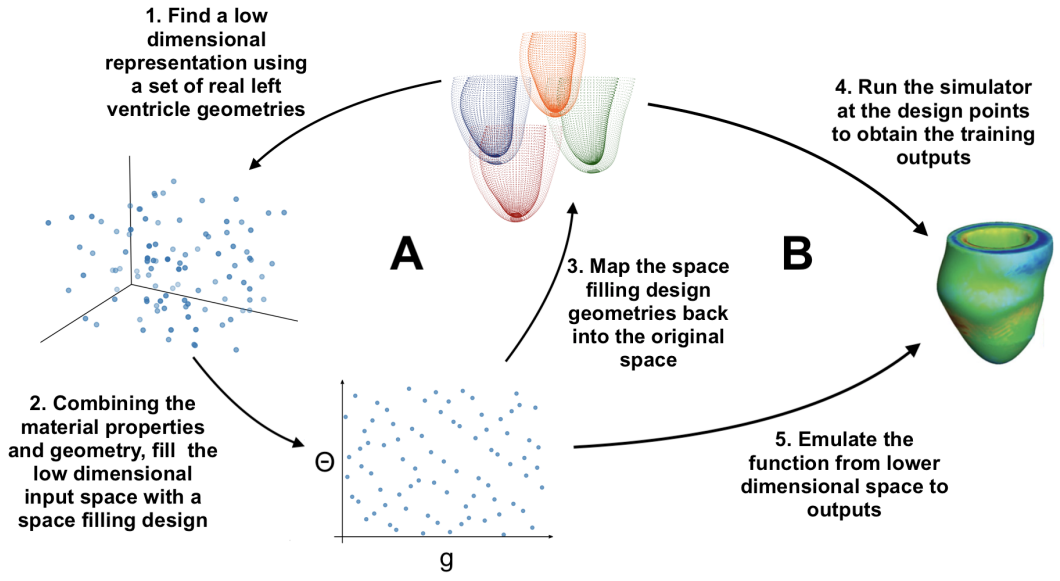


Figure 6.2: Emulation of the LV model for full geometries is not possible. Demonstrated in **part A** is the generation of a design for a reduced LV geometry emulator, where we must find a mapping into a lower dimensional space that, combined with the space of material properties, enables a sufficiently informative emulator design. **Part B** shows that the emulator can be trained to replicate the function from this lower dimensional space since our training geometries can be fully described in this space.

6.2 Methods

6.2.1 Generalising the statistical emulator

Let us start by considering the aim of dimension reduction in our context, describing several mappings that will be referred to in the remainder of this chapter. We may consider the exact, or ground truth, simulator model, $\zeta(\cdot)$, as providing the following map:

$$\zeta : \mathcal{X}, \mathcal{G} \to \mathcal{Y} \tag{6.1}$$

where all inputs and outputs to the model are exact i.e. the LV geometry is represented by the exact shape extracted from CMR scans[1], which belongs to a space *GeomSpace*. The space of material parameters $\mathcal{X}$ is assumed to be the 4 dimensional theta representation from Section 4.1.2. The first approximation to the model in (6.1) would be, imagining the possibility of design and emulation in the original input space, to approximate the map $\zeta(\cdot)$ by training on variables in the domain and codomain of (6.1) to obtain the approximation:

$$\upsilon : \mathcal{X}, \mathcal{G} \to \mathcal{Y}. \tag{6.2}$$

The domain and codomain of this function are identical to (6.1), but the function between them is provided by an emulator. Constructing an accurate approximation in this manner is implausible due to our ignorance of the manifold on which the LV geometries lie. An alternative approximation of (6.1) considers an approximation of the input space:

$$\tilde{\zeta} : \mathcal{X}, \hat{\mathcal{G}} \to \mathcal{Y} \tag{6.3}$$

where the LV geometry space has been replaced by the space of LV geometry approximations, $\hat{\mathcal{G}}$, that is of lower dimension than $\mathcal{G}$. Unlike (6.2), this function is no cheaper to evaluate than (6.1). However, because an element from this new space $\hat{\mathbf{g}}$ can be fully specified by a low-dimensional representation $\mathbf{h}$, we can consider emulating (6.3), obtaining an approximation of the map from (6.1):

$$\eta : \mathcal{X}, \mathcal{H} \to \mathcal{Y} \tag{6.4}$$

where the accuracy of our approximation to (6.3) is determined by a combination of the dimension of $\mathcal{H}$ and the number of training simulations. The accuracy of this mapping, relative to (6.1), is bounded by the accuracy of (6.3). This new model is represented by part B in Figure 6.2.

[1] In Chapter 9 we will see that this notion of an exact shape is misleading due to errors introduced by different operators.

At this point, we should consider how to recover the low-dimensional representation of the LV geometry. One could hypothesise an optimal solution to this task as visualised in Figure 6.3. Given a training set of LV geometries $\mathbf{G}$, the idea would be to construct a parametric function from LV geometry $\mathbf{g} \in \mathcal{G}$ to LV geometry reconstruction $\hat{\mathbf{g}} \in \hat{\mathcal{G}}$ (indicated by the encoder and decoder). Optimising the parameters of this model using the sum-of-squared differences between the outputs of the simulator evaluated for the actual LV geometries and the outputs of the simulator evaluated at the approximate LV geometries would provide an encoded form of the LV geometry, $\mathbf{h} \in \mathcal{H}$. This method, which properly optimised over a training set of LV geometries would give the optimal encoding of the LV geometries for the purpose of biomechanical modelling, is not attainable due to the high computational costs required in the forward pass of the system. A compromise on efficiency and optimality was attempted in Appendix C.1, but this did not prove successful.

The computational cost incurred by the optimal methodology stems from the coupling of the LV geometry reduction and the computational simulator. Gradient based optimisation of the parameters requires the derivative of a loss function that depends on the simulator model. Requiring a number of forward passes in the order of thousands, training of this system is impossible in the current context where one simulation requires around ten minutes of computation time. Instead, we must detach the ideas of dimensionality reduction and simulator error, selecting the best dimensionality reduction approach based on LV geometry reconstruction error. Such an approach was taken in Chapter 5, with PCA proving to perform at least as well as competing methods. Albeit a compromise on the optimal reconstruction approach, the reconstruction error criterion ensures an accurate fit to the actual geometry while avoiding the expensive coupling discussed above.

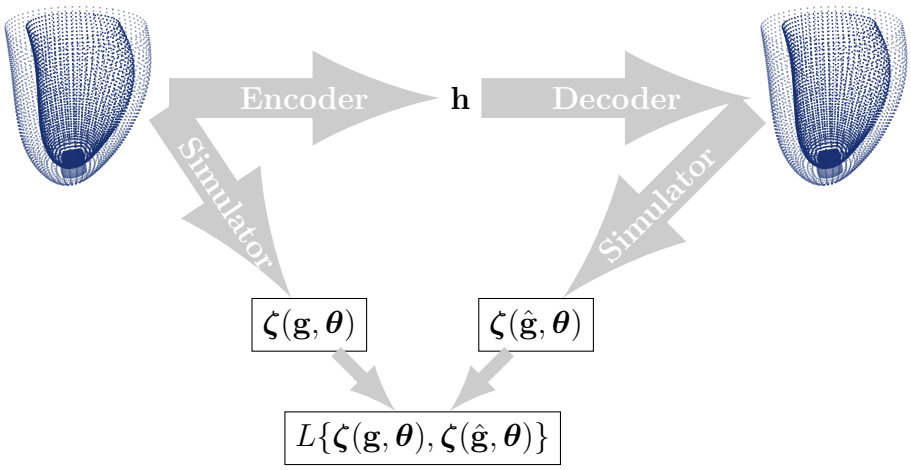


Figure 6.3: In the optimal dimension reduction for cardio mechanical modelling, we would find the encoding and decoding maps that minimise the error between the simulator evaluated at the original geometry and the simulator evaluated at the reconstructed geometry. The computational costs of the simulator make this method non-applicable.

Dimension reduction for the purpose of cardio-mechanical modelling has been explored in the literature. Di Achille et al. (2018) used an idealised geometrical shape in a model of the inverse LV dynamics in an attempt to recover the unloaded configuration (the LV geometry without any pressure loading). This dimension reduction method was explored in Chapter 5 but was found to perform quite poorly. More similar to the work in this chapter is that of Maso Talou et al. (2020), who adopted PCA to obtain a low-dimensional representation of the LV, permitting the inclusion of a geometry component in the input of a model estimating the deformation field of the LV during diastole. However, the authors did not benchmark the method against simulations from actual LV geometries, and gave no justification for their choice of lower-dimensional space (in that case 2 PCs).

6.2.2 Building the emulator

How many principal components to use?

The construction of an emulator that allows for variation in the LV geometry requires that we choose a dimension of the LV geometry encoding. The larger the number of PCs used, the more information we encode in our low-dimensional representation and the closer our reconstructions move to the ground truth LV geometry. Although the corresponding improvement in the fit of (6.3) to (6.1) may not be monotonic (this is due to the fact that our objective function for dimension reduction is independent of the simulator), we would expect that, in general, as reconstruction error decreases, so too does the error in our forward simulator. Unfortunately, the opposite is true for our second approximation: as we increase the number of PCs the emulation task becomes more difficult as a result of the higher dimension and the approximation of (6.3) with (6.4) becomes more crude.

Although the optimal dimension reduction from Figure 6.3 is too expensive, we can use the LV simulator to compare the accuracy of different sizes of LV geometry encodings in the cardio-mechanical model. This was done by running a batch of forward simulations with LV geometry reconstructions obtained using different numbers of principal components. While our final goal is model calibration, accuracy of the forward model should provide a good proxy for suitability of low-dimensional representations for parameter estimation. In total, 42 LV geometries were used, each reconstructed in a LOOCV implementation of PCA. Note that the PCA included the full training set of LV geometries from Chapter 5 but for reasons of computational cost, only 42 of these were run in the simulator. For the material parameters, a Sobol sequence of length 42×3 was obtained and each LV geometry was combined with three unique material parameter configurations to get a set of 126 ground-truth simulations[2]. Running the PCA reconstructions at the same material parameter configurations allows us to assess the impact that PCA has on the mechanical

[2]During simulation, of the 126 examples, 2 led to an error in Abaqus as a result of excessive distortion of the LV mesh

model. To account for the fact that the early diastole volume is different with the mean mesh, the standardised volume is used in the error evaluation, where the end diastole cavity volume (LVV) is standardised by the early diastole volume:

$$\bar{V}_{\text{end}} = \frac{V_{\text{end}} - V_{\text{early}}}{V_{\text{early}}}, \tag{6.5}$$

this gives a dimensionless volume quantity that has similar interpretation to the circumferential strains. This is also a fairer comparison if we are concerned about the effects on model calibration since unequal early diastole volumes can easily be accounted for by including the above dimensionless quantity in our likelihood function.

Table 6.1 gives the results of running the simulator with LV geometries reconstructed using 0,2,4,5,7 and 10 principal components in Cartesian space[3]. Despite the improvement in reconstruction error as we increase the number of PCs (see Table 5.2), the circumferential strain predictions do not really change in accuracy as we increase the number of PCs, while the error in the LVV prediction decreases quite rapidly. With 5PCs, the LVV RAE is less than half that of the mean mesh. It is worth emphasising that even with a constant MM representation, the median absolute relative error in volume is only 4.4%. In other words, there is not that much room for improvement in terms of average performance. In light of these results, we would not expect drastic changes in the accuracy of our parameter estimates as we increase the number of PCs.

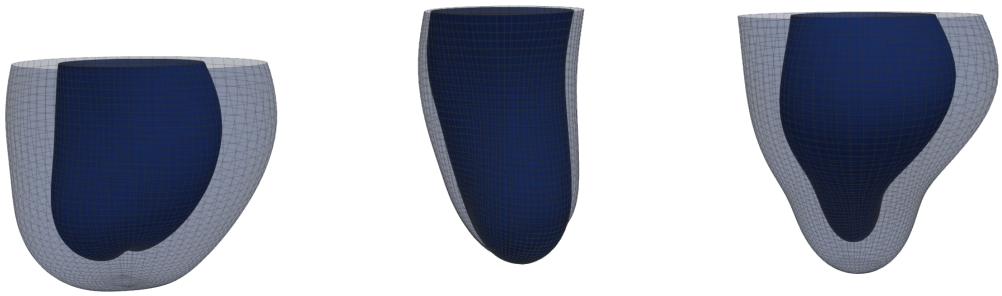
Table 6.1: Comparing the accuracy of simulations with different LV geometry approximations using relative absolute error (RAE). There is not a dramatic improvement in accuracy when we increase the number of principal components.

Reconstruction	LVV RAE 0.5 (0.25,0.75) quantiles	Circumferential strain RAE 0.5 (0.25,0.75) quantiles
0PC	0.044 (0.022,0.069)	0.065 (0.030,0.120)
2PC	0.028 (0.010,0.050)	0.062 (0.029,0.118)
4PC	0.028 (0.011,0.054)	0.056 (0.025,0.106)
5PC	0.021 (0.010,0.036)	0.052 (0.023,0.104)
7PC	0.016 (0.007,0.031)	0.049 (0.023,0.092)
10PC	0.013 (0.007,0.024)	0.042 (0.020,0.080)

An issue to consider in the context of a low-dimensional geometry representation is the consistency of the synthetically constructed LV geometries with respect to physiological constraints. As we begin to consider higher numbers of principal components, we begin to consider aspects of the LV shape that are not representative of the population as a whole,

[3]A prolate coordinate system representation of the LV geometry could be proposed instead of a Cartesian one. In such a system, we only require a single radial coordinate for each 3D Cartesian coordinate, so the initial dataset dimension reduces to 17376/3. However, when assessing the results of PCA in prolate coordinates in the simulator, the results were similar to those obtained with Cartesian coordinates.

possibly including artefacts from operator errors that should not be there in the first place. Given that we will be sampling synthetic LV geometries from our lower dimensional space, the increased flexibility could cause problems. To demonstrate this, Figure 6.4 presents three samples obtained using random sampling in the 10 PC space, projected back into the original 17k dimensional space. As suspected, some of the generated LV geometries are physiologically unrealistic, displaying issues such as overlapping endocardial and epicardial walls and concavity at the apex. In the end we could consider checks to assess whether an LV geometry is realistic, but it is likely that some irregularities will not be accounted for. This issue is partly caused by an assumption of a uniform distribution in the low-dimensional space, which is likely an oversimplification of the problem. However, in the absence of a large enough training set of LV geometries, taking a conservative approach and assuming a uniform distribution seems like the correct strategy. A more



(a) (b) (c)

Figure 6.4: Random samples in 10PC space lead to physiologically unrealistic LV geometries. In the plot on the left, the apex region is concave, in the middle the endocardium (inner surface) intersects the epicardium (outer surface) and in the right plot the bulge and shape of the apex are unrealistic.

obvious consequence of increasing the dimension of the LV geometry representation is the deterioration of our emulator performance. This, combined with the problem posed by the flexibility of the low-dimensional representation, means that we have to make a compromise on the number of PCs to include in the input space. Based on results in Table 6.1, and the discussion of the excessive flexibility permitted by increasing numbers of PCs, a 5 dimensional geometry space will be assumed. This number of PCs will be further justified by a study performed in Chapter 9.

The Design

In Chapter 4, a space filling design was used to fill the 4-dimensional design space efficiently before running the simulator. With the inclusion of a low-dimensional LV geometry representation, the new simulator function has 9 input dimensions: the first four are ma-

terial parameters and the final 5 provide the low-dimensional representation of the LV geometry. Therefore, the trained emulator will map a set of material parameters $\boldsymbol{\theta}$, and low-dimensional LV geometry representation $\mathbf{h}$, to a vector of simulator outputs $\boldsymbol{y}$ that contains the LVV and 24 circumferential strains:

$$\eta : \boldsymbol{\theta}, \mathbf{h} \mapsto \mathbf{y} \tag{6.6}$$

For further details on the extraction of circumferential strains from the deformed mesh, see Figure 2.3.

In light of what we saw in Section 4.4.3, it appears that uniform coverage is somewhat suboptimal for our problem: if a function is nonstationary then a uniform coverage of input space will be a suboptimal design. This is because we require more data in the region of space where the function is more varied. As a solution, one might propose sequential design schemes that allow us to gradually learn the regions of space where the design should be focused. However, to achieve noticeable improvements in the design of computer experiments for the LV simulator, we would require the use of a nonstationary GP such as the one proposed by Heinonen et al. (2016). These methods increase the computational cost of sequential design even further since they rely on expensive MCMC procedures for hyperparameter estimation, and cannot realistically be used with dataset sizes in the order of thousands. Another issue with such approaches is the increase in computational costs since the generation of the design is no longer as parallelisable.

At this point in the thesis, the interest lies in determining the accuracy with which we can infer the material parameters of the HO law given only an approximation of the LV geometry. Exploration of more sophisticated design schemes could lead to a more efficient design in terms of the size of training set required, but the computational cost of generating such a design will be higher. Instead, we can just consider the same design scheme as Section 4.3.1, which allows for massive parallelisation of the forward simulations. This scheme also allows for the possibility of extending the emulator design in the event that the approximation is not sufficient. Later in this thesis more efficient methods for design will be considered that use a simple adaptation of uniform coverage to focus on the more complex part of the function.

In total, 16,000 training data were generated for training the emulator, with a further 615 used for testing. These numbers were selected based on the number of successful runs from a large batch of forward simulations. The bounds used for parameter space were $[0.1, 5]^4$, which was also used in Section 4.3.1. For PCA space, the bounds were obtained based on the projection of the training set of LV geometries into the lower dimensional space, extending the space in each direction by 10% of the most extreme projected value.

Training the emulator

Dalton et al. (2020) compared several different statistical emulation methods for approximation of the LV simulator. They found that local GPs provide the best results in the inverse estimation, with similar results obtained using a neural network. Relevant results from that study are provided in Table 6.2, where the 0.25,0.5,0.75 quantiles of the errors in the parameter estimates for a population of test cases are provided for three different emulation approaches[4]. It should be noted that the optimisation routines for each method were not the same. The neural network and sparse GP were fitted in Python using Tensorflow (Abadi et al., 2015) and GPflow (Matthews et al., 2017), respectively. The model calibration for these two models used gradient descent in Tensorflow, initiated from 100 different start points. On the other hand, the local GP, which used 500 nearest neighbours in function space similar to Algorithm 2, was fitted using fitrgp in MATLAB and model calibration was performed using GlobalSearch in MATLAB. More general information on these regression methods can be found in Chapter 3. The benefit of neural networks over

Table 6.2: Error in parameters estimated from a test dataset of simulations. The table gives the 0.25,0.5,0.75 quantiles of the mean squared error.

Method	Quantile		
	0.25	**0.5**	**0.75**
Sparse GP	7.1×10^{-3}	2.6×10^{-2}	6.1×10^{-2}
Local GP	4.3×10^{-3}	1.6×10^{-2}	4.7×10^{-2}
Neural network	6.6×10^{-3}	2.1×10^{-2}	6.0×10^{-2}

local GPs is their global nature, which is beneficial for the implementation of MCMC approaches to uncertainty quantification where we require accurate evaluation of the likelihood over the entire prior support. For this reason, a neural network emulator will be used as an approximation to the ground truth function. Cross validation was used to select the network architecture, with mini-batch optimisation performed using the Adam optimiser (Kingma and Ba, 2015). It was found that a network with two hidden layers of 200 nodes, with initial weights sampled from a $\mathcal{N}(0, 0.1)$ distribution, gives the lowest validation error for this problem. The small standard deviation in the Gaussian distribution is unsurprising when we consider the simplicity of the simulator function. Based on the discussion in Appendix B.2, the benefit of including emulator uncertainty in the model calibration scheme (which was small in that case) is judged to be outweighed by the extra computational costs involved in both training a Bayesian neural network and predicting from the resulting ensemble of models. The performance of the neural network emulator on the held-out test data set can be found in Figure 6.5, where the predictions from the model are compared with the outputs of the test dataset. The test data were obtained by

[4]The sparse GP results have been updated having adjusted the parameter optimisation routine.

running forward simulations for a sample of synthetic LV geometries created using random samples in the low-dimensional space, so the emulator should be able to match the data perfectly.

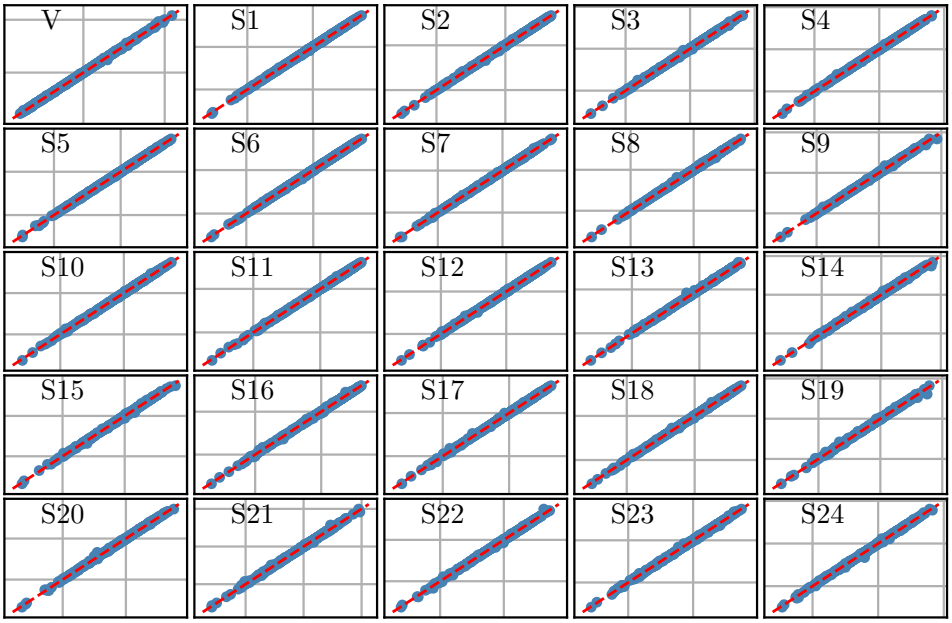


Figure 6.5: Prediction with the 5PC emulator at a held out dataset. Points lying close to $y = x$ indicates an accurate approximation.

6.2.3 Quantifying our uncertainty about the material parameters with MCMC

Model specification

In Chapter 4, optimisation was used to infer the material parameters of the HO law, and a bootstrap approach was used to quantify our uncertainty about the parameters in the model. In this chapter, a Bayesian approach to uncertainty quantification will be adopted, which was discussed in Section 3.2.2. Of interest is the accuracy with which we can infer the material parameters in light of the error introduced by the LV geometry approximation.

Let us consider inferring the parameters from measured data, obtained from a patient with an LV geometry $\mathbf{g}$ that has a 5 PC representation $\mathbf{h}$. These data consist of a measured LVV y_0^* and the 24 circumferential strains measured at end of diastole: $\boldsymbol{y}^* = (y_0^*, y_1^*, ..., y_{24}^*)$. We can assume that these data are corrupted with iid Gaussian noise with variance σ^2,

giving a log-likelihood function of the form:

$$p(\boldsymbol{y}^*|\boldsymbol{\theta}, \sigma^2) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\|\boldsymbol{y}^* - \boldsymbol{\eta}(\boldsymbol{\theta}, \mathbf{h})\|_2^2 \tag{6.7}$$

where $\boldsymbol{\eta}(\boldsymbol{\theta}, \mathbf{h})$ is the patient-specific prediction from the emulator at material parameters $\boldsymbol{\theta}$ (see (6.6) for further details). Within the likelihood function, the LVV is standardised according to (6.5) and the data are all standardised using the means and standard deviations of the emulator training outputs.

As discussed in Section 4.1.1, the parameters of the HO law exhibit poor identifiability. For this reason, the same parameterisation given in Section 4.1.1 and used by Noè et al. (2019) and Davies et al. (2019) will be adopted for the work in this simulation study where the 8 parameters in the HO law are replaced by $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4)$, which relate to the original HO law parameterisation through population average values found in Gao et al. (2017). For each of the θ parameters, a $\mathrm{u}(0.1, 5)$ prior is assumed, which is considered to be the physiological range of the parameters. Furthermore, an inverse Gamma prior is assumed for the noise variance. The posterior distribution of the material parameters, $\boldsymbol{\theta}$, is not available in closed form, so MCMC is used to sample from the posterior distribution Specifically, the NUTS variant of HMC (Hoffman and Gelman, 2014) will be used (see Section 3.2.5). In each implementation, ten chains are initiated from a Sobol sequence and the convergence is assessed using the Gelman-Rubin statistic (Gelman and Rubin, 1992) with a value less than 1.01 indicative of convergence.

A note on existing methods to speed up inference with HMC

Given that HMC will be used to sample from the posterior distribution of the parameters using an emulator-based likelihood function, it is worth considering existing methods for speeding up inference in HMC. Rasmussen (2003) proposed the use of Gaussian processes to speed up HMC simulations. Their method still relies on expensive forward simulations, but these are only used in an initial exploratory phase and in the acceptance of a proposed move. In other words, the leapfrog steps move along the GP approximated function so each iteration of the sampling phase requires only one evaluation of the expensive simulator. A similar approach was proposed by Conrad et al. (2016), but using local approximations in Metropolis Hastings instead of HMC. For the simulator considered in this study, these procedures would require days of simulation, because they both require sequential calls to the simulator that cannot be performed in parallel. For this reason, they are not applicable in the current problem and we must instead consider the sole use of an emulator-based log-likelihood.

6.2.4 Including a simple bias correction in the likelihood function

In Table 6.1 we saw that approximation of the LV geometry with a reconstruction obtained using PCA introduces an error to the simulator. In that analysis, the different subjects were grouped together under one quantitative assessment of accuracy. If we instead consider the subjects individually, a pattern begins to emerge in the comparison of the approximate simulations and ground truth simulations. In Figure 6.6, simulations using the 5PC LV geometry approximation (circles) and the MM geometry approximation (crosses) are plotted against simulations obtained from the actual LV geometry. Three simulations obtained from four different ground truth LV geometries are plotted, with the colour of the points distinguishing between the different LV geometries. The line at $y = x$ indicates perfect correspondence between the simulations obtained with an LV geometry approximation and those obtained with the actual LV geometry. As well as the obvious deviations from the line at $y = x$, points of the same colour are more or less parallel in each individual plot and, in cases where they deviate from parallel, the bias present between ground truth and approximate simulation is at least in the same direction. This begs the question of how well we can infer the parameters if we allow for a simple bias correction in the predictions from the 5PC model.

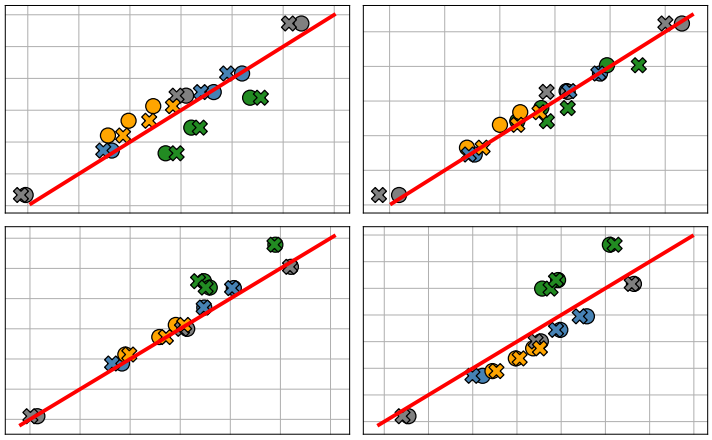


Figure 6.6: Simulations obtained from LV geometry approximation versus simulations obtained with exact LV geometry. The colour of points distinguishes between different ground truth LV geometries and we see that points of the same colour all tend to lie on the same side of the line at $y = x$. The different symbols correspond to a 5PC LV geometry approximation (circles) and a mean mesh LV geometry approximation (crosses). Each square on the grid corresponds to a difference of 0.025 in circumferential strain (in both the horizontal and vertical axes).

To allow for a bias correction in the emulator, we can update (6.7) to account for deviations in the emulator. This update provided by the inclusion of $\mathbf{b}$, which contains a

separate bias correction for each different output of the model:

$$p(\boldsymbol{y}^*|\boldsymbol{\theta}, \sigma^2) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\|\boldsymbol{y}^* - (\boldsymbol{\eta}(\boldsymbol{\theta}, \mathbf{h}) + \mathbf{b})\|_2^2 \tag{6.8}$$

Obtaining b_i relies on running forward simulations, but this can be done with very few parallel simulations, with total run time equal to around ten minutes. The total computation time, including the time for obtaining the bias correction and running the MCMC samplers, is still comfortably below 15 minutes. However, use of the method now has the extra burden of requiring an Abaqus license at inference time, which might not be so readily available. Although the simple bias correction proposed here does help to correct for errors in the generalised emulator, the actual correction required is inevitably more complex than a constant. In Appendix F an alternative, more accurate, correction mechanism is proposed that relies on more forward simulations than the simple correction that is considered here.

6.3 Simulation study

6.3.1 Implementation details

Parameter inference was performed in Tensorflow probability (TFP), allowing for the implementation of HMC and NUTS for emulator models constructed in Tensorflow. The run times required for 1000 samples were typically lower than 10 seconds[5]. For comparison, using simple Metropolis Hastings with the simulator would require 6×10^6 seconds, which equals roughly 69 days. To give some idea of the efficiency of the sampling routine, 1000 samples obtained using NUTS typically gives an effective sample size of 300-400. This could be improved by tuning of the mass matrix in NUTS, allowing for better fit to the posterior surface. However, mass matrix adaptation was not implemented in TFP at the time of this analysis.

6.3.2 Quantifying our uncertainty with Bayesian inference

Of principal concern is the effect that the LV geometry approximation has on the modelling of a patient's LV dynamics in diastole. With regards to the forward problem, the results in Table 6.1 suggested that the improvement in accuracy is small when we include LV geometry information obtained from PCA. An understanding of the effect this has on the inverse problem is best obtained through a simulation study. For this task, the emulator discussed in Section 6.2.2 will be used to provide a computationally efficient evaluation of

[5]The computation time was always in the order of seconds, whether running on a computing cluster or laptop.

the likelihood function from (6.7), allowing us to sample the material parameters of the model using either a 5 PC representation of the LV geometry or a mean mesh representation. Given a set of 40 LV geometries, a total of 120 synthetic test points were generated as follows: obtain 120 samples from a Sobol sequence in 4-dimensional material parameter space and evaluate in the simulator three different parameters with every one of the 40 different LV geometries[6]. This process gives us 120 synthetic datasets obtained from 120 unique parameter-geometry pairs. Conditional on each of these synthetic datasets, we can sample the material parameters according to the details given in Section 6.2.3.

Using the MCMC samples, the posterior distributions are summarised in Figure 6.7. The left plot shows the 120 median absolute errors of the marginal posterior distributions for each parameter and the right plot shows the corresponding interquartile ranges. Comparing the MM and 5PC representations, we see that both provide similar results, with the 5PC representation slightly outperforming the mean mesh. Looking at the different parameters, the IQR and MAE are larger for θ_3 and θ_4 and smallest for θ_2.

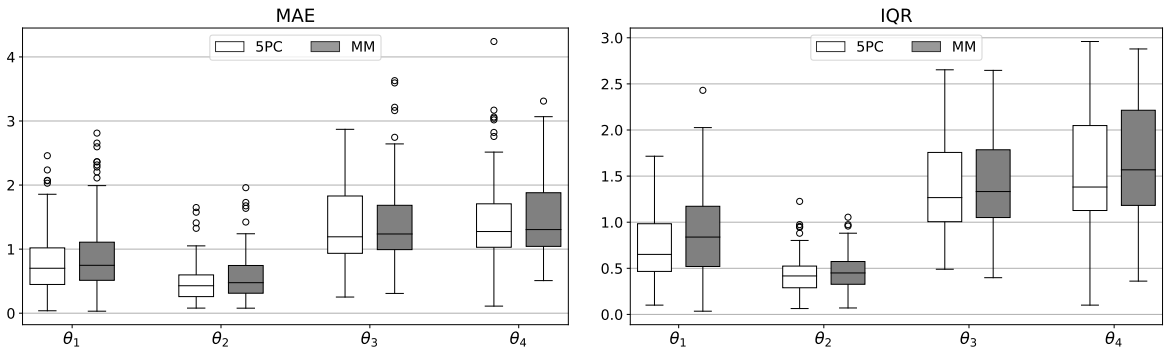


Figure 6.7: Comparing results of parameter estimation with a mean mesh LV geometry representation and an LV geometry representation obtained from 5 principal components.

In Figure 6.8, some example inference results are plotted in more detail in the form of marginal posterior density plots. The spread of the posterior densities varies quite substantially, especially for θ_1 and θ_2. One reason for this could be that certain LV geometries are better approximated using 5PCs, so the error and uncertainty in the posterior distributions would decrease. However, another possibility is that this is the result of the nonstationarity discussed in Section 4.4.3. In such a scenario, the uncertainty would be lower in the region of space where the function varies more rapidly. This possibility will now be explored in greater detail.

[6]Note that the test LV geometries were not out of sample. This means that the training set used to generate the principal components of the emulator contained the test LV geometries. However, it was found that the in sample and out of sample performance of PCA was similar for these test cases.

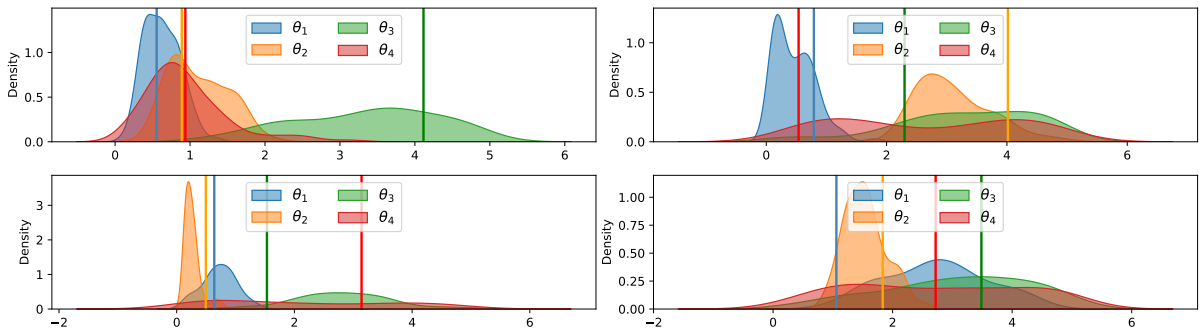


Figure 6.8: A more detailed look at the posterior inference of 4 test examples. The different colours indicate the material parameter, with matching vertical lines showing the ground truth value of that parameter.

6.3.3 Investigating the likelihood function

Let us first consider how the LV geometry approximation affects the loss function. In Figure 6.9, two typical loss functions, approximated using the 5PC emulator, are visualised with the ground truth parameter shown as a red point and the new optimum shown in black. The effect that the LV geometry approximation has on the loss function changes quite substantially between the two plots, displaying a more fundamental property of the simulator. On the left, the optimum shifts by as much as 0.3 in the θ_1 direction as a result of the LV geometry approximation and on the right the change in the optimum is negligible. Reconsidering Figure 6.1, we can see that the rate of change of the function over the parameter space is not constant, which was modelled by a nonstationary kernel in Appendix B.2. The relationship for the simulator, $\zeta(\cdot)$, can be expressed as follows:

$$\zeta(\boldsymbol{\theta}_1 + \delta\boldsymbol{\theta}) - \zeta(\boldsymbol{\theta}_1) > \zeta(\boldsymbol{\theta}_2 + \delta\boldsymbol{\theta}) - \zeta(\boldsymbol{\theta}_2)$$

where $\boldsymbol{\theta}_1$ is a parameter configuration corresponding to a soft material and $\boldsymbol{\theta}_2$ corresponds to one that is stiff. The consequence this has on the parameter estimation, in the presence of errors incurred by LV geometry approximation, is intuitive: for soft parameters, the deviation resulting from an LV geometry approximation has less effect on the optimum of the loss function than with stiffer parameters. This change in the function variation manifests itself in other ways: for ground truth parameters in the stiffer region, we see a long ridge forming at the optimum of a flatter loss function. This latter feature is also indicative of correlations between the model parameters.

It is worth investigating in more detail how the varying rate of change of the simulator outputs can affect the parameter identifiability. The procedure to do so is outlined in Algorithm 3. This will be repeated for every combination of parameters, each time sampling all four parameters together but only summarising the results for the two parameters that vary during synthetic data generation. The uncertainty in the parameters

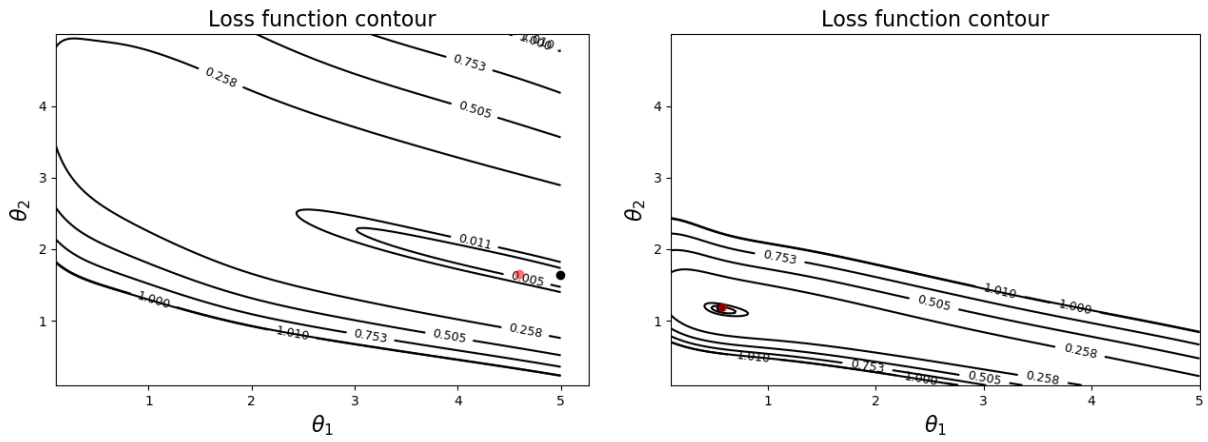


Figure 6.9: Visualising the effect of PCA on the loss function. By reducing the geometry to a 5PC representation we change the loss function. This causes the global minimum to shift from the red point (the minimum obtained with the full LV geometry) to the black point.

will be summarised by the IQR in the marginal posterior distribution of each parameter.

Algorithm 3 Producing a contour plot of IQR over $\theta_1 - \theta_2$ space

1: Generate grid of $\theta_1 - \theta_2$ values in the range 0.1-5.
2: For each point in the grid, obtain synthetic data by evaluating the emulator, with θ_3 and θ_4 fixed equal to 1.
3: Add $N(0, 0.1)$ noise to the standardised test data (standardised by the mean and standard deviation of the emulator training data).
4: For each test point, obtain samples of the material parameters using HMC and use these to approximate the IQR of the posterior distribution.
5: Create contour plot of the IQR at each point on the grid of material parameter values.

Figure 6.10 shows the results of this analysis. Each row shows how the IQR of an individual material parameter, the *target parameter*, changes as we vary the ground truth parameter set over the joint space of the target parameter and one of the other material parameters, the *extra parameter*, while keeping the other parameters fixed equal to 1. In each plot, the target parameter varies along the vertical axis, the extra parameter varies along the horizontal axis and the title is arranged as follows: *main parameter-extra parameter*. For θ_1, shown in the top row of plots, we see that the uncertainty increases as we increase the target parameter without much effect from the value of the extra parameter. The exception is θ_2, where small values of the extra parameter make θ_1 less identifiable if θ_1 is large. The uncertainty in θ_2, shown in the second row, appears to depend not just on its value but also the value of the extra parameter. When the extra parameter is θ_1 or θ_3 then increasing both parameters generally increases the uncertainty in θ_2. The only time this changes is when the extra parameter is θ_4, which has less effect on the uncertainty of θ_2. Interestingly, the uncertainty in θ_2 appears to have less variation than that of θ_1, which

can be seen by comparing the values in the $\theta_1 - \theta_2$ and $\theta_2 - \theta_1$ plots. The uncertainty in θ_1 changes from 0.016 to 0.088 as we vary the value of the parameters while the uncertainty in θ_2 changes from 0.036 to 0.078. The pattern in variation of θ_3 is similar to that of θ_2 although we see that the uncertainty has increased quite substantially. This is consistent with the findings in Figure 6.7. The pattern in θ_4 uncertainty is quite different to the others. When θ_1 is the extra parameter, the pattern is more difficult to interpret and it appears as though for small values of the parameters, uncertainty in θ_4 increase as both parameters increase. Moreover, the value of θ_1 appears to have a substantial effect on the identifiability of θ_4, with the uncertainty increasing from 0.24 to 0.72 over the joint space. When θ_2 and θ_3 are the extra parameters, the uncertainty in θ_4 appears to be lowest for small values of θ_4 and large values of the extra parameter, and increases as the extra parameter decreases and θ_4 increases. Overall, from comparing typical values found in the contour plots, θ_3 and θ_4 are less identifiable than θ_1 and θ_2.

6.3.4 Further exploration of the effect of LV geometry approximation

Figure 6.10 showed that parameter identifiability changes over the parameter space. Parameters that describe a soft material tend to be more identifiable than those describing a stiff material and, as we saw in Figure 6.9, the optimum of the loss function is less affected by the LV geometry approximation when the ground truth parameters are softer. Since the material parameters for testing were generated according to a uniform distribution, the parameters in the stiffer region could be hijacking the inference results in Figure 6.7. These stiffer parameters could even be configurations that we would never find in reality and we should investigate the effect of the LV geometry approximation for inferring parameters that are more realistic.

One easy check to determine the type of material the parameters describe is to determine the corresponding ejection fraction (EF):

$$\frac{\text{End diastolic volume-End systolic volume}}{\text{End diastolic volume}} \tag{6.9}$$

This quantity is often used to assess cardiac function in the clinic. According to Hudson and Pettit (2020), normal ejection fraction is greater than 55%. In the test data used for Figure 6.7, 61% of cases had an EF lower than 55%. To determine the effect this has on the inference results, we can break the results up according to the EF. These results are given in Figure 6.11, with MAE on the left and IQR on the right. In each subplot, boxes to the right of the red dashed line are the results for test data with EF>55%. We see that the results do improve, particularly for θ_1, but we still have a median error of around 0.4. The uncertainty about the parameters appears to change more drastically, especially for

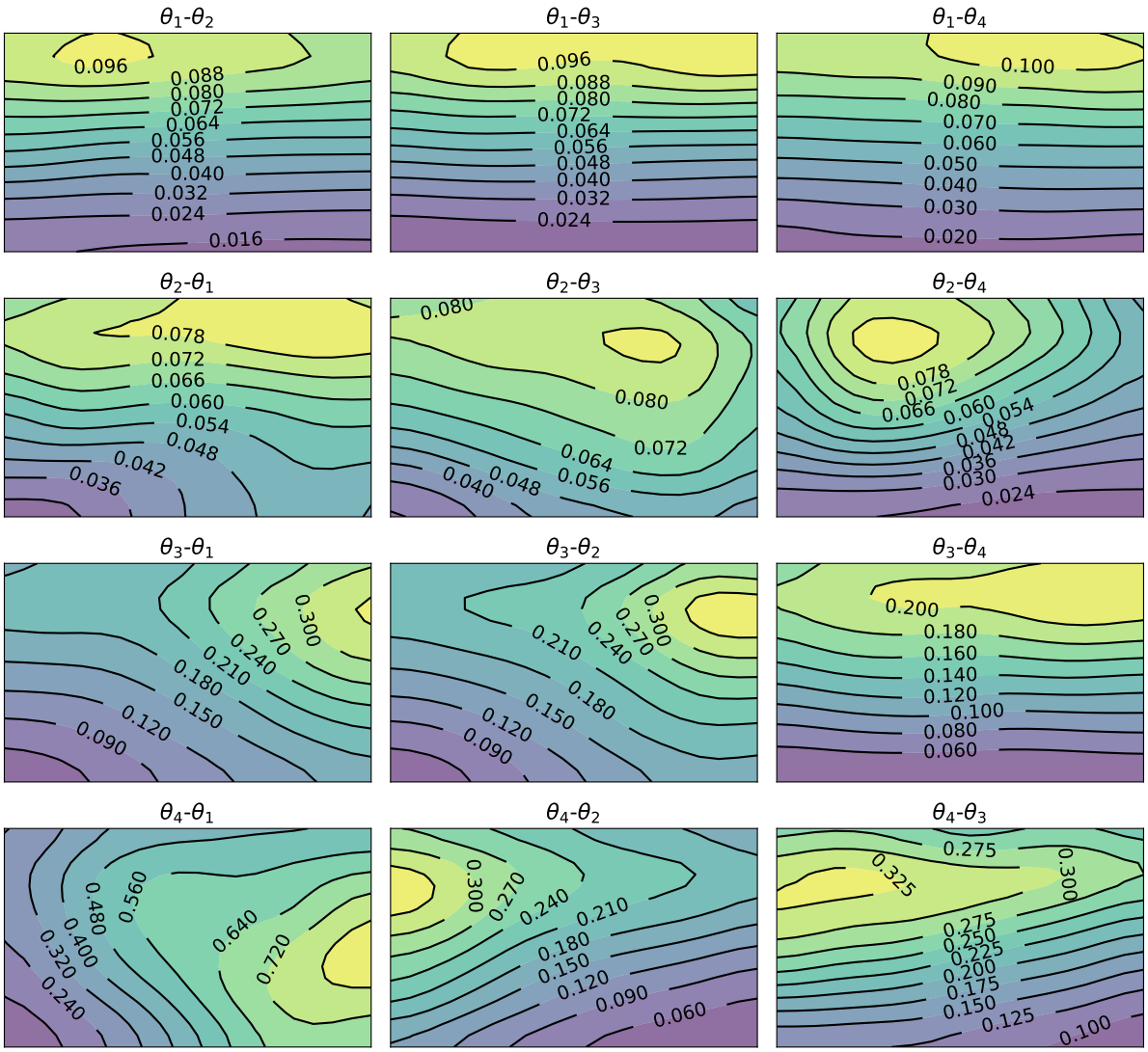


Figure 6.10: Showing the effect of the ground truth parameter value on the uncertainty in our parameters. Each row shows how our uncertainty about the first parameter in the title of the plot changes over the given space, with bounds on the parameters being 0.5 and 5.0. The uncertainty is measured as the IQR in the marginal posterior distribution of the parameter.

θ_1. The change in results depending on the EF is another indication of the effects of the function nonstationarity on our inference since a high EF occurs in the region of space where the function varies more rapidly.

6.3.5 Inference while accounting for bias correction

The final analysis will consider the Bayesian inference task while adopting the likelihood function from Section 6.2.4. Thirty of the LV geometries that have been reconstructed from PCA in a LOOCV manner will be used. For each LV geometry, 3 simulations were obtained at different material parameter values for the analysis in Table 6.1. These

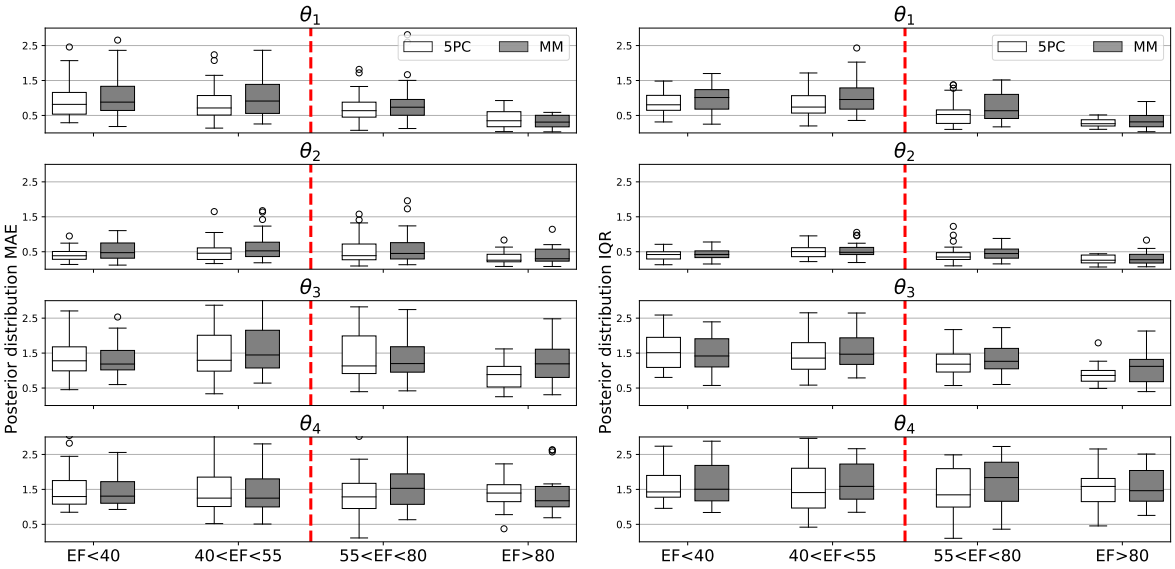


Figure 6.11: Using the ejection fraction we can distinguish realistic parameter configurations (with EF greater than 55%) and unrealistic ones (EF less than 55%), before assessing the results of the posterior inference in both cases, as divided by the vertical dashed line.

simulations were obtained with a 5PC LV geometry approximation $\{\boldsymbol{y}_5^{(i)}\}_{i=1}^3$, a mean mesh LV geometry approximation $\{\boldsymbol{y}_M^{(i)}\}_{i=1}^3$, and the ground truth LV geometry, $\{\boldsymbol{y}_T^{(i)}\}_{i=1}^3$. The testing procedure for assessing the adequacy of a bias correction is as follows: treating the jth simulation of LV geometry i as the test data for inference, obtain a bias correction as follows:

$$\mathbf{b}_5 = \frac{1}{2}\sum_{k\in\{1,2,3\}\setminus\{j\}} (\boldsymbol{y}_T^{(k)} - \hat{\boldsymbol{y}}_5^{(k)}) \quad \mathbf{b}_M = \frac{1}{2}\sum_{k\in\{1,2,3\}\setminus\{j\}} (\boldsymbol{y}_T^{(k)} - \hat{\boldsymbol{y}}_M^{(k)}) \tag{6.10}$$

The success of this correction is assessed in two ways: first by comparing the predictions, corrected using a bias correction, with the ground truth simulation and then using parameter inference, with the bias included in the log-likelihood from (6.8). The comparison based on predictions is found in Table 6.3, where the bias corrected prediction errors are compared with baseline results from the 10 PC LV geometry approximation without any bias correction (note that the results are slightly different to those found in Table 6.1 because a subset of the data is now used). By correcting for the simple bias in the predictions from the approximate LV geometry, we achieve better results than with the 10PC LV geometry approximation.

The results in parameter space, comparing the likelihood function with and without bias correction, are found in Figure 6.12. There is a quite substantial difference in the results when we introduce this simple correction, in some cases the median absolute error is reduced by as much as a factor of 0.5. The change in IQR is equally as large, indicating the reduction in discrepancy of the model predictions from the ground truth (recall that

Table 6.3: Comparing the accuracy of simulations with different LV geometry approximations, allowing for corrections in the model predictions.

Reconstruction	End diastolic volume RAE 0.5 (0.25,0.75) quantiles	Circumferential strain RAE 0.5 (0.25,0.75) quantiles
0PC corrected	0.007, (0.006, 0.01)	0.02, (0.009, 0.04)
5PC corrected	0.007 (0.003, 0.01)	0.02 (0.008, 0.04)
10PC	0.013 (0.002,0.049)	0.032 (0.02,0.07)

because data are generated from actual LV geometries without any added noise, the only noise entering the problem is the result of the discrepancy caused by the LV geometry approximation).

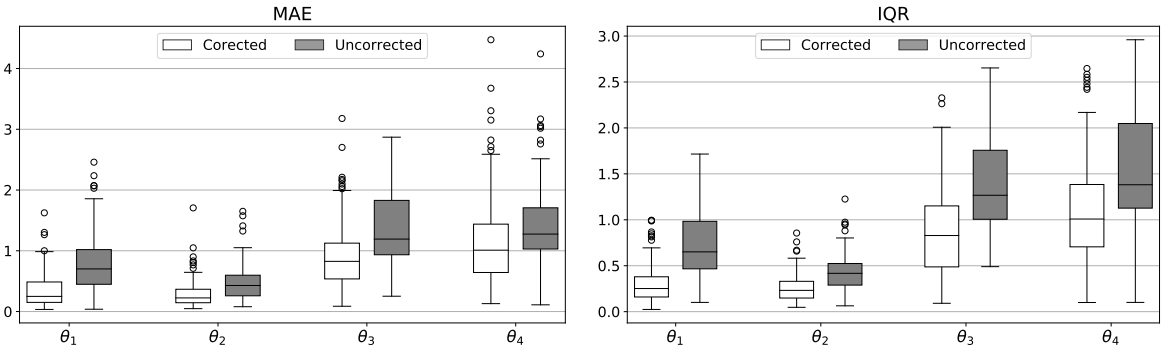


Figure 6.12: Comparing results of parameter estimation with a 5PC representation of the LV geometry when we include a simple bias correction in the likelihood function (white) and when we do not include a bias correction (grey).

6.4 Discussion and conclusion

Representation of the LV geometry with five principal components, although far superior in terms of reconstruction error, does not give much improvement over the mean mesh in terms of accuracy in simulated circumferential strains, as seen in Table 6.1. Even increasing the number of principal components to ten does not appear to provide much improvement over the mean mesh for this purpose. This is surprising when considered in the context of the results from Table 5.2, which showed that there is a large improvement in MSE—by a whole order of magnitude—when we introduce LV geometry information in the form of 5 PCs. A possible justification for these results is the fact that geometry reconstruction methods attempt to capture global shape changes, whereas the circumferential strains will likely be influenced by local properties of the LV geometry.

By quantifying the uncertainty in the parameters, the identifiability of the parameters was explored in the presence of LV geometry errors. These results were shown in Figure 6.7, where the median absolute error and interquartile ranges of the 120 posterior

distributions were summarised. These results showed that θ_3 and θ_4 are less identifiable than θ_1 and θ_2. Given knowledge of the physiological meaning of the parameters, these results make sense: θ_3 relates to behaviour in the high stretch region and θ_4 largely governs the behaviour of components of the system that are not currently included in the likelihood function evaluation. The absence of information in the high-stretch region is apparent if we consider our inference results in stress–stretch space. Figure 6.13 shows a posterior distribution of stress–stretch curves obtained by evaluating the H-O model for the samples of material parameters represented by the densities in the left of the figure. The spread in the distribution of stress values increases as we increase the stretch, resulting from the lack of information about large-stretch tissue behaviour.

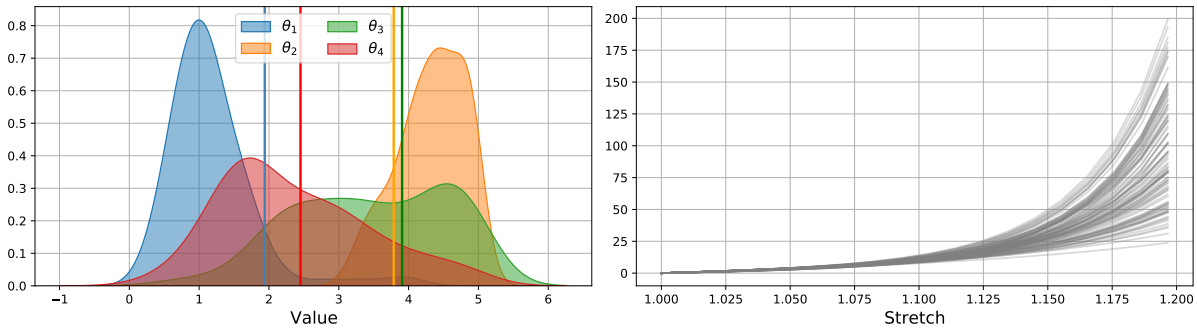


Figure 6.13: Typical inference results. θ_1 and θ_2 are well identified but θ_3 and θ_4 are subject to large uncertainty. Translating to the tissue properties on the right, we see that the uncertainty is large for high stretch.

In Figure 6.10, the parameter identifiability was shown to change over the parameter space, affecting the results of inference with synthetic data generated from real LV geometries in Figure 6.11. Again, this result is intuitive: with stiff materials, the small deviation in simulations between an approximated LV geometry and the ground truth LV geometry corresponds to large deviations in parameter space, as a result of the slowly varying function in this stiffer parameter regime. With softer materials and a faster varying simulator function, the small deviations that result from approximation with PCA or the mean mesh no longer correspond to large deviations in parameter space. These results show the effect that the nonstationarity observed in Section 4.4.3 can have on our parameter identifiability.

The accuracy of the parameter inference when adopting a representation of the LV geometry obtained using PCA is worse than the results of Chapter 4. At first this would appear concerning, pointing to the new LV geometry representation as the source of the issue, or perhaps the new emulator, which will inevitably be subject to higher error as a result of the increase in dimension of the input space. However, the results from Chapter 4 have to be considered with a word of caution. In that study, only one LV geometry was considered and we had zero discrepancy between the emulator and the test data. The

result of LV geometry approximation is effectively to introduce extra noise, likely in the form of structured noise. While this will introduce bias to the inference results, the errors incurred in simulator output space as a result of an LV geometry approximation can only really be considered in the context of actual measured volume and strain data. Moreover, proper assessment of the errors incurred by PCA would require a comparison with the inference results of LV geometries reconstructed by different operators. If the level of error incurred by PCA is lower than that incurred when changing operator then we can have more confidence in our approximation. This will be explored in Chapter 9.

Overall, the results of this chapter have shed light on the task of patient-specific parameter estimation. In particular, the use of MCMC methods has shown the poor identifiability of θ_3 and θ_4. Moreover, we have seen that the identifiability of the tissue changes depending on the ground truth value of the parameters. When these describe a soft material, the tissue properties are easier to learn than when the tissue is stiff. This highlights the importance of generating data from realistic ground truth parameters in order to assess parameter inference methods. Moreover, it appears that the inclusion of information about the tissue at higher stretches will be necessary to improve the identifiability of the parameters. With this in mind, the next chapter will consider the inclusion of information from ex vivo experiments in an attempt to improve parameter identifiability.

Chapter 7

Improved inference by combining in vivo strain data with ex vivo volume-pressure data

A key component of a Bayesian inference model is the prior distribution. Expressing knowledge gained from previous experiments, this can help to constrain our posterior distribution to regions that make physiological sense. Until this point, this thesis has considered the non-informative case of a uniform prior, which expresses the prior belief that all regions in input space are equally likely. The current chapter proposes to introduce prior information to the parameter estimation problem by way of ex vivo data from the literature. Two alternative priors, an empirical Bayesian prior (where the prior depends on the data) and a nonempirical prior, will be proposed before comparing these on a series of problems using both synthetic and measured data. The work in this chapter is based on work that has been submitted to the Journal of the Royal Statistical Society Series C and received minor revisions.

7.1 Background

Until this point, the focus of this thesis has been the likelihood function. For the purpose of efficient estimation of the parameters, emulation was used to expedite the calculation of the likelihood. The theta parameterisation of the H-O model was originally introduced in Chapter 4 (specifically (4.2)), where a fixed LV geometry emulator was used to successfully estimate the parameters from synthetically generated observations and real observations measured from CMR scans. In Chapter 6, the emulator was generalised to different LV geometries by introducing a PCA representation of the LV geometry to the emulator's input space. Using MCMC, the parameters were inferred, showing that the uncertainty in θ_3 and θ_4 was higher than in θ_1 and θ_2. This is the problem that is of interest in the

current study.

The behaviour of the myocardium, which was discussed in Section 2.1.2, is governed by components operating at a microscopic scale within the tissue whose influence changes with the level of stretch in the muscle (Voorhees and Han, 2015). This behaviour is replicated by the H-O model from (8.1), where different mathematical terms take effect at different stretches. In particular, some of the parameters in this expression relate to the behaviour at small stretch and some to the behaviour at larger stretch. This property is useful in the forward problem because it allows us to simulate the behaviour of the tissue at stretches beyond those typically observed in vivo. However, it also introduces identifiability problems to the inverse problem because the in vivo data do not inform us about the behaviour of the material at higher stretches. To overcome this issue, the current study proposes to introduce ex vivo information to the inference routine through a relationship known as the Klotz curve (Klotz et al., 2006).

Klotz et al. (2006) investigated the dependence between pressure and end-diastolic volume of the LV cavity (LVV) over a large population of various diseased patients and healthy volunteers. In their study, the end-diastolic volume-pressure curves varied substantially and spanned a wide range of volumes. However, they found that on standardising the LVV, the curves become nearly identical, and that despite different disease status, the shape of the curve is nearly the same (Figure 2 in Klotz et al. (2006)), with a functional form provided by:

$$P \;=\; \alpha \exp\left(\beta \tilde{V}\right), \text{ where } \tilde{V} \;=\; \frac{V - V_0}{V_{30} - V_0}. \tag{7.1}$$

V is the LVV at pressure P, V_{30} is the LVV at pressure 30 mmHg and V_0 is the volume of the unloaded left ventricle, at pressure 0 mmHg. Throughout this chapter, V_P will be used to denote the LVV at pressure P mmHg, and $\tilde{V}_P$ will be used to denote the corresponding normalised volume $\tilde{V}_P = \frac{V_P - V_0}{V_{30} - V_0}$.

The Klotz curve has already been explored in the context of cardio mechanical modelling. In particular, Palit et al. (2018a); Peirlinck et al. (2019); Krishnamurthy et al. (2013b) used it in combination with in vivo data to estimate the cardio-mechanic parameters, while Hadjicharalambous et al. (2015) considered it in a comparison of different constitutive laws. The current study will expand on the current literature, particularly with regards to parameter estimation, in the following ways: firstly, a prior distribution that is based on the Klotz curve will be introduced, allowing us to correctly introduce this information in a Bayesian inference scheme for the material parameters. The prior will be implemented using statistical emulators, enabling the introduction of the Klotz information to the parameter inference procedure while avoiding any large computational costs. Secondly, these priors will be used to improve parameter inference in the presence of LV geometry errors and information loss caused by the dimension reduction. The result is a reduction in the posterior uncertainty and improving accuracy of the parameter

estimation. Thirdly, this study contributes a comparative performance assessment of two alternative prior representations, using both synthetic data and real CMR images, and quantifying the effect that the inclusion of the empirical Klotz law has on the inference.

7.2 Methodology

As discussed above, two alternative prior distributions will be formulated for the introduction of information from the Klotz curve, each derived from the parametric form suggested by Klotz et al. (2006) and provided in (7.1). The first distribution, which is called the *nonempirical Klotz prior*, will relate two emulated volumes at different end diastolic pressures while the second—called the *empirical Klotz prior*—will require the introduction of the LVV to the expression, resulting in a prior distribution that depends on the measured data. Both of these priors will require samples of the parameters from (7.1), for which a rearrangement of the original model is considered, expressing the normalised volume as a function of pressure. For now, consider this relationship to be of the form $\tilde{V} = f(p; \boldsymbol{\phi})$ where f depends on some parameters $\boldsymbol{\phi}$. The form of this function is deliberately not made explicit at the moment so that the discussion generalises to both of the Klotz prior distributions. Assuming a heteroscedastic noise model:

$$\tilde{V} \sim \mathcal{N}\left(\tilde{V} \middle| f(P; \boldsymbol{\phi}), \omega(P)\right), \tag{7.2}$$

where

$$\log(\omega(P)^2) \;=\; a + bP + cP^2 \tag{7.3}$$

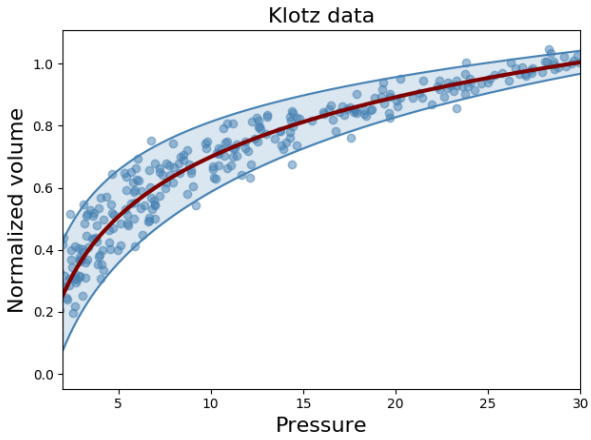
the parameters $\boldsymbol{\phi}, a, b$ and c are sampled from the posterior distribution conditional on the data from Figure 7.1. The posterior mean values of a, b and c are used to provide a heteroscedastic noise model estimate. Prediction of the Klotz curve is shown in Figure 7.1, along with uncertainty bands obtained as $\pm 2\omega(P)$.

7.2.1 Emulating the likelihood function

The data available from in vivo CMR images are $K = 24$ end diastolic circumferential strains at well-defined positions on the endocardial wall (see Figure 2.3) and the LVV. In what follows, the measured quantities are denoted by:

$$\boldsymbol{y}^* = (y_0^*, y_1^*, \ldots, y_K^*)^\mathsf{T}. \tag{7.4}$$

Given an emulator training set, the mean and standard deviation of the simulations $\boldsymbol{\mu_y}$ and $\boldsymbol{\sigma_y}$, can be used to standardise these data. The corresponding true quantities, which



(a)

Figure 7.1: A heteroscedastic noise model is fitted to the normalised volume as a function of pressure. The data, represented by the scatter points, were obtained from Klotz et al. (2006) using the Grabit function in MATLAB. Pressures lower than 2 mmHg are excluded from the model fitting. The shaded area shows $\pm 2\omega(P)$.

are modelled by the neural network emulator $\boldsymbol{\eta}$ from Chapter 6, are:

$$\boldsymbol{\eta}(\boldsymbol{\theta}, \mathbf{h}) = \Big(\eta_0(\boldsymbol{\theta}, \mathbf{h}), \eta_1(\boldsymbol{\theta}, \mathbf{h}), \ldots, \eta_K(\boldsymbol{\theta}, \mathbf{h})\Big)^{\mathsf{T}}, \tag{7.5}$$

where the dependence of the emulator on the cardio-mechanic parameters $\boldsymbol{\theta}$ and a 5 PC approximation of the LV geometry $\mathbf{h}$ has been made explicit.

Under the assumption that the measurement noise is iid additive Gaussian with variance σ_m^2:

$$\boldsymbol{y}^* \;=\; \boldsymbol{\eta}(\boldsymbol{\theta}, \mathbf{h}) + \boldsymbol{\xi}; \quad \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \sigma_m^2 \mathbf{I}), \tag{7.6}$$

we obtain the log likelihood:

$$\log p\Big(\boldsymbol{y}^* | \boldsymbol{\eta}(\boldsymbol{\theta}, \mathbf{h})\Big) \;=\; \frac{-1}{2\sigma_m^2} \sum_{i=0}^{K} \Big(y_i^* - \eta_i(\boldsymbol{\theta}, \mathbf{h})\Big)^2 - \frac{K+1}{2} \log(2\pi\sigma_m^2). \tag{7.7}$$

In Chapter 6, this log-likelihood function was used in combination with a $\mathrm{u}(0.1, 5)$ prior on $\boldsymbol{\theta}$ to sample from the posterior distribution of the parameters. The aim in the current study is to combine the in vivo likelihood function with an informative prior derived from the Klotz curve.

7.2.2 Method A: the nonempirical Klotz prior

The emulated form of the cardio-mechanic model from Section 7.2.1 is used to model the in vivo data, represented by the right branch in Figure 7.2 with an arrow from the cardio mechanic model to $\boldsymbol{y}^*$ (the in vivo data). To include extra information, obtained from the

ex vivo study performed by Klotz et al. (2006), information from the relationship in (7.1) will be introduced to the inference problem. Eliminating α from the Klotz curve function in (7.1) (see Appendix D.1), we can evaluate the following function for some parameter value β, providing a prediction of the normalised volume at pressure 20 mmHg based on the Klotz relationship:

$$\tilde{V}_{20} = \frac{1}{\beta}\log\left(\frac{2}{3}\right) + 1 \tag{7.8}$$

Recalling the general form of the Klotz noise model in (7.2), we have the following function:

$$f(P;\boldsymbol{\phi}) = \frac{1}{\beta}\log\left(\frac{P}{30}\right) + 1 \tag{7.9}$$

so that $\boldsymbol{\phi} = \beta$. Information from the ex vivo data is introduced by inferring the parameters of this model β, a, b and c, from the Klotz data plotted in Figure 7.1. Assuming priors of the form $\beta \sim \mathrm{u}(0, 10)$[1], $a, b, c \sim \mathrm{u}(-100, 100)$, where the bounds are deliberately overly conservative, these parameters are sampled using MCMC. The posterior distribution on β can be approximated with a Normal distribution:

$$\pi(\beta) = \mathcal{N}(\beta|\mu_\beta, \sigma_\beta^2).$$

The validity of which was confirmed using the Shapiro-Wilk test.

Estimating the mean μ_β and standard deviation σ_β using the posterior samples, we can use the delta method (see Appendix D.2) to propagate the uncertainty in β through to the normalised volume via (7.8):

$$\begin{aligned}\pi_{Kl}(\tilde{V}_{20}) &\approx \mathcal{N}\left(\tilde{V}_{20}\middle|\tilde{V}_{20}(\mu_\beta), \left[\frac{\partial\tilde{V}_{20}}{\partial\beta}\right]_{\mu_\beta}^2 \sigma_\beta^2\right) \\ &= \mathcal{N}\left(\tilde{V}_{20}\middle|\frac{1}{\mu_\beta}\log\left(\frac{2}{3}\right) + 1, \left[\frac{1}{\mu_\beta^2}\log\left(\frac{2}{3}\right)\right]^2 \sigma_\beta^2\right)\end{aligned} \tag{7.10}$$

giving a distribution on $\tilde{V}_{20}$ induced by the Klotz data and function. This part of the model is represented by the left branch of Figure 7.2, where we relate the normalised volume with the ex vivo data, $\boldsymbol{y}_{ex}$, through the *Klotz curve model*.

Reconsidering Figure 7.1, we see that there is fairly substantial variance in the data. Due to the restrictive nature of the model in (7.8) and the large quantity of data, much of the variation is inferred as noise during inference instead of an increased uncertainty in

[1]Positivity is a result of physiological knowledge that the normalised volume increases as pressure increases.

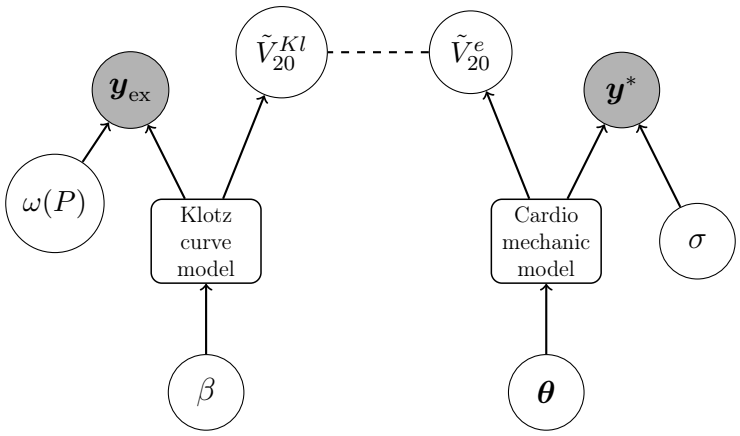


Figure 7.2: Including the nonempirical Klotz prior in the cardio-mechanic parameter estimation. Observed quantities are in grey circles, functions are in boxes and variables are in white circles. In addition to using the cardio-mechanic model for modelling in vivo data ($\boldsymbol{y}^*$) at the in vivo reference pressure (right branch), it is also used to model the LVV at higher ex vivo reference pressures, providing a distribution on the normalised volume, which is labelled as $\tilde{V}_{20}^{e}$. In addition, an empirical law based on the function from (7.1) fitted to ex vivo data ($\boldsymbol{y}_{\text{ex}}$) provides a distribution on the normalised volume at the same reference pressure (left branch), with noise model assumed to be heteroscedastic (see details below (7.2)). Despite the fact that the normalised volume resulting from the emulator and the Klotz relationship are actually the same variable, we choose to distinguish between these based on their distributions ($\tilde{V}_{20}^{e}$ for emulator induced distribution and $\tilde{V}_{20}^{Kl}$ for Klotz induced.) The two distributions are combined via a product-of-experts approach (dashed line).

the parameters. This could lead to a slight underestimation of the true variation in the Klotz curve so an alternative form of (7.10) will be considered, where the heteroscedastic noise at pressure 20 mmHg, given by equation (7.3) with $P = 20$ mmHg, replaces the variance of the Normal distribution. This model is referred to as the nonempirical Klotz prior with model mismatch.

Instead of predicting the normalised volume using the Klotz curve, we can predict this quantity for a given set of material parameters using the simulator from Figure 2.2. This allows us to relate the normalised volume from (7.1) with the material parameters. Training Gaussian process emulators, we learn a probabilistic representation of the LVV at pressure 20 mmHg and 30 mmHg. Using the delta method, we can propagate the uncertainty in V_{20} and V_{30}—induced by the probabilistic nature of the emulators—through to the normalised volume:

$$\pi_e(\tilde{V}_{20}) \;=\; \mathcal{N}(.|\mu, \sigma^2), \tag{7.11}$$

where

$$\begin{aligned}\mu=&\frac{\hat{V}_{20}(\boldsymbol{\theta},\tilde{\mathbf{h}})-V_0}{\hat{V}_{30}(\boldsymbol{\theta},\tilde{\mathbf{h}})-V_0},\\ \sigma^2=&\left(\frac{1}{\hat{V}_{30}(\boldsymbol{\theta},\tilde{\mathbf{h}})-V_0}\right)^2\hat{\sigma}^2_{20}(\boldsymbol{\theta},\tilde{\mathbf{h}})+\left(\frac{\hat{V}_{20}(\boldsymbol{\theta},\tilde{\mathbf{h}})-V_0}{(\hat{V}_{30}(\boldsymbol{\theta},\tilde{\mathbf{h}})-V_0)^2}\right)^2\hat{\sigma}^2_{30}(\boldsymbol{\theta},\tilde{\mathbf{h}}).\end{aligned} \tag{7.12}$$

A more detailed derivation can be found in Appendix D.4. Here, $\hat{V}_i(\boldsymbol{\theta},\tilde{\mathbf{h}})$ and $\hat{\sigma}^2_i(\boldsymbol{\theta},\tilde{\mathbf{h}})$ are the mean and variance of the predictive distribution of the pressure i volume emulator, evaluated at $(\boldsymbol{\theta},\tilde{\mathbf{h}})$ where $\tilde{\mathbf{h}}$ is a low dimensional summary of the LV geometry that will be defined in Section 7.3.1. This distribution is represented by the branch leading to $\tilde{V}^e_{20}$ in Figure 7.2, where a dependence of the normalised volume on the material parameters, $\boldsymbol{\theta}$, is introduced through the underlying *cardio-mechanic model*.

We now have two different distributions for $\tilde{V}_{20}$: $\pi_{Kl}(\tilde{V}_{20})$, from fitting the Klotz function to the ex vivo data, and $\pi_e(\tilde{V}_{20})$ from the cardio-mechanic emulators. Intuitively, we want to reward cardio-mechanic parameters for which the agreement between these two distributions is high, and penalise parameters for which the agreement is poor. There are different ways to quantify the agreement between the distributions. In the present study, the so-called "product of experts" approach is used, which is widely used in the Machine Learning literature; see e.g. Calderhead et al. (2008) and Dondelinger et al. (2013). In the present case, it leads to a Gaussian convolution integral with a closed-form solution. Define:

$$\mathcal{F} \;=\; \int \pi_{Kl}(\tilde{V}_{20})\pi_e(\tilde{V}_{20})d\tilde{V}_{20}. \tag{7.13}$$

Inserting (7.10) and (7.11) into (7.13) and using standard results for Gaussian integrals, we obtain:

$$\mathcal{F}(\boldsymbol{\theta},\tilde{\mathbf{h}}) \;=\; \mathcal{N}\Big(\mu\Big|\frac{1}{\mu_\beta}\log\left(\frac{2}{3}\right)+1,\sigma^2_K+\sigma^2\Big) \tag{7.14}$$

where σ^2_K is the variance of the distribution in (7.10). This is represented by the dashed line connecting the two ex vivo reference volume terms in Figure 7.2. Note that while $\mathcal{F}$ interpreted as a function of μ is a normal distribution, $\mathcal{F}$ interpreted as a function of the cardio-mechanic parameters, $\boldsymbol{\theta}$, is a complex non-linear function where:

$$Z \;=\; \int \mathcal{F}(\boldsymbol{\theta},\tilde{\mathbf{h}})d\boldsymbol{\theta}, \tag{7.15}$$

does not have a closed-form solution. We obtain, for a given LV geometry representation

$\tilde{\mathbf{h}}$, the prior distribution of $\boldsymbol{\theta}$ via normalisation:

$$\pi(\boldsymbol{\theta}|\tilde{\mathbf{h}}) = \frac{\mathcal{F}(\boldsymbol{\theta}, \tilde{\mathbf{h}})}{Z}, \tag{7.16}$$

where MCMC sampling techniques have to be applied due to the intractability of Z. We have now obtained a prior distribution of the cardio-mechanic parameters $\boldsymbol{\theta}$ by fitting an empirical law for the pressure-volume dependence to ex vivo data from the literature, combining it with the cardio-mechanic model via the emulation of the LV cavity volumes at two ex vivo reference pressures – at 20 mmHg and 30 mmHg; see (7.12) – and taking explicitly the uncertainty predicted by the emulator into consideration.

7.2.3 Method B: the empirical Klotz prior

The first Klotz model was developed under the framework of the normalised Klotz curve, incorporating directly the relationship between normalised volume and pressure in (7.1) through a prior that measures the adherence of the predicted LV dynamics to the Klotz law. An alternative model, referred to as the *empirical Klotz prior*, will now be considered, where the level of fit with the Klotz curve is measured as a function of the measured LVV. Recall that the measured LVV is the cavity volume at end of diastole, measured from CMR scans and denoted as y_0^* in (7.4). The empirical Klotz prior formulation is more in line with the use of the Klotz curve in the literature, which usually depends on an in vivo volume measurement.

Consider a rearrangement of the original Klotz function from (7.1), where $\epsilon_P \sim \mathcal{N}(0, \omega(P))$ arises from an assumption of heteroscedastic Gaussian noise on the Klotz data (as presented in (7.2)):

$$\tilde{V}_P = \frac{\log\left(\frac{P}{\alpha}\right)}{\beta} + \epsilon_P \Rightarrow V_{30} = \frac{V_P - V_0}{\frac{\log\left(\frac{P}{\alpha}\right)}{\beta} + \epsilon_P} + V_0. \tag{7.17}$$

This equation expresses the volume at pressure 30 mmHg as a function of the volume at some pressure P. Choosing $P = 20$ mmHg and plugging in the volume prediction from an emulator will result in a prior the same as the one from Section 7.2.2. Instead, $P = 8$ mmHg will be used, which means plugging in the LVV from the measured dataset y_0^*. This assumes that EDP=8 mmHg, which was justified in Section 4.1.1.

From the above relationship, and with reference to (7.2), we have:

$$f(P, \boldsymbol{\phi}) = \frac{\log(\frac{P}{\alpha})}{\beta}. \tag{7.18}$$

Similar to the model from Section 7.2.2, ex vivo information can be introduced to the

empirical Klotz model by inferring the parameters α, β and $\omega(P)$ conditional on the data in Figure 7.1 (recall from (7.3) that estimating $\omega(P)$ involves inferring the parameters a, b and c). For estimation, the priors on the parameters were as follows: $\alpha, \beta \sim \mathrm{u}(0, 10)$ and $a, b, c \sim \mathrm{u}(-100, 100)$.

Using the posterior samples of α and β, the uncertainty in α and β can be propagated through to V_{30} via the Gauss error propagation model (see Appendix D.3) to give a distribution on V_{30}:

$$\pi_{Kl}(V_{30}) = N(V_{30}|\mu_{Kl}, \sigma^2_{Kl}). \tag{7.19}$$

with parameters:

$$\mu_{Kl} = \frac{V_8 - V_0}{\frac{\log\left(\frac{8}{\alpha}\right)}{\beta} + \epsilon_8} + V_0\Bigg|_{\alpha=\mu_\alpha,\beta=\mu_\beta,\epsilon_8=0} = \frac{y_0^* - V_0}{\frac{\log\left(\frac{8}{\alpha}\right)}{\beta} + \epsilon_8} + V_0\Bigg|_{\alpha=\mu_\alpha,\beta=\mu_\beta,\epsilon_8=0}, \tag{7.20}$$

$$\sigma^2_{Kl} = \left[\frac{\partial V_{30}}{\partial \alpha}\right]^2_{\mu_\alpha,\mu_\beta} \sigma^2_\alpha + \left[\frac{\partial V_{30}}{\partial \beta}\right]^2_{\mu_\alpha,\mu_\beta} \sigma^2_\beta + \left[\frac{\partial V_{30}}{\partial \epsilon_8}\right]^2_{\mu_\alpha,\mu_\beta} \omega(8)^2 \tag{7.21}$$

$$= \left[\frac{\mu_\beta(y_0^* - V_0)}{\mu_\alpha(\log(\frac{8}{\mu_\alpha}))^2}\right]^2 \sigma^2_\alpha + \left[\frac{y_0^* - V_0}{\log(\frac{8}{\mu_\alpha})}\right]^2 \sigma^2_\beta - \left[\frac{\mu^2_\beta(y_0^* - V_0)}{\log(\frac{8}{\mu_\alpha})^2}\right]^2 \omega(8)^2 \tag{7.22}$$

where $P = 8\,\mathrm{mmHg}$ has been used. Substituting $\omega(P)^2 = 0$ gives the zero Klotz model mismatch case where only the uncertainty in α and β contributes towards the Klotz prior evaluation. The final result—displayed in the left branch of Figure 7.3—is a distribution on V_{30} with dependence on the *Klotz curve model* and the measured LVV y_0^*. In the clinical setting, we must also account for noise in the y_0^* measurement. Assuming that y_0^* is the mean of a distribution on measured LVV values with standard deviation $\sigma_8 = 5\,\mathrm{ml}$ (for justification of this value see Table E.1), the uncertainty on the volume measurement is propagated through the model via the delta method and σ^2_{Kl} includes the extra term:

$$\left[\frac{\partial V_{30}}{\partial V_8}\right]^2_{\mu_\alpha,\mu_\beta} \sigma^2_8 = \left[\frac{\mu_\beta}{\log\frac{8}{\mu_\alpha}}\right]^2 \sigma^2_8 \tag{7.23}$$

While the distribution in (7.19) was derived from the Klotz curve, we can train a surrogate model on simulated volume data at pressure 30 mmHg and obtain a distribution on V_{30} obtained from the predictive distribution of the Gaussian process:

$$\pi_e(V_{30}) \;=\; \mathcal{N}(.|\mu_{30}(\boldsymbol{\theta}, \tilde{\mathbf{h}}), \sigma^2_{30}(\boldsymbol{\theta}, \tilde{\mathbf{h}})), \tag{7.24}$$

which, unlike $\pi_{Kl}(\cdot)$, depends explicitly on the material parameters. This distribution is represented by the right branch of Figure 7.3 (leading to V^e_{30}), which shows its dependence on the material parameters and the *cardio mechanical model*. We now have two distribu-

tions on V_{30}: $\pi_{Kl}(V_{30})$ from the Klotz function in (7.19) and the other, $\pi_e(V_{30})$, from the V_{30} emulator with mean $\mu_{30}(\boldsymbol{\theta}, \tilde{\mathbf{h}})$ and variance $\sigma^2_{30}(\boldsymbol{\theta}, \tilde{\mathbf{h}})$. Considering a statistic similar to that from (7.13):

$$\mathcal{F}_{EK} = \int \pi_{Kl}(V_{30})\pi_e(V_{30})dV_{30}, \tag{7.25}$$

we obtain:

$$\mathcal{F}_{EK}(\boldsymbol{\theta}, \tilde{\mathbf{h}}) = N\big(\mu_{30}(\boldsymbol{\theta}, \tilde{\mathbf{h}})|\mu_{Kl}, \sigma^2_{Kl} + \sigma^2_{30}(\boldsymbol{\theta}, \tilde{\mathbf{h}})\big). \tag{7.26}$$

Letting $Z = \int \mathcal{F}_{EK}(\boldsymbol{\theta}, \tilde{\mathbf{h}})d\boldsymbol{\theta}$, we retrieve a prior distribution:

$$\pi_{EK}(\boldsymbol{\theta}|\tilde{\mathbf{h}}) = \frac{\mathcal{F}_{EK}(\theta, \tilde{\mathbf{h}})}{Z}, \tag{7.27}$$

where the intractability of the denominator, Z, does not affect the inference if we use MCMC.

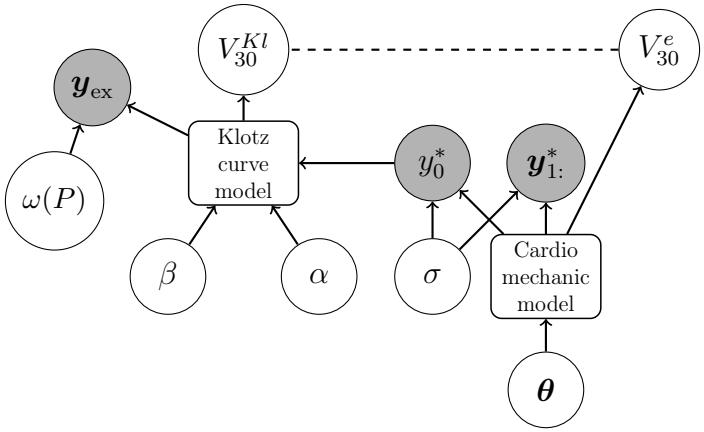


Figure 7.3: Visualising the model with the empirical Klotz prior. The central branch, leading to y_0^* and $\boldsymbol{y}_{1:}^*$, provides the in vivo data model where we match the simulator predictions with the in vivo measured volume (y_0^*) and strains ($\boldsymbol{y}_{1:}^*$), which are assumed to be obtained at EDP 8 mmHg. Based on the cardiac model, an emulator has been trained to predict the volume at pressure 30 mmHg and the predictive distribution provides a distribution on the volume, indicated by V_{30}^e. Inferring the Klotz relationship from the ex vivo data, $\boldsymbol{y}_{\text{ex}}$, and estimating V_{30} from the in vivo volume using (7.17) gives another distribution on V_{30}, denoted by V_{30}^{Kl} in the diagram. Using a product of experts approach, we assess the similarity of the two different distributions, indicated by the dashed line. Although the formulation of this model is different from the nonempirical Klotz prior, including a dependence on both α and β in a rearranged expression for predicting V_{30} from the in vivo volume (compare (7.17) with (7.8)), the important difference with the model in Figure 7.2 is the introduction of a connection between the measured volume and the Klotz curve model, represented by the horizontal arrow exiting the node y_0^*.

To summarise: the important difference between the empirical Klotz curve prior and

the nonempirical Klotz prior is the dependence of the former on the in vivo measured LVV. The two priors are described in Table 7.1.

Table 7.1: Description of the two different Klotz curve prior models.

Prior	Description
nonempirical Klotz prior	This uses the normalised form of the Klotz function from (7.1), not depending on any in vivo end of diastole measurements.
Empirical Klotz prior	This uses a rearrangement of (7.1) where we evaluate the Klotz predicted V_{30} based on (7.17) using the measured LVV y_0^*.

7.3 Implementation

This section discusses the implementation of the Klotz prior, beginning with the emulators used to speed-up inference and then the generation of test data to assess the utility of the priors. The test data will be used in several simulation studies in Section 7.4.

7.3.1 Construction of the in clinic emulators

Efficient inference with the Klotz prior and in vivo likelihood requires two “in-clinic" emulators that can generalise to unseen LV geometries. Both of these are described in Table 7.2. The multi-geometry emulator emulates the cardio-mechanical model at in vivo EDP (8 mmHg), taking as input an approximation of the LV geometry at early diastole and the four material parameters. This emulator is identical to the one explored in Chapter 6, which required a 5 PC representation of the LV geometry and the four parameter values of the theta parameterisation of the H-O model, found in (4.2). For training the emulator, 16,000 training points were generated and a neural network was optimised to be used as the emulator. Further details about this emulator can be found in Chapter 6. For the Klotz prior evaluation, we require generalised Klotz emulators that are used to predict V_{20} and V_{30} for different LV geometries. The input space for these models is 6 dimensional: 4 dimensions for material properties, one for cavity volume and one for wall volume at early diastole. This means that $\tilde{\mathbf{h}}$ is a two-dimensional vector containing the cavity and wall volumes at early diastole. To train the emulator, 9000 simulations were used, on which the sparse GP of Titsias (2009) was trained. This GP approximation method was described in Section 3.4.4. For both sparse GP emulators, 200 inducement points were used.

Table 7.2: Description of the different emulators used in this study.

Emulator	Description	Use
Generalised Klotz emulator	Emulator constructed over an input space of 4 material properties, early diastolic cavity volume and early diastolic wall volume.	In clinic prediction of volumes at ex vivo pressure ranges for implementation of Klotz priors.
Multi geometry emulator	The emulator of the cardio mechanic model, generalised over 5 dimensional LV geometry representations obtained with PCA.	In clinic efficient approximation of the simulator model

7.3.2 Test data generation

Exploring the effects of LV geometry approximation

Using six different LV geometries that were excluded from the original PCA analysis during emulator training, the first simulation study will assess the ability of the empirical and nonempirical Klotz priors to reduce the error incurred by the LV geometry approximation in the emulator, as well as the intrinsic identifiability problems in the model. To do so, we require the gold standard parameters so synthetic data were generated that are consistent with the Klotz priors to assess the improvement in inference accuracy that can be obtained under the assumption that the Klotz model is physiologically realistic (which is based on the evidence provided in the literature cited in Section 7.1). In the generation of these test data, θ_1 and θ_4 were fixed at population based averages (this corresponds to fixing them equal to 1). For each of the six LV geometries, five different parameter configurations were used, giving 30 test cases in total. Further details on the sampling procedure for test data generation can be found in Appendix D.5.

Measured data study

As a second study, the task of inferring the material properties from measured CMR scans is considered. In this case, we do not possess the ground truth parameters. Instead, the effect the Klotz curve priors have on cardio-mechanic parameter estimation will be assessed, including the physiological plausibility of this effect. For five different healthy volunteers, four positions of short-axis cine images from the base to the middle ventricle were chosen for strain measurements so that the position of the measured strain matched the position of the model prediction. A B-spline deformable registration approach was used for estimating segmental averaged circumferential strains at each location (Mangion et al., 2016a). Following the clinical convention, 6 peak diastolic strains are calculated at four different short axis slices from the base- to mid-LV. The end-diastolic LV cavity volume is calculated from short-axis cine images at the end of diastole after manually delineating

the endocardial boundaries. The LV cavity volume at early diastole was calculated from the reconstructed LV geometry using the MATLAB function `convhull`. In total there were 24 peak diastolic strains and 2 LV cavity volumes (one at early diastole, and the LVV).

7.3.3 Simulations

Sampling of the material properties is carried out using NUTS with dual averaging step size adaptation (Hoffman and Gelman, 2014; Betancourt, 2017). The likelihood function is evaluated using the statistical emulator so no forward simulations are required during inference. A burnin phase of length at least 5000 was used, continuing until a potential scale reduction factor (Gelman and Rubin, 1992) of 1.1 was reached and then samples were taken until an effective sample size (see Section 11.5 of Gelman et al. (2013)) of 100 was achieved. The burnin was then discarded and the sample was used for analysis of the different methods.

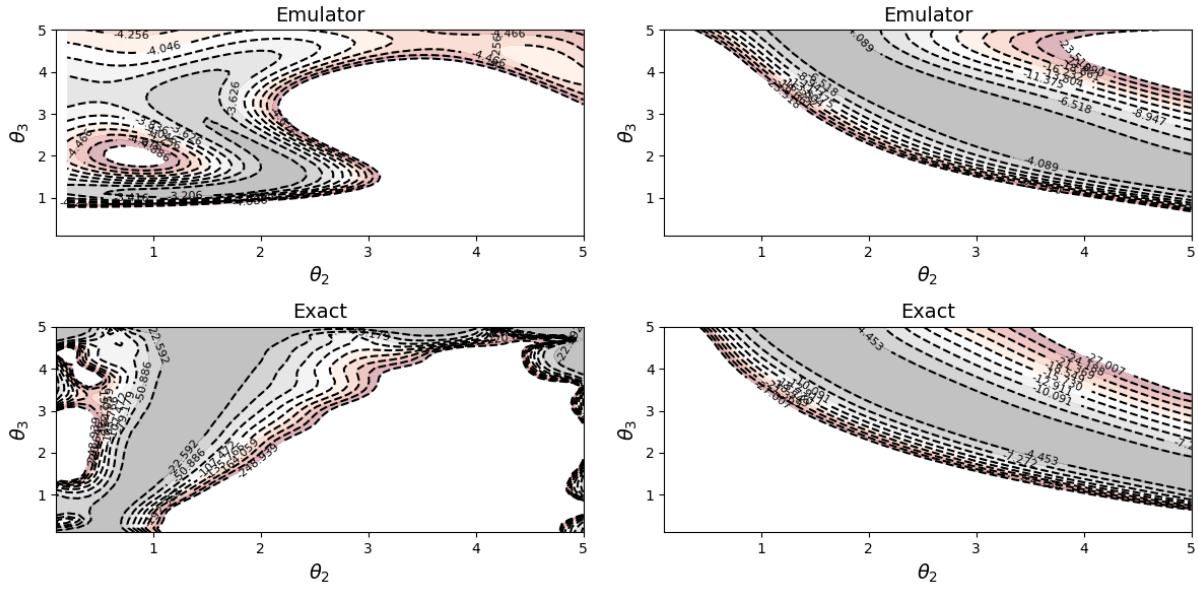
7.4 Results

7.4.1 Comparing low fidelity and high fidelity Klotz priors

In Section 7.3.1, multiple emulators were discussed and summarised in Table 7.2. As discussed above, the multi geometry emulator was explored extensively in Chapter 6, where it was found to introduce errors compared to benchmark simulations with a full LV geometry representation. A similar assessment of the generalised Klotz emulators is required because these also rely on a summary of the LV geometry at early diastole. Of interest is the use of these emulators in the Klotz prior models, which is compared to evaluation of the prior using the high fidelity emulators that were developed for the generation of the synthetic test data. As discussed in Appendix D.5, these high fidelity emulators were trained on data in the 2 dimensional $\theta_2-\theta_3$ space, with simulations obtained using the full LV geometry.

Figure 7.4 shows the comparison of the Klotz prior evaluations. For the nonempirical prior, this assessment compares evaluations of (7.14) and for the empirical prior (7.26) was used. The prior surfaces are only compared in the two dimensional $\theta_2-\theta_3$ space, the latter of which is the parameter we expect to be affected most by the prior information. In the nonempirical Klotz prior, which is on the left of Figure 7.4, there is quite a large change in the prior surface. For the empirical Klotz prior, which is on the right of Figure 7.4, the prior does not change substantially when using the generalised Klotz emulator compared with a high fidelity emulator. The difference in accuracy of the prior evaluation when using generalised Klotz emulators is almost certainly related to the fact that the empirical Klotz

prior depends on only one emulator, while the nonempirical prior depends on emulators for both V_{20} and V_{30}.



(a) (b)

Figure 7.4: Comparing the evaluation of the empirical and nonempirical Klotz priors when evaluated using the generalised Klotz emulators (labelled *Emulator*) that rely on LV geometry approximations with high fidelity emulators (labelled *exact*) that do not rely on an LV geometry approximation. On the left is the nonempirical Klotz prior (evaluated using (7.14)) and on the right is the empirical Klotz prior (compared by evaluating (7.26)). In each plot, the horizontal and vertical axes show the entire physiological range for the parameters.

7.4.2 Initial prior exploration with high fidelity emulators

Building on Section 7.4.1, this section considers sampling in 2D space, with priors evaluated using the high fidelity Klotz emulators. The top row of Figure 7.5 provides visualisations of the prior surfaces, obtained by evaluation of (7.14) for the nonempirical Klotz prior and (7.26) for the empirical Klotz prior. In the bottom row, the evaluation of the log likelihood from (7.7) over the parameter space is plotted. For the estimation of the cardio-mechanic parameters the data extracted from the CMR scans are informative to different degrees, generally allowing for accurate estimation of θ_2 but not θ_3[2]. In light of this, the contour plot in the top left panel of Figure 5 suggests that with the nonempirical

[2]Although this general trend is observed, the ability to infer the material parameters changes depending on their ground truth value, as we saw in Section 6.3.3. For instance, parameters describing a soft material are easier to learn than those of a stiffer material as a result of the faster varying simulator outputs in the softer region of space.

Klotz prior we will achieve little improvement in our inference, in that on estimating θ_2 from the CMR data, the conditional probability of θ_3 conditional on θ_2 remains diffuse.

The problems witnessed in the nonempirical Klotz prior are largely alleviated by consideration of the empirical Klotz prior on the right of Figure 7.5. Here, the introduction of the measured LVV means that only V_{30} is predicted using an emulator (see left branch of Figure 7.3). As a result, the ridge in the prior distribution is re-oriented so that conditioning on θ_2 (which can usually be easily estimated from the data as shown in the bottom right panel of this figure) substantially reduces the uncertainty for θ_3. The most likely reason for this improvement is the fact that one of the two LV volume emulators required for the nonempirical Klotz prior has become obsolete as a consequence of explicitly including the measured LVV. In particular, as we increase pressure, the contours gradually align more with the θ_2 axis. This is precisely what we hope will happen. However, by standardising V_{20} with V_{30} we partially negate this realignment and end up with a density that is often aligned with the θ_3 direction. This effect is visualised in Figure 7.6.

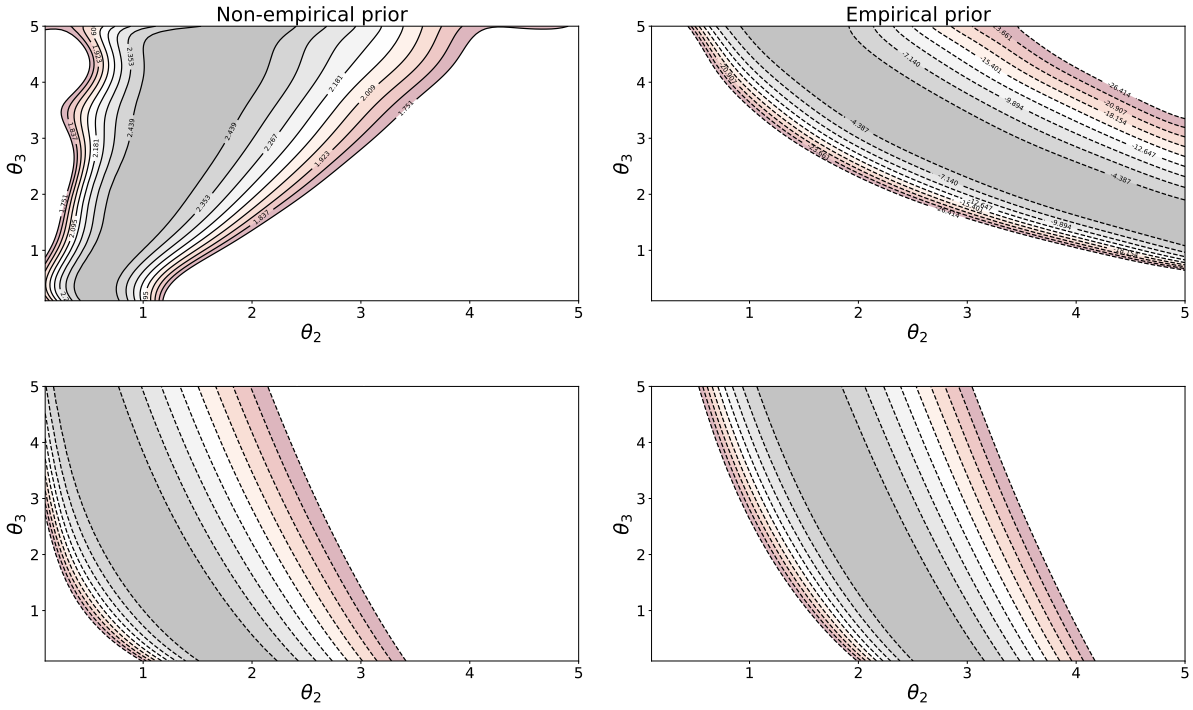


Figure 7.5: Comparing the shapes of the nonempirical and empirical Klotz priors, from (7.14) and (7.26) respectively, (top row) and the RSS functions (bottom row) for the in vivo data: nonempirical (evaluation of (7.14)) on the left, and empirical (evaluation of (7.26)) on the right. The orientation of the mode in the empirical Klotz prior (top right) , along with identifiability of θ_2 based on the in vivo data, should allow for a reduction in the uncertainty of the parameters in the model. In the case of the nonempirical Klotz prior (top left) , the high density ridge is aligned with the θ_3 direction. In each plot, the horizontal and vertical axes show the entire physiological range for the parameters, which is $[0.1, 5]$.

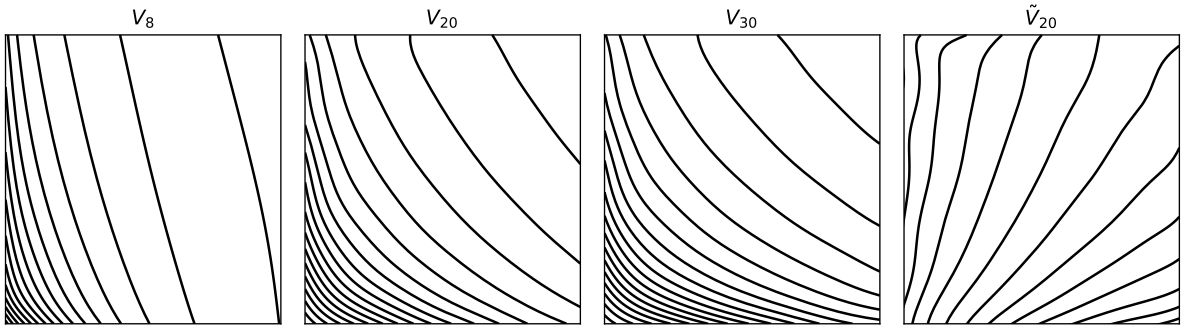


Figure 7.6: Change in the shape of the LVV simulated function over parameter space as the EDP varies. The horizontal axis is $\boldsymbol{\theta}_2$ and the vertical axis is $\boldsymbol{\theta}_3$. Increasing the EDP from 8 to 30 mmHg aligns the contours more with the θ_2 axis. The final plot shows that by normalising the volume at EDP=20 mmHg with the volume at EDP=30 mmHg we end up with a completely new alignment of the surface that is often aligned with the θ_3 direction.

Combining the priors from Figure 7.5 with a likelihood function obtained using the multi geometry emulator, we can sample from the two dimensional posterior of θ_2 and θ_3 while fixing θ_1 and θ_4 to their true values. As a benchmark, this is compared with posterior samples when adopting a non-informative $\mathrm{u}(0.1, 5)$ prior in each dimension. The posterior density, when using the nonempirical Klotz prior, is shown in Figure 7.7a. For data generated in the way described in Section 7.3.2, the nonempirical Klotz prior appears to reduce the uncertainty in the parameters compared with the non-informative prior in Figure 7.7c but some bias is present in the posterior samples. Using the empirical Klotz prior, with results shown in Figure 7.7b, the uncertainty in θ_3 is reduced compared with the results of the non-informative prior in Figure 7.7d. In this case, bias is not introduced to the posterior samples. This initial study demonstrates the potential the empirical Klotz prior has for improving cardio-mechanic parameter inference.

7.4.3 Assessing the Klotz models with synthetic data

A key question of interest is the following: can the introduction of ex vivo information help to alleviate any error that is introduced to the inference problem by the LV geometry approximation while reducing uncertainty in the parameters? With test data generated as described in Section 7.3.2, this question will now be explored using the generalised Klotz emulators to efficiently evaluate the Klotz prior models and the multi geometry emulator to evaluate the likelihood function, both described in Table 7.2. The results will be assessed both in parameter space and in stress-stretch space.

Beginning with parameter space, the results are evaluated using the squared error,

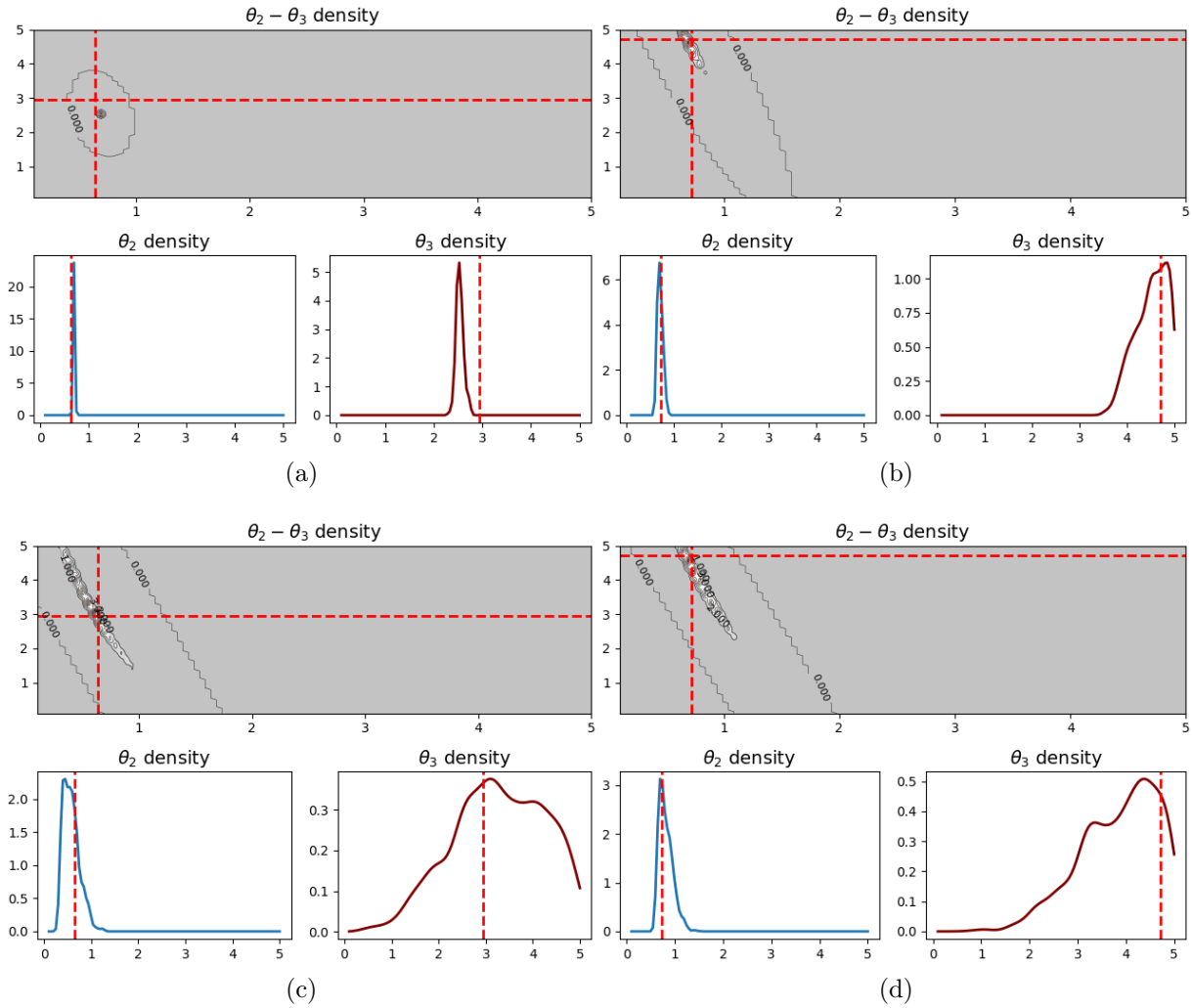


Figure 7.7: Kernel density estimates obtained using samples from the posterior of the material properties when ex vivo information is introduced in (a) the nonempirical Klotz prior (b) the empirical Klotz prior. The red dashed lines show the true parameter values. Panels (c) and (d) show posterior density plots obtained from samples when assuming a uniform prior on each of the parameters. In two dimensional plots, θ_2 is on the x-axis and θ_3 is on the y-axis, with the entire physiological range plotted. For the 1D density plots, the horizontal axis shows the entire physiological range.

which is defined as follows:

$$\mathrm{SE}_i = (\hat{\theta}_i - \tilde{\theta}_i)^2. \tag{7.28}$$

In this error function, $\hat{\theta}_i$ is an estimate of material parameter i and $\tilde{\theta}_i$ is the ground truth value of this parameter. For each of the six LV geometries, synthetic data have been generated from five different material parameter configurations. MCMC is then used to obtain a sample from the posterior distribution of the material parameters for each of these 30 test cases, providing a distribution of squared errors for each of the four

material parameters. Given that the performance is measured in absolute terms via the squared error, the results for all these test cases are combined, separately for each LV geometry, into one summary in the boxes of Figures 7.8 to 7.11. That is to say that, in total, each box provides the distribution of SEs obtained from 5 (the number of parameter configurations per LV geometry used for synthetic data generation) $\times$ 100 samples of the material parameters, where each subset of size 100 is the posterior distribution of squared errors corresponding to different test parameter configurations. For LV geometry i, these distributions are labelled as Gi, with separate plots for each of the four material parameters.

The nonempirical Klotz model

Figures 7.8 and 7.9 display the squared error in parameter samples when adopting the nonempirical Klotz prior. Considering first Figure 7.8, which shows the results without including model mismatch, the estimation accuracy of θ_2 and θ_3 is improved very slightly by introducing the Klotz information. The distributions of θ_1 and θ_4 are unchanged, the latter of which is consistent with our expectations based on the physical interpretation of this parameter, which governs the shear response of the material. Neither the circumferential strains nor the volume captures this property. These results are mirrored in Figure 7.9, where the Klotz model mismatch term was included in the prior, albeit with a smaller difference between inference with the Klotz prior and inference without. It is worth noting that the lack of improvement observed with the nonempirical Klotz prior could be partially attributed to errors observed in the approximation of the LVV, which was shown to affect the prior evaluation in Section 7.4.1.

The empirical Klotz model

The results obtained in the empirical Klotz prior analysis are shown in Figures 7.10 and 7.11. The former shows the performance when not accounting for the mismatch in the Klotz model (ϵ_P from (7.17)), compared against results with a non-informative prior on the four parameters. Quite clearly the estimation of θ_2 and θ_3 is improved while sampling of θ_4 is largely unchanged. Information from the in vivo data is already enough to identify θ_1, so we observe little difference in this parameter. In Figure 7.11, we observe less improvement upon inclusion of the Klotz model mismatch term (recall that this stems from the inclusion of the heteroscedastic noise term in the Klotz prior expression, as discussed below (7.22)). This is exactly as expected given the weaker form of the prior distribution. The derivation of the empirical Klotz prior used the Gauss error propagation model. Instead of this, the multivariate delta method could have been used, which would include the covariance between the two Klotz parameters in the variance calculation in (7.22). A comparison between the two is made in Appendix D.6 where we see that there

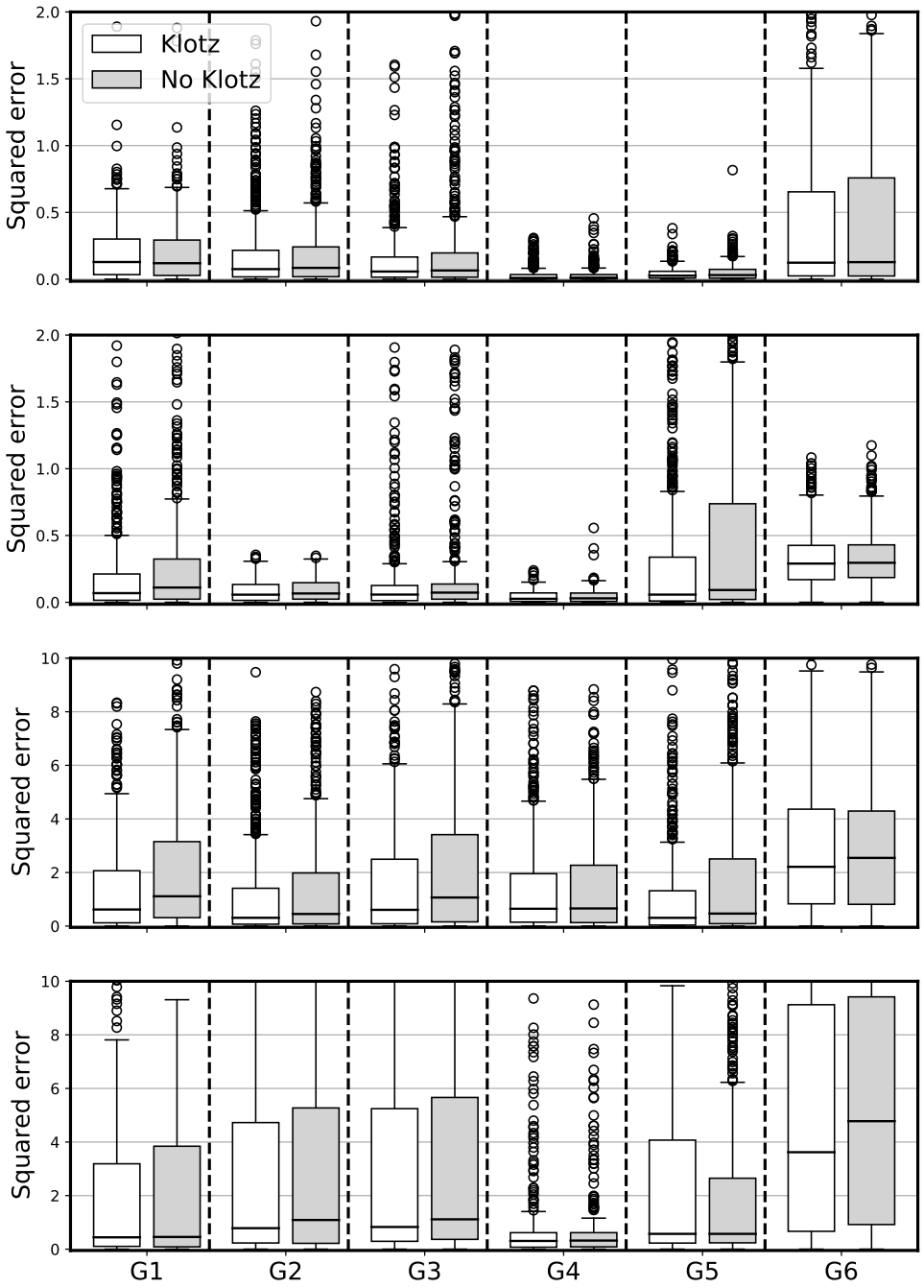


Figure 7.8: Squared error in samples from the posterior distribution of the material parameters when using the nonempirical Klotz prior without the Klotz model mismatch term. Row i gives the results for θ_i, with the boxplots arranged into the 6 different LV geometries, denoted by Gi for the ith geometry, with samples obtained while adopting the nonempirical Klotz prior and a non-informative prior shown side by side in each pair. For each LV we have 5 parameter configurations and the squared errors from the 100 posterior samples for each configuration are combined into one distribution of 500 squared errors. The physiological range for each parameter is $[0.1, 5]$

is only a small difference in the marginal posterior distributions for each test case.

In light of the superior results with the empirical Klotz prior, the effect of this prior

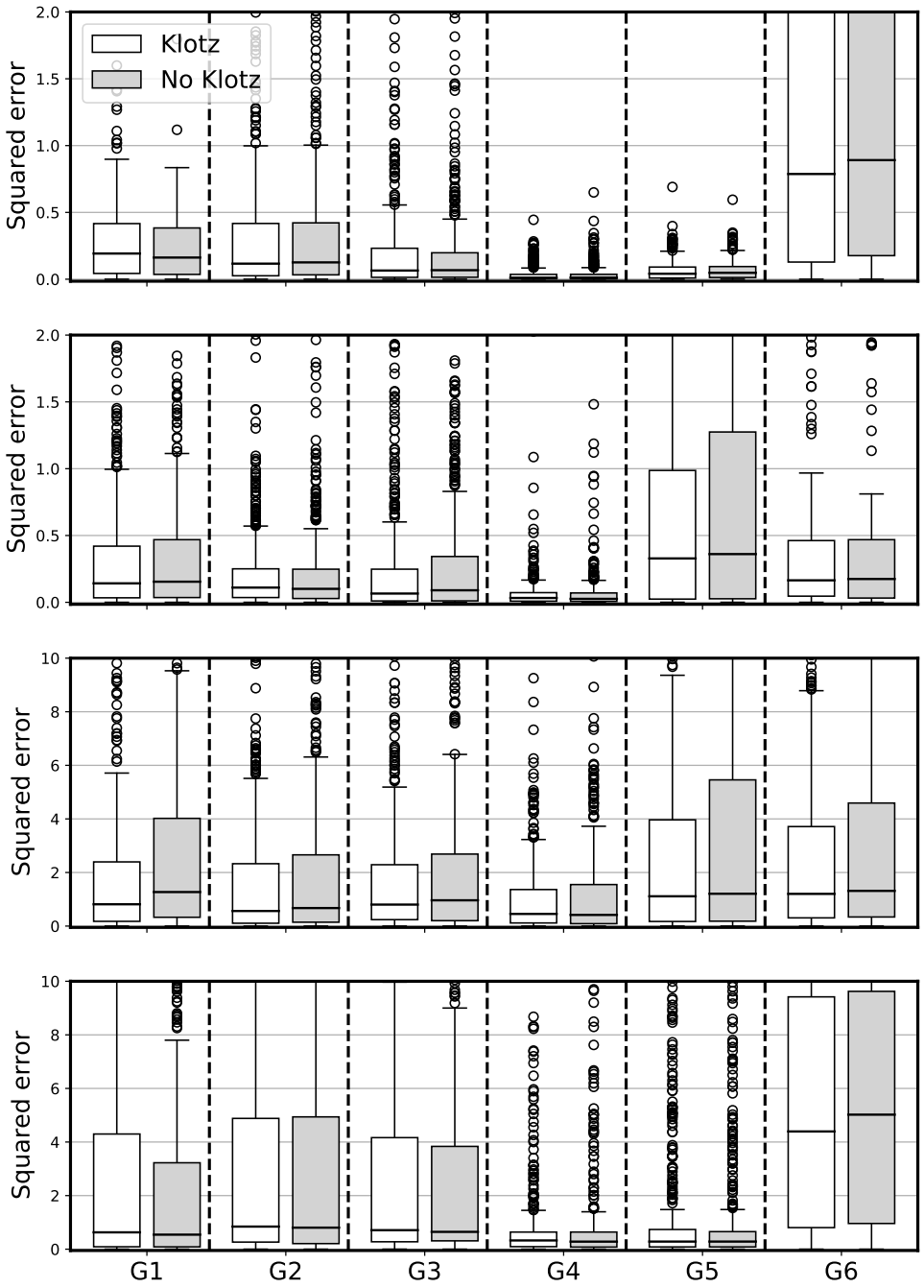


Figure 7.9: Squared error in samples from the posterior distribution of the material parameters when using the nonempirical Klotz prior with the Klotz model mismatch term. Row i gives the results for θ_i, with the boxplots arranged into the 6 different LV geometries, denoted by Gi for the ith geometry, with samples obtained while adopting the nonempirical Klotz prior and a non-informative prior shown side by side in each pair. For each LV we have 5 parameter configurations and the squared errors from the 100 posterior samples for each configuration are combined into one distribution of 500 squared errors. The physiological range for each parameter is $[0.1, 5]$

model is now assessed in more detail. For each test case, we have a sample from the 4 dimensional posterior distribution of the model parameters (the four θ parameters).

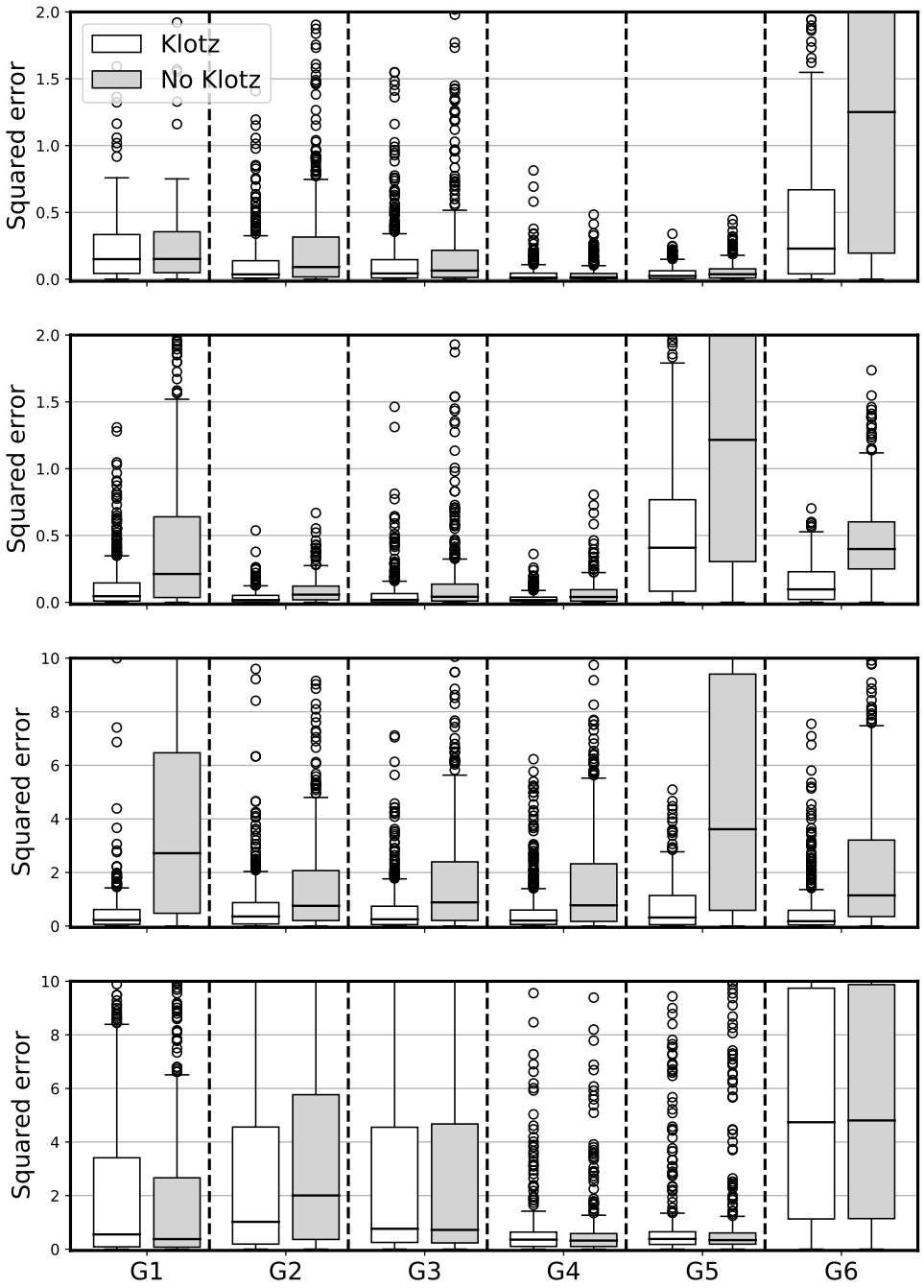


Figure 7.10: Squared error in samples from the posterior distribution of the material parameters when using the empirical Klotz prior without the Klotz model mismatch term. Row i gives the results for θ_i, with the boxplots arranged into the 6 different LV geometries, denoted by Gi for the ith geometry, with samples obtained while adopting the nonempirical Klotz prior and a non-informative prior shown side by side in each pair. For each LV we have 5 parameter configurations and the squared errors from the 100 posterior samples for each configuration are combined into one distribution of 500 squared errors. The physiological range for each parameter is $[0.1, 5]$

Physiologically, these parameters characterise the tissue stiffness, which is more easily viewed in the stress-stretch space, i.e. by virtually stretching a myocardial strip along

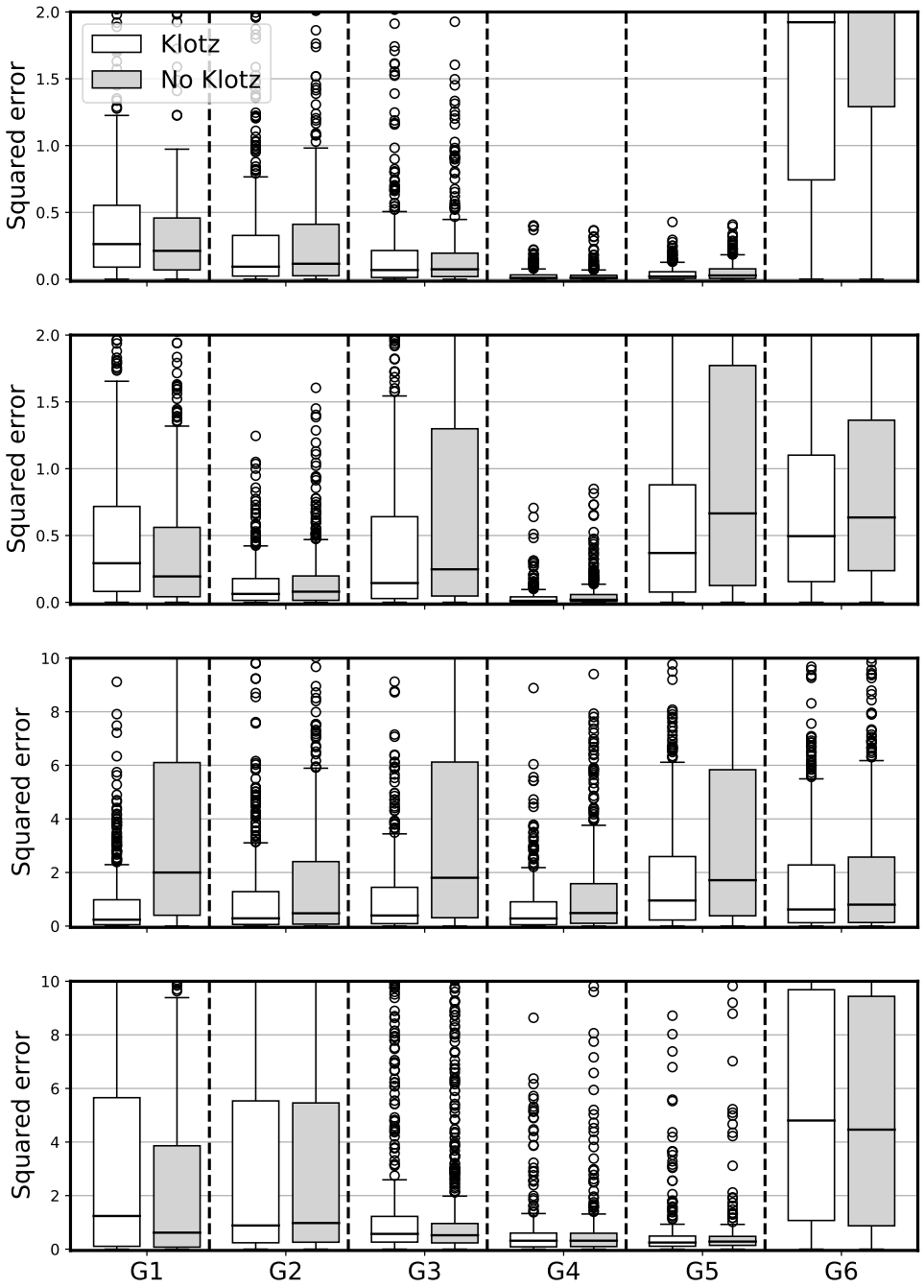


Figure 7.11: Squared error in samples from the posterior distribution of the material parameters when using the empirical Klotz prior with the Klotz model mismatch term. Row i gives the results for θ_i, with the boxplots arranged into the 6 different LV geometries, denoted by Gi for the ith geometry, with samples obtained while adopting the nonempirical Klotz prior and a non-informative prior shown side by side in each pair. For each LV we have 5 parameter configurations and the squared errors from the 100 posterior samples for each configuration are combined into one distribution of 500 squared errors. The physiological range for each parameter is $[0.1, 5]$

a specific direction uni-axially (Gao et al., 2015). This provides a metric with which to assess the health of the cardiac tissue that could be used in the clinical setting. Note

that stress here means the Cauchy stress, which is the actual stress that the myocardium experiences.

From the parameter samples of each of the 30 test cases, a distribution of stress–stretch curves was obtained, with one curve for each sample from the posterior distribution of the material parameters. Similarly, knowledge of the ground truth parameters provides a ground-truth stress–stretch curve for each test case. Measuring the distance between each sampled curve and the ground truth curve at each stretch level provides a distribution of distances that measure the accuracy with which the sampled material parameters can replicate the tissue behaviour of the ground truth parameters at a given stretch. Plotted in Figure 7.12 is the proportion of the 30 test cases (recall we have 6 LV geometries each with 5 test parameter configurations) where the distribution of curves at a given stretch, obtained from material parameters sampled under an assumption of the empirical Klotz prior, is significantly closer than the distribution of curves obtained from material parameters sampled under an assumption of a uniform prior, where closeness is measured by distance to the ground truth stress–stretch curve. A significant difference between the distributions of distances, which is assessed via the one-sided Mann-Whitney U test at a 5% level (Mann and Whitney, 1947), signifies that the stress–stretch curves obtained from the samples under an assumption of the empirical Klotz prior are significantly closer to the ground-truth stress–stretch curve than the stress–stretch curves obtained under an assumption of a uniform prior on the material parameters (note that a lack of significance does not imply a deterioration). This same procedure is repeated—checking for significant improvements in the stress–stretch curve estimation—to check for improvement when using the uniform prior with the proportion of significant results also plotted in Figure 7.12. At the majority of stretch locations we observe an improvement upon introduction of the empirical Klotz prior, but there is a dip in the proportion of significant improvements around stretch 1.12 for the myocyte direction and 1.2 for the sheet direction. Looking at the number of significant improvements when adopting the uniform prior we see a reverse of this pattern, but the proportion of significant improvements with the uniform prior is low, reaching a peak of around 0.2 in the no model mismatch case. The corresponding stretch, where proportion of improvements with the empirical Klotz prior is low, is at a level where behaviour of the myocardium can be inferred from the in vivo data alone. Moreover, the fact that we do see a few cases where the inference is better with a uniform prior is not surprising since the effect of the Klotz prior distribution is to act as a regulariser, potentially introducing a small bias to the parameter estimates at the in vivo location while allowing for improved estimation in the ex vivo domain. This will be further discussed in the discussion section.

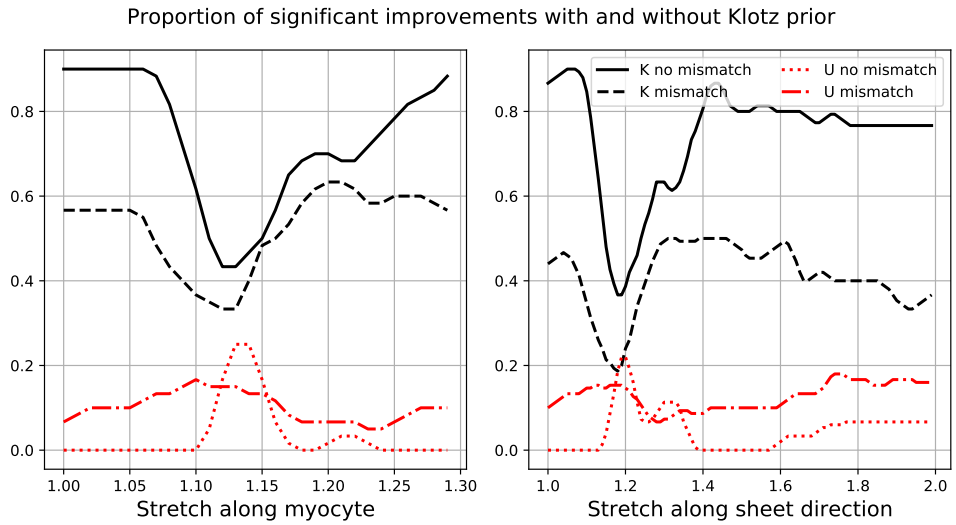


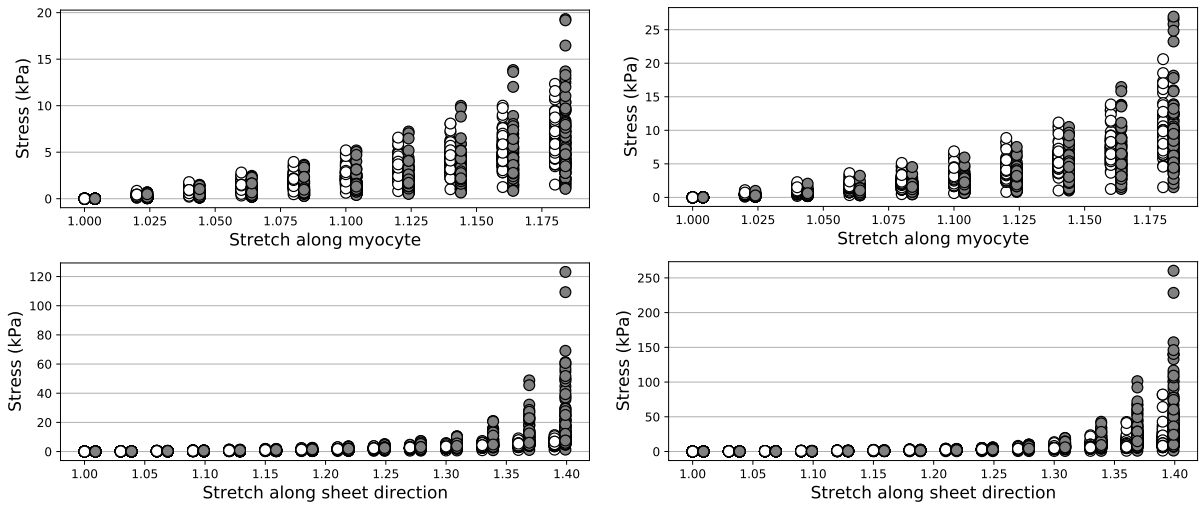
Figure 7.12: Using stress–stretch curves to compare the performance of the empirical Klotz prior, with and without model mismatch as discussed below (7.22), and a uniform prior. Two different cases are considered: 1. if the empirical Klotz prior without model mismatch improves upon a unif$(0.1, 5)$ prior (K *no mismatch*) and 2. if the empirical Klotz prior with model mismatch improves upon a unif$(0.1, 5)$ prior (K *mismatch*). The plots show the proportion of test cases where we find a significant improvement in our inference when using the empirical Klotz prior, assessed by comparing the distribution of distances of the sample of stress–stretch curves obtained under an assumption of the different priors. To complete the study, improvement was also tested in the other directions: 1. if a unif$(0.1, 5)$ prior improves upon the empirical Klotz prior without model mismatch (U *no mismatch*) and 2. if a unif$(0.1, 5)$ prior improves upon the empirical Klotz prior with model mismatch (U *mismatch*), assessing improvement as before.

7.4.4 Inferring parameters from real data

Finally, the task of inferring material properties from real measured data of five healthy volunteers is considered. The data measuring procedure was outlined in Section 7.3.2. This time, the strains and LV volume are subject to measurement error, inevitably affecting the parameter estimation both with and without the Klotz prior. Results for two of the examples are shown in Figure 7.13 (one subject in (a) and one in (b)). In Appendix D.7, the full samples of curves for all five real data cases are provided. Again, only the empirical Klotz prior is considered and, given that real data are used, we do not possess ground truth material properties so inference will only be assessed in stress–stretch space. In both test cases shown in Figure 7.13, which show inference with the introduction of the empirical Klotz prior without Klotz model mismatch, the stress in the upper stretch region is reduced, signifying a softer material than is described when we do not introduce the Klotz curve.

An important difference between the results with and without the Klotz prior is that without the Klotz prior, mismatch between the measured strains and volumes can lead

us to sample material properties at which the model predicted V_{30} is smaller than the observed V_8, violating the "Frank–Starling" law of the heart (Moss and Fitzsimons, 2002). Essentially, the problem is that the measured circumferential strains imply material parameters that are far stiffer than those suggested by the LVV measurement. More detail on this mismatch will be provided in Chapter 9. Introducing the Klotz information helps to alleviate this problem by balancing the weights between the measured strains and volumes. Thus the material property prediction with Klotz information will be physiologically more realistic within a wide range of LV pressures covering from in vivo to ex vivo.



(a) (b)

Figure 7.13: Stress–stretch samples for two real test cases (left and right). From a sample of material parameters, obtained from the posterior distribution of the material parameters for each real test case, we obtain a sample of stress–stretch curves. These stress values are then plotted as points at the different stretch locations along the myocyte (top) and sheet direction (bottom). Inference results are compared with prior knowledge expressed through the uniform prior on material parameters (grey dots) and the empirical Klotz prior on the material parameters (white dots). The physiological range for the stress is less than 100 kPa.

7.5 Discussion and conclusion

7.5.1 Discussion of results

The work in this chapter has adopted an approximation to the in vivo likelihood function in the form of a statistical emulator. This provides a computationally efficient approximation to the true model, but was shown in Chapter 6 to introduce some error as a result of the LV geometry approximation and the difficult emulation problem in nine dimensions. The work in the current chapter aimed to mitigate any errors through the introduction of ex vivo

information to the parameter inference scheme in the form of Klotz prior distributions. The results of the study showed that the introduction of this extra information does allow us to reduce the posterior uncertainty in θ_3, which characterises myofibre stiffening under higher pressure. The idea of introducing information from the Klotz curve to assist with inferring this behaviour is not novel, but the formulation of this information as a prior distribution has never been presented before. Importantly, this allowed for the implementation of Bayesian inference schemes that account for the uncertainty in the material parameters.

The first Klotz paradigm presented in this chapter considered the normalised form of the Klotz curve in a prior named the nonempirical Klotz prior. The presence of both V_{20} and V_{30} in the Klotz function in (7.8) via (7.1) is potentially the cause of two issues in the prior. The first of these is a problem of identifiability, which is a fundamental property of the prior, and not a result of the particular implementation in this study. As shown in Figure 7.6, the normalisation of V_{20} by V_{30} gives a function such that material parameter configurations that give very different V_{20} and V_{30} predictions, and therefore represent different physiological behaviours, are evaluated equally. While the nonempirical prior was shown to massively reduce the uncertainty in the 2D problem in Figure 7.5, it was found to give very little benefit in four dimensions. The problems with the nonempirical prior motivated the development of the empirical Klotz prior, where V_{30} is expressed as a function of the empirical LVV measurement at pressure 8 mmHg. The results showed that, in the context of inference with a statistical emulator, the inclusion of an in vivo LVV measurement in the Klotz prior proves to be important for achieving a performance improvement in the inference scheme. The second issue with the nonempirical prior relates to the way that it was implemented for the work in this chapter. By summarising the LV geometry with the cavity volume and wall volume, an error was introduced to both Klotz prior evaluations, and this was more substantial in the case of the nonempirical prior, as shown in Figure 7.4. The reason for the extra error in the nonempirical Klotz prior is the presence of two emulated volumes in this model. While the presence of extra emulator error is a confounding factor in the comparison of the Klotz priors, we could also regard the absence of the V_{20} emulator from the empirical prior as one of its strengths.

The empirical Klotz prior was found to improve inference in parameter space in Section 7.4.3, while in Figure 7.12 it was shown that this leads to significant improvements in the estimation of stress–stretch behaviour at the majority of stretch locations. While the Klotz curve is quite widely used in the literature to assist with parameter estimation, it does remain a slightly contentious subject. The conditions in which the Klotz data were measured (ex vivo) inevitably change the dynamics of the LV, and may affect the compatibility of any Klotz-based loss functions or prior distributions with models of the in vivo LV. In reality, any mismatch between the behaviour of the in vivo and ex vivo left ventricle—or the consideration of a left ventricle that violates the Klotz curve relation—

will lead to a bias in the parameter estimates, particularly those parameters that relate to the behaviour of the tissue in the high stretch regime such as θ_3. It is important to note, however, that the effect of the Klotz prior is also to reduce the variance in the posterior samples. Considering the bias-variance decomposition of the generalisation error of the model, the prior will act to reduce the variance term while incurring a small increase in the bias (see Section 3.2 of Bishop (2006) for more details). In this sense, the overall effect of the Klotz prior will be to improve the generalisation error in much the same way as any regularisation method from statistics.

There are some limitations in the methodology. Currently, the statistical emulator for the in vivo predictions does not account for differing patient pressures in the input space. This limits the applicability of the model to healthy volunteers, since this pressure assumption is only valid for this cohort. It is likely that the generalisation of the input to different end diastolic pressures would increase the uncertainty in the material properties since the uncertainty in the pressure measurement must also be accounted for in the parameter inference scheme. A second limitation of the work in this study is the use of the LV geometry at early diastole as the unloaded state (with zero pressure in the LV cavity). This approximation, which is required for evaluation of the Klotz priors in (7.14) and (7.26) where both $\mu_i(\boldsymbol{\theta}, \tilde{\mathbf{h}})$ terms depend on V_0, also introduces error to the in vivo likelihood for measured data and future work will aim to estimate the unloaded configuration, as has been done in the recent literature (Wang et al., 2020). Another issue encountered was the informativeness of the measured data, particularly with reference to parameters θ_3 and θ_4. Of course, this is precisely what we attempt to overcome with the use of the Klotz curve, but it would be preferable to use information in the measured data to resolve the issue. An interesting line of future research would consider other strain measurements, like radial and longitudinal strains, when estimating the parameters, but current methods for extraction from CMR scans are inaccurate. The sensitivity of the radial and longitudinal strains to the inputs of the H-O model will be explored further in the next chapter.

7.5.2 Conclusion

The use of statistical emulators is critical for the translation of cardio-mechanical models to the clinical setting. In particular, these models are thought to hold potential for personalised healthcare, which requires patient specific measurements of LV geometry, pressure and other anatomical features. Unfortunately, the inclusion of the LV geometry in the emulator is confounded by the dimension of the representation, necessitating the use of a generalised emulator like the one explored in Chapter 6, and implemented in this chapter. The approximation of the LV geometry introduces errors to the model that exacerbate the weak identifiability of the H-O model. With the introduction of ex vivo information

via the Klotz curve, the methods in this study have partially alleviated these issues. In particular, the prior was successful in reducing the uncertainty in the weakly identifiable θ_3 parameter. By generating synthetic data that are consistent with the empirical Klotz function, it was shown that this ex vivo information leads to a significant bias and uncertainty reduction for one of the two Klotz prior models proposed (the empirical Klotz prior). For measured datasets, the ability of the Klotz curve to alter the inferred behaviour of the myocardium in the ex vivo domain was presented, reducing the uncertainty about this behaviour. The results of this study, which confirm that certain parameters affect the behaviour of the tissue at higher pressures, motivate a proper sensitivity analysis of the H-O model that accounts for different end diastolic pressures in the model. This will be the subject of the next chapter of this thesis, returning to the original parameter space of the H-O model from (8.1).

Chapter 8

A combined sensitivity analysis and inverse uncertainty quantification study of the H-O model

Up to this point a fixed reparameterisation of the H-O model, originally given in (4.2), has been considered. This was motivated by the work of Gao et al. (2015), as well as our understanding of the physiological meaning of the parameters. Instead of taking this parameterisation as a given, it is important that a proper, global, sensitivity study is carried out, providing a quantitative justification for a new parameterisation of the H-O model. Additionally, we should also be interested in the practical identifiability of the parameters of the model, which tells us how well the parameters can be estimated in light of the measured data. This chapter considers these two critical analyses for the H-O model, using global sensitivity analysis (SA) to assess structural identifiability, and inverse-uncertainty quantification (I-UQ) to assess practical identifiability. The SA is carried out on a range of different input parameters to the LV model, making use of computationally efficient GP surrogate models in place of the numerical forward simulator. The results of the SA are then used to inform a low-order reparameterisation of the constitutive law for passive myocardium. The quality of this parameterisation in the context of an inverse problem having observed noisy experimental data is then quantified with an I-UQ study, which again makes use of GP surrogate models. The I-UQ is carried out in a Bayesian manner using Markov Chain Monte Carlo, which allows for full uncertainty quantification of the material parameter estimates.

This chapter reveals insights into the relation between SA and I-UQ, elucidates the dependence of parameter identifiability on external factors, like LV cavity pressure, and sheds new light on cardio-mechanic model formulation, with the particular focus on the H-O myocardial model. This work has been submitted for review to *Biomechanics and Modelling in Mechanobiology*, and is joint work with David Dalton, Dirk Husmeier and

Hao Gao. All authors contributed to the design and building of the emulators while David carried out the SA and I carried out the inverse uncertainty quantification. All authors were involved in discussions throughout the project and contributed to all studies in this manner. The extensive literature review was carried out by the entire group.

8.1 Background

Ever since the publication of the seminal white paper by Mirams et al. (2016), the cardiophysiological modelling community has appreciated the importance of allowing for variability and uncertainty in their computational and mathematical models. The present work distinguishes between sensitivity analysis (SA) and uncertainty quantification (UQ) in the way depicted in Figure 8.1. SA focuses on structural identifiability, i.e. whether parameters are unidentifiable as a result of the model structure per se. This is closely related to forward uncertainty quantification (F-UQ), where the aim is to establish how uncertainties in model inputs (such as parameters) affect the model output, and is represented by the upper arrow in Figure 8.1. Since structural identifiability is an intrinsic feature of the model, the analysis is *a priori* in nature, meaning that it can be carried out based on the model alone, without the need for any measurements or experimental data.

I-UQ focuses on practical identifiability. When parameters are unidentifiable as a result of practical restrictions related to limited availability and quality of data, they are referred to as practically unidentifiable. I-UQ is therefore *a posteriori* in nature, meaning conditionally dependent on the experimental data, and this is represented by the bottom arrow in Figure 8.1.

SA can be conceptually divided into local and global SA; see Smith (2014) and Morio (2011) for a review. Local SA merely analyses how small perturbations near an input space value influence the output of the model. Global SA, originally introduced by (Sobol, 2001), is a more powerful global approach that has been promoted by Saltelli et al. (2010) via the design of computationally efficient quasi-Monte Carlo sampling techniques, and is increasingly being used for the analysis of complex biological systems (Jarrett et al., 2015).

SA techniques have been widely applied in haemodynamic and cardio-vascular modelling. Melis et al. (2017) have carried out global SA for a range of 1D vascular circulation models, following the procedure of Saltelli et al. (2010) and using Gaussian process emulation to reduce the computational complexity. Eck et al. (2016) have compared Monte Carlo and polynomial chaos methods for global SA in 0D and 1D cardiovascular circulation models. Marquis et al. (2017) and Colebank et al. (2019) have combined SA and I-UQ for cardiovascular modelling, by first using SA to select a subset of identifiable parameters, and then using both frequentist and Bayesian I-UQ techniques to obtain confidence and

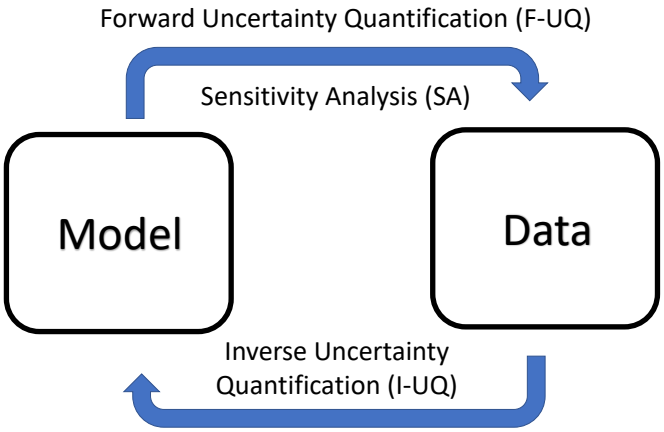


Figure 8.1: Illustration of sensitivity analysis (SA) and uncertainty quantification (UQ). SA and forward uncertainty quantification (F-UQ) are *a priori* in nature and focus on structural parameter identifiability. Inverse uncertainty quantification (I-UQ) is *a posteriori* in nature and focuses on practical parameter identifiability.

credible intervals of the identifiable parameters. While the first study was restricted to local SA, the second study applied both local and global SA. SA and F-UQ have also been applied to arterial wall mechanics for evaluation of vascular drug therapies (Heusinkveld et al., 2018).

There have been various applications of SA and F-UQ in cardiac electrophysiology (Clayton et al., 2020; Pagani and Manzoni, 2021; Mirams et al., 2016), cardiac mechanics (Rodriguez-Cantano et al., 2018; Campos et al., 2019; Kallhovd et al., 2019) and coupled electromechanics (Hurtado et al., 2017; Levrero-Florencio et al., 2020; Rodero et al., 2021). The focus of the present article is on cardiac mechanics, which requires a review of the recent work on SA and UQ in this field in more detail. An overview can be found in Table 8.1.

Osnes and Sundnes (2012) have investigated F-UQ and global SA for the LV passive filling phase using an idealised geometry and a Fung-type constitutive law, with a focus on replacing intrusive methods (i.e. methods that modify the simulation code) with non-intrusive methods (i.e. methods that do not require changes in the simulation code). The impact of constitutive parameters and fibre orientation uncertainties on ventricular dynamics (LV cavity volume, apex lengthening and rotation, wall thickness) were studied, and they found that the overall stiffness and cross-fibre stiffness had the greatest influences on selected model outputs, while fibre orientation had a minor influence on apex rotation. Rodriguez-Cantano et al. (2018) have extended this work with a focus on the impact of uncertainties in material parameters and fibre orientation field on LV mechanics during diastolic filling. The fibre field variation was approximated using a truncated Karhunen-Loeve expansion, and a polynomial chaos expansion-based method was adopted for F-UQ

Table 8.1: Summary of UQ and SA studies on cardiac mechanics

Studies	Cardiac Model	Uncertain Inputs	Quantities of interest (QoIs)	UQ and SA	Results
Osnes H. & Sundnes J. (2012)	1. Idealized LV 2. Passive filling 3. Fung-type SEF	1. constitutive parameters 2. fibre rotation angle	1. LV Cavity volume 2. apex lengthening & rotation 4. wall thickness	1. Polynomial chaos expansion 2. forward UQ	The overall stiffness and cross-fibre stiffness have the greatest influences on QoIs
Rodriguez-Cantano et al. (2018)	1. image-derived LV 2. Passive filling 3. Fung-type SEF	1. constitutive parameters 2. random fibre filed using a truncated Karhunen-Loeve expansion	1. LV cavity volume 2. apex lengthening 3. wall thickness 4. wall volume	1. polynomial chaos expansion 2. Sobol sensitivity 3. forward UQ	The overall stiffness is the most sensitivity parameter; QoIs are relatively insensitive to fibre angle, but to local variations
Campos J. et al. (2019)	1. personalized LV 2. passive filling 3. Fung-type SEF	1. constitutive parameters 2. fibre/sheet angles 3. wall thickness	1. wall thickness 2. LV cavity volume 3. the torsion 4. mean fibre stress/strain	1. Polynomial chaos expansion 2. forward UQ 3. Sobol sensitivity	Stress is highly sensitivity to wall thickness; Wall thickness has similar proportion of impact on QoIs as material stiffness
Campos J. et al. (2020)	1. personalized LV 2. whole cardiac cycle 3. Fung-type SEF	1. constitutive parameters 2. fibre orientations 3. wall thickness 4. contractility 5. the circulatory model	1. LV torsion 2. mean fibre stress/strain 3. Ejection Fraction 4. end-systolic pressure 5. max pressure variation	1. polynomial chaos expansion 2. forward UQ 3. Sobol sensitivity	The most sensitive inputs are wall thickness and contractility
Kallhovd S. et al. (2019)	1. image-derived LV 2. Fung-type SEF 3. passive filling with active contraction	selected 21 published sets of parameters	1. LV cavity volume 2. fibre stress/strain 3. pressure-volume loop	uncertainty analysis	Similar P-V loops could be obtained by tuning contractility; Local stress is less sensitive to passive parameters, but not strain.
Barbarotta L. et al. (2021)	1. average LV geometry 2. Fung-type SEF 3. passive filling with active contraction	fibre orientation (helix angle and transverse angle) modelled using a 5-parameter rule-based model	End-systolic strain	1. Mean and standard deviation based on the elementary effects method 2. coefficient of variation	Shear strains are more sensitive to fibre orientation than normal strains
Hurtado E. et al. (2017)	1. electromechanics model in a bar geometry 2. isotropic Neo-Hookean material	1. electrophysiological parameters (conductance, etc) 2. passive parameters	1. action potential duration 2. peak intracellular transient 3. fibre stretch 4. active tension	Polynomial chaos expansion	Passive parameters have little impact on the duration of action potential and peak calcium transient
Rodero C. et al. (2021)	1. electromechanics model 2. shape analysis using PCA 3. Fung-type SEF	Geometry variation by adding +-2/3 SD of each PCA mode to the average mesh	1. ventricular pressure/volume 2 peak pressure variation 3. contraction duration	1. global sensitivity analysis 2. forward finite element simulations	PCA modes 2 and 9 are the most important geometrical features in determining LV mechanics
Gao H. et al. (2015)	1. personalized LV 2. passive filling 3. the H-O model	Constitutive parameters	1. circumferential strains 2. LV cavity volume	local sensitivity analysis	The isotropic stiffness and the myofibre stiffness are the most sensitive parameters
Levrero-Florencio F. et al. (2020)	1. electromechanical model 2. idealized LV geometry 3. the H-O model	1. cross-fibre stiffness 2. contractility 3. fibre rotation angle 4. boundary conditions	1. LV ejection fraction 2. systolic long-axis shortening 3. systolic wall thickening 4. end-systolic pressure	1. local sensitivity analysis 2. global sensitivity analysis using a partial rank correlation coefficient	LV ejection fraction is strongly affected by contractility; The systolic long-axis shortening is sensitive to fibre rotation angle and cross-fibre contraction

analysis in a more realistic LV model compared to Osnes and Sundnes (2012). Similar results were reported in Barbarotta and Bovendeerd (2021). Campos et al. (2019) have extended global SA and F-UQ analysis in LV passive mechanics by considering geometrical uncertainties like wall thickness variation, and they suggested that LV passive mechanics may be more affected by wall thickness than material properties. Later, they (Campos et al., 2020) further extended the similar analysis to a full cardiac cycle by taking into account uncertainties in active stress and the circulatory model. They found that LV ejection fraction and ventricular torsion were very sensitive to active stress, wall thickness and fibre direction, but not the passive material parameters, which was different from the SA and F-UQ in LV passive mechanics.

The aforementioned studies (Osnes and Sundnes, 2012; Rodriguez-Cantano et al., 2018; Osnes and Sundnes, 2012; Campos et al., 2019, 2020) used a transversely isotropic Fung-type constitutive law. Recent studies have demonstrated that the myocardium is an orthotropic material with three mutually orthogonal principal axes (Dokos et al., 2002; Holzapfel and Ogden, 2009; Sommer et al., 2015), and for that reason, a few orthotropic nonlinear constitutive laws have been proposed, including the widely used H-O model developed by Holzapfel and Ogden (2009).

For the H-O model, Gao et al. (2015) performed local SA for the passive myocardial properties using a patient-specific LV geometry and found that the LV passive mechanics are highly sensitive to the isotropic stiffness and myofibre stiffness. They also found that the parameters in the H-O model are highly correlated. This correlation between the parameters has motivated various reparameterisations of the model that will be discussed in Section 8.2.2, as well as the parameterisation used in the preceding chapters of this thesis. Levrero-Florencio et al. (2020) studied local/global SA in an idealised LV electromechanical model with a focus on LV pump function by treating the cross-fibre stiffness as the only uncertain parameter in the H-O model. To the best of our knowledge, a comprehensive global SA and I-UQ analysis using the H-O model has not been reported in the literature despite its wide application in cardiac mechanics (Gao et al., 2017; Sahli Costabal et al., 2019).

This work aims to complement previous work on SA and F-UQ in cardio-mechanics models (Osnes and Sundnes, 2012; Rodriguez-Cantano et al., 2018; Campos et al., 2020) by including I-UQ with a focus on the H-O model in the LV passive filling. To paraphrase this with reference to Figure 8.1: while the previous studies cited above have carried out UQ that is intrinsically *a priori* in nature (F-UQ), focusing on structural parameter identifiability corresponding to the top arrow in the diagram. This work is complemented with a dimension that is *a posteriori* in nature, corresponding to the bottom arrow in Figure 8.1. This makes it possible to additionally quantify the degree of practical parameter identifiability, in light of experimental data. This work extends the work of Gao et al. (2015)

by carrying out global rather than local SA, and systematically combining it with I-UQ. This study is conceptually related to the work of Marquis et al. (2017) and Colebank et al. (2019), but shifts the focus from cardiovascular to cardio-mechanics modelling. Besides gaining methodological insights into the relation between structural and practical parameter identifiability in the context of the LV passive mechanics, this work leads to a better understanding of the appropriate level of complexity for constitutive cardiac mechanics models, with a particular focus on the H-O strain energy function.

8.2 Methods

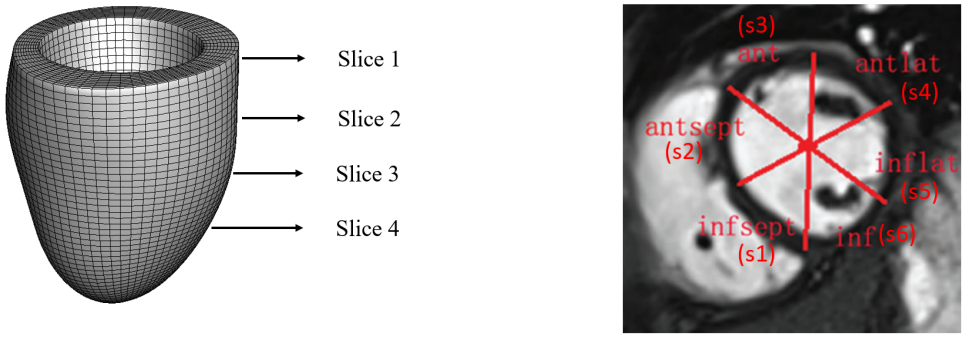
The objective of this chapter is to perform SA and I-UQ for the parameters of the H-O model, with the SA proposing a new parameterisation that can be explored using I-UQ. Section 8.2.1 revisits the model of interest, namely the H-O model, while briefly discussing some existing reparameterisations that have been adopted in the literature. Section 8.2.3 gives a summary of the global approach to SA that will be used to quantify the importance of the input parameters of this model. A more detailed overview of SA was provided in Section 3.1.3. Section 8.2.4 describes the concept of surrogate modelling, which allows for the global SA to be carried out at significantly lower computational expense, and finally Section 8.2.5 details the Bayesian approach used to perform I-UQ.

8.2.1 Left Ventricular Forward Model

Patient-specific LV Geometry

In this study, a cardiac magnetic resonance (CMR) imaging derived LV model of a healthy volunteer (male, 43 years) is chosen from previous work (Gao et al., 2017), which was reconstructed from both the short and long axial CMR cine images at early diastole following existing studies (Genet et al., 2014b) and the method outlined in Section 5.2.1. The LV geometry, which is displayed in Figure 8.2 (a), was discretised using hexahedral elements. Further details of LV geometry reconstruction can be found in Li et al. (2020a) and the discussion in Section 5.2.1.

It is common practice to incorporate the layered myofibre structure in LV models. Due to the extreme difficulty of imaging myofibres in vivo, a rule based method (RBM) was adopted for describing the myofibre structure (Wang et al., 2013a), in which a local material coordinate system was defined, the so-called fibre (**m**)–sheet (**s**)– sheet-normal (**n**) system that was shown in Figure 2.4. In the present study, the fibre angle varied linearly from α_{endo} at the endocardium to α_{epi} at the epicardium, with the sheet orientation along the transmural direction. This same method was used to generate the fibre directions in the simulator for the preceding work in this thesis.



(a) (b)

Figure 8.2: A reconstructed LV geometry with indications of 4 short-axis slices from the base to the mid-ventricle (a), and schematic illustration of 6 segmental regions for a selected short-axis slice following the AHA division convention. infsept: inferior septum; antsept:anterior septum; ant: anterior; antlat: anterior lateral; inflat: inferior lateral; inf: inferior. The strain regions were similarly visualised in Figure 2.3.

8.2.2 Constitutive Law of the Myocardium

The passive myocardium is considered an incompressible, anisotropic and hyperelastic material, described by the constitutive law introduced by Holzapfel and Odgen (Holzapfel and Ogden, 2009):

$$\begin{aligned}\boldsymbol{\Psi} = & \frac{a}{2b}\left[\exp\left\{b\left(I_1-3\right)\right\}-1\right] \\ & + \sum_{i\in\{\mathrm{f,s}\}} \frac{a_i}{2b_i}\left[\exp\left\{b_i\left(\max(I_{4i},1)-1\right)^2\right\}-1\right] \\ & + \frac{a_{\mathrm{fs}}}{2b_{\mathrm{fs}}}\left\{\exp\left(b_{\mathrm{fs}}I_{8\mathrm{fs}}^2\right)-1\right\}\end{aligned} \tag{8.1}$$

Implementation in the simulator requires the specification of 8 different material parameters: a, b, a_{f} and b_{f}, a_{s}, b_{s}, a_{fs} and b_{fs}, all related to different properties of the myocardial tissue. The H-O model is combined with a set of momentum balance equations to simulate the behaviour of the LV in diastole. Further details on this model can be found in Section 2.2.

The H-O model has been used widely in personalised cardiac modelling in recent years (Gao et al., 2017; Palit et al., 2018b; Baillargeon et al., 2014; Sahli Costabal et al., 2019; Guan et al., 2019). Due to the poor identifiability of some parameters, different reduced forms have been proposed in the literature, which generally fall into two frameworks. The first of these is to introduce scaling parameters that link multiple parameters together. For example, Palit et al. (2018b) used a four dimensional parameterisation, a, b, Ka, Kb where Ka and Kb scale the remaining a and b parameters of the H-O model. Further examples of this type of reparameterisation can be found in the literature (Gao et al.,

2015) and in Section 4.1.2. The alternative approach is to also ignore several parameters, either by fixing them to constant values or removing them from the model altogether. Krishnamurthy et al. (2013a) removed the terms depending on $a_{\mathrm{s}}.b_{\mathrm{s}}, a_{\mathrm{fs}}, b_{\mathrm{fs}}$ from the model. Hadjicharalambous et al. (2015) set a_{s} and a_{fs} equal to 0 and fixed b and b_{f} to constant values, leaving two unknown parameters (a and a_{f}). They further showed the bijectivity of the map from this two dimensional parameter space. However, this bijectivity only holds in the case where the end diastolic pressure of blood within the cavity (EDP) is known and b and b_{f} are fixed. With this in mind, they further proposed learning a ratio of a and a_{f} in the real data case when EDP is not known in advance. This method has been adopted in more recent work (Asner et al., 2016; Hadjicharalambous et al., 2017, 2021). None of the above reparameterisations are based on a comprehensive global sensitivity study, but from empirical observations and intuitive insights of myocardial behaviours. The current study is the first to motivate a reparameterisation using a proper sensitivity analysis.

Input Parameters

A total of eleven input parameters to the H-O model are considered as random in the SA experiments. They are the eight material parameters from (8.1): $a, b, a_{\mathrm{f}}, b_{\mathrm{f}}, a_{\mathrm{s}}, b_{\mathrm{s}}, a_{\mathrm{fs}}, b_{\mathrm{fs}}$, EDP, and two fibre rotation angles of the RBM for fibre generation: α_{endo} and α_{epi}. Each input parameter is considered independently and randomly distributed within the intervals given in Table 8.2 based on a previous study (Gao et al., 2017). Two different forms of input parameter distributions are considered in the SA experiments, as will be discussed in Section 8.3.1. For the I-UQ study, interest lies in inferring the material parameters for fixed fibre angles over a range of different pressures. The reasoning for fixing the fibre angles will be discussed in Section 8.2.5.

Table 8.2: Model input parameters with bounds based on Gao et al. (2017)

		Bounds	
Input	Unit	Lower	Upper
a	kPa	0.1	10
b	-	0.1	10
a_{f}	kPa	0.1	10
b_{f}	-	0.1	10
a_{s}	kPa	0.1	10
b_{s}	-	0.1	10
a_{fs}	kPa	0.1	10
b_{fs}	-	0.1	10
EDP	mmHg	4	30
α_{endo}	Degrees	-90	0
α_{epi}	Degrees	0	90

Quantities of Interest

Four different types of end-diastolic response of the LV model will be considered as quantities of interest (QoIs) for the experiments in this study: LV cavity volume at end diastole (LVV), as well as segmental circumferential (ε_{cc}), radial (ε_{rr}) and longitudinal (ε_{ll}) strains. These were all visualised in Figure 2.3, and are formally defined as follows:

$$\begin{cases} \varepsilon_{\text{cc}} = \mathbf{c} \cdot \left(\frac{1}{2}(\mathbf{F}_{\text{ed}}^T \mathbf{F}_{\text{ed}} - \mathbf{I})\mathbf{c}\right), \\ \varepsilon_{\text{ll}} = \mathbf{l} \cdot \left(\frac{1}{2}(\mathbf{F}_{\text{ed}}^T \mathbf{F}_{\text{ed}} - \mathbf{I})\mathbf{l}\right), \\ \varepsilon_{\text{rr}} = \mathbf{r} \cdot \left(\frac{1}{2}(\mathbf{F}_{\text{ed}}^T \mathbf{F}_{\text{ed}} - \mathbf{I})\mathbf{r}\right), \end{cases} \tag{8.2}$$

where $\mathbf{c}$, $\mathbf{l}$, $\mathbf{r}$ are the unit circumferential, longitudinal and radial directions, and $\mathbf{F}_{\text{ed}} = \partial \mathbf{z}_0 / \partial \mathbf{z}_1$ with $\mathbf{z}_0$ the position at early diastole and $\mathbf{z}_1$ the position at end diastole. Note that all strains are calculated with respect to the end-diastolic phase to be consistent with clinical convention.

To extract the segmental strains from the deformed mesh at end of diastole, the clinical convention (which is the convention used throughout this thesis) is used, dividing the LV circumferentially into 6 segments, as visualised for a CMR scan in Figure 8.2 and presented in Figure 2.3. Normally, 4 short-axial slices are available from the most basal region to the middle of the LV, and dividing each slice into the 6 segments gives a total of 24 segments on the LV. For each of these segments an averaged peak strain is calculated, providing 24 simulated strains at end of diastole. In the SA study only one segmental strain region is used because the sensitivity is similar for all segments. Specifically, and with reference to Figure 8.2, the segmental circumferential strain, segmental longitudinal strain and radial strain at the inferior lateral segment from Slice 2, denoted as $\varepsilon_{\text{cc}}^*$, $\varepsilon_{\text{ll}}^*$, $\varepsilon_{\text{rr}}^*$, respectively, were used. Here * indicates that the strain is taken from a preselected segment. For the I-UQ study (see Section 8.4), all 24 segmental circumferential strains are used, which means an inverse problem identical to the ones in the preceding chapters of this thesis.

8.2.3 Sensitivity Analysis

An overview of SA was provided in Section 3.1.3 and a high level summary will be provided in this section. The cardio-mechanic forward model described in Section 8.2.1 is implemented as a simulator, ζ, that, for a fixed LV geometry, maps a set of input parameter values $\boldsymbol{\theta} = (\theta_1, \theta_2, ...\theta_D)$ to associated output quantities of interest, $y_1, y_2, ...y_M$:

$$f(\theta_1, \theta_2, ...\theta_D) = (y_1, y_2, ...y_M). \tag{8.3}$$

In this case, $D = 11$, corresponding to the eleven input parameters outlined in Section 8.2.2, while the M outputs can be chosen to be any of the quantities of interest

detailed in Section 8.2.2. The first-order sensitivity index, denoted S_{ji}, of input parameter θ_i for output y_j is defined as:

$$S_{ji} = \frac{\mathbb{V}_{\theta_i}\left[\mathbb{E}_{\boldsymbol{\theta}_{\sim i}}\left[y_j \mid \theta_i\right]\right]}{\mathbb{V}[y_j]}, \tag{8.4}$$

where the notation $\boldsymbol{\theta}_{\sim i}$ denotes the vector of all the random inputs except from θ_i. The first order index does not account for interaction effects between different input parameters. For this purpose, we must use the total-effect index, T_{ji}, for each input θ_i, which is defined with respect to output y_j as:

$$\begin{aligned} T_{ji} &= \frac{\mathbb{V}[y_j] - \mathbb{V}_{\boldsymbol{\theta}_{\sim i}}\left[\mathbb{E}_{\theta_i}\left[y_j \mid \boldsymbol{\theta}_{\sim i}\right]\right]}{\mathbb{V}[y_j]}, \\ &= \frac{\mathbb{E}_{\boldsymbol{\theta}_{\sim i}}\left[\mathbb{V}_{\theta_i}\left[y_j \mid \boldsymbol{\theta}_{\sim i}\right]\right]}{\mathbb{V}[y_j]}. \end{aligned} \tag{8.5}$$

Due to the computational costs of the simulator, an emulator will be used for the SA, as has been done in the literature (see Melis et al. (2017); Rodriguez-Cantano et al. (2018); Gramacy (2020, Chapter 8)). This means replacing y_j with its emulated quantity.

8.2.4 Surrogate Modelling

A GP will be used to emulate the simulator function, requiring the selection of mean and covariance functions. In the present study, a zero mean function was used as all outputs were standardised to mean zero and unit variance before training. For the covariance function, the squared exponential function will be used, with separate lengthscales for each input dimension allowing for automatic relevance determination (recall that this allows for the changing influence of the input dimensions on the function):

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left(-\frac{1}{2}\sum_{k=1}^{D}\frac{(x_{ik} - x_{jk})^2}{\lambda_k^2}\right) + \eta^2\delta_{ij}. \tag{8.6}$$

The quantity η^2 is a small nugget term added to ensure numerical stability when calculating the predictive distribution of the GP. As discussed in Section 3.4.1, the properties of the kernel are governed by its hyperparameters $\sigma_f, \lambda_1, \lambda_2, ...\lambda_k, \eta$, and their values can be tuned to allow the GP to best represent the true underlying function. In the current study, this will be done by setting the hyperparameters to those values that maximise the log marginal likelihood of the training data. While fully Bayesian approaches are possible for fitting the hyperparameters, obtaining a point estimate in this manner is the most common approach in the GP literature. This is because the number of hyperparameters is relatively low when compared to flexible parametric models, and for sufficiently large

training sets the posterior distributions tend to be highly peaked.

The posterior mean of the GP gives the best estimate for the unknown true function values under quadratic loss. Using this point estimate to compute the Sobol indices allows for efficient calculation that is multiple orders of magnitude quicker than directly using the numerical forward model. However, in addition to a best point estimate, the GP returns a full posterior probability distribution over the unknown function values. Oakley and O'Hagan (2004) incorporate this uncertainty analytically in their Bayesian approach to SA, but this requires assumptions are made about the distribution of the inputs to the model. To include this extra uncertainty we can follow the approach detailed in Gramacy (2020, Chapter 8) and, instead of plugging in a single point estimate to compute the Sobol indices, draw an ensemble of samples from the posterior distribution of the GP, and compute the Sobol indices for each sample. The uncertainty in the value of the indices can then be quantified by calculating summary statistics of the resulting ensemble of index values.

8.2.5 Posterior Parameter Inference

For a given patient, a fixed but unknown set of material parameters is assumed. Measurement of these parameters would require mechanical tests on a sample of dissected myocardium, so these can instead be inferred from measurements of circumferential strains and LVV obtained in vivo. Parameter estimation for a specific patient requires that we specify suitable physiological conditions—the EDP and fibre angles—which were treated as random variables in the SA study. As discussed in Section 8.2.2, the fibre angles will be fixed for the I-UQ study. This is motivated by the fact that in vivo estimation of these angles will be available with further development of already existing techniques (Toussaint et al., 2013; Khalique et al., 2020).

From the model, we obtain 24 circumferential strains $\mathbf{s} = \{\varepsilon_{\text{cc}}^i, i = 1, ..., 24\}$, and a prediction of the LVV V. The measured quantities, $\boldsymbol{y}^*$, are assumed to relate to these predictions through the following noise model:

$$\boldsymbol{y}^* = (V, \mathbf{s}) + \boldsymbol{\xi}, \tag{8.7}$$

where $(V, \mathbf{s})$ is the concatenation of the volume prediction and strain predictions and $\boldsymbol{\xi}$ is a random variable representing the additive noise. Future work could also include segmental radial and longitudinal strains in this model, but currently there is no method for obtaining these at high precision from CMR images. Therefore, excluding these from the model better resembles what would be feasible with currently available measured data. In the preceding chapters of this thesis, this model was assumed to be iid Gaussian, with the standardised volume and strains corrupted by the same level of Gaussian noise. However,

a more accurate representation of reality is to assume that the volume is corrupted by a separate, smaller, variance than the circumferential strains:

$$y_0^* = V + \xi_0, \quad \boldsymbol{y}_{1:}^* = \mathbf{s} + \tilde{\boldsymbol{\xi}}, \tag{8.8}$$

where $\xi_0 \sim \mathcal{N}(0, \sigma_0^2)$ is a random variable representing the additive Gaussian noise on the volume and $\tilde{\boldsymbol{\xi}} \sim \text{MVN}(\mathbf{0}, \tilde{\sigma}^2\mathbf{I})$ is a random variable representing the additive iid Gaussian noise on the circumferential strains. Given that σ_0^2 is the variance of the noise on only one measurement, this parameter is non-identifiable. As a result, we must fix this variance during inference to a value obtained in empirical studies. The assumption that the 24 circumferential strains are all corrupted by Gaussian noise with the same variance ensures that the other variance parameter, $\tilde{\sigma}^2$, can be inferred along with the material parameters. The noise model from (8.8) gives rise to the following log-likelihood function:

$$\begin{aligned} l(\boldsymbol{y}^* \mid \boldsymbol{\theta}, \tilde{\sigma}^2) = &-\frac{1}{2}\log(\sigma_0^2) - \frac{1}{2\sigma_0^2}(y_0^* - \zeta_0(\boldsymbol{\theta}))^2 \\ &- \frac{24}{2}\log(\tilde{\sigma}^2) - \frac{1}{2\tilde{\sigma}^2}\sum_{i=1}^{24}(y_i^* - \zeta_i(\boldsymbol{\theta}))^2, \end{aligned} \tag{8.9}$$

where $\zeta_i(\boldsymbol{\theta})$ is the ith circumferential strain prediction at material parameters $\boldsymbol{\theta}$ and $\zeta_0(\boldsymbol{\theta})$ is the simulated LVV. Unlike the SA study, $\boldsymbol{\theta}$ does not contain all the 8 parameters from the H-O model (8.1). Instead, the SA study results are used to inform a new parameterisation of the H-O model where some parameters are held fixed. Different from previous chapters in this thesis, the dependence of the simulator on the LV geometry is not explicit here. This is because the LV geometry was fixed throughout the study.

As in previous chapters, the high computational cost of the simulator necessitates the use of a surrogate model. For this purpose, a GP model will be used in order to be consistent with the choice of surrogate model for the SA study. In Appendix B.2, the integration of the predictive variance out of the loss function was considered, and a similar approach could be used here to give a likelihood function that incorporates both the mean and variance of the GP posterior distribution. Unfortunately, the computational costs of this approach are huge when we consider large training datasets. For this reason, the uncertainty in the posterior distribution of the GP will not be accounted for in the I-UQ.

A Bayesian approach will be taken to parameter inference, using MCMC to sample from the posterior distribution of the material parameters. This requires the selection of a prior distribution, for which a uniform prior will be used with bounds selected according to Table 8.2. Combining the prior and the likelihood, samples are obtained from the posterior distribution of the parameters using NUTS (see Section 3.2.5), with 5 parallel chains used to check for convergence using the PSRF (see Section 3.2.4). In total, 1000

samples were drawn using MCMC, before thinning to obtain the final summary of the posterior distribution.

8.3 SA Simulation Studies

Two SA experiments have been carried out for the H-O model. Section 8.3.1 gives the details of the two experimental studies, before Sections 8.3.2 and 8.3.3 describe how GP surrogate models were trained and validated. Finally, the experimental results obtained using the surrogates are presented and discussed in Section 8.3.4.

8.3.1 Experimental Setup

Sensitivity Analysis One

In the first numerical experiment, which is referred to as Sensitivity Analysis One (SA1), an SA is performed with all eleven input parameters detailed in Section 8.2.2 considered as random variables. The analysis is performed for four different output quantities: LVV, $\varepsilon^*_{\text{cc}}$, $\varepsilon^*_{\text{ll}}$ and $\varepsilon^*_{\text{rr}}$. For brevity, each strain value is considered only in the inferior lateral segment at slice 2 from Figure 8.2, as the SA results for each strain type are similar across the 24 segmental strain regions. To examine the influence on the SA results of the prior distribution for the eight material parameters, two prior distributions are considered. In the first experiment, the material parameters are considered to be independently uniformly distributed between the bounds given in Table 8.2. This is referred to as the uniform-prior. In the second experiment, the material parameters are considered to be independently uniformly distributed on the *log-scale* between these bounds. This is referred to as the log-uniform prior. For both priors, the non-material parameter inputs: EDP, α_{endo} and α_{epi}, were considered independently uniformly distributed between the bounds stated in Table 8.2. The computation of the Sobol indices (8.4) and (8.5) was performed using the Python package SAlib (Herman and Usher, 2017).

Sensitivity Analysis Two

In Sensitivity Analysis Two (SA2), the effect of EDP on the sensitivity scores of the material parameters of the H-O model is explored. Specifically, the value of EDP is fixed to each of the values $\{5, 7.5, 10, ...25\}$ mmHg, before conducting an emulator-based sensitivity analysis for each fixed pressure value. This experiment is informative because the results of SA1 considered EDP as a random variable, with a wide range of uncertainty between 4 and 30 mmHg. However, for a given subject EDP could be a measured value. For this reason, this study examines whether the sensitivity scores of the material parameters vary for different fixed EDP values. As discussed in Section 8.2.5, currently only LVV

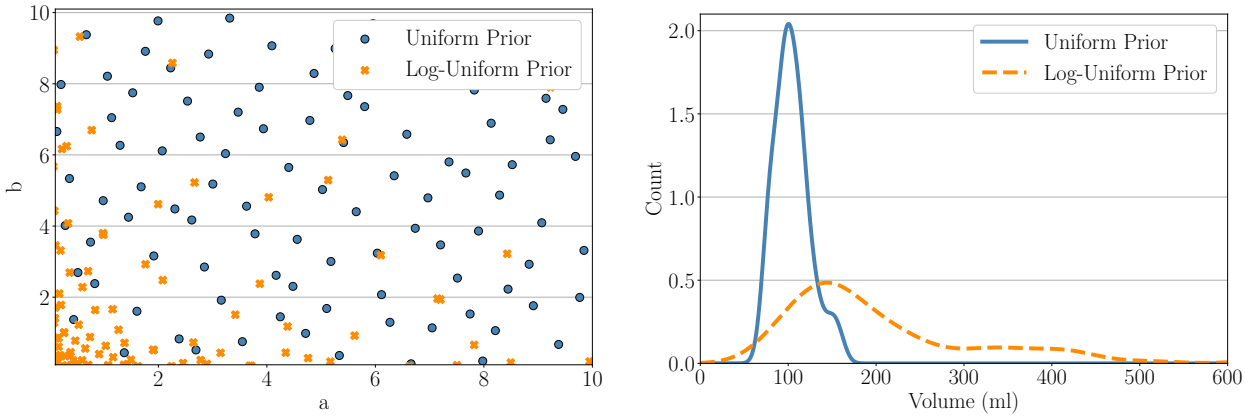
and ε_{cc} can be measured with high accuracy from cardiac imaging scans (Mangion et al., 2016b), so this analysis is restricted to these two output quantities. In addition, because the material parameters are of primary interest in this experiment, we do not consider the two RBM fibre generation parameters as random variables for SA2. Instead, α_{endo} and α_{epi} were fixed to the values $60°$ and $-90°$ (Lombaert et al., 2011; Land et al., 2015), respectively.

8.3.2 Surrogate Model Training

SA experiments are performed using GP surrogate models, as discussed in Section 3.4.1; one surrogate each for the four output quantities under consideration. Constructing these surrogate models requires training data sets of input-output pairs to be created for each output. This in turn requires the *location* and *number* of training inputs to be specified. Given the high computational costs incurred with each forward simulation of the LV diastolic filling process, it is important to choose a set of input points that cover the parameter domain described in Section 8.2.2 as efficiently as possible. As was discussed in Section 3.1.2, a Sobol sequence can be used for this purpose. Since two different material parameter priors will be considered for the SA, the first 1000 material parameter samples were taken in standard parameter space, and the final 1000 were taken in log-parameter space. The simulator was then evaluated at each of these input points and the output quantities of interest were extracted from each simulation result, yielding a training data set of 2000 input-output pairs on which the GP surrogates for each output QoI could be trained.

The effect of the two alternative material parameter sampling procedures is illustrated in Figure 8.3 (a), which plots the first 100 sampled a and b values in original Euclidean space, and the first 100 sampled in log-warped Euclidean space. By sampling in log space, lower stiffness material parameter configurations are favoured. In this lower stiffness region where the cardiac tissue is softer, the magnitude of the expansion of the left ventricle is more sensitive to small changes in the material stiffness parameters than is the case at higher stiffness levels. As a result, the variance observed in the end-diastolic quantities of interest is higher under the log-uniform prior than for the uniform prior. This is illustrated in Figure 8.3 (b), which displays empirical density plots for the simulated LVV obtained when running from the input points of Figure 8.3 (a). With the log-uniform sampled points, the distribution of LVV values exhibits a significantly longer tail. This analysis hints at the benefits of a log-uniform emulator design, which will now be discussed in more detail.

The nonstationarity of the simulator has been acknowledged in previous parts of this thesis (see Section 4.4.3 and Section 6.3.3) and the emulator design will now be discussed in this context. In Figure 8.4, the output functions of the simulator are visualised in



(a) (b)

Figure 8.3: (a) - Distribution of first 100 a/b values for uniform design (blue), and log-uniform design (orange). (b) - Density plots for LVV for simulations run from input points in (a)

different two-dimensional spaces. These functions have a sigmoidal shape, and we can separate a surface into two regimes: fast varying and slow varying. As a rule of thumb, large parameters lead to a slowly varying regime and small parameters lead to a fast varying one. In other words, the function is non-stationary. For such a function, a space filling design is suboptimal because it assumes that each point in input space is equally informative about the shape of the function. If we have a region of space where the function varies more than in others, then points in this fast varying region of space are more informative about the function shape. A simple way to account for this property is to use a manifold GP approach (Calandra et al., 2016) to find a transformation that makes the function more stationary. If we then build a space filling design in the transformed space and transfer back to the original space, then the design will be more concentrated in regions where the function is more varied. To perform this second task, we require that the transformation to the new space is injective. In Figure 8.5 we see that a log transformation performs similarly to more complicated neural network transformations (labelled as MLP-nhidX for a neural network with X hidden nodes) for this purpose. As was discussed in Section 3.1.2, we must combine with this design a model that accounts for the nonstationarity. If we were to fit a GP regression model with standard SE kernel, the posterior distribution of the GP would be heavily influenced by the fast varying portion of space, with the log marginal likelihood peaking at smaller lengthscale hyperparameters. This would result in a function with too much variation and overly-conservative uncertainty intervals in the slowly varying region of space. Instead, when training the GPs, the *log* of the eight material input parameters was used as inputs, rather than the raw values. This approach is similar in principle to the one adopted by Calandra et al. (2016), but with a more context-specific transformation.

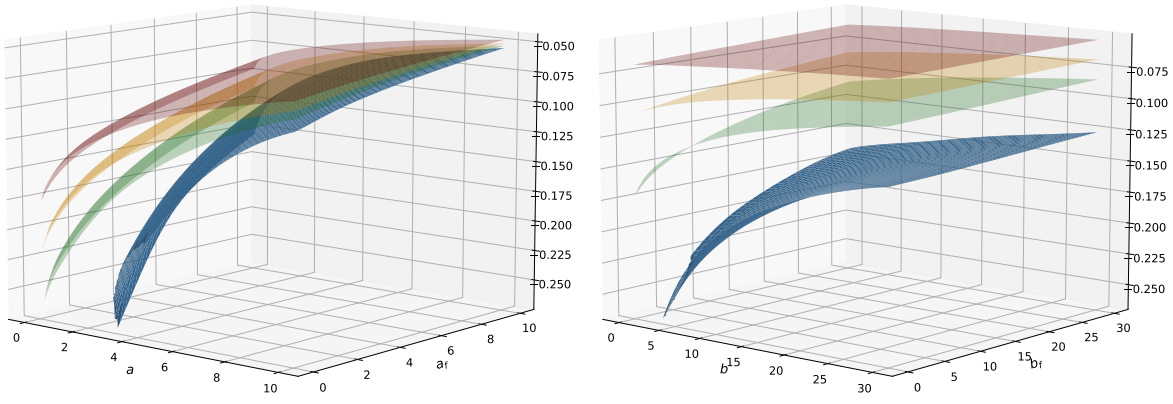


Figure 8.4: Plots of the simulator function over different two dimensional input spaces. In each subplot, two parameters are fixed and the different surfaces correspond to different values of the fixed parameters. As the value of the fixed parameters decreases, the surface shifts downwards and the non linearity increases.

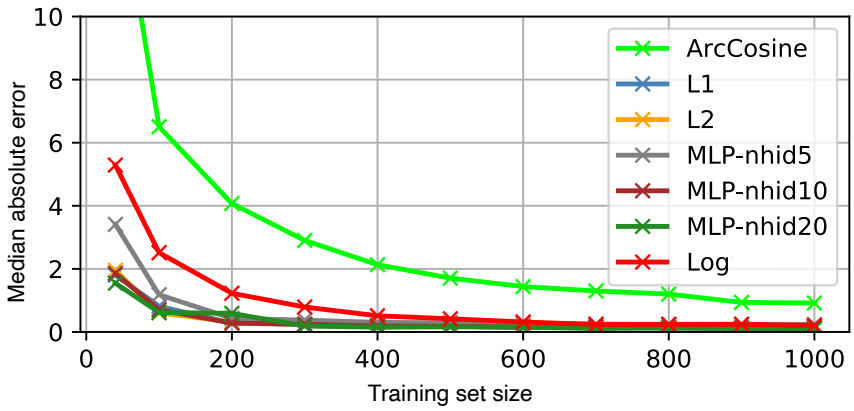


Figure 8.5: Comparison of different input warped GPs. The non-stationary ArcCosine kernel is also included in the comparison. Each of the other models uses some form of transformation of the input space. Of particular interest are the neural network transformations, labelled as MLP-nhidX for a neural network (multilayer perceptron) with X hidden nodes.

8.3.3 Surrogate Model Validation

By performing the SA experiments with surrogate models rather than the true forward model, we will introduce some error in the calculation of the Sobol indices (8.4) and (8.5). Therefore, it is essential that the accuracy of the surrogates is validated. This can be done by evaluating their accuracy on a set of 100 independent test simulations from the forward model. The input locations that the test simulations were run from were found by continuing the Sobol sequence used to generate the training data for a further 100 points. Note that, for the material stiffness parameters, the sequence was run in log space, which favours less stiff material parameter configurations, as discussed in Section 8.3.2. The

accuracy of the surrogates was quantified using the Q^2 coefficient. If we denote the true outputs from the simulator as $\mathbf{f}^*$, and the predictions of the GP surrogate as $\hat{\mathbf{f}}^*$, where $\hat{\mathbf{f}}^*$ is set to be the posterior mean of the GP, then the Q^2-coefficient is defined as:

$$Q^2 = 1 - \frac{(\mathbf{f}^* - \hat{\mathbf{f}}^*)^\top (\mathbf{f}^* - \hat{\mathbf{f}}^*)}{(\mathbf{f}^* - \bar{\mathbf{f}}^*)^\top (\mathbf{f}^* - \bar{\mathbf{f}}^*)} \tag{8.10}$$

where $\bar{\mathbf{f}}^*$ is the vector of the mean value of the test outputs. This definition is analogous to that for the R^2 coefficient, except it is calculated on out-of-sample, rather than in-sample data. The Q^2-coefficient gives the proportion of the variance in the test outputs that is accounted for by the surrogate model, with values close to one indicating that the surrogate has high predictive accuracy.

8.3.4 Results

Surrogate Model Verification

The SA experiments were performed using GP surrogate models with squared-exponential covariance functions, as described in Section 8.2.4. For the four output quantities of interest in the SA: LVV, $\varepsilon^*_{\text{cc}}$, $\varepsilon^*_{\text{ll}}$ and $\varepsilon^*_{\text{rr}}$, the accuracy of the surrogates is validated on the 100 test simulations detailed in Section 8.3.3. Table 8.3 presents the Q^2-coefficient for the four outputs, to two decimal places. The Q^2 values for LVV, $\varepsilon^*_{\text{cc}}$ and $\varepsilon^*_{\text{ll}}$ are each 0.98, indicating very strong agreement between the true outputs and the predictions of the surrogate model. The surrogate model for $\varepsilon^*_{\text{rr}}$ is less accurate, with a Q^2-coefficient of 0.86. Nevertheless, this value still indicates good predictive accuracy, and for this reason these trained surrogates can be trusted to perform the two SA experiments described in 8.3.1.

Table 8.3: **Surrogate Model Verification Results**: Q^2-coefficient values for the four output quantities considered for the SA experiments, calculated on a set of 100 independent simulations from the forward model.

Output	Q^2
LVV	0.98
$\varepsilon^*_{\text{cc}}$	0.98
$\varepsilon^*_{\text{ll}}$	0.98
$\varepsilon^*_{\text{rr}}$	0.86

Sensitivity Analysis One

The results of SA1 are displayed in Figure 8.6. All eleven input parameters detailed in Section 8.2.2 are assumed to be random variables in this experiment. As discussed in

Section 8.3.1, the analyses are repeated for two prior distributions over these inputs: the uniform prior and the log-uniform prior, where the log-uniform prior favours less stiff material parameter configurations. Four output quantities of interest at end diastole are considered: LVV, $\varepsilon^*_{\mathrm{cc}}$, $\varepsilon^*_{\mathrm{ll}}$ and $\varepsilon^*_{\mathrm{rr}}$. The four rows of Figure 8.6 correspond to each of these output quantities, while the left column shows the results under the uniform prior, and the right column shows the results under the log-uniform prior. The analyses were carried out using the sampling approach detailed in Section 8.2.4. That is, rather than plug in the point estimate from the GP surrogate when numerically evaluating the Sobol indices, 250 samples were drawn from the posterior distribution of the GP. The Sobol indices were then evaluated for each sample, resulting in an ensemble of estimated indices. The bars in Figure 8.6 correspond to the ensemble mean values, and a box-plot is then overlaid on each bar, showing the uncertainty in the index values.

Comparing the two columns of Figure 8.6 allows the effect of the two different prior distributions on the SA indices to be assessed. Under the log-uniform prior, more probability mass is placed on less stiff material parameter configurations than is the case for the uniform prior, which in turn leads to a significant tail in the distribution of the magnitudes of the resulting simulated displacements. This effect is illustrated in Figure 8.3. Nevertheless, the SA results under the two priors indicate qualitatively reasonable agreement. That is, inputs which have close to zero influence under one prior tend to have close to zero influence under the other, with the same agreement observed for highly influential inputs. The exception to this is the influence of EDP, which exhibits significantly less influence on each QoI under the log-uniform prior. This issue is discussed in more detail below.

Considering the sensitivity indices observed for the material stiffness input parameters, we can see that a, b, a_{f} and b_{f} are clearly the most influential on the output quantities of interest. For all analyses except ε^*_{cc} under the uniform prior, either a or b has the highest total-effect index, with the effect of a tending to be larger under the uniform prior, while b is generally more influential under the log-uniform prior. Parameters a_{f} and b_{f} tend to have lower, but still significant sensitivity indices, and for LVV under the log-uniform prior they are the third and second most influential inputs, respectively. By contrast, the total-effect sensitivity indices are always lower for a_{s}, b_{s}, a_{fs} and b_{fs} than the other material parameters, for each combination of output quantity and material prior distribution, and in many cases their indices values are very close to zero. Given that these four parameters have very little influence on the variation in the model outputs, if we now consider the inverse problem of trying to estimate their values given from noisy experimental data, they will be essentially non-identifiable. For this reason, in subsequent experiments these are fixed to values from the literature (Gao et al., 2017).

EDP has significant first-order and total-effect sensitivity indices for each output quan-

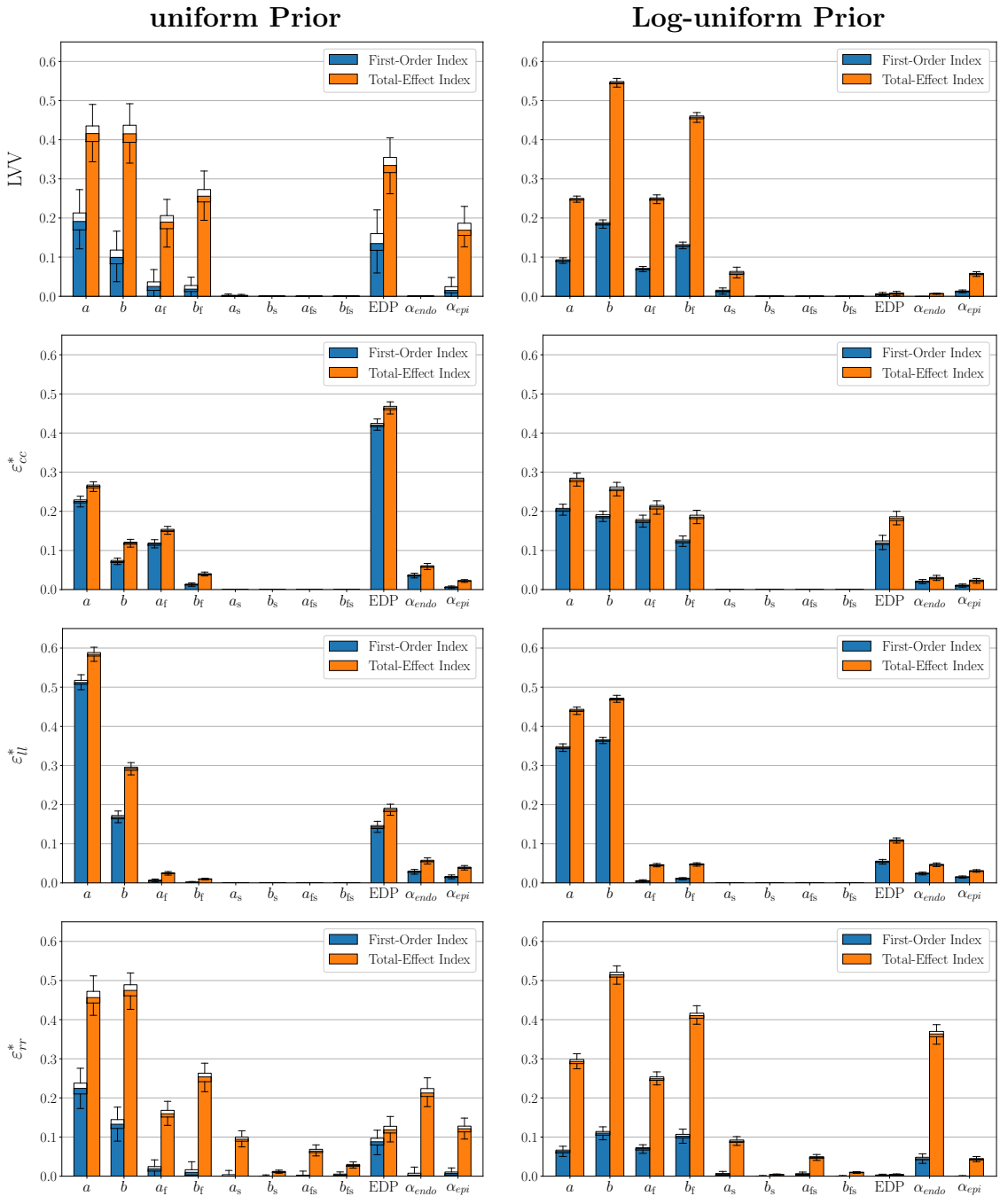


Figure 8.6: SA1 Results. The left column shows the SA results under the uniform material parameter prior, and the right column shows the results under the log-uniform prior. Each row corresponds to one of the four output QoI respectively.

tity under the uniform prior. However, under the log-uniform prior, the influence of EDP is lower for all outputs. For LVV and $\varepsilon^*_{\mathrm{rr}}$ in particular, the influence of EDP is close to zero. This is because the log-uniform prior places more weight on low material stiffness parameter configurations, as illustrated in Figure 8.3 (a), aggregated in the left bottom corner. For sufficiently low material stiffness parameter values, the magnitude of LVV

and $\varepsilon^*_{\text{rr}}$ values at the end of diastole are very high, across the entire range of EDP values considered in this study. As a result, the influence of EDP on observed variance for LVV and $\varepsilon^*_{\text{rr}}$ is suppressed by the large influence of the material stiffness parameters, resulting in Sobol indices that are close to zero. By contrast, $\varepsilon^*_{\text{cc}}$ and $\varepsilon^*_{\text{ll}}$ do not exhibit the same level of skewness under the log-uniform prior, meaning that the influence of EDP on these two QoI remains significant.

For both choices of prior, the fibre orientation angles α_{endo} and α_{epi} tended to have a low, but still clearly non-zero, impact on the end-diastolic output quantities. The exception to this was $\varepsilon^*_{\text{rr}}$ under the log-uniform prior. Here, α_{endo} shows a small first-order sensitivity index, but a large total-effect index.

Sensitivity Analysis Two

In SA2, EDP was not assumed to be a random variable. Instead, a range of fixed EDP values (5, 7.5, ... 25) mmHg were considered, and a separate SA was carried out for each fixed value. The objective of this experiment was to quantify how this fixed EDP value impacts the Sobol indices of the material parameters. This is because, for a given subject, EDP will not be a random variable over the range [4, 30] mmHg as in SA1, but will be a fixed value, which can in principle be measured. Two output QoI are analysed: LVV and $\varepsilon^*_{\text{cc}}$. Since the material parameters a_{s}, b_{s}, a_{fs} and b_{fs} exhibited very low influence on LVV and $\varepsilon^*_{\text{cc}}$ in SA1 (see Figure 8.6) these parameters were fixed in this study, and the analysis concentrates on the material parameters a, b, a_{f} and b_{f}. The fixed values are those published in (Gao et al., 2017), namely $a_{\text{s}} = 0.69\,\text{kPa}$, $b_{\text{s}} = 1.11$, $a_{\text{fs}} = 0.31\,\text{kPa}$ and $b_{\text{fs}} = 2.58$. In addition, because the material parameters are of primary interest here, the RBM parameters α_{endo} and α_{epi} are also considered fixed, to the values $-60°$ and $90°$, respectively.

The results of this experiment are displayed in Figure 8.7 for the uniform prior, and Figure 8.8 for the log-uniform prior. As in SA1, the GP sampling approach discussed at the end of Section 8.2.4 was used to generate an ensemble of total-effect Sobol index values at each fixed EDP value. The solid lines in Figures 8.7 and 8.8 show the ensemble mean values, while the dashed lines indicate 95% credible intervals (CIs). These results exhibit stability under the different choices of prior. That is, the effect of EDP on the Sobol indices of each material parameter is qualitatively similar for both the uniform and log-uniform prior distributions. For LVV under the uniform prior the effect of a decreases with rising EDP, while the effect of b increase slightly, whereas the sensitivity scores of a_{f} and b_{f} do not exhibit large variation as a function of EDP. Under the log-uniform prior, the influence of all four material parameters remain constant as a function of EDP. This is consistent with the results of SA1, where the Sobol indices of EDP for LVV were found to be close to zero with the log-uniform prior. For $\varepsilon^*_{\text{cc}}$, the effect of EDP on the sensitivity

indices of the material parameters tends to be larger than for LVV. Specifically, the total-effect scores of b and b_f on $\varepsilon^*_{\mathrm{cc}}$ *increase* as EDP rises, whereas the total-effect scores of a and a_f *decrease* with rising EDP.

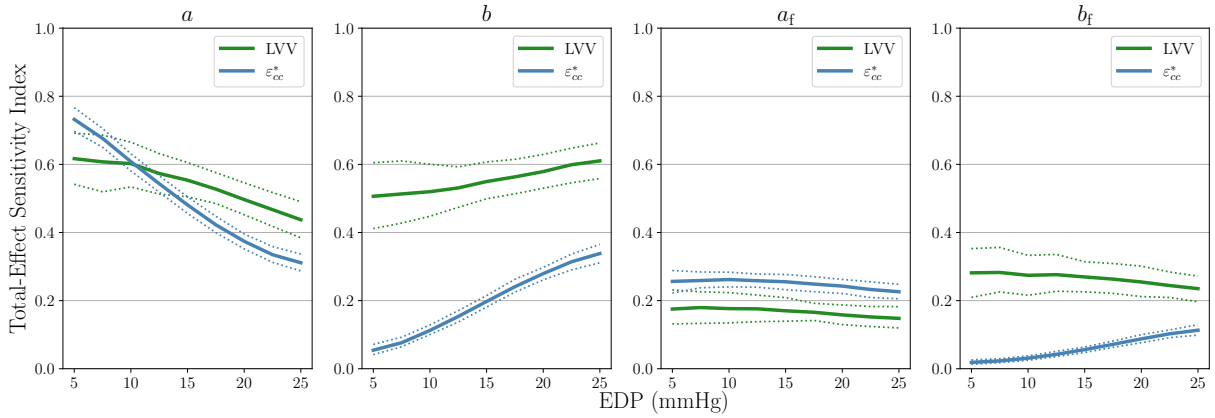


Figure 8.7: SA2 Results under the uniform material parameter prior. The dashed lines indicate 95% credible intervals.

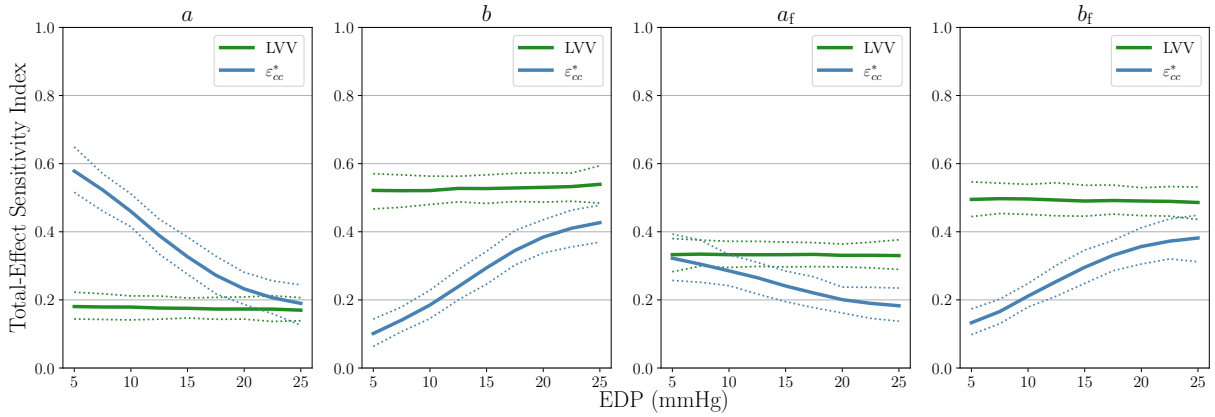


Figure 8.8: SA2 Results under the log-uniform material parameter prior. The dashed lines indicate 95% credible intervals.

8.4 I-UQ Simulation Studies

We can now use an I-UQ study to build on the results of the SA. The details of the study to be carried out are given in Section 8.4.1, the implementation and results of which are discussed in Section 8.4.2 and Section 8.4.4, respectively.

8.4.1 Experimental setup

Inputs and outputs of the emulator

As explained in Section 8.2.5, only the circumferential strains and LVV were included in the outputs of the I-UQ emulator. Motivated by the results of Section 8.3.4, a reduced parameterisation of the H-O model from (8.1) was adopted, where a_{s}, b_{s}, a_{fs} and b_{fs} were held fixed to values obtained from the literature (Gao et al., 2017). Additionally, the fibre angles were held fixed at $\alpha_{\mathrm{endo}} = -60^o$ and $\alpha_{\mathrm{epi}} = 90^o$, as discussed in Section 8.2.5. The resulting five dimensional input space, containing $a, b, a_{\mathrm{f}}, b_{\mathrm{f}}$ and P, is identical to the one considered in Section 8.3.4.

Test data generation

For the I-UQ, we require a set of test data to be used as observations for the posterior inference. A total of 100 test simulations were obtained at EDP of 5,10,15,20 and 25 mmHg. For each EDP, the same 100 unique material parameter configurations were used for the simulations, generated from a log-uniform distribution with bounds 0.1 and 10. Out of the 100 simulations, one led to an error in the simulator as a result of excessive distortion of the LV mesh causing numerical instabilities and lack of convergence of the numerical solver. For this reason, the final test set contained 99 data points. Of interest is I-UQ in the presence of noisy data, so Gaussian noise with standard deviation 5 ml was added to the simulated volumes and Gaussian noise with standard deviation 0.03 was added to the circumferential strains, motivated by results in the literature (Zhang et al., 2021). The volume standard deviation was informed by an empirical study that is provided in Table E.1. The use of different noise variances on volumes and circumferential strains is consistent with the noise model in (8.8).

8.4.2 Surrogate Model Training

As discussed in Section 8.2.5, the computational costs of the simulator make statistical emulation (see Section 8.2.4) a necessity for I-UQ in any reasonable time frame. For each of the 25 outputs (24 circumferential strains and end diastolic volume) an independent GP emulator is built, which requires a set of training data. The design for statistical emulation was discussed in Section 8.3.2 and, motivated by the discussion there, a Sobol sequence in log-material parameter space was used. In total, 2000 training simulations were used and the GP emulator was trained over the joint space of pressure and log-material parameters.

8.4.3 Surrogate model validation

The surrogate model was validated on a set of 100 points generated from a Sobol sequence in log-material parameter space. The Q^2 metric, which was discussed in Section 8.3.3, was used with the results provided in Table 8.4. Due to the reduction in dimension of the emulator input space the performance is slightly better than with the SA emulator in Table 8.3.

Table 8.4: **Surrogate Model Verification Results**: Q^2 coefficient value for the two outputs considered for the I-UQ experiments, calculated on a test-set of 100 simulations from the forward model. The values are rounded to three digits.

Output	Q^2
LVV	1.00
ε_{cc}	1.00

8.4.4 Results

I-UQ Results in Parameter Space

Of interest is the certainty with which we can infer the parameters and how this is affected by EDP. For each of the 99 test cases at all five values of pressure, the material parameters were sampled conditional on the synthetic observed data, generated according to Section 8.4.1. The MCMC samples are obtained in the joint space of the four material parameters, but to summarise the results the marginal posterior of each of the four parameters is considered. In the case of high practical identifiability, we expect a more peaked distribution, which can be quantified using the interquartile range (IQR). Taking the inverse of the IQR (denoted as I-IQR) provides a quantity that is analogous to the sensitivity index used in the SA, with large values indicating better practical identifiability of the parameters, as demonstrated in Figure 8.9. Since there are 99 different test cases, a set of 99 different I-IQR values at each of the five different pressure values is obtained, giving us five different distributions of I-IQR values as plotted in each subplot of Figure 8.10. In the plots of a and a_f, we see the general trend of the I-IQR decreasing as EDP increases. For b and b_f, the I-IQR of the marginal posterior distribution increases as EDP increases, suggesting that the practical identifiability improves. These results are in general agreement with the results of SA2 (see Section 8.3.4), as shown in Figure 8.11.

In Figure 8.10, we saw the change in the spread of $p(I_\theta|P)$ over P, where I_θ is the I-IQR of the marginal posterior distribution of θ and P is EDP. For a and a_f, the spread of this distribution decreases while for b and b_f, the spread increases as we increase P. In Figure 8.12, this is taken a step further, considering the distribution $p(I_\theta|P, \mathcal{V})$ where $\mathcal{V}$ is

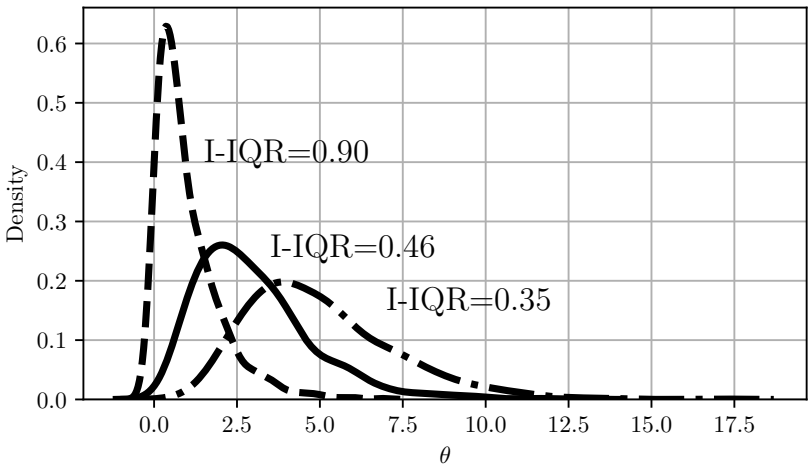


Figure 8.9: Demonstrating the I-IQR metric. For more peaked distributions (lower uncertainty), the I-IQR is larger and indicates improved practical identifiability.

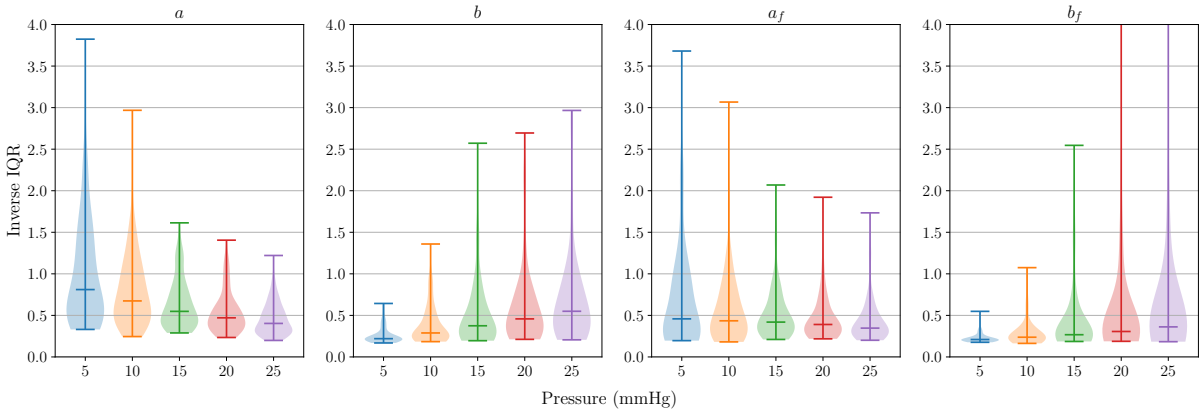


Figure 8.10: Distributions of I-IQRs of marginal posterior distributions, conditional on data obtained at each of the five different EDPs.

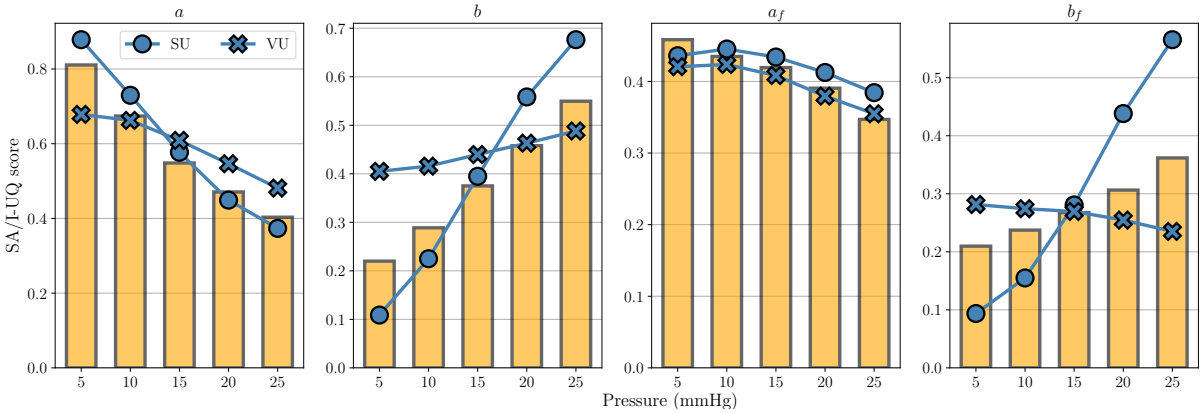


Figure 8.11: A comparison of SA and I-UQ. The I-UQ results are presented by orange bars showing the median I-IQR from the 99 test cases at each pressure. The SA results with the uniform prior (strains=SU and volume=VU) are provided as trend plots, allowing us to see the agreement between the results of the two studies.

the LVV, simulated at pressure 10 mmHg. To do so, the test cases were arranged in bins according to their LVV, with:

- v_1 : LVV < 102
- v_2 : $102 <$ LVV < 126
- v_3 : $126 <$ LVV < 153
- v_4 : LVV > 153.

These values were selected using the 0.25, 0.5 and 0.75 quantiles of the test set LVVs (at EDP=10 mmHg). The ith column of plots show the distributions $p(I_\theta|P, \mathcal{V} = v_i)$ over the range of pressures. That is to say that within subplot i, the different boxes show the distribution $p(I_\theta|P, \mathcal{V} = v_i)$, with the EDP indicated by the labels of the horizontal axes in the bottom row. Therefore, if we wish to compare the distributions $p(I_a|P = 15, \mathcal{V})$ over varying $\mathcal{V}$, we should compare the 3rd box in each subplot of the first row of Figure 8.12.

All the distributions in the first column of boxplots (corresponding to LVV<102 ml) have a low median value and very little spread. This indicates the difficulty of inferring the parameters when the material is very stiff. As we increase $\mathcal{V}$, the I-IQR gradually moves away from zero and the pattern in $p(I_\theta|P, \mathcal{V} = v_i)$ begins to resemble the pattern we saw in Figure 8.10. These plots suggest that the value of the parameters, which influences the LVV, impacts their identifiability. This feature was explored for the theta parameterisation from (4.2) in Section 6.3.3, and will now be considered for the new parameterisation.

Figure 8.13 shows 2D projections of the 99 test inputs from Section 8.4.1. These 2D projections show the value of a primary parameter (the first parameter in the plot title) in the horizontal axis and a secondary parameter (the second parameter in the title) in the vertical axis. The shading of the points indicates the value of the I-IQR of the marginal posterior density of the primary parameter, where the parameters have been inferred conditional on the test data obtained by simulating to EDP as indicated by the title of the subplot. In general, if the ground truth value of a parameter is large (moving left to right on the horizontal axis of a subplot) the I-IQR of the marginal posterior distribution is small. Inconsistencies in the pattern of I-IQR values suggests there are other factors influencing the practical identifiability and this will now be explored with several specific test cases.

To display some of the effects of parameter values on the I-UQ, several examples are highlighted in Figure 8.13. The different edge colours and shapes allow us to see the value of the material parameters for 5 different configurations, which are also given in Table 8.5. The results for these configurations are considered in more detail in Figure 8.14, where the colour and shape of the scatter points match those in Figure 8.13. $\boldsymbol{\theta}_4$ gives an example where all parameters clearly follow the general pattern seen in Figure 8.10. In the case of $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, the large value of b contributes to a low I-IQR of b_f at all pressures. This

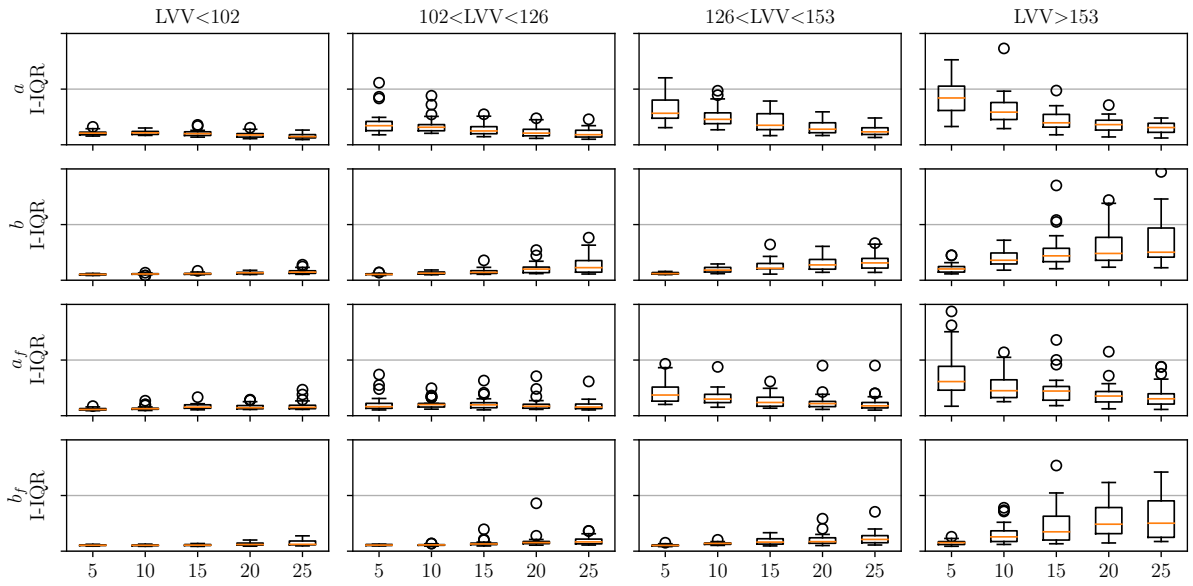


Figure 8.12: We can split up the I-IQR distributions based on LVV and consider the changes in the distribution of I-IQR with pressure and LVV. Each subplot shows the I-IQR of the marginal posterior for a particular material parameter (per row) for parameter configurations that give a simulated LVV at pressure 10 mmHg in a particular range (per column). For instance, the subplot in the second column from the left of the second row from the top provides the distribution of I-IQR values of the marginal posterior distributions of b for test parameter configurations with simulated LVV between 102 and 126 ml at pressure 10 mmHg. The limits of the vertical axes are all the same and have been removed because they are not required for interpretation of the plots.

effect of b on the inference of b_f at pressure 25 mmHg is also seen in the $b_\mathrm{f} - b$−P=25 plot in Figure 8.13, where all points in the upper left corner (b_f small and b large) are lightly coloured. In $\boldsymbol{\theta}_3$, we see an example where despite a being small, the I-IQR is fairly small. This shows the influence b and a_f can have on the identifiability of a. When we push the model to extreme configurations, such as $\boldsymbol{\theta}_5$ where a is large and all other parameters are small, we get strange patterns in the I-IQR distributions. In this case, the I-IQR of b and b_f is huge at high pressure and the I-IQR of a_f increases as pressure increases. This last feature is only observed in these more extreme cases, while a marginal posterior distribution for b and b_f with large I-IQR relies on these parameters being very small. Overall, these different examples show the dependence of the parameter identifiability on the values of the other parameters. In particular, this leads to a large spread of the distribution of I-IQRs of a and a_f at low pressure and b and b_f at large pressures.

I-UQ Results in Stress–Stretch Space

Any interpretation of inference results in terms of summaries of marginal posterior distributions is limited by the correlations that exist between the material parameters. In

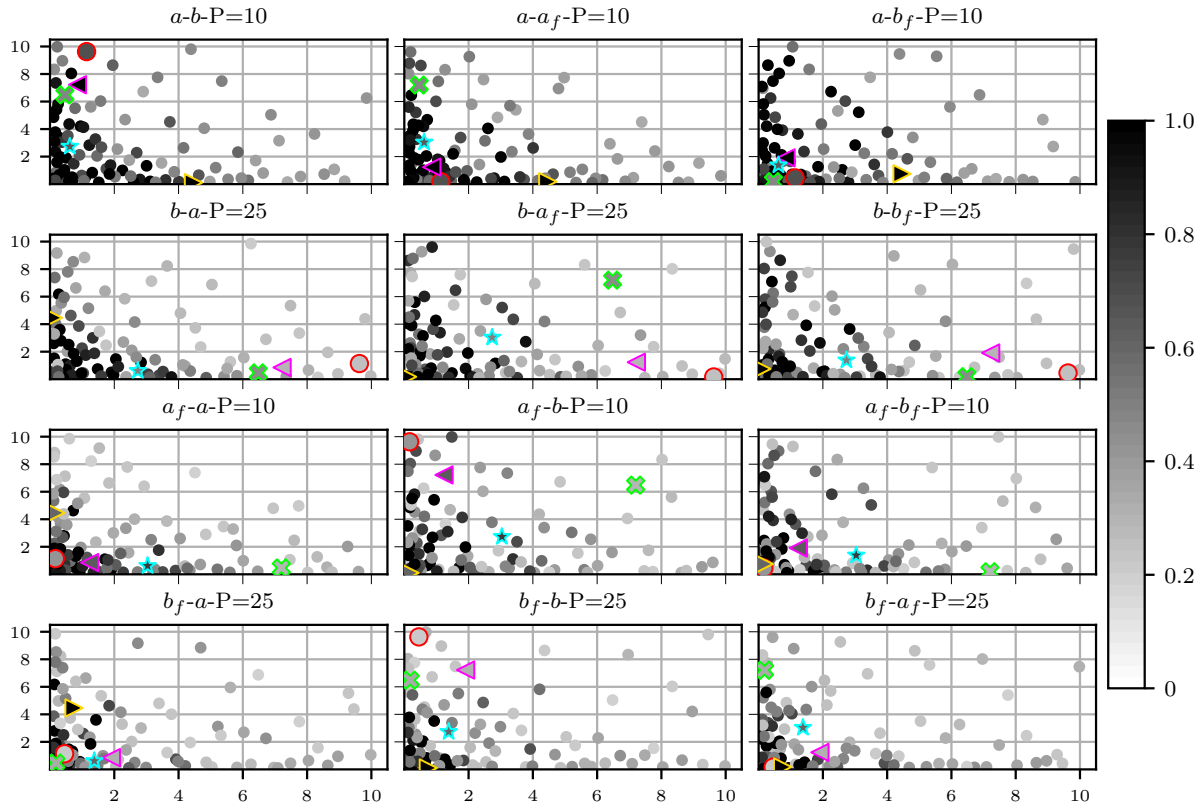


Figure 8.13: Scatter plot of test points in 2D, coloured based on I-IQR of the marginal posterior distribution. The title of each plot, $x - y - P = z$ gives the parameter of the horizontal axis (x), the parameter of the vertical axis (y) and the pressure at which the parameters are inferred (z). The points are shaded based on the I-IQR of the marginal posterior distribution of parameter x. Several points are highlighted (by colour and shape) to be looked at more extensively in proceeding visualisations in Figure 8.14, corresponding to the five configurations listed in Table 8.5. All subplots share the same bounds on the vertical and horizontal axes so these are only provided for the outer plots.

Table 8.5: The four parameter configurations being considered in greater detail.

Config	a (kPa)	b	a_f (kPa)	b_f
$\boldsymbol{\theta}_1$	1.13	9.63	0.16	0.45
$\boldsymbol{\theta}_2$	0.85	7.22	1.24	1.91
$\boldsymbol{\theta}_3$	0.46	6.48	7.20	0.18
$\boldsymbol{\theta}_4$	0.62	2.73	3.04	1.38
$\boldsymbol{\theta}_5$	4.46	0.10	0.18	0.75

an attempt to overcome this, the inferred tissue properties were also considered in stress–stretch space, where we simulate the deformation of a myocardial strip under a given set of material parameters from the H-O model (8.1). From a posterior sample of material parameters obtained using MCMC, we can determine a posterior distribution of stress–stretch curves, which is repeated for each of the 99 test cases at each of the five pressures. These distribution of curves are obtained by simulating the deformation of a sample of tis-

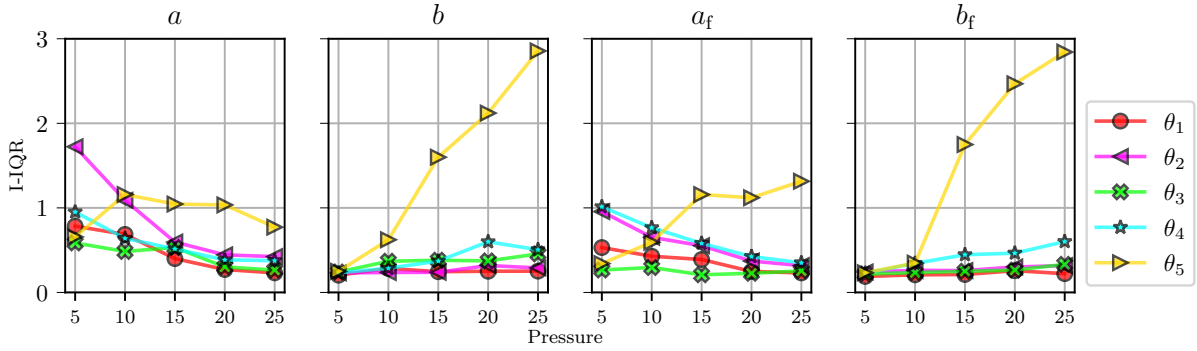


Figure 8.14: Plotting the change in I-IQR of the marginal posterior distributions over different EDP (mmHg) range for the test cases highlighted in Figure 8.13. The colours and shapes of the symbols match those in Figure 8.13, and the x-axis represents various EDP range.

sue as characterised by the H-O model (8.1). Repeating for each sample from the posterior distribution of parameters gives a posterior distribution of stress–stretch curves.

Firstly, we are interested in the uncertainty in the inferred tissue properties, which will be measured using the I-IQR of the stress distribution at given levels of stretch. From the 99 test cases we get a distribution of I-IQR at each level of stretch, as shown in Figure 8.15. The value decreases for all end diastolic pressures as the stretch location increases but the rate of this decrease is different for the different end diastolic pressures. This means that at low stretch the lowest uncertainty in the stress distribution was achieved using low pressure data (higher I-IQR scores, in general) while at high stretch, the lower uncertainty was achieved using high pressure data. The median inverse absolute error, plotted in Figure 8.16, shows a similar pattern: at low stretch, MAE is lowest for pressure 5 and highest for pressure 25. As we increase the stretch, the pattern gradually reverses.

8.5 Discussion and conclusion

The objective of this chapter has been to perform SA and I-UQ experimental studies of the passive mechanics of the left ventricle using the H-O myocardial model. As illustrated in Figure 8.1, these analyses are complementary; the SA quantifies how the structure of the model affects observed data (forward modelling effect), while the I-UQ quantifies how well-observed data can be used to identify the model structure (inverse modelling effect).

8.5.1 Discussion of SA

A total of eleven model input variables were considered across all studies: the eight material parameters of the H-O constitutive law, EDP, and two inputs to the RBM fibre generation algorithm. In addition, four different output quantities of interest were consid-

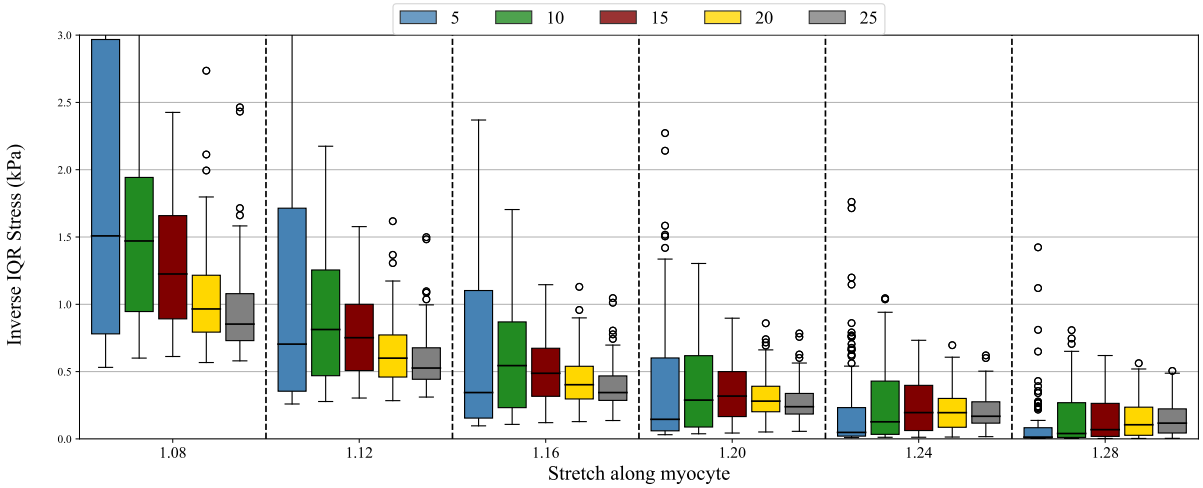


Figure 8.15: Representing our uncertainty about the tissue properties at different pressures. From each posterior distribution of material parameters, we obtain a distribution of stress-stretch curves (one curve for each sample). At different stretch locations, we calculate the inverse interquartile range of the stress distribution. Based on the 99 test cases, we get a distribution of these inverse interquartile ranges as found in the boxplots. This can be repeated for test data at different end diastolic pressures to assess the certainty of our estimation of the tissue properties at different pressures.

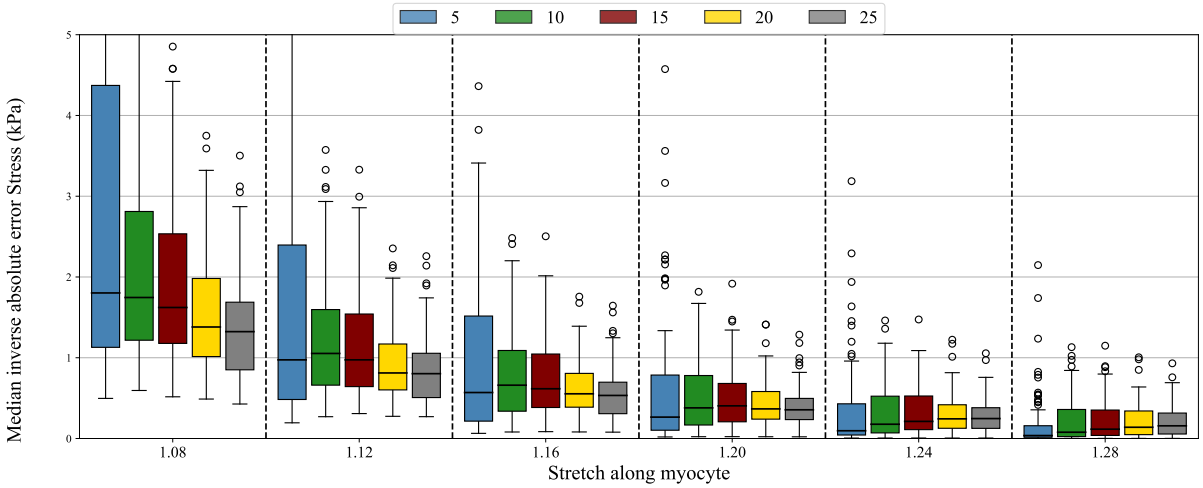


Figure 8.16: The error in the inferred tissue properties at different pressures. From each posterior distribution of the material parameters, we obtain a distribution of stress–stretch curves (one curve for each sample). We possess the ground truth stress-stretch curves and can obtain the median inverse absolute error in the distribution of stress–stretch curves. Based on the 99 test cases, we get a distribution of these median inverse absolute errors. This can be repeated for test data at different end diastolic pressures to assess the accuracy of our estimation of the tissue properties at different pressures.

ered in total: LVV, $\varepsilon^*_{\mathrm{cc}}$, $\varepsilon^*_{\mathrm{ll}}$ and $\varepsilon^*_{\mathrm{rr}}$, all measured at end diastole. These input and output quantities were discussed in Sections 8.2.2 and 8.2.2, respectively. All experiments were performed with respect to the LV geometry of a healthy volunteer. The SA experiments focused on assessing how uncertainty in each of the *a-priori* uncertain input variables

accounted to the uncertainty observed in the output quantities of interest. The I-UQ analysis then built on the results of the SA, to quantify how identifiable the most important material stiffness parameters were from noisy data, with emphasis on quantifying how these identifiability levels were affected by changing EDP values.

This study is the first to perform a comprehensive global sensitivity analysis and I-UQ of the H-O model in LV biomechanics. The clear takeaway from the results of the SA is that the material parameters a, b, a_{f} and b_{f} are of primary importance to the four output quantities of interest considered. While the relative importance of these parameters varies under the two different prior distributions, Figure 8.6 shows that they are consistently the most important inputs for each QoI. In contrast, the material parameters a_{s}, b_{s}, a_{fs}, b_{fs} have either zero or close to zero influence for each output quantity and prior distribution configuration. We see this low influence as justification for a low-order reparameterisation of the H-O model, where these parameters are set to fixed values, rather than discarding the two related terms in Equation (8.1), since such reduction may not fit the experimentally measured stretch-stress data well (Holzapfel and Ogden, 2009; Guan et al., 2019).

By using the Guccione myocardial model in LV diastolic filling, Rodriguez-Cantano et al. (2018) showed that the general stiffness parameter C has the highest sensitivity for their chosen quantities of interest, around 70%, much higher than other constitutive parameters. They further found that the total Sobol indices for the parameters associating with the fibre direction, the sheet direction and the shear response were clearly not zero. Similar results were reported in their subsequent study (Campos et al., 2019). While it was found that the parameters associated with the sheet direction and shear response, in general, have very low sensitivity, the parameters for the isotropic response and fibre reinforcement were found to have high sensitivity, see Figure 8.6. Those different findings may be due to different constitutive laws, QoI and parameter ranges used in the two studies.

Computational studies have shown that myofibre architecture, such as fibre angles, can have a substantial impact on LV mechanics, but not the sheet angle (Wang et al., 2013a; Palit et al., 2018b; Guan et al., 2021). For this reason, only the fibre rotation angles were included in the inputs in this study. These angles were found to generally have a low influence on the observed variations in the output quantities, but not zero in general, especially for radial strains. Similar results have been reported in (Rodriguez-Cantano et al., 2018). Fibre angles have been found to potentially have an even higher impact on myocardial active contraction (Campos et al., 2019). In the literature, nonrotationally symmetric distributions (Holzapfel et al., 2019), such as π-periodic von Mises distributions, have been used to model fibre dispersion in the myocardium (Guan et al., 2021). Due to limited measured data for describing local fibre distribution variations, the uncertainty in fibre dispersion was not included in the current study. Future sensitivity studies should

incorporate both uncertainties in the fibre rotation angles (i.e. a truncated Karhunen-Loeve expansion (Rodriguez-Cantano et al., 2018)) and fibre dispersion, in particular for myocardial contraction.

LV cavity pressure usually needs to be measured in an invasive way, so it is not available for most cardiac patients. Instead, many existing studies rely on population-based EDP (Gao et al., 2017; Genet et al., 2014b). LV EDP was found to have high importance for the uniform prior, while its influence declined under the log-uniform prior, as discussed in Section 8.3.4. The high sensitivity of EDP during the diastolic filling highlights the need for accurate measurement of EDP, in particular by using non-invasive approaches. One such recent development is using 4D cardiac CMR flow imaging to estimate realistic ventricular relative pressure (Marlevi et al., 2021). We also see from the results of SA2 in Figures 8.7 and 8.8 respectively that the influence of the material parameters a, b, a_f and b_f on circumferential strains can vary significantly as a function of the value of EDP when it is considered a known input. This has important implications when we consider the inverse problem of trying to estimate these parameters from experimental data for a given patient with a measured LV EDP.

8.5.2 Discussion on I-UQ

Considering the results of the I-UQ study, in Figure 8.10 we saw that as we vary EDP the uncertainty in the marginal posterior distributions changes as follows: the uncertainty of a and a_f *increases* as we increase pressure while the uncertainty of b and b_f *decreases*. This uncertainty was measured using the I-IQR metric (see Figure 8.9). This trend in identifiabilities over EDP agrees with the SA results from Figure 8.7, especially if we consider the fact that our inverse estimation is based on a set of measurements containing 24 circumferential strains and only one LVV, meaning that the circumferential strains carry extra weight in the likelihood function from (8.9). These results are also in line with our physical interpretation of the parameters, which is that a and a_f govern low stretch behaviour of the myocardium while b and b_f govern high stretch behaviour in the nonlinear regime.

The parameter space results of the I-UQ study showed that there is quite a large change in the spread of the distribution of I-IQR values, suggesting variation in the identifiability of the parameters. There were two reasons for this change. First of all, we must refer to Figure 8.12. Here it was shown that when pressure is large, a and a_f tend to be poorly identified regardless of the ground truth parameter values and when pressure is small b and b_f are poorly identified regardless of ground truth parameter value. This explains the low spread of the I-IQR distributions for b and b_f at low pressure and a and a_f at large pressure. Now, in Figure 8.13 it was shown that at high pressure, there are variations in the I-IQR of the marginal posteriors of b and b_f that result from the ground

truth value of the parameters. For a and a_{f}, this same figure showed a similar variation at low pressure. Another feature is that, in general, parameter inference in the case of large material parameters tends to lead to lower I-IQR values. These variations are a result of the nonstationarity discussed in Section 8.3.2, leading to improved parameter identifiability in the region where the function varies more rapidly. Overall, this leads to large spread in the distribution of I-IQR values for b and b_{f} at higher pressures, and high spread in the I-IQR distributions of a and a_{f} at lower pressure. For emulation purposes, a log-transformation was proposed to account for this nonstationarity in the GP model.

The interpretation of inference results in terms of marginal posterior distributions is confounded by the strong coupling that exists between the parameters of the H-O model. To overcome this, the results were also considered in stress-stretch space, providing a more direct measurement of tissue behaviour that takes into account the entire set of material parameters. Figure 8.15 showed the change in distributions of inverse IQRs for different EDP at different stretches. As we expect, the uncertainty in stress locations increases as the stretch increases and the pressure that provides the lowest uncertainty in the stress distribution increases. This pattern in the pressures is intuitive, since larger pressures give us information about the behaviour of the tissue at higher stretch locations. The stress–stretch curves were also used to assess the accuracy of our inference in Figure 8.16. These results show the same behaviour as the IQR results, with high pressure data giving lower error in the high stretch region.

8.5.3 Limitations

One limitation with the SA experiments presented in this study was the choice of an uninformative prior distribution for the model input parameters, in the absence of knowledge of the true population distribution. Future work could improve on this with a rejection sampling approach, whereby input parameter configurations which lead to unrealistic outputs, for example extremely large values of LVV, are discarded. A prior distribution obtained in this manner is likely to better approximate the true population distribution than those used in this study, which assumed independence between the inputs.

In the I-UQ study, the noise added to the circumferential strains and LVV makes an assumption of Gaussianity. Additionally, for the circumferential strains a fixed standard deviation on the noise was assumed, which may be an inaccurate assumption. For instance, the noise levels in a CMR scan at different regions of the LV can be different due to partial volume effects and wall motion, etc. These noise distributions would be worth studying in more detail in future work, allowing the generation of more realistic synthetic data.

In this study, only the diastolic filling of the LV biomechanics was considered with a limited set of quantities of interest. The reason for such a choice is because the passive property has very low influence on active contraction, as demonstrated in (Campos et al.,

2020), but highly determines the passive response of the myocardium during the diastolic filling process (Campos et al., 2019; Rodriguez-Cantano et al., 2018). The quantities of interest are chosen based on routinely available in vivo CMR measurements. For example, LVV and circumferential strains have been widely used for clinical diagnosis. Future studies shall extend the model to consider both passive filling and active contraction (Campos et al., 2020), and to include other measurements such as ejection fraction, the rate of systolic pressure increase, the torsion, etc., and further extend it to a coupled electromechanics full-heart model (Levrero-Florencio et al., 2020). Another limitation is that the geometrical uncertainty has not been considered, which can arise from the imaging protocol, the segmentation procedure and the reconstruction. The effect of LV geometry segmentation errors on parameter estimation will be explored in Chapter 9. For a detailed review of uncertainties associated with cardiac models, the reader is referred to (Mirams et al., 2016) and the presentations given at the workshops of "The Fickle heart" programme held at the Isaac Newton Institute, 2019 (Mirams et al., 2020).

8.5.4 Conclusion

The work in this chapter has provided a comprehensive sensitivity analysis of the H-O model, proposing a new parameterisation that has been studied using I-UQ. This involved the use of Bayesian inference, with a likelihood function approximated using a statistical emulator. To our knowledge, the final parameterisation is different from any previously proposed in the literature and, unlike the existing models, it comes with a quantitative justification from a sensitivity analysis. The results of the I-UQ study were found to be consistent with those of the SA. In particular, good identifiability of a and a_{f} was observed in the in vivo pressure range. These parameters are associated with the toe region of the myocardial mechanical response. It was also found that b and b_{f} cannot be reliably inferred without the inclusion of high pressure data. This should motivate future work on the inclusion of high-stretch behaviour of the tissue in the inference framework, using priors similar to those derived in Chapter 7. For these methods to be reliable in a clinical setting, this must all be done within a proper statistical inference framework, taking account of the uncertainties introduced by the data and the model.

Chapter 9

Studying the effects of physiological conditions on model calibration

The end goal of model calibration in mathematical models of cardiac function is to provide real world information on cardiac health. By linking the underlying behaviour of the heart with observable quantities, tuning these models to match a given dataset can provide insight into the system at a level that is naked to the human eye. While fundamentally informative, the utility of our parameter estimation depends on the accuracy of the mathematical model: if the model is imperfect, we may only ever obtain a representation in the parameters that is imperfect. Such imperfections—which can be the result of measurement error or suboptimal modelling— are an inevitability of the translation of mathematical models to the real world. Acknowledging these imperfections necessitates the use of statistical methodology that can correctly account for the discrepancy between the model and the data. This chapter will consider this task in the context of parameter estimation for data measured from cardiac magnetic resonance (CMR) imaging scans. In addition, the effect that operator errors during LV geometry segmentation have on the parameter estimation task will be explored. The LV operator study is joint with David Dalton and Arash Rabbani, who acted as the other two operators.

9.1 Model discrepancy background

9.1.1 Motivation with a set of measured data

Let us begin with an example that shows the typical results of parameter inference for measured LV data. Directly from CMR scans we can measure a set of 24 circumferential strains and the end diastolic volume of the cavity (LVV), as well as the LV geometry G, which has vector representation $\mathbf{g}$. The parameterisation of the H-O model used for this study is the one proposed by the SA in Chapter 8. Adopting the likelihood

function used in Section 8.2.5, predictions of the LVV and strains are provided by an emulator $\boldsymbol{\eta}(\boldsymbol{\theta}, \mathbf{h})$, where $\mathbf{h}$ is the 5 PC representation of $\mathbf{g}$. The material parameters a, b, a_{f} and b_{f} can be sampled from the posterior distribution under an assumption of iid zero-mean Gaussian noise on the circumferential strains and a noise standard deviation on the volume held fixed at 5 ml, which is consistent with the results of an empirical study that is found in Appendix E.1. The parameter sampling is performed using NUTS, before thinning to get the final sample to be used for the analysis. In Figure 9.1, the posterior distributions of $\boldsymbol{\eta}(\boldsymbol{\theta}, \mathbf{h})$, obtained by predicting with the emulator at the thinned material parameter samples, are plotted as solid black density curves for each model output. The vertical dashed line, showing the measured value of that output, allows us to identify any discrepancy between the posteriors and the observations. For comparative purposes, the inference task is repeated with a likelihood function that contains only the LVV, with the results displayed in the same plots using dashed density curves. Recall that strain measurements are negative, so a density lying to the right of the dashed vertical line shows an underestimation of the strain. For the volume (shown in the top left plot), the opposite is true: density to the left of the vertical dashed line indicates an underestimation of the LVV.

The solid density lines in Figure 9.1 (which result from inference conditional on volume and strains) present a familiar theme, and one that was touched on in Section 7.4.4: in order to match the measured strains, we must underestimate the volume. The word “match" is used rather loosely here because, as we can see in the plot, many of the densities do not match the observed values. The level of noise in the measured data is so large that the measurements span the majority of the space of possible strain values (in some cases between -0.05 and -0.25). In the presence of such high levels of noise, the best we can do is match the mean strain and this is always in disagreement with the LVV measurement, which can only be matched at parameter configurations that correspond to predicted circumferential strains that are of greater (in absolute value) than 0.2.

In Figure 9.2, the measured circumferential strains for 8 different test cases are compared with simulated circumferential strains obtained at material parameter values that allow us to match the LVV. Comparing the observations and simulations in each test case presents two consistent findings. The first of these is the high degree of noise in the measured data. In the majority of test cases, the IQR of the measured data is significantly larger than that of the simulated data. The second finding is that the simulated circumferential strains correspond to a greater deformation than the measured circumferential strains. This pattern suggests that some combination of the following is occurring: the measurement noise is not iid zero-mean Gaussian (related to the measurement process) or there is a systematic mismatch between the simulator and underlying physical process (related to the model building process). Each of these problems can manifest in a similar

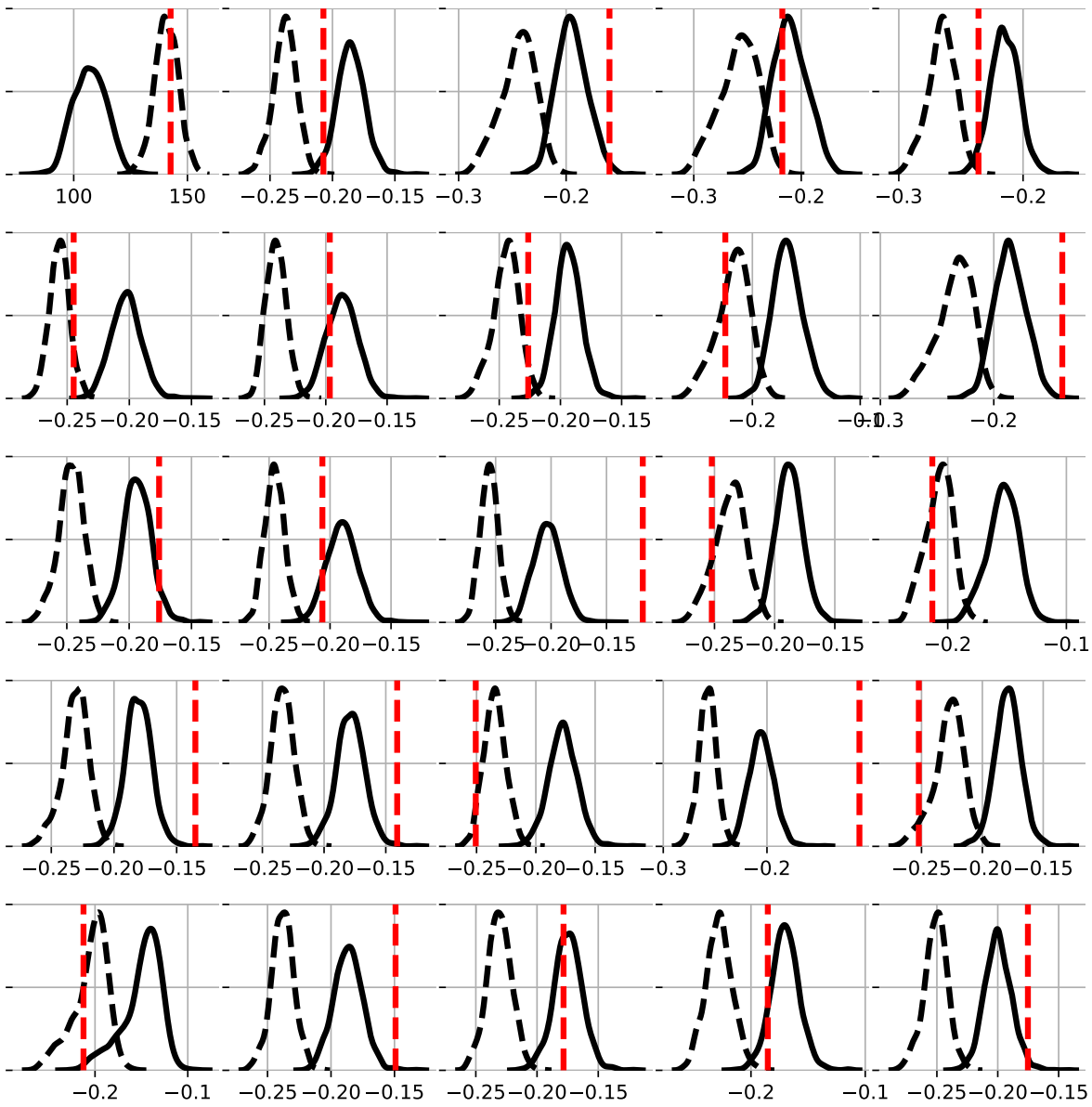


Figure 9.1: Posterior distributions of $\eta(\boldsymbol{\theta})$ for each of the 25 model outputs in two situations: material parameters sampled conditional on volume alone (dashed lines) and material parameters sampled conditional on the measured data. The upper left plot gives results for the LVV and the others show circumferential strains, with each plot containing two different predictive distributions. The solid line is the predictive distribution obtained when inferring the parameters conditional on both volume and circumferential strains while the dashed line is obtained from parameters estimated using only the volume measurement. The vertical dashed lines show the measured value of the output. Notice that the volume is underestimated in order to match the strain measurements.

way: through a departure of the residuals from normality. This means that their influence can be difficult to disentangle. To allow for a proper analysis of discrepancy introduced by the simulator, this chapter will use a series of synthetic data studies to study the effects that different components of the simulator can have on our parameter inference.

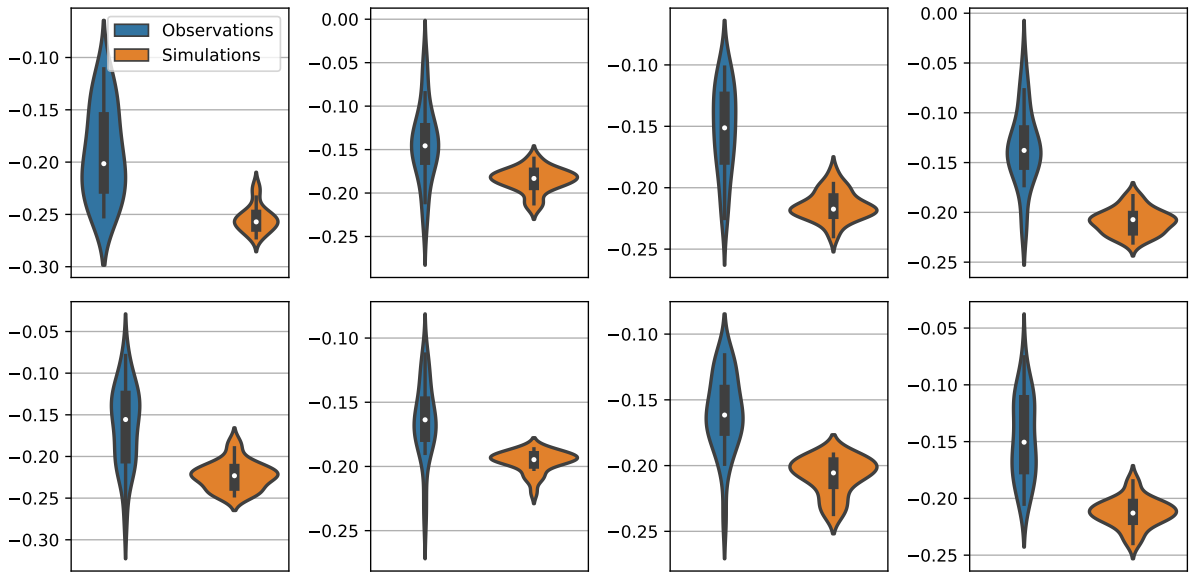


Figure 9.2: Each subplot compares the measured circumferential strain data for a single test case with simulated circumferential strains obtained at parameters that allow us to match the LVV measurement. In all plots the spread of the distributions of measurements is larger than that of the simulations and the median of the simulations is consistently below the median of the measurements.

9.1.2 The presence of model discrepancy in the LV simulator

For a new subject, with end diastolic pressure (EDP) P (either measured or assumed) and LV geometry $\mathbf{g}$, we may relate a set of measured data $\boldsymbol{y}^*$, obtained from a CMR scan of a subject's LV, with the true underlying strains and standardised LVV, gathered in the term $\boldsymbol{\psi}$:

$$\boldsymbol{y}^* = \boldsymbol{\psi} + \boldsymbol{\xi}, \tag{9.1}$$

When modelling the measured LV data, we approximate the unknown $\boldsymbol{\psi}$ with a simulator, $\zeta(\cdot)$. Similar to the exposition of Goldstein and Rougier (2009), we can consider this simulator as consisting of several parts. The *biomechanical model*, which includes the H-O model and the momentum balance equations, provides the system of equations to be solved during simulation. This is assumed to be fixed, selected by the mathematical modeller during the forward modelling. This model takes as input a set of *constitutive parameters*, $\boldsymbol{\theta}$, and depends on a set of conditions, $\mathbf{u}$, that must be selected in order to evaluate the simulator for a given set of parameters. These *physiological conditions*— which allow us to specify the model for a given subject— include the LV geometry, $\mathbf{g}$, and boundary conditions such as the EDP of the blood within the cavity at end of diastole. Finally, the simulator also requires a *solver*, which consists of the finite element solver and the discretisation of the LV geometry into a set of finite element nodes. Like the model,

the solver is fixed by the mathematical modeller and built into the simulator code. The simulator was discussed in greater detail in Section 2.2.

9.1.3 A closer look at the physiological conditions

Accurate model calibration depends on the accuracy with which we can represent the set of physiological conditions for a patient. In principle, these can all be measured in the clinical setting and included in the simulator. In practice, the absence of accurate non-invasive measurement methods, or the vulnerability of existing ones to operator subjectivity, introduces extra uncertainties to the modelling of the LV that decrease the confidence we have in the model and subsequent inferences. These sources of error, which will be explored in simulation studies later in this chapter, will now be discussed in more detail.

Fibre angles

The ventricular myocardium is built up of a stack of fibres, the orientation of which varies as we move from endocardium to epicardium. Advanced imaging techniques, such as diffusion tensor magnetic resonance imaging (DTMRI), have been used to show that the fibres in the myocardium follow a mean fibre direction, subject to some dispersion (random fluctuations around the mean direction) (Guan et al., 2021). This mean orientation can vary from patient to patient and is affected by ventricular remodelling (Palit et al., 2015). In the simulator used for this work, a Laplace Dirichlet rule based approach (Bayer et al., 2012; Wang et al., 2013a) is adopted for the fibre orientation, assuming that the mean fibre angle varies linearly between the endocardium and epicardium. The effect of these bounds was explored in the sensitivity study carried out in Chapter 8, but no consideration was given to how they could impact the inverse problem. Instead, fixed boundaries of $\alpha_{\text{endo}} = -60$ and $\alpha_{\text{epi}} = 90$ were used for the endocardium and epicardium, respectively.

The effect of fibre orientation on the estimation of passive tissue properties has been explored by Palit et al. (2018a), who inferred the passive properties while allowing for different angles in a rule based approach. They found that the fibre angles affected the parameter estimates when optimising based on an objective function that included the LVV[1]. More recently, Guan et al. (2021) studied the effect of fibre dispersion, which accounts for random perturbation of the fibre orientation around the mean fibre angle. They found that the dispersion especially affects the behaviour in the active contraction part of the cardiac cycle, but still had some effect on the passive filling process. Unfortunately, dispersion is still being explored by the mathematical modellers (Guan et al., 2021) and could not be included in the current simulator code. For this reason, it will not be explored in this work.

[1]This objective included LVV and an extra term measuring adherence to the Klotz curve, which was discussed extensively in Chapter 7

Pressure

Easily accounted for in the emulator, as done in Chapter 8, is the EDP of the blood within the LV cavity. Problematic, however, is the invasiveness of measurement methods, which means that we can only rely on measured values when dealing with diseased patients. For healthy volunteers, a value is usually assumed, which in most of the work in this thesis has meant setting the EDP to a value of 8 mmHg. This value, motivated by Bouchard et al. (1971), inevitably introduces some bias to the parameter estimation. However, the work of Palit et al. (2018a) showed that the parameter estimates were fairly robust when varying pressure around values that we expect in healthy volunteers (between 8 and 12 mmHg).

LV geometry

Calibrating the LV simulator to a particular patient depends, importantly, on the LV geometry. Unlike pressure and fibre angles, non-invasive methods currently exist for measuring the LV geometry in vivo, using the procedures outlined in Section 5.2.1. In previous chapters, a PCA representation of the LV geometry was adopted, while acknowledging in Chapter 6 that the failure of this method to take the biomechanical model into account makes it suboptimal for LV geometry representation in the simulator. This idea of the LV geometry representation being suboptimal was built on the premise that the LV geometry extracted from a CMR scan can be treated as a ground truth. However, in reality this makes little sense because the final shape extracted from the CMR scan will be subject to both inter and intra observer (operator) errors. As a result, the effect that these different operators have on the biomechanical model is of interest for two reasons. Firstly, the variation introduced to the inferred cardio-mechanics as a result of operators is of interest. Secondly, the utility of a PCA basis for representing the LV geometry in the simulator depends critically on its comparison with the performance of different operators: if LV modelling errors incurred using a PCA representation of the LV geometry are lower than those incurred with a new operator then PCA is a sufficient representation of the LV geometry. At the time of this work the effect of operators on the mechanical modelling of the LV had not been explored, but recently this was carried out by Hasaballa et al. (2021) who found that the operator could have a substantial effect on the results of the parameter estimation. However, they did not benchmark this operator effect against the performance of PCA. In this chapter, it will be shown that errors introduced to parameter estimation as a result of PCA are in line with the error introduced by a new operator.

9.1.4 Effect of erroneous physiological conditions and further modelling errors

It is worth considering the consequences of erroneous physiological conditions on the noise model from (9.1). Assuming that the simulator has the capacity to perfectly replicate the underlying behaviour of the real world, we can take the ground truth constitutive parameters, $\tilde{\boldsymbol{\theta}}$, and physiological conditions, $\tilde{\mathbf{u}}$, to be those that provide the perfect fit to the true underlying process. The simulator can be incorporated into (9.1) by replacing $\boldsymbol{\psi}$ with ζ evaluated at this parameter combination:

$$\boldsymbol{y}^* = \zeta(\tilde{\boldsymbol{\theta}}, \tilde{\mathbf{u}}) + \boldsymbol{\xi}, \tag{9.2}$$

Note that the LV geometry $\mathbf{g}$ has been absorbed by $\tilde{\mathbf{u}}$ because the LV geometry is also considered as a physiological condition. The noise model is the same as before due to the perfect correspondence between simulator and reality. In any use of the simulator model, we rely on measured or assumed values for $\mathbf{u}$, which we may represent as $\hat{\mathbf{u}}$. When these are measured, such as the case of the LV geometry, they will be subject to measurement error (or possibly approximation error in the case of the statistical emulator with PCA) and when they are assumed known and fixed to population based values, they will be subject to some bias. In the presence of these suboptimal modelling conditions, the best fitting constitutive parameters will no longer be the same as the "ground truth" values. At the new best fitting parameters $\hat{\boldsymbol{\theta}}$, we could argue that it no longer makes sense to think of the noise model the same as before, instead considering the following model:

$$\boldsymbol{y}^* = \zeta(\hat{\boldsymbol{\theta}}, \hat{\mathbf{u}}) + \boldsymbol{\xi}^* \tag{9.3}$$

where $\boldsymbol{\xi}^*$ will often not have the same variance and may no longer satisfy the independence assumption. For instance, in the presence of incorrect fibre angles we might expect a relationship between the error in two neighbouring regions of the model. Critically, if we were to assume the same noise model as was considered in (9.2), then a bias would be introduced to the estimated parameters.

It turns out that the inconsistencies between the model and reality are more fundamental than erroneous physiological conditions. Built into all parts of the simulator are several simplifying assumptions and chief among these is the model itself. The H-O model is not a fundamental law of nature to which the myocardium abides. Instead, it is a mathematical description of the myocardium, constructed by modellers and validated against experimental data to prove its ability to match the behaviour of the myocardium tissue at different ranges of stretch (Holzapfel and Ogden, 2009). As such, we can be confident that, although accurate, this description of the tissue will not perfectly replicate the en-

ergy build-up in the myocardium. A further inconsistency is in our use of this law in the simulator, where we make the assumption that the myocardium tissue can be described by homogeneous material properties. Even for healthy tissues we would expect this assumption is violated, but any introduction of heterogeneity to the model is highly complex and methods do not currently exist to allow for it in the simulator. This reality of heterogeneity does introduce the question of how we can justify an interpretation of homogeneous tissue properties, especially if we do not account for any resulting mismatch in the simulator. A final limiting assumption is that the model considers the LV in isolation. In reality, the LV motion at the septum (see Figure 2.3) will be limited by the presence of the right ventricle (RV). Using a bi-ventricular model would better resemble the boundary conditions on the wall of the LV, without which we will incur a discrepancy between the reality and the simulator, as studied by Palit et al. (2015).

9.2 Methodology

In Figures 9.1 and 9.2, we saw evidence of systematic mismatch between the simulator and reality. The possible sources of this discrepancy, specifically related to the mathematical model, were discussed in Section 9.1.2. This section will present an approach that allow us to acknowledge this discrepancy in our noise model, largely based on the methods for parameter calibration in the presence of model discrepancy that were discussed in Section 3.2.7.

9.2.1 Acknowledging model discrepancy in Bayesian model calibration

Consider the situation where we wish to infer the material parameters from measured data $\boldsymbol{y}^*$, with an LV geometry vector $\mathbf{g}$ that has a 5 PC projection $\mathbf{h}$. In order to acknowledge model discrepancy, a likelihood of the following form will be considered:

$$p(\boldsymbol{y}^*|\boldsymbol{\theta}) = l\mathrm{V}(\boldsymbol{\theta}) - \frac{k}{2}\log(\sigma_{\mathrm{S}}^2) - \frac{1}{2\sigma_{\mathrm{S}}^2}\sum_{i=1}^{k}(y_{\mathrm{i}}^* - (\eta_i(\boldsymbol{\theta}, \mathbf{h}) + b_i))^2, \tag{9.4}$$

where $\eta_i(\boldsymbol{\theta}, \mathbf{h})$ is the emulator prediction for output i at parameters $\boldsymbol{\theta}$ and 5 PC representation $\mathbf{h}$. The first term, which is the log-likelihood term associated with the volume measurement, has the following form:

$$l\mathrm{V}(\boldsymbol{\theta}) = -\frac{1}{2}\log(\sigma_V^2) - \frac{1}{2\sigma_V^2}(y_0^* - \eta_0(\boldsymbol{\theta}, \mathbf{h}))^2, \tag{9.5}$$

and σ_{S}^2 is the variance associated with the strain measurements. The volume variance σ_V^2, which is non-identifiable given only one volume measurement, will be fixed during inference to a value that is consistent with an operator error study (see Table E.1) while the strain variance will be inferred along with the constitutive parameters. Within the likelihood in (9.4), an extra term b_i has been included. This extra parameter, which must be inferred along with the constitutive parameters, adds an offset to the prediction for strain i, making the strain more consistent with the volume. To summarise: the model in (9.4) assumes that there is no systematic mismatch between the predicted and measured LVV, which is necessary because of the absence of prior knowledge about the model discrepancy. Moreover, in the presence of only one volume measurement we cannot expect to learn a model discrepancy term on this quantity.

9.2.2 GP model for circumferential strain discrepancy

We have 24 circumferential strains, each of which will be subject to a separate bias correction. Modelling the corrections independently would lead to overfitting and a model that is non-identifiable. To overcome this, we must model some dependence between the correction variables, encouraging smoothness that will help to limit the flexibility of the function. Following the work of Brynjarsdóttir and O'Hagan (2014), this can be done by introducing a GP prior on the discrepancy function, allowing for the correction to be jointly sampled with the constitutive parameters according to the following likelihood function:

$$p(\boldsymbol{y}^*|\boldsymbol{\theta}) = l\mathrm{V}(\boldsymbol{\theta}) - \frac{k}{2}\log(\sigma_{\mathrm{S}}^2) - \frac{1}{2\sigma_{\mathrm{S}}^2}\sum_{i=1}^{k}(y_{\mathrm{i}}^* - (\eta_i(\boldsymbol{\theta}, \mathbf{h}) + b_i(\mathbf{x}) + v))^2, \tag{9.6}$$

where the parameter v is needed due to a zero-mean assumption on the GP. Assigning a GP prior and treating $\mathbf{b}$ as a function, we require a domain $\mathcal{X}$ that defines the space over which the discrepancy varies. On this domain we can then specify a kernel function that provides the covariance between realisations of $\mathbf{b}$ at two separate points, as required for the GP prior. For identifiability purposes, we require that $\mathcal{X}$ is independent of $\boldsymbol{\theta}$.

To account for model discrepancy in the LV problem, the domain of $\mathbf{b}$ is assumed to be the strain location. The location of the strains on the LV wall was visualised in Figure 2.3, where the colouring indicated a different circumferential positioning. One possible choice for $\mathcal{X}$ is to consider the strain values defined over 1D space, with an ordering that accounts for their location on the LV surface. However, any ordering we consider here will be suboptimal because we cannot fully define the positioning of 3D strain locations in only 1 dimension. Instead, we could use 3D coordinates in Euclidean space to define the positioning of the strain locations and use a standard GP kernel, like

the squared exponential, to define the correlations between locations. The issue with such an approach is that the distance measure can jump across the centre of the cavity and misrepresent the distance between two strain locations. Ideally, we want to account for the circular shape of each short axis slice, for which we require a GP kernel defined on polar coordinates.

Let us assume the strain measurements lie on a grid in two dimensional cylindrical coordinates with a circumferential coordinate from $\{0, \frac{\pi}{3}, \frac{2\pi}{3}, \pi, \frac{4\pi}{3}, \frac{5\pi}{3}\}$ and z coordinate from $\{0, 1, 2, 3\}$. Thinking first about how to define the correlation in the circumferential direction, we can consider the geodesic distance between two regions, which is given by $d(\gamma, \gamma') = \text{acos}(\cos(\gamma - \gamma'))$ where γ and γ' are the circumferential positioning of the two regions of interest. Naturally, we can apply the polar GP kernel that was discussed in Section 3.4.2 to specify correlations in this domain. In the z-direction, the strain locations are defined in Euclidean space and we can use standard kernel functions like squared exponential or Matérn class kernels. Combining this kernel in the z-direction and the polar kernel in the circumferential direction provides us with a kernel function defined on the surface of a cylinder. These can be combined additively or multiplicatively if we wish to account for interactions between the two dimensions. Multiplying two kernels can be thought of as an AND operation. That is, if you multiply together two kernels, then the resulting kernel will have high value only if both of the two base kernels have a high value. Adding two kernels can be thought of as an OR operation. That is, if you add together two kernels, then the resulting kernel will have high value if either of the two base kernels have a high value. In this work, an additive structure will be assumed as it was found to lead to smoother corrective functions in preliminary runs. The final kernel has the following form:

$$k(\{\gamma, z\}, \{\gamma', z'\}) = \sigma_{f_1}^2 W_c(d(\gamma, \gamma'); c, \tau) + \sigma_{f_2}^2 k_z(z, z'; \lambda) \tag{9.7}$$

with the hyperparameters of the model summarised in Table 9.1.

9.2.3 Inference for the model with discrepancy correction

Inference will be carried out using MCMC. Specifically, a HMC within Gibbs sampler will be adopted, where sampling of the hyperparameters is performed adjacent to sampling of the main model parameters. Table 9.1 provides a summary of the parameters present in the model along with the corresponding prior that is used during inference. The bounds for the uniform priors on $\boldsymbol{\theta}$ were as follows: $a, a_{\text{f}} \in [0.05, 10]$ and $b, b_{\text{f}} \in [0.1, 30]$.

The posterior of the discrepancy function $\mathbf{b}$ is highly correlated due to the multivariate normal prior that results from the assumption of a GP. To improve efficiency of sampling,

Table 9.1: Description of all parameters from the model in (9.6) along with priors used in Bayesian inference. IG represents an inverse Gamma distribution.

Parameter	Description	Prior ($\pi(\cdot)$)	Rationale
σ_S^2	Noise variance of strains	IG(0.001,0.001)	Conjugate and vague
$\boldsymbol{\theta}$	Material parameters $(a, b, a_{\mathrm{f}}, b_{\mathrm{f}})$	Uniform	Physiological bounds
v	Global bias of strains	Normal(0.05)	Knowledge of small global discrepancy
λ	Lengthscale of z-component of kernel function	Gamma(2,2)	Encourage smoother functions
$\sigma_{f_1}^2$	Signal variance of polar kernel	IG(0.001,0.001)	Vague
$\sigma_{f_2}^2$	Signal variance of z-direction kernel	IG(0.001,0.001)	Vague
c	Compact support parameter of polar kernel	u$(0, \pi)$	Physical bound
τ	Smoothness parameter of polar kernel	u$(4, 10)$	Encourage smoother functions

this is sampled in the whitened space:

$$\tilde{\mathbf{b}} = \mathbf{L}^{-1}\mathbf{b} \tag{9.8}$$

where $\mathbf{b} \sim \mathrm{GP}(\mathbf{0}, \mathbf{K})$ and $\mathbf{K} = \mathbf{L}\mathbf{L}^\top$ is the Cholesky decomposition. This method was used by Kuss and Rasmussen (2005) to improve sampling efficiency, and depends on the selection of the hyperparameters to determine the matrix $\mathbf{L}$. Due to the Gibbs step, we only compute the Cholesky decomposition once per MCMC sample because the GP hyperparameters have been sampled before the model parameters. The procedure is outlined in Algorithm 4.

Algorithm 4 HMC within Gibbs for the polar GP model.

1: Sample from $p(\sigma_S^2|\boldsymbol{\theta}, v, \lambda, \sigma_f^2, c, \tau, \mathbf{b}, \boldsymbol{y}^*)$, which is possible in closed form given conjugate inverse Gamma prior.
2: Sample from $p(\lambda, c, \tau, \sigma_{f_1}^2, \sigma_{f_2}^2|\boldsymbol{y}^*, \boldsymbol{\theta}, \sigma^2, \mathbf{b}, v)$ using HMC.
3: Calculate updated covariance matrix $\mathbf{K}$ and corresponding Cholesky decomposition: $\mathbf{K} = \mathbf{L}\mathbf{L}^\top$.
4: Sample from $p(\boldsymbol{\theta}, \mathbf{b}, v|\boldsymbol{y}^*, \sigma_S^2, \lambda, c, \tau, \sigma_{f_1}^2, \sigma_{f_2}^2)$ using HMC, where $\mathbf{b}$ is sampled in whitened space.

The posterior distributions used in steps 2 and 3 of Algorithm 4 have the following

form:

$$p(\lambda, c, \tau, \sigma_f^2 | \boldsymbol{y}^*, \boldsymbol{\theta}.\sigma_S^2, \mathbf{b}, v) \propto \mathcal{MVN}(\mathbf{b}|0, \mathbf{K}(\lambda, c, \tau, \sigma_{f_1}^2, \sigma_{f_2}^2))\pi(\lambda)\pi(c)\pi(\tau)\pi(\sigma_f^2)$$
$$p(\boldsymbol{\theta}, \mathbf{b}, v | \boldsymbol{y}^*, \sigma_S^2, \lambda, c, \tau, \sigma_{f_1}^2, \sigma_{f_2}^2) \propto p(\boldsymbol{y}^* | \boldsymbol{\theta}, \mathbf{b}, v)\mathcal{MVN}(\mathbf{b}|0, \mathbf{K}(\lambda, c, \tau, \sigma_{f_1}^2, \sigma_{f_2}^2))\pi(v)$$

where the various prior distributions, represented by $\pi(\cdot)$, are found in Table 9.1 and $\mathbf{K}(\lambda, c, \tau, \sigma_{f_1}^2, \sigma_{f_2}^2)$ is the GP covariance matrix.

9.2.4 The emulator model

Based on the results from Chapter 8, a four dimensional parameterisation of the H-O model is adopted where $a, b, a_\mathrm{f}, b_\mathrm{f}$ are considered as random variables while $a_\mathrm{s}, b_\mathrm{s}, a_\mathrm{fs}, b_\mathrm{fs}$ are fixed to population based values from the literature (Gao et al., 2017). To allow for variations in the LV geometry, a 5PC representation of the LV geometry is included in the input space of the emulator, resulting in a 9 dimensional regression problem. The same simulator outputs as previous chapters were used: 24 circumferential strains and the LVV. Essentially, this emulator is the same as the one from Chapter 6 but a different parameterisation of the H-O model is adopted.

In Chapter 6, a Sobol sequence was used to generate the training data for the emulator, despite the evidence from Chapter 4 that the function is nonstationary. In Chapter 8, it was shown that a warping of input space provided by a log transformation performed similarly to more flexible neural network transformations of the input space, motivating the use of a Sobol sequence on log-uniform space. This method will be used to generate the training data for the current emulator. In total, 30,000 simulations were obtained for training the emulator model and, as in Chapter 6, a neural network was used. To decide on a final network, different combinations of 50,100,120 and 200 nodes were used in two hidden layers, each containing tanh activation functions. The optimal architecture was selected based on the MSE on a validation set of 3000 simulations. Training was performed using the Adam optimiser with a decaying learning rate starting at 0.01. As with other neural network models in this thesis, the validation set also allowed for experimentation with different batch sizes. Based on previous results, the weights were initialised using samples from a $\mathcal{N}(0, 0.1)$ distribution.

9.3 Implementation

Section 9.1.2 discussed the idea of discrepancy as a result of physiological conditions. These included the LV geometry, the fibre orientation and the EDP. In order to explore the effects of these on our model calibration, we can generate synthetic data by running the simulator at a series of different physiological conditions. Several simulation studies

to determine the effect of physiological conditions on the model calibration will now be outlined.

9.3.1 Joint optimisation of fibre angles and constitutive parameters

The data in Figure 9.1 are subject to a substantial degree of noise. Despite this, we could see that the model and reality are in disagreement: in order to match the volume, most of the circumferential strains must be overestimated. As we learnt in Section 9.1.2, there are several sources of error in the model related to assumptions being made in the physiological conditions and the typical approach taken in this thesis was to fix the physiological conditions to sensible values based on knowledge from the literature. In light of this, it is worth questioning whether we can get closer to the measured data by estimating the physiological conditions from the data. Not all conditions can be considered here: due to its high-dimensional representation the LV geometry is difficult to tune to the data, and jointly inferring EDP and material parameters is an ill-posed problem. On the other hand, the fibre angles can be represented in a two dimensional space and, until now, their effect on the parameter estimation has not been thoroughly explored. In this study, the angles will be jointly optimised with the material parameters using Bayesian optimisation (BO) (Shahriari et al., 2016), which has been used to optimise the material parameters by Romaszko et al. (2021). In doing so, we treat the angles as tuning parameters, and their value is only a secondary concern. Primarily, we are hypothesising that the fixing of the fibre angles represents a primary source of discrepancy between the model and reality and wish to see whether treating these parameters as tunable can help us to better match the LVV and circumferential strains. This will be investigated for each of the 8 measured test sets.

The general idea in BO is to replace the iterative optimisation of an objective function $f(\boldsymbol{\theta})$, with iterative optimisation of an acquisition function $A(\boldsymbol{\theta})$. Starting with a set of objective function evaluations $\{\boldsymbol{\theta}_i, f(\boldsymbol{\theta}_i)\}_{i=1}^{n}$, a GP (or similar regression model) is fitted to approximate f. Using information from the fitted GP, an acquisition function is defined that will be optimised to select a new query point at which to evaluate the objective function. This acquisition function typically has a multi-modal landscape, but is very computationally cheap to evaluate and optimise. Once the query point is added to the training set $\{\boldsymbol{\theta}_i, f(\boldsymbol{\theta}_i)\}_{i=1}^{n+1}$, the GP is updated. Updating the GP that is fitted to the objective function evaluations also updates the acquisition function, which can then be reoptimised to find the next query point. Importantly, the form of $A(\boldsymbol{\theta})$ is such that there is a trade-off between exploring unseen locations in input space, and exploiting areas where we think the objective function is low (assuming minimisation). The fact that the acquisition function encourages exploration as well as exploitation means that

the crudeness of the objective function approximation, especially at early stages, is not detrimental. As long as the posterior distribution of the function correctly represents our state of belief, we will explore areas where the uncertainty is high and exploit areas where the posterior mean is low. In particular, the algorithm will initially explore the input space, before exploiting as the fitted GP becomes more confident. Further details can be found in the literature (Shahriari et al., 2016).

To use BO in this study, an initial set of 300 points (from a LHD) were run in parallel before initialising the BO algorithm. The benefit of such a large number of initial runs is that we can easily exploit parallelisation at this stage of the algorithm. This LHD gives a matrix of design points $\mathbf{X}$ that is $300 \times p$, where p is 4 for the fixed physiological condition runs (only material parameter space) and 6 for the optimised angles runs (material parameters plus two angles). To get the initial design for BO $\tilde{\mathbf{X}}$, the first 4 dimensions of $\mathbf{X}$ are transformed as follows:

$$\tilde{\mathbf{x}}_i = \exp(\mathbf{x}_i)$$

and rescaled to $[0.1, 10]$, which was the physiological range of the parameters in this study. As we see in Figure 9.3, this newly transformed design (labelled as the exp design) is a compromise between the highly skewed log-uniform design and the uniform design.

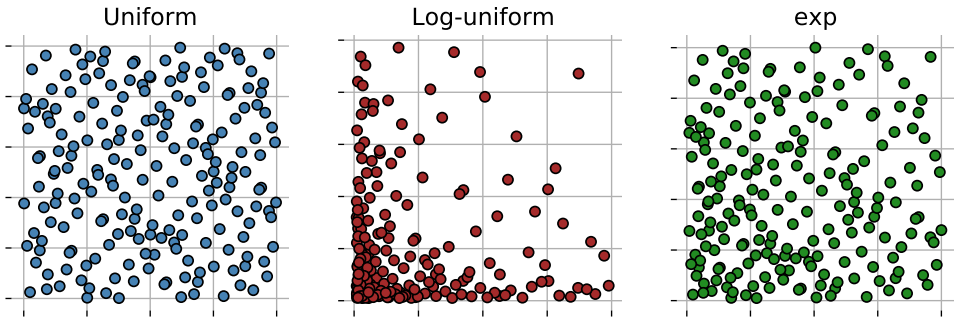


Figure 9.3: Comparing the different designs. The log-uniform design is better for building an emulator because most points are in the highly varying portion of the function, but many of these parameter values will not be physiologically realistic. The exp design used in this chapter is a compromise between the standard Sobol sequence and the Sobol sequence in log-Uniform space.

From the initial points, a set of simulations $\mathbf{Y}$ was obtained, and the LVV was standardised using the early diastole volume. The mean and standard deviation of these simulations (one for each of the 25 outputs) were used to standardise the initial data and the test data. The residual sum of squares was used as the objective function, but preliminary runs suggested that with volume and strains equally weighted there was not enough incentive to match the volume. Therefore, the contribution of the LVV to the RSS was

upweighted by a factor of 2, ensuring that the optimiser tries to match the LVV as well as the circumferential strains. The BO was run for 300 iterations after the initial set of data were generated, with the lower confidence bound acquisition function (Shahriari et al., 2016) used to propose the next query point in the BO algorithm. In the LCB acquisition function, the default weighting between mean and standard deviation of the GP predictive distribution in the MATLAB BO toolbox was used:

$$A(\boldsymbol{\theta}) = \mu(\boldsymbol{\theta}) - 2\sigma(\boldsymbol{\theta}) \tag{9.9}$$

This acquisition function promotes exploration, which makes sense when we are in such a high dimensional space.

9.3.2 Accounting for fibre angle discrepancy

To explore the effect of fibre angle approximation, the emulator from Section 9.2.4 will be used to infer the material parameters. Recall that the emulator was trained on simulations obtained from different combinations of constitutive parameters and 5PC representations of the LV geometry, with the fibre angles held fixed at $\alpha_{\text{endo}} = -60$ and $\alpha_{\text{epi}} = 60$. These are standard values of the fibre angles, used in the work in the earlier chapters of this thesis and found in the literature (Noè et al., 2019). To introduce discrepancy between the model and the test data as a result of fibre angle discrepancy, test data were generated with fibre angles and constitutive parameters set to combinations of the following values:

$$\begin{aligned}(\alpha_{\text{endo}}, \alpha_{\text{endo}}) &= \{(-60, 90), (-30, 60), (0, 90)\} \\ (a, b, a_{\text{f}}, b_{\text{f}}) &= \{(0.1, 5, 1, 6), (0.5,\ 8,\ 1.5,\ 6), (1,\ 1,\ 1,\ 1), (1,\ 5,\ 1,\ 5)\}\end{aligned} \tag{9.10}$$

These material parameter vectors were selected to give physiologically realistic diastolic behaviour. Each combination of material parameters and fibre angles was used, giving 12 synthetic test sets in total. Gaussian noise with standard deviation equal to 5 ml was added to the volume (motivated by the study in Table E.1) and iid Gaussian noise with a standard deviation of 0.03 was added to the circumferential strains (as discussed in Chapter 8, this is consistent with results from the literature). In order to remove the effect of LV geometry error on the fibre angle exploration, the simulated data were generated from a synthetic LV geometry generated in the 5 PC space.

Of interest is the effect that erroneous fibre angles can have on the estimation of the material parameters. In addition to this, we can investigate the use of methods from Section 9.2.2 to correct for the discrepancy introduced by the erroneous fibre angles. Given the nature of the discrepancy model, we might expect that inference is driven by the LVV component of the model. This concern will be addressed by comparing with inference results based on LVV alone. For each model, inference will be performed using MCMC.

For the volume only model and the model without discrepancy correction, this is a simple application of NUTS in line with the procedures used throughout this thesis. In the case of the discrepancy correction model, the procedure outlined in Section 9.2.3 will be used. Convergence is assessed using the PSRF, with 1000 samples obtained after convergence.

9.3.3 Effect of LV geometry

In order to infer the material parameters from measured data, we require the LV geometry. During parameter inference, we fix the LV geometry as an input to the simulator and optimise or sample the constitutive parameters. In the real data case, the measured data are obtained from CMR scans with the LVV obtained by reconstruction of the LV geometry at end of diastole (this follows the same procedure as segmentation of the early diastole LV geometry, which was outlined in Section 5.2.1) and the circumferential strains obtained using an independent measurement procedure called deformation tracking (Mangion et al., 2016a). Ideally, the effect of operator error would be assessed using data measured from CMR scans. For multiple different subjects, we could get multiple operators to segment the LV geometry at early and end of diastole and then measure the circumferential strains from the CMR scans. Unfortunately, the inconsistencies between model and data that we witnessed in Section 9.1.1 make this a difficult analysis to perform. Moreover, the difficulty in measuring circumferential strains means that the errors in their measurement would overshadow the errors in the LV geometry representation, which is of principal interest here. As a result, a different study had to be constructed.

Of interest is the effect that different operators (different people segmenting the LV geometry) have on the biomechanical model. In total, the LVs of 6 different patients[2] were segmented by three different operators who will be referred to as operators 1,2 and 3. The operators had different levels of experience in segmentation of the LV, but all were trained for the task. To investigate the effect of the operator on parameter estimation, the procedure visualised in Figure 9.4 was used.

Each operator will be treated as a *main operator*, with the other two considered as *test operators*. The main operator's LV geometry G, which has vector representation $\mathbf{g}$, will be used to generate a set of synthetic data by evaluating the simulator using $\mathbf{g}$ and material parameters $\tilde{\boldsymbol{\theta}} = (a, b, a_{\mathrm{f}}, b_{\mathrm{f}}) = (0.60, 1.30, 1.00, 3.60)$[3]. This is repeated for each of the 6 different patients, creating 6 "measured" datasets for each main operator. To infer the parameters from these datasets, several test geometries $\hat{G}$, which have vector representation $\hat{\mathbf{g}}$, will be used. These are the LV segmentations of the test operators, as well as the 5 and 10 PC reconstructions of the main operator's segmented LV geometry. If

[2]These were obtained from a study of COVID patients.

[3]The material parameters were selected because the simulated LVV corresponds to an EF roughly equal to 60%, which is a normal value (Hudson and Pettit, 2020).

the error introduced to the parameter estimates by PCA is lower than the error introduced by a new operator, then we have evidence that the inclusion of any extra LV geometry information will not improve the parameter inference accuracy, given the current methods for measuring the LV geometry.

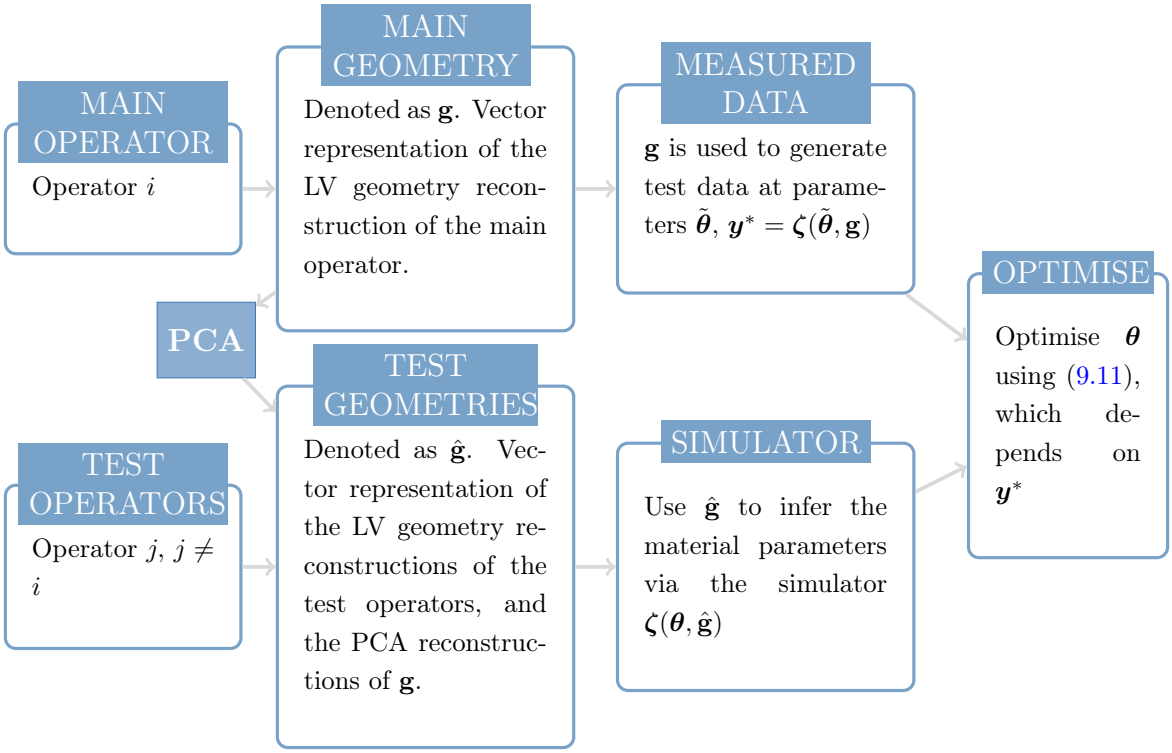


Figure 9.4: Diagram of the optimisation procedure for the LV geometry study. In total there are 4 test LV geometries, coming from the two test operators and the 5 and 10 PC reconstruction of the main geometry. Therefore, the optimisation is repeated 4 times.

For the following outline of the optimisation procedure, let $\boldsymbol{y}^* = \zeta(\tilde{\boldsymbol{\theta}}, \mathbf{g})$ be the measured data. The first output, y_0^*, is the LVV and the remaining 24, $\{y_i^*\}_{i=1}^{24}$, are the measured circumferential strains. To optimise the material parameters, the following objective function is used:

$$L(\boldsymbol{\theta}) = \left(\frac{V_{\text{early}} - y_0^*}{y_0^*} - \frac{\hat{V}_{\text{early}} - \zeta_0(\boldsymbol{\theta}, \hat{\mathbf{g}})}{\zeta_0(\boldsymbol{\theta}, \hat{\mathbf{g}})} \right)^2 + \sum_{i=1}^{24} (y_i^* - \zeta_i(\boldsymbol{\theta}, \hat{\mathbf{g}}))^2, \tag{9.11}$$

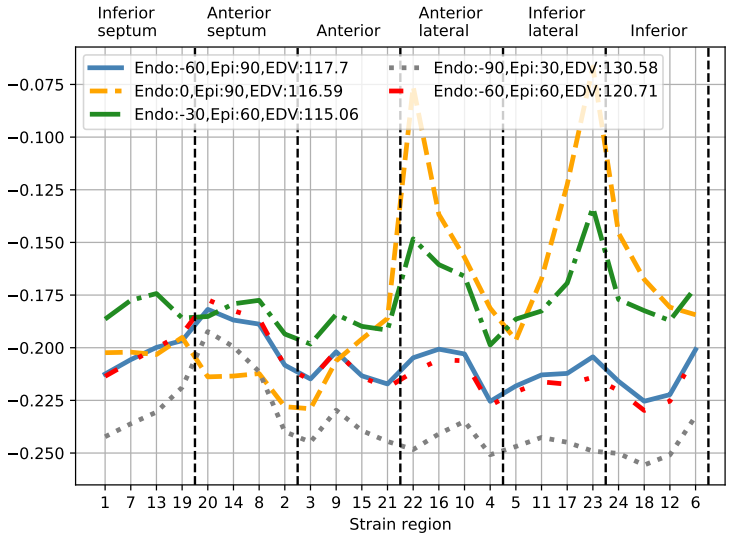
where V_{early} is the volume of G at early diastole and $\hat{V}_{\text{early}}$ is the volume of $\hat{G}$ at early diastole (this is in contrast to y_0^*, which is the LVV). The volume component of this loss function is unconventional and requires some discussion. The motivation behind this term is to negate the effect that the initial volume of a reconstruction has on the parameter optimisation, instead focusing on the effect of the shape of the LV geometry. This does not remove any contribution of LVV from the loss function, instead it means that the contribution from LVV is not dominated by the difference in early diastole volume. The

optimisation was performed in MATLAB using *fmincon* with the sequential quadratic programming method, which was also used by Gao et al. (2015). Each optimisation is initialised from the ground truth material parameter value, so the optimiser measures the perturbation introduced as a result of the LV geometry approximation.

9.4 Results

9.4.1 An initial look at the effect of fibre angles

In Figure 9.5, simulations obtained using the same LV geometry and material parameters but with a range of different fibre angles are plotted. The line plots show the trend in the circumferential strains, which can vary quite substantially as the angle changes. The corresponding angles are provided in the legend along with the LVV (in the legend it is labelled as EDV). The effect of the angle change is similar no matter the LV geometry, which is that some circumferential strains are affected quite substantially, especially the anterior lateral and inferior lateral regions. An important observation to take from this plot is as follows: by varying the angle from $\alpha_{\text{endo}} = -60, \alpha_{\text{epi}} = 90$ to $\alpha_{\text{endo}} = -30, \alpha_{\text{epi}} = 60$, we get only a 2 ml difference in LVV but a more substantial increase in the circumferential strains, particularly in the anterior lateral, inferior lateral and inferior region. In other words, it appears that the effect of fibre angle on volume and circumferential strain is not always in the same direction. This is precisely the type of effect that could be missing when we calibrate the simulator to new data, as shown in Figure 9.1.



(a)

Figure 9.5: Effect of fibre angle on the simulated circumferential strains and LVV.

9.4.2 Bayesian optimisation of angles and material parameters

The second exploration of the fibre angles considers the task of learning the material parameters and fibre angles together from data measured from CMR scans. Using BO, the four constitutive parameters a, b, a_f, b_f were optimised, with final results provided in Table 9.2. As well as providing the optimal configuration of the constitutive parameters, the table provides the error in the LVV at the optimum as well as the error in the circumferential strains, assessed as the mean absolute error in the 24 circumferential strains. We see a common inability to fit both the circumferential strains and the LVV. Both the angles in the table are constant because these were held fixed during optimisation. Table 9.3 gives the results of the 6 dimensional BO problem, including the four constitutive parameters and the two fibre angles from the RBM. Comparing the volume errors, we see that by optimising the fibre angles along with the constitutive parameters we can better match the volume measurements while still reaching the same (or lower) errors in the circumferential strains. Typically, this requires that the two angles are quite small compared with the values used to obtain the results in Table 9.2 and, as discussed above, these were treated as tuning parameters with no attention paid to their physiological validity.

Table 9.2: Results of Bayesian optimisation of material parameters, with fibre angles fixed at $\alpha_{\text{endo}} = -60$ and $\alpha_{\text{epi}} = 90$.

	Fixed angles Vweight=2							
ID	a	b	a_f	b_f	α_{endo}	α_{epi}	Absolute LVV error	Median absolute strain error
HV1	0.30	0.54	0.60	9.74	-60	90	20.58	5.05×10^{-2}
HV2	0.12	9.91	8.32	0.92	-60	90	12.48	4.0×10^{-2}
HV3	0.10	5.88	3.17	9.94	-60	90	23.8	3.67×10^{-2}
HV4	0.18	9.80	2.68	10.0	-60	90	35.50	4.18×10^{-2}
HV5	0.10	6.88	1.98	9.96	-60	90	17.2	4.71×10^{-2}
HV6	2.35	0.27	3.83	1.59	-60	90	16.0	2.62×10^{-2}
HV7	0.14	9.90	1.29	9.8	-60	90	15.0	3.08×10^{-2}
HV8	0.12	3.18	3.76	9.98	-60	90	29.8	4.24×10^{-2}

9.4.3 An attempt to account for fibre angle discrepancy

Based on the results of the BO study, it appears that fixing of fibre angles introduces a source of model discrepancy. To explore how this discrepancy can be acknowledged during parameter inference, the model from Section 9.2.3 will now be considered, inferring the parameters from the test data that were generated according to Section 9.3.2. Comparisons in parameter space are confounded by the strong correlations between the parameters of the H-O model, so another method for comparison is required. Recall from (9.10) that the test data were generated using each pair of material parameter-fibre angle pairs from

Table 9.3: Results of Bayesian optimisation of fibre angles and material parameters.

Optimised angles Vweight=2								
ID	a	b	a_f	b_f	α_endo	α_epi	Absolute LVV error	Median absolute strain error
HV1	0.36	1.27	3.00	0.58	-11.2	24.8	1.23	4.48×10^{-2}
HV2	0.12	9.95	5.84	1.01	-33.7	89.7	3.13	3.93×10^{-2}
HV3	0.18	3.52	5.01	9.91	-17.7	5.5	1.11	3.79×10^{-2}
HV4	0.12	5.55	4.63	6.26	-36.2	25.9	16.78	3.82×10^{-2}
HV5	0.16	5.79	3.37	0.96	-1.9	36.2	8.7	4.13×10^{-2}
HV6	1.43	3.67	2.19	0.54	-33.0	52.7	8.94	2.64×10^{-2}
HV7	0.15	4.31	3.03	5.19	-7.3	5.8	3.12	2.86×10^{-2}
HV8	1.69	0.35	3.65	2.90	-2.73	19.3	12.5	3.76×10^{-2}

sets of 4 material parameters configurations and 3 fibre angle configurations. Letting $\tilde{\boldsymbol{\alpha}}$ be a particular fibre angle configuration from this set of 3 and $\tilde{\boldsymbol{\theta}}$ be a material parameter vector from the set of 4 different parameter configurations, then the test data are:

$$\boldsymbol{y}^* = \boldsymbol{\zeta}(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\alpha}}, \mathbf{g}) + \boldsymbol{\xi} = \mathbf{z} + \boldsymbol{\xi} \tag{9.12}$$

where $\mathbf{g}$ is the representation of an LV geometry generated from a point, $\mathbf{h}$, in 5 PC space. The variable $\boldsymbol{\xi}$ corresponds to the Gaussian noise, with standard deviation of 5 ml for the volume and 0.03 for circumferential strains. If we were to plug $\tilde{\boldsymbol{\theta}}$ and $\mathbf{h}$ into the emulator η, we would recover:

$$\tilde{\boldsymbol{y}} = \boldsymbol{\eta}(\tilde{\boldsymbol{\theta}}, \mathbf{h}) \neq \mathbf{z} \tag{9.13}$$

where the lack of equality stems from the fact that the emulator was trained on simulations that assume a fixed fibre angle configuration of $\alpha_\mathrm{endo} = -60°$, $\alpha_\mathrm{epi} = 60°$, which are different from the three possible values of $\tilde{\boldsymbol{\alpha}}$.

In the preceding work of this thesis, two noise models have been considered: the first was iid Gaussian noise on the 25 standardised outputs (LVV and 24 circumferential strains) and the other considered iid Gaussian noise on the circumferential strains with a unique noise standard deviation of 5 ml on the LVV. Were we to assume this model and sample the material parameters from test data that, like $\boldsymbol{y}^*$, were produced by a process with fibre angles different from those used to generate the training data for the emulator, then the inferred parameters $\hat{\boldsymbol{\theta}}$ would be biased, as discussed in Section 9.1.4. As a result, the following would hold:

$$\tilde{\boldsymbol{y}} \neq \boldsymbol{\eta}(\hat{\boldsymbol{\theta}}, \mathbf{h}). \tag{9.14}$$

where $\tilde{\boldsymbol{y}}$ comes from (9.13). The lack of equality here is solely because $\hat{\boldsymbol{\theta}} \neq \tilde{\boldsymbol{\theta}}$, and ideally

we would infer material parameters $\hat{\boldsymbol{\theta}}$ such that (9.14) is an equality. This is what we hope to achieve by sampling parameters using the likelihood function from (9.6), where the term $\mathbf{b}(\mathbf{x}) + v$ corrects for discrepancy between $\boldsymbol{\eta}(\hat{\boldsymbol{\theta}}, \mathbf{h}) = \tilde{\boldsymbol{y}}$ and $\boldsymbol{y}^*$. Therefore, when sampling parameters using (9.6) as the likelihood function, we can assess the success of the model by comparing the distribution of predictions from the emulator at each posterior sample with $\tilde{\boldsymbol{y}}$. These simulations $\tilde{\boldsymbol{y}}$, which are produced according to (9.13), are now referred to as *benchmark simulations*. Note that we would ideally use the simulator to obtain the distribution of the simulator evaluations over the posterior of the parameters but due to computational costs, the emulator will be used instead.

First, we can perform a posterior predictive check by comparing the posterior predictive distribution with the test data. Since there are many test cases, this is only done for a single example (with $\alpha_{\text{endo}} = 0°, \alpha_{\text{epi}} = 90°$). Figure 9.6 shows the posterior predictive distributions of the circumferential strains in the case where parameters were sampled with the likelihood with model discrepancy (blue dashed lines) and the model that does not account for model discrepancy (red solid lines). Focusing first on the model without discrepancy correction, we see that the variance in the posterior predictive is quite large. Because the discrepancy introduced in $\boldsymbol{y}^*$ is not fully within the capacity of the emulator, the noise standard deviation is overestimated (by a factor of 2) and very little is learnt about the material parameters of the model. For the model with discrepancy correction, the variance in the posterior predictive is smaller. Indeed, the noise variance is now slightly underestimated, with the median of the posterior distribution of the standard deviation equal 0.024, which underestimates the true value of 0.03 as shown in Figure 9.7.

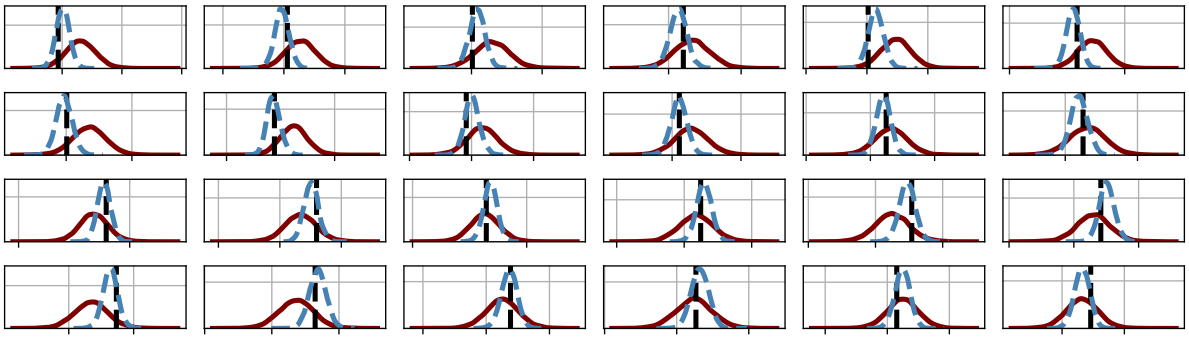


Figure 9.6: Comparing the posterior predictive distribution of the model that does not account for model discrepancy (red solid lines) and the model that does account for model discrepancy (blue dashed). Each subplot gives the result for a different circumferential strain, with the vertical dashed line showing the measured strain value.

We can now assess the performance of the different models by comparing the posterior distribution of $\eta(\boldsymbol{\theta}, \mathbf{h})$ with the benchmark simulations that were discussed previously. This time the volume only model can be considered as well. For all models, a distribution of model predictions is created by predicting from the emulator at each posterior sample.

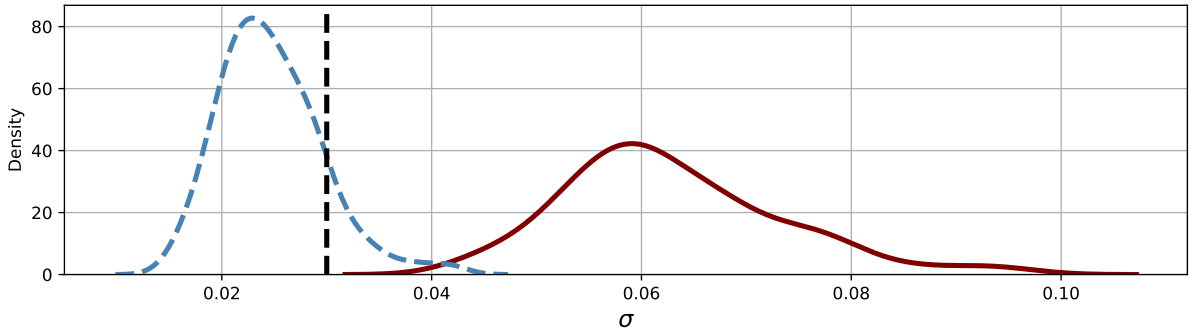


Figure 9.7: Comparing the posterior distribution of the noise standard deviation with the ground truth value, which is indicated by a vertical dashed line at 0.03. Without model discrepancy (red solid line), there is a huge inflation in noise standard deviation because the discrepancy between model and data is outwith the capacity of the model. With the inclusion of model discrepancy, this noise standard deviation is slightly underestimated as the model discrepancy term models some of the extra structure in the data.

According to the discussion at the beginning of this section, this distribution should contain the benchmark simulation if the model discrepancy correction is successful. Figure 9.8 provides detailed results for three different test cases (in each column) using each of the different models (in the different rows), as indicated in the legend. Note that the first column shows results for the same test case for which the posterior predictive distribution was visualised in Figure 9.6. The volume and polar GP models perform similarly, both outperforming the model without any discrepancy correction, which is particularly bad in the last example. The results for all test cases are compared in Figure 9.9, which shows the distribution of errors between the benchmark simulation and the posterior of η over the 12 different test cases. These are all combined into one distribution for each method of inference. From these plots we see that the volume only model and the model with discrepancy correction provided by the polar GP perform almost identically, both of which outperform inference with the standard normal likelihood function ((9.6) without any discrepancy correction included on the circumferential strains).

9.4.4 LV geometry study

An initial look at the effect of LV geometry

It is already clear from previous work that the behaviour of the LV in diastole can vary depending on the LV geometry. This was made evident by the error introduced to the outputs of the simulator when adopting a PCA representation of the LV geometry in Chapter 6. In Figure 9.10, the variations that result from running the simulator at the same material parameters for different LV geometries are presented. From this plot we see that the LV geometry can have a fairly substantial effect on the simulator outputs

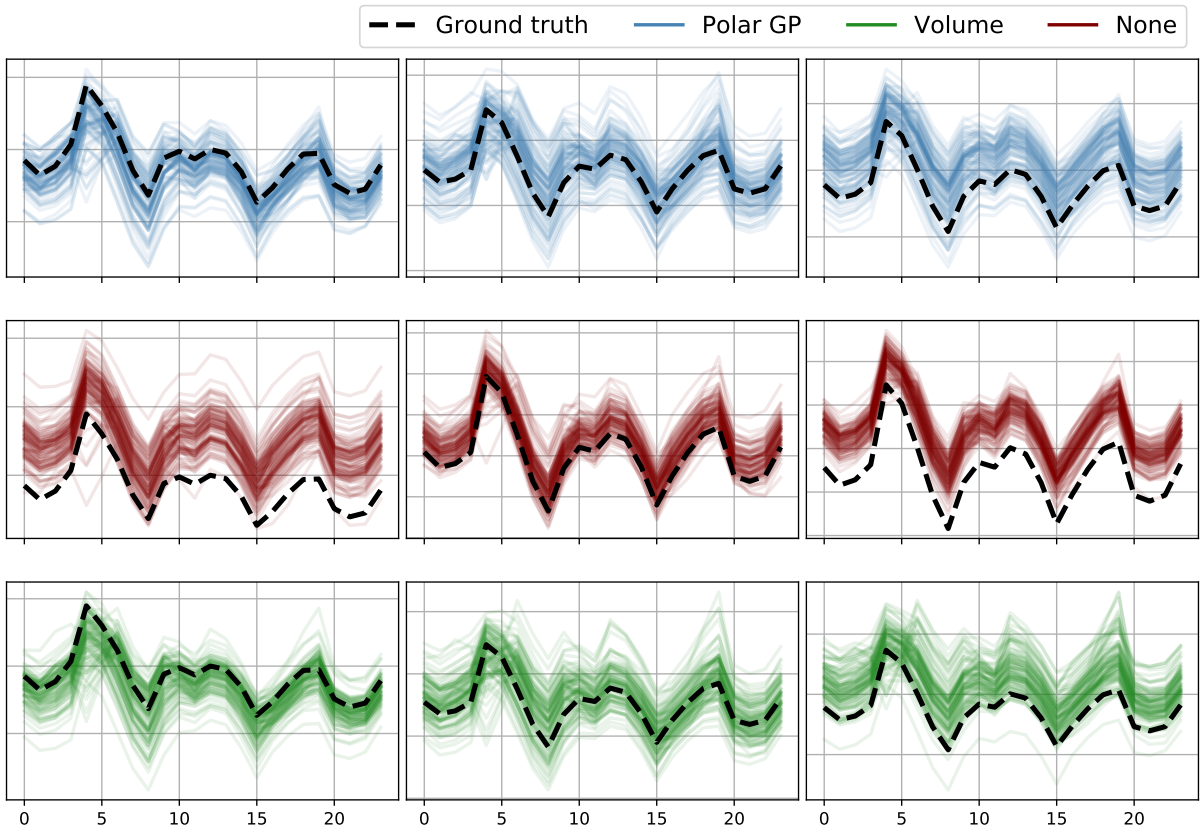


Figure 9.8: Comparing the posterior distribution of $\eta(\boldsymbol{\theta}, \mathbf{h})$ with the benchmark simulations. Three different test cases are shown (in each column), with the different rows giving the results when parameters are inferred using each of the different models, as indicated in the legend. “None" corresponds to the model that includes LVV and circumferential strains without accounting for any model discrepancy.

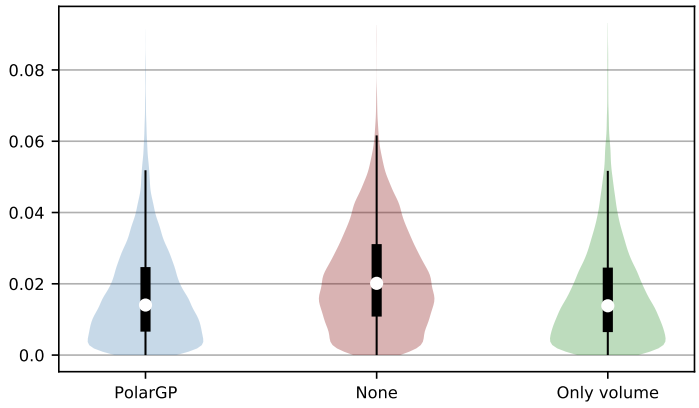
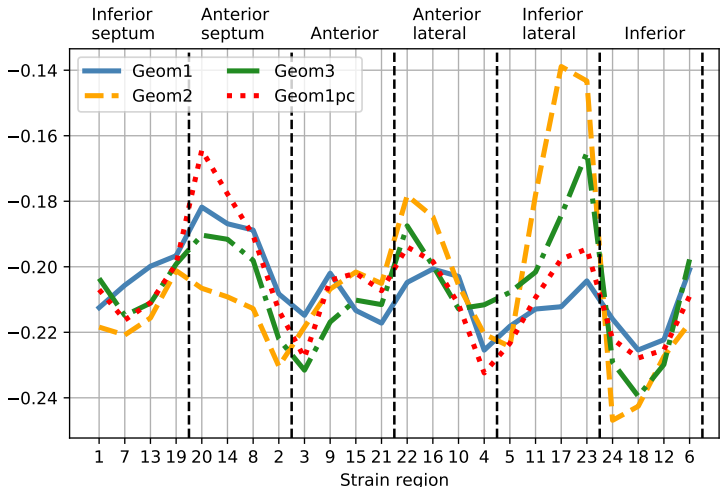


Figure 9.9: Distribution of absolute errors between the posterior distribution of $\eta(\boldsymbol{\theta}, \mathbf{h})$ and the benchmark simulations for all 12 test cases combined, across the three different noise models. The Polar GP and model with only volume perform similarly, both outperforming the standard iid Gaussian noise model (labelled as “None”).

and, similar to the fibre angles, the effect of LV geometry variation seems to be most pronounced in the inferior lateral region.



(a)

Figure 9.10: The effect of LV geometry on the simulated circumferential strains and LVV.

Operator effect in the geometry domain

As discussed in Section 9.3.3, 3 operators segmented 6 different LV geometries. A visual comparison of the LV geometry reconstructions of the different operators is provided in Appendix E.1, along with comparisons of the cavity volumes. Quantitative differences between the segmentations are provided in Table 9.4, where each subtable gives the results when a different operator is used as the main operator (recall that the main operator's segmentations are taken to be the benchmark). Also provided is the reconstruction error of a PCA reconstruction of the main operator's LV geometry segmentation, with the values highlighted in bold indicating the number of PCs needed to outperform both of the test operators in a particular example. In general, it appears that the shapes of operator 1 and 3 are in good agreement (at least in terms of MSE), but 8PCs are enough to outperform the test operators in 5 out of 6 test cases. The first segmentation of operator 1 appears to be a bit of an outlier, suggesting some error has been made during the segmentation. Indeed, upon further inspection it was discovered that an obvious error had been made during motion correction (where we align the short and long axis of the LV geometry reconstructions). The bottom row of the table provides the proportion of cases where PCA outperforms either operator. For instance, the value 0.44 below the 5PC column shows that in 44% of the 18 cases, 5PCs give lower reconstruction error than either operator. In all but one example, 10PCs (and often less) is enough to outperform a different operator in terms of reconstruction error.

Table 9.4: Comparing reconstruction errors of three different operators. Each of the 6 rows gives the results for an individual patient segmented by the operators. The main operator's reconstruction is taken to be the benchmark, compared with its PCA reconstruction and those of the other two operators. The first reconstruction with PCA that gives lower MSE than either operator is in bold.

Main operator: 1							
Oper 2	Oper 3	5PC	6PC	7PC	8PC	9PC	10PC
0.025	0.030	**0.002**	0.002	0.002	0.001	0.001	0.001
0.003	0.003	0.012	0.007	0.006	0.005	0.004	**0.003**
0.009	0.006	0.008	0.007	0.007	**0.006**	0.004	0.004
0.020	0.005	**0.005**	0.005	0.004	0.002	0.002	0.002
0.016	0.008	0.013	0.011	**0.007**	0.007	0.006	0.006
0.005	0.005	**0.003**	0.003	0.003	0.003	0.003	0.003
Main operator: 2							
Oper 1	Oper 3	5PC	6PC	7PC	8PC	9PC	10PC
0.025	0.004	**0.003**	0.003	0.002	0.002	0.002	0.002
0.003	0.006	0.014	0.007	0.007	0.006	0.005	0.004
0.009	0.013	**0.003**	0.003	0.003	0.002	0.002	0.002
0.020	0.013	**0.012**	0.012	0.004	0.004	0.004	0.004
0.016	0.023	0.021	0.020	**0.011**	0.010	0.008	0.008
0.005	0.004	**0.002**	0.002	0.002	0.002	0.002	0.002
Main operator: 3							
Oper 1	Oper 2	5PC	6PC	7PC	8PC	9PC	10PC
0.030	0.004	0.006	0.005	**0.002**	0.002	0.002	0.002
0.003	0.006	0.006	0.004	0.004	0.003	**0.002**	0.002
0.006	0.013	0.007	0.007	0.007	**0.005**	0.004	0.004
0.005	0.013	0.010	0.009	0.006	**0.004**	0.003	0.003
0.008	0.023	0.015	0.014	**0.008**	0.008	0.007	0.007
0.005	0.004	**0.002**	0.002	0.002	0.002	0.002	0.002
-	-	0.44	0.44	0.67	0.83	0.89	0.94

Operator effect in the biomechanical domain

The primary interest lies in the effect that the operator induced variations in the LV geometry have on the simulator. First considered using a comparison in output space, the results of this analysis are found in Figure 9.11. As before, each operator is individually treated as a main operator, and the simulated circumferential strains obtained using the main operator's LV geometry are compared with simulated quantities obtained using the 2 test operators and the 5 and 10 PC reconstructions. The labels on the horizontal axis of the violin plots indicate which test reconstruction we are considering. Note that the distributions combine the results for all 6 patients segmented by the operators.

From Figure 9.11, we see that operators 1 and 3 are in quite good agreement: if either 1 or 3 is the main operator and the other is the test operator, the simulated circumferential strain error is roughly the same as that obtained using a 10 PC reconstruction. Operator

2 tends to perform similarly to the 5 PC reconstruction, both of which are worse than the other two test operators. The real assessment of these errors is in the parameter estimation, which is the ultimate interest in this study.

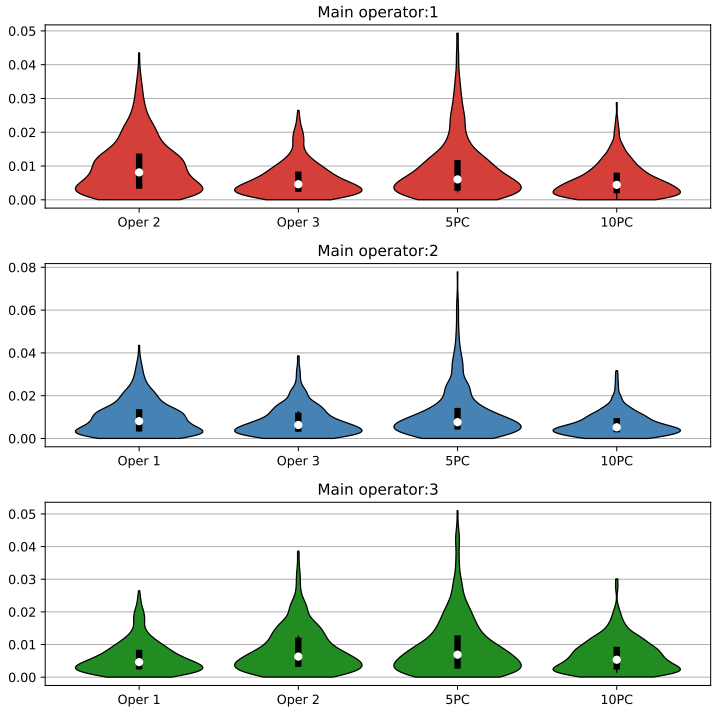


Figure 9.11: Distribution of simulated strain errors with different LV geometry approximations. In each subplot, a different operator is treated as the main operator (as specified in the title) and the simulations with the main operator's LV geometries are compared with the simulations using the other two operator's LV geometries and the 5 and 10 PC reconstructions of the main operator's LV geometry.

The effect of operators on the parameter estimation is explored using the simulation study outlined in Section 9.3.3. In this section the results will be summarised, with the exact numerical results found in Appendix E.2. The plots in Figure 9.12 compare the inference results using the absolute error in each individual parameter, with each row corresponding to a different main operator (recall that the six LV geometry reconstructions obtained by the main operator are used to generated test data for parameter optimisation). The results in the plots suggest that optimisation using each different test geometry gives similar results no matter the main operator.

A comparison that depends on the Euclidean distance of each individual parameter from its ground truth value is not a fair indicator of performance because it fails to take into account the relationship between the different material parameters. This inter-dependence is indicated by the ridges in Figure 9.13, where the loss function is plotted as a function of different combinations of material parameters. When assessing the success of parameter inference, a perturbation of the parameters by distance D along a ridge should be viewed

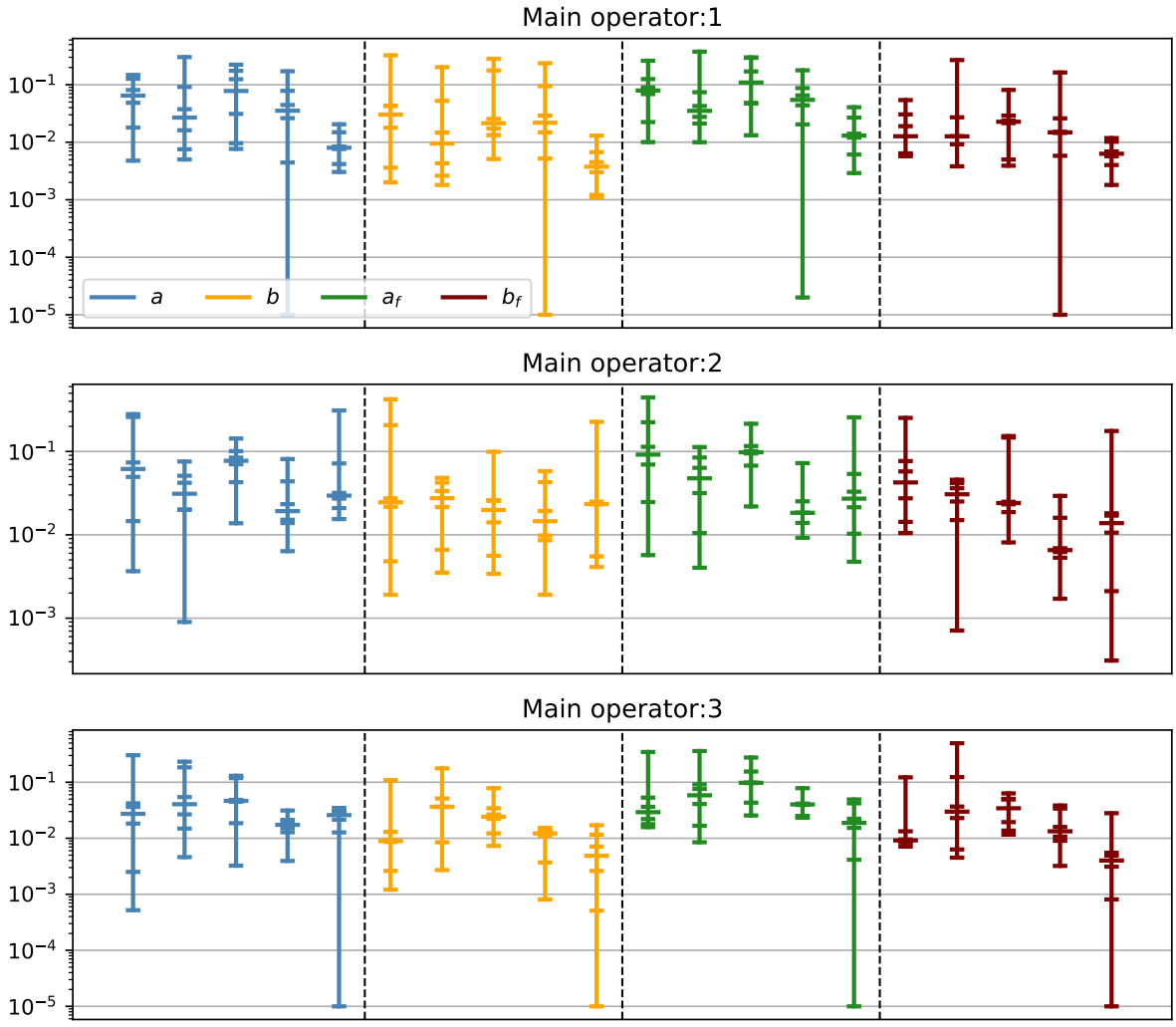


Figure 9.12: Results of parameter estimation. Each subplot shows the absolute errors in material parameters when a particular operator is treated at the main operator, whose segmentation is used to generate synthetic test data. In each subplot, the summaries are split according to each constitutive parameter with the sets of 4 error bars showing results for test operators, 5PC and 10PC. For instance, in the top subplot the first 4 (blue) error bars show the error in parameter estimates for a when using the reconstructions of operator 2, operator 3, 5PC and 10 PC where the segmentations of operator 1 are used to generate the test data. Each error bar shows the results for the 6 different LV geometries as narrow tick marks with the median shown as a wider tick mark.

differently to perturbation by distance D off a ridge. One metric that could overcome this issue is the Mahalanobis distance, which accounts for correlations between variables of interest. However, this is not straightforward to compute because it relies on a numerical estimate of the Hessian to provide an approximate covariance matrix. Instead of this, a metric in simulator output space can be used.

Recall that the data used in this study were generated using the main operator's LV geometry segmentation and the ground truth material parameters $\tilde{\boldsymbol{\theta}}$. For the current

comparison, this simulation is referred to as a benchmark simulation. For each test LV geometry, the material parameters were optimised to obtain $\hat{\boldsymbol{\theta}}$, and in general $\hat{\boldsymbol{\theta}} \neq \tilde{\boldsymbol{\theta}}$ because $G \neq \hat{G}$. In the current comparison, these parameters will be compared not in parameter space, but in function space. To do so, the simulator is evaluated using the main operator's LV geometry segmentation and $\hat{\boldsymbol{\theta}}$. This simulation can then be compared with the benchmark simulation to assess the accuracy of inference. By assessing the accuracy of the parameter estimates in simulator output space, movements along a ridge in parameter space will be penalised less than movements off a ridge.

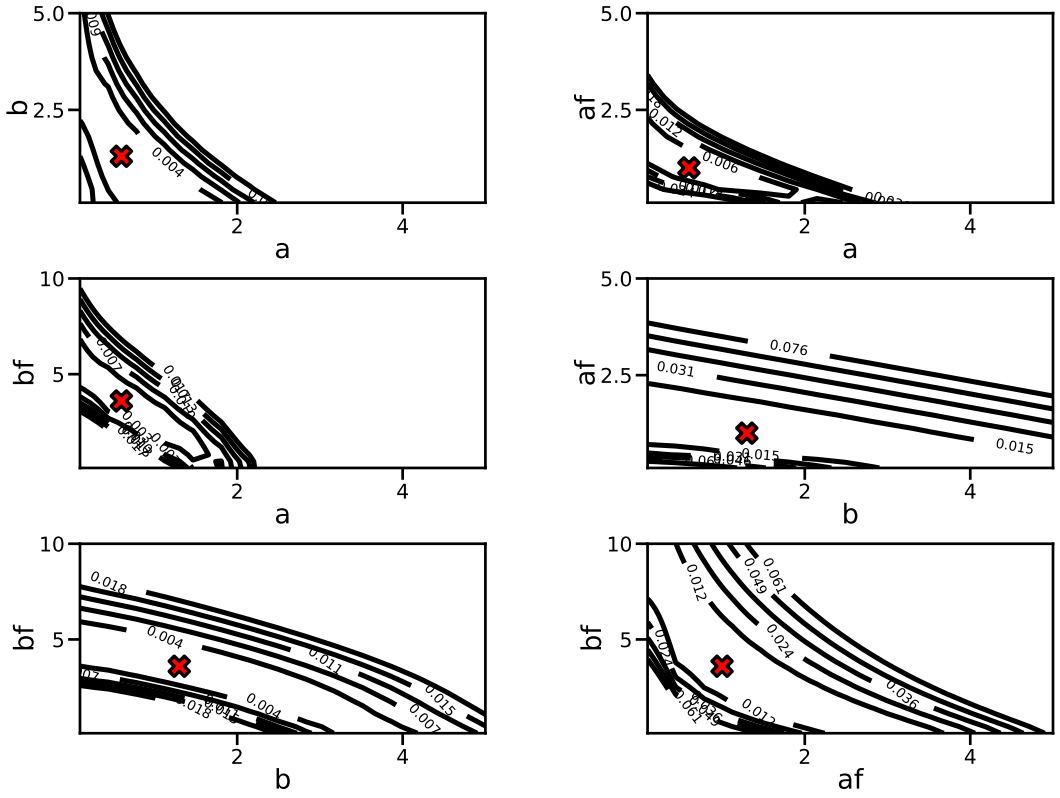


Figure 9.13: Contour plot of the RSS function over different combinations of parameters. The red X marks the ground truth parameter value.

The results of the output comparison are shown in Figure 9.14. Each subplot shows the results when a different operator is treated as the main operator, as indicated by the titles of the plots. The first two distributions correspond to results with the 2 test operators (indicated by the label of the violin plot) and the final 2 distributions are those corresponding to 5 and 10PC reconstructions of the LV geometry. Although it is difficult to see the details of some of the distributions, it is clear that the two different PC representations provide results that are at least as consistent as the new operators. This

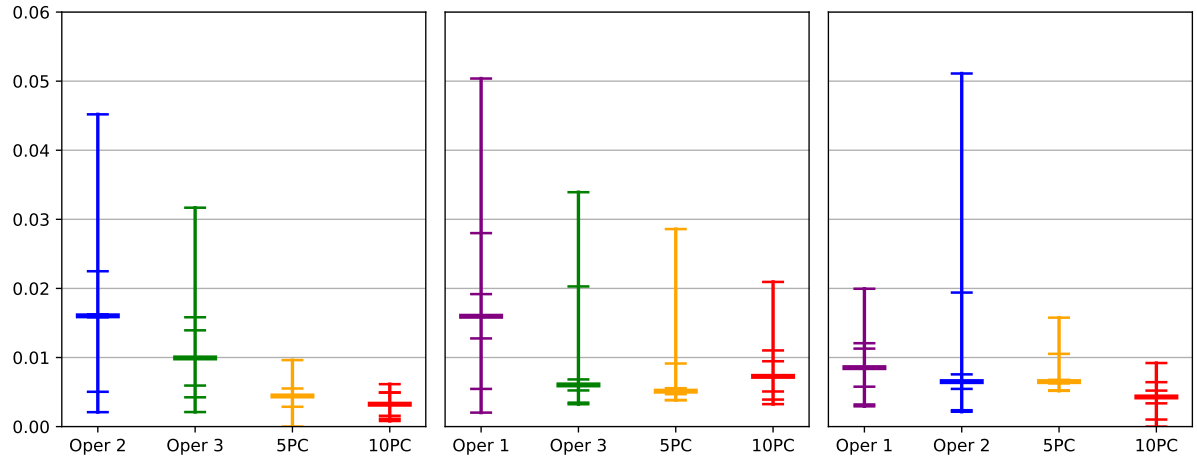


Figure 9.14: Comparing the parameter estimates in output space using Euclidean distance between circumferential strains. Each plot gives the results for a different main operator. The simulator is evaluated, with the ground truth LV geometry, at the optimised parameters (the test simulation) and ground truth parameters (the benchmark simulation). Comparing the test and benchmark simulation allows for a comparison of parameter estimates that takes into account correlations between the parameters.

result suggests that we will not gain anything by including large numbers of PCs in the parameter inference scheme, because the errors introduced by new operators lead to larger errors than with just 5 PCs.

9.5 Discussion and conclusion

Having acknowledged an obvious mismatch between the cardio-mechanic model and measured CMR data in Section 9.1.1, the work in this chapter considered the effect of erroneous physiological conditions on the model calibration task. The first study showed the change in the fit of the model to the measured data when the fibre angles are allowed to vary during optimisation. Building on this, a synthetic data study was carried out where test data were generated using deliberately incorrect angles in the rule-based fibre generation algorithm. Finally, the effect of the LV geometry on the parameter estimation task was considered. In particular, the knock-on effect of LV segmentation operator errors on the parameter estimation was studied.

In Section 9.4.2, the fibre angles were treated as extra variables during inference and jointly optimised along with the material parameters. This was motivated by the fact that the volume tends to be underestimated during inference, and the exploratory analysis in Figure 9.5 that showed that a change to the fibre angles can affect the circumferential strains more than the cavity volume. Therefore, it was postulated that one reason for the LVV-circumferential strain mismatch could be the fibre angles. This joint-optimisation study showed that tuning the parameters and the angles together does allow for an im-

provement in the fit to the measured CMR data. These results should motivate future work on inferring the material parameters and fibre angles jointly, especially given the difficult of measuring the fibre angles in vivo. To do so reliably would require more precise or more plentiful datasets than those that are currently available.

The next stage in studying the effect of fibre angles was to consider a synthetic data study. In replacement of the cardio-mechanical model, an emulator was used where the training data had been generated assuming a fixed set of fibre angles (which was the case with all other emulators used for I-UQ in this thesis). To introduce model discrepancy, the synthetic data were generated using fibre angles different from the ones used when generating the training set. This introduced a mismatch between the emulated circumferential strains and the measured circumferential strains, but resulted in no systematic mismatch in the LVV. In total, three different models were used to estimate the parameters from the test data. The first was the standard Gaussian noise model used in the previous work in this thesis. In particular, this assumed a fixed noise variance on the LVV and iid Gaussian noise on the circumferential strains, the latter of which could be inferred during inference. The second was a model that accounted for discrepancy on the circumferential strains via a correction function on the circumferential strains that was modelled using a GP as outlined in Section 9.2.2. To ensure model identifiability, the LVV was assumed to be corrupted with Gaussian noise with a fixed noise variance, but no model discrepancy. Finally, a model that included the LVV alone was considered, again with fixed noise standard deviation. The fixed noise standard deviation for the LVV was the same across the three models. By assuming the first noise model, a bias was introduced to the parameter estimates as a result of the mismatch between the fibre angles used in the emulator and the fibre angles used to generate the test data. In addition, there was an inflation in the inferred noise variance, resulting from the fact that there was extra discrepancy between model and data that could not be accounted for by the model. This discrepancy was corrected by including a discrepancy term on the circumferential strains, allowing for a reduction in the inferred noise variance on the circumferential strains that was closer to the ground truth value, and a reduction in the bias of the parameter estimates. Lastly, the model with discrepancy correction and the model with only LVV were shown to provide similar inference results. While this might motivate the idea of using the LVV alone for parameter inference, the benefit of the model with discrepancy correction is that it is automatic, without requiring us to study the relationship between circumferential strains and LVV. In the future, it would be worth exploring ways of incorporating a more informative discrepancy correction for both the circumferential strains and the LVV.

One of the main tasks in this thesis has been to allow for the generalisation of a statistical emulator to different LV geometries. Previous work, in particular Chapter 6, suggested that 5 PCs were not sufficient for representation of the LV geometry in parameter

inference. Making use of three different LV segmentation operators, the final study in this chapter has compared inter-operator LV geometry errors with those obtained using PCA. The comparison was performed first in geometry space and then using the biomechanical model. Considering first the geometry space comparison, the results in Table 9.4 showed that in all but one example, a 10 PC reconstruction gave a more accurate approximation to an LV geometry than a segmentation obtained by a new operator. The next stage of the comparison considered the biomechanical model, first in terms of the forward model and then in terms of inverse modelling. In the forward model, operators 1 and 3 were found to be fairly consistent, both achieving results similar to 10 PCs. On the other hand, Operator 2 and the 5 PC reconstructions were noticeably worse. The real test of these different LV geometry approximation methods was in the parameter estimation, which is ultimately our main interest. Comparing parameter optimisation results in parameter space, the different LV geometry approximations were found to give similar results. It was suggested that this parameter space comparison might be suboptimal because it does not account for correlations between parameters. As a result, an output space comparison of the parameter estimates was performed. The result of this comparison showed that 5 and 10 PCs give better results than the new operators, which is a quite substantial result in the context of this thesis. In particular, this suggests that there is no advantage to improving the accuracy of the LV geometry reconstruction in the generalised emulator.

While the results in the current chapter suggest that there is no need to improve the accuracy of the generalised emulators that were built in this thesis, it is possible that the accuracy of LV segmentation will improve to be more objective in the future. In this case, a more accurate emulator would be beneficial for parameter inference. Appendix F suggests an alternative approach to emulation of the simulator model, where a set of basis functions is learnt from a training set of LV geometries (with corresponding simulations) and adapted to a new test subject. Fitting to the new test subject relies on a small number of simulations that can be run in parallel, but preliminary results show that the model can be very accurate with only 20 simulations. Further exploration of the design and training of the model would likely lead to further improvements in accuracy.

In conclusion, the present study has highlighted several problems with the translation of mathematical models to the clinic. First of all, the large mismatch between measured data and the LV model was highlighted, acknowledging that this could be the result of either model-reality discrepancy or noise introduced by the measurement process. The former situation was studied in the remainder of the chapter, which sheds light on the effect that the fibre angles of the rule based method and the error introduced by operators to the LV geometry can have on the parameter estimation. In particular, a model that accounts for model discrepancy was found to provide better results than one that assumes iid measurement errors, while providing no noticeable difference with a model that included

LVV alone. This study assumed that there is no bias introduced to the LVV, which was the case for the fibre angles used to generate the test data. In the LV geometry study, the effect that operator segmentation variation has on the parameter estimation in the LV model was found to exceed that that was introduced by PCA. This should motivate multiple lines of future study. Principal among these is the collection of larger and more reliable datasets for the estimation of the model parameters, without which the parameter estimates obtained for these models cannot really be trusted. Secondly, a better informed discrepancy correction that accounts for bias in the LVV and circumferential strains would allow for more trustworthy inferences than are currently available.

Chapter 10

Discussion and Conclusion

10.1 Main findings of this thesis

The work in this thesis considered the task of statistical emulation of a patient-specific mathematical model of the LV in diastole. To this end, several important contributions have been made. Principal among these was the proposal of a generalisable emulator for the mathematical model, allowing for generalisation to previously unseen LV geometries. While the PCA representation of the LV geometry that was used in the model was not perfect, a comparison with inter-operator errors has suggested that the errors incurred are within those experienced when varying the operator that performs the segmentation of the LV geometry. A second key contribution was the construction of a prior distribution from the information contained in the Klotz curve. This is an ex vivo study that has often been used in the literature as part of an objective function for optimisation of passive myocardial parameters. The formation of a prior distribution allowed for the incorporation of the Klotz curve in a Bayesian inference model. Finally, a combined SA/I-UQ study was used to explore the structural and practical identifiability of the H-O model. This study proposed a new four-dimensional parameterisation of the H-O model as a result of the low sensitivity of some of the model's parameters.

Starting from a simplified simulator in Chapter 4, it was shown that emulation of the outputs of the simulator allows for improved parameter inference accuracy compared with emulation of the loss function. In this same chapter, a neural network model, which provides an efficient global emulator of the simulator function, was shown to perform similarly to local GP models. This, together with results found in Chapter 6, motivated the use of neural networks for emulation of the outputs of the simulator in the remainder of the thesis. As opposed to GPs, neural networks only allow regression rather than interpolation; the ability to do the latter is one of the main advantages of GPs for emulating deterministic functions. However, the results from Chapter 9 suggest that interpolation is not what we want in practice, due to the noise in the data and the intrinsic model mismatch.

This provides a motivation for the use of parametric regression models, like NNs, due to their lower computational complexity. The simplification of the simulator in Chapter 4 involved fixing the LV geometry, which meant that the emulator could not generalise to new subjects. As a solution to this, the proceeding two chapters considered first the task of dimension reduction of the LV geometry in Chapter 5 and then the extension of the emulator input space to include a low dimensional representation of the LV geometry in Chapter 6. During the exploration in Chapter 6, the identifiability of the parameters was shown to display two properties. Firstly, as a result of nonstationarity of the simulator function, it was found that accurate parameter estimation is more attainable in certain parts of the configuration space. Secondly, certain parameters of the model were found to exhibit poor identiability, which is the result of their physiological meaning. In more detail: certain parameters in the H-O model are helpful for providing accurate prediction of the myocardium's behaviour at stretches beyond the in vivo range, and this introduces identifiability problems when we consider the task of inferring parameters from in vivo measured data.

In Chapter 7, a technique that is commonly used in optimisation routines in the literature was formalised as a prior distribution. Specifically, the Klotz curve has previously been used to introduce a dependence of an objective function on the high stretch behaviour of the myocardium. In Chapter 7, the uncertainty in the Klotz function, obtained through MCMC sampling of the Klotz curve function parameters, was propagated through to the LVV at different pressure values. Using a product of experts approach, two different prior distributions were derived: the empirical Klotz prior depended on the measured end diastolic volume of the cavity while the nonempirical Klotz prior depended only on the predicted end diastolic volume at different cavity pressures. The empirical Klotz prior was shown to reduce the uncertainty in the parameter estimates when compared with a standard uniform prior on the physiological range of the parameters, and outperformed the nonempirical Klotz prior for this purpose.

While the results of Chapter 7 were encouraging, we ideally wish to extract as much information as possible about the parameters from the in vivo data. To this end, one hypothesis was that the radial and longitudinal strains could help in the estimation of the parameters that are weakly-identified using only circumferential strains and LVV. In Chapter 8, the sensitivity of the LVV and radial, longitudinal and circumferential strains to the H-O model inputs was assessed. This showed that, while the sensitivity indices for a, b, a_{f} and b_{f} (see (8.1)) varied slightly across the different outputs from the LV model, the remaining parameters of the H-O model have near-zero sensitivity regardless of the output considered. This motivated a new parameterisation of the H-O model in which the first 4 parameters are treated as random variables and the final 4 are fixed to population-based values. The use of I-UQ in this study showed the variation in

the practical identifiability of the parameters with varying end diastolic pressure, which provides retrospective motivation for the use of the Klotz prior. The I-UQ study showed that the identifiability of the parameters depended on their locations in the parameter space, which was similar to the results found for the previous theta parameterisation.

Chapter 9 tackled some issues that are of particular relevance to the translation of mathematical models of cardiac function to the clinical setting. These were motivated by apparent inconsistencies between the LV model and the measured data during inference. Based on this, the issues of model discrepancy as a result of the physiological conditions were explored. A synthetic data study revealed that without accounting for model discrepancy, inconsistencies between the assumed fibre angles and the actual fibre angles can lead to a bias in our parameter estimates. The bias in the parameter estimates was corrected by accounting for model discrepancy in the noise model. Finally, a study of the effect that operator variations in the LV segmentation have on the parameter estimates was explored, showing that the error incurred with a 5 PC reduction of the LV geometry is within the error obtained by changing the operator. This provides empirical evidence that the generalisable emulator proposed in this thesis is sufficiently accurate given the current methods for reconstructing the LV geometry manually. Assuming that LV geometry measurement approaches improve sufficiently, the preliminary exploration of a new method to emulation with in clinic correction has proven encouraging in Appendix F, but this method is still in the early stages of development.

10.2 Future work

In all studies throughout this thesis, the LVV and 24 circumferential strains at well defined locations on the LV wall were used to summarise the deformation of the LV in diastole for the purpose of inverse estimation. Given the high levels of noise we typically find in these datasets, it would be beneficial to increase the number of observations. Recall from Section 2.2 that the mathematical model of the LV produces the entire deformed LV mesh, from which the circumferential strains and LVV are extracted. Rather than using these summaries of the mesh, an alternative would be to attempt to match the actual deformed mesh with one that is extracted from the CMR scans at end of diastole. To define a loss or likelihood function from these data and model, we must find a correspondence between the nodes of the simulator deformed mesh and the meshes extract from the CMR scans, which is a non-trivial task. If possible, the larger dataset provided by this approach would allow us to better infer the underlying signal, while also allowing for better exploration of the discrepancy correction methods that were studied in Chapter 9. Also related to the measured data is the LV geometry segmentation, which was shown in Chapter 9 to be subject to operator subjectivity. An interesting direction of future work would

be to consider accounting for the uncertainty in the LV geometry segmentation in the forward modelling, which could be done using a probabilistic treatment of the segmented boundaries.

The inclusion of LV geometry uncertainties in the inverse estimation requires a model that can more accurately generalise to new LV geometries. The work in Appendix F is promising in this regard, but we ideally would use a method that requires zero simulator invocations, which is also desirable for in clinic inference approaches. To this end, recent efforts have been made to apply graph neural networks (GNNs) to the LV simulator (Dalton et al., 2021). GNNs replace costly finite element numerical integration by message passing on graphs, which has been widely studied by the machine learning community (see for instance Bishop (2006, Section 8.4); Murphy (2012, Section 22.4)). This produces a model that can generalise to unseen graphs, which in this case are new LV finite element meshes. While the results of Dalton et al. (2021) are promising, these methods have yet to be applied to full-scale anatomical LV meshes.

One of the disadvantages of data-driven modelling approaches, including the emulators used throughout this thesis, is that they perform poorly outwith the vicinity of the training data. This was the fundamental reason why the model developed in Chapter 4 could not generalise to new LV geometries. Physics informed neural networks (PINN) are recently proposed methods that overcome this issue by taking the physics of the problem into account by using the form of the PDEs to train the network weights, turning what is typically a black-box method into a "grey-box" method. These models require no simulator runs for training, and have already been applied in the context of cardiac modelling (Sahli Costabal et al., 2020) and in the specific context of the LV in diastole (Buoso et al., 2021). Successful implementation of either PINNs or GNNs to model anatomical LV geometries would be a huge step forward, with lasting impact in other cardiac models.

The ultimate goal of mechanical modelling of the heart is clinical translation. While mathematical and statistical modelling of cardiac function are progressing to the point of patient-specific modelling, further developments in terms of model accuracy could ultimately be in vain without greater consideration of how these models will be used in the clinic. An impact area in this regard is the provision of clinical biomarkers (Chabiniok et al., 2016; Niederer and Smith, 2016). For this task, the inferred parameters would be used to provide an indicator of disease, and an important consideration here is the parameterisation of the model. While more heavily parameterised models can provide higher fidelity simulations, these models often become practically unidentifiable, making the interpretation of our inference results more difficult. The parameterisation proposed in Chapter 8, while motivated by the SA study, could probably be reduced further with this in mind. Ultimately, this model selection task would require the consideration of diseased patients to see which parameters have diagnostic potential. Assuming the selection

of an appropriate parameterisation, we also require greater consideration of how to communicate or summarise the inferred mechanical behaviour of the heart in a way that can be interpreted by clinicians. For instance, while uncertainty quantification is vital given the uncertainties associated with all aspects of cardiac models, results summarised in the form of parameter space posterior distributions would be of little use to clinicians. More useful would be to devise a system that takes the uncertainty into account to provide some interpretable, complementary information to the clinician when assessing the health of a patient.

Appendix A

Appendix for Chapter 3

A.1 GP prediction with precomputation

Recall the predictive equations for a GP model:

$$m(\theta^*) = \mathbf{k}_{\theta^*\boldsymbol{\theta}}[\mathbf{K}_{\boldsymbol{\theta}\boldsymbol{\theta}} + \sigma^2\mathbf{I}]^{-1}\boldsymbol{y}$$
$$v(\theta^*) = k_{\theta^*\theta^*} - \mathbf{k}_{\theta^*\boldsymbol{\theta}}[\mathbf{K}_{\boldsymbol{\theta}\boldsymbol{\theta}} + \sigma^2\mathbf{I}]^{-1}\mathbf{k}_{\boldsymbol{\theta}\theta^*} + \sigma^2$$

With $\mathbf{K}_{\boldsymbol{\theta}\boldsymbol{\theta}} = \mathbf{L}\mathbf{L}^\top$, which requires $\mathcal{O}(n^3)$ computations, the predictive mean becomes:

$$\mu(\theta^*) = k_{\theta^*\boldsymbol{\theta}}\mathbf{L}^{-\top}\mathbf{L}^{-1}\mathbf{y}$$

where $\mathbf{b} = \mathbf{L}^{-\top}\mathbf{L}^{-1}\mathbf{y}$ can be precomputed using forward and backward substitution. At a test point, the predictive mean $\mu(\theta^*) = \mathbf{k}_{\theta^*\boldsymbol{\theta}}\mathbf{b}$ requires $\mathcal{O}(n)$ computation. For the predictive variance, only $[\mathbf{K}_{\boldsymbol{\theta}\boldsymbol{\theta}} + \sigma^2\mathbf{I}]^{-1}$ can be precomputed, meaning the predictive variance requires computational cost $\mathcal{O}(n^2)$ rather than $\mathcal{O}(n^3)$

A.2 Sparse GP prediction with precomputation

The predictive equations of the sparse GP proposed by Titsias (2009), with training inputs $\boldsymbol{\theta}$, training outputs $\boldsymbol{y}$ and inducing inputs $\tilde{\boldsymbol{\theta}}$ are:

$$m(\theta^*) = \mathbf{k}_{\theta^*\tilde{\boldsymbol{\theta}}}\mathbf{K}_{\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}}^{-1}\boldsymbol{\mu} \tag{A.1}$$
$$v(\theta^*) = k_{\theta^*\theta^*} - \mathbf{k}_{\theta^*\tilde{\boldsymbol{\theta}}}\mathbf{K}_{\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}}^{-1}\mathbf{k}_{\tilde{\boldsymbol{\theta}}\theta^*} + \mathbf{k}_{\theta^*\tilde{\boldsymbol{\theta}}}\mathbf{K}_{\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}}^{-1}\mathbf{A}\mathbf{K}_{\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}}^{-1}\mathbf{k}_{\tilde{\boldsymbol{\theta}}\theta^*} \tag{A.2}$$

where θ^* is the test input. $\mathbf{A}$ and $\boldsymbol{\mu}$ have the following form:

$$\boldsymbol{\mu} = \mathbf{K}_{\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}}\boldsymbol{\Sigma}\mathbf{K}_{\tilde{\boldsymbol{\theta}}\boldsymbol{\theta}}\mathbf{y} \tag{A.3}$$

$$A = \mathbf{K}_{\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}}\boldsymbol{\Sigma}\mathbf{K}_{\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}} \tag{A.4}$$

$$\boldsymbol{\Sigma} = [\sigma^2\mathbf{K}_{\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}} + \mathbf{K}_{\tilde{\boldsymbol{\theta}}\boldsymbol{\theta}}\mathbf{K}_{\boldsymbol{\theta}\tilde{\boldsymbol{\theta}}}]^{-1} \tag{A.5}$$

Plugging (A.5) into (A.3), and (A.3) into (A.1) gives:

$$m(\theta^*) = \mathbf{k}_{\theta^*\tilde{\boldsymbol{\theta}}}\mathbf{K}_{\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}}^{-1}\mathbf{K}_{\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}}[\sigma^2\mathbf{K}_{\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}} + \mathbf{K}_{\tilde{\boldsymbol{\theta}}\boldsymbol{\theta}}\mathbf{K}_{\boldsymbol{\theta}\tilde{\boldsymbol{\theta}}}]^{-1}\mathbf{K}_{\tilde{\boldsymbol{\theta}}\boldsymbol{\theta}}\mathbf{y} \tag{A.6}$$

With $\mathbf{K}_{\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}} = \mathbf{L}_m\mathbf{L}_m^\top$, we have the following:

$$\begin{aligned} m(\theta^*) &= \mathbf{k}_{\theta^*\tilde{\boldsymbol{\theta}}}[\sigma^2\mathbf{L}_m\mathbf{L}_m^\top + \mathbf{K}_{\tilde{\boldsymbol{\theta}}\boldsymbol{\theta}}\mathbf{K}_{\boldsymbol{\theta}\tilde{\boldsymbol{\theta}}}]^{-1}\mathbf{K}_{\tilde{\boldsymbol{\theta}}\boldsymbol{\theta}}\mathbf{y} \\ &= \mathbf{k}_{\theta^*\tilde{\boldsymbol{\theta}}}\mathbf{L}_m^{-\top}[\mathbf{I} + \sigma^{-2}\mathbf{L}_m^{-1}\mathbf{K}_{\tilde{\boldsymbol{\theta}}\boldsymbol{\theta}}\mathbf{K}_{\boldsymbol{\theta}\tilde{\boldsymbol{\theta}}}\mathbf{L}_m^{-\top}]^{-1}\mathbf{L}_m^{-1}\mathbf{K}_{\tilde{\boldsymbol{\theta}}\boldsymbol{\theta}}\mathbf{y} \end{aligned}$$

Letting $\mathbf{b} = \mathbf{L}_m^{-1}\mathbf{K}_{\tilde{\boldsymbol{\theta}}\boldsymbol{\theta}}$, we have the following:

$$\begin{aligned} m(\theta^*) &= \mathbf{k}_{\theta^*\tilde{\boldsymbol{\theta}}}\mathbf{L}_m^{-\top}[\mathbf{I} + \sigma^{-2}\mathbf{b}\mathbf{b}^\top]^{-1}\mathbf{L}_m^{-1}\mathbf{K}_{\tilde{\boldsymbol{\theta}}\boldsymbol{\theta}}\boldsymbol{y} \\ &= \mathbf{k}_{\theta^*\tilde{\boldsymbol{\theta}}}\mathbf{L}_m^{-\top}[\mathbf{L}_B\mathbf{L}_B^\top]^{-1}\mathbf{L}_m^{-1}\mathbf{K}_{\tilde{\boldsymbol{\theta}}\boldsymbol{\theta}}\boldsymbol{y} \\ &= \mathbf{k}_{\theta^*\tilde{\boldsymbol{\theta}}}\mathbf{L}_m^{-\top}[\mathbf{L}_B\mathbf{L}_B^\top]^{-1}\tilde{\boldsymbol{y}} \end{aligned}$$

Letting $\mathbf{z} = \mathbf{L}_B^{-1}\tilde{\boldsymbol{y}}$, we have:

$$\begin{aligned} &= \mathbf{k}_{\theta^*\tilde{\boldsymbol{\theta}}}\mathbf{L}_m^{-\top}\mathbf{L}_B^{-\top}\mathbf{z} \\ &= \mathbf{k}_{\theta^*\tilde{\boldsymbol{\theta}}}\mathbf{a} \end{aligned}$$

where $\mathbf{a}$ is an $m \times 1$ vector that can be precomputed. Therefore, the predictive mean is $\mathcal{O}(m)$ with precomputation.

As in the standard GP, there is a limit on precomputation of the predictive variance of the sparse GP.

$$v(\theta^*) = k_{\theta^*\theta^*} - \mathbf{k}_{\theta^*\tilde{\boldsymbol{\theta}}}\mathbf{K}_{\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}}^{-1}\mathbf{k}_{\tilde{\boldsymbol{\theta}}\theta^*} + \mathbf{k}_{\theta^*\tilde{\boldsymbol{\theta}}}\mathbf{K}_{\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}}^{-1}\mathbf{A}\mathbf{K}_{\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}}^{-1}\mathbf{k}_{\tilde{\boldsymbol{\theta}}\theta^*} \tag{A.7}$$

$$= k_{\theta^*\theta^*} - \mathbf{k}_{\theta^*\tilde{\boldsymbol{\theta}}}\mathbf{K}_{\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}}^{-1}\mathbf{k}_{\tilde{\boldsymbol{\theta}}\theta^*} + \mathbf{k}_{\theta^*\tilde{\boldsymbol{\theta}}}[\sigma^2\mathbf{K}_{\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}} + \mathbf{K}_{\tilde{\boldsymbol{\theta}}\boldsymbol{\theta}}\mathbf{K}_{\boldsymbol{\theta}\tilde{\boldsymbol{\theta}}}]^{-1}\mathbf{k}_{\tilde{\boldsymbol{\theta}}\theta^*} \tag{A.8}$$

Now, without repeating the process of writing out matrix decompositions to introduce stable matrix calculations, we can just read off the computational costs of the predictive variance. In the term $\mathbf{k}_{\theta^*\tilde{\boldsymbol{\theta}}}\mathbf{K}_{\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}}^{-1}\mathbf{k}_{\tilde{\boldsymbol{\theta}}\theta^*}$, the matrix inverse can be precomputed, but $\mathbf{k}_{\theta^*\tilde{\boldsymbol{\theta}}}$ and $\mathbf{k}_{\tilde{\boldsymbol{\theta}}\theta^*}$ cannot. Therefore, the cost of the matrix multiplications, which is $\mathcal{O}(m^2)$ is incurred at prediction time. In the second term $\mathbf{k}_{\theta^*\tilde{\boldsymbol{\theta}}}[\sigma^2\mathbf{K}_{\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}} + \mathbf{K}_{\tilde{\boldsymbol{\theta}}\boldsymbol{\theta}}\mathbf{K}_{\boldsymbol{\theta}\tilde{\boldsymbol{\theta}}}]^{-1}\mathbf{k}_{\tilde{\boldsymbol{\theta}}\theta^*}$, the matrix inverse

can be precomputed so at prediction time this also requires $\mathcal{O}(m^2)$ computational costs, and the total cost of the predictive variance is $\mathcal{O}(m^2)$. Note that many GP packages do not include precomputation in their prediction routines and require repeated inversion of the covariance matrix at predict time.

Appendix B

Appendix for Chapter 4

B.1 Multioutput GP

Continuing from the main text, assume a linear form for the mean function, with $\mathbf{h}(\boldsymbol{\theta}) = [1, \boldsymbol{\theta}]$ an $m \times 1$ vector and coefficients $\mathbf{B}$, which is an $m \times q$ matrix. At point $\boldsymbol{\theta}$ the mean is calculated as $m(\boldsymbol{\theta}) = \mathbf{B}^\top \mathbf{h}(\boldsymbol{\theta})$. Letting $\mathbf{H}$ be a matrix with ith row containing $\mathbf{h}(\boldsymbol{\theta}_i)^\top$ leads to the matrix Normal distribution as a generalisation of the multivariate Gaussian:

$$\mathbf{Y}|\mathbf{B}, \boldsymbol{\Sigma}, \mathbf{r} \sim \mathrm{MN}(\mathbf{HB}, \boldsymbol{\Sigma} \otimes \mathbf{K}_{\boldsymbol{\Theta}\boldsymbol{\Theta}}) \tag{B.1}$$

which combined with the prior in (3.47) gives a full posterior for the latent variable of the form:

$$\boldsymbol{\zeta}(\cdot)|\mathbf{B}, \boldsymbol{\Sigma}, \mathbf{r}, \mathbf{Y} \sim \mathcal{N}(\boldsymbol{\mu}(\cdot), v(\cdot,\cdot)\boldsymbol{\Sigma}) \tag{B.2}$$

where $\boldsymbol{\mu}(\cdot)$ and $v(\cdot,\cdot)$ can be found using standard Gaussian identities to be:

$$\boldsymbol{\mu}(\boldsymbol{\theta}^*) = \mathbf{B}^\top \mathbf{h}(\boldsymbol{\theta}^*) + (\mathbf{Y} - \mathbf{HB})^\top \mathbf{K}_{\boldsymbol{\Theta}\boldsymbol{\Theta}}^{-1} \mathbf{k}_{\boldsymbol{\Theta}\boldsymbol{\theta}^*} \tag{B.3}$$

$$v(\boldsymbol{\theta}^*, \boldsymbol{\theta}^*) = k_{\boldsymbol{\theta}^*\boldsymbol{\theta}^*} - \mathbf{k}_{\boldsymbol{\theta}^*\boldsymbol{\Theta}} \mathbf{K}_{\boldsymbol{\Theta}\boldsymbol{\Theta}}^{-1} \mathbf{k}_{\boldsymbol{\Theta}\boldsymbol{\theta}^*} \tag{B.4}$$

$\mathbf{B}$ and $\boldsymbol{\Sigma}$ can be integrated out of the model, giving a process of the following form:

$$\boldsymbol{\zeta}(\cdot)|\boldsymbol{\Sigma}, \mathbf{r}, \mathbf{Y} \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}(\cdot), \tilde{v}(\cdot,\cdot)\boldsymbol{\Sigma}) \tag{B.5}$$

where

$$\tilde{\boldsymbol{\mu}}(\boldsymbol{\theta}^*) = \hat{\mathbf{B}}^\top \mathbf{h}(\boldsymbol{\theta}^*) + (\mathbf{Y} - \mathbf{H}\hat{\mathbf{B}})^\top \mathbf{K}^{-1} \mathbf{k}_{\boldsymbol{\Theta}\boldsymbol{\theta}^*} \tag{B.6}$$

$$\tilde{v}(\boldsymbol{\theta}^*, \boldsymbol{\theta}^*) = v(\boldsymbol{\theta}^*, \boldsymbol{\theta}^*) + [\mathbf{h}(\boldsymbol{\theta}^*) - \mathbf{H}^\top \mathbf{K}_{\boldsymbol{\Theta}\boldsymbol{\Theta}}^{-1} \mathbf{k}_{\boldsymbol{\Theta}\boldsymbol{\theta}^*}]^\top (\mathbf{H}^\top \mathbf{K}_{\boldsymbol{\Theta}\boldsymbol{\Theta}}^{-1} \mathbf{H})^{-1} [\mathbf{h}(\boldsymbol{\theta}^*) - \mathbf{H}^\top \mathbf{K}_{\boldsymbol{\Theta}\boldsymbol{\Theta}}^{-1} \mathbf{k}_{\boldsymbol{\Theta}\boldsymbol{\theta}^*}] \tag{B.7}$$

where $\hat{B} = (\mathbf{H}^\top \mathbf{K}_{\boldsymbol{\Theta}\boldsymbol{\Theta}}^{-1} \mathbf{H})^{-1} \mathbf{H}^\top \mathbf{A}^{-1} \mathbf{Y}$. The mean of the above distribution can be used as the emulator.

To sample the hyperparameters of the GP kernel using MCMC, the following prior:

$$\pi(\mathbf{B}, \boldsymbol{\Sigma}, \mathbf{r}) = \pi(\mathbf{r}) \pi(\mathbf{B}, \boldsymbol{\Sigma} | \mathbf{r}) \propto \prod_{i=1}^{p} (1 + r_i^2)^{-1} |\boldsymbol{\Sigma}|^{-\frac{q+1}{2}} \tag{B.8}$$

where p=dimension of inputs and q=number of outputs, is combined with the likelihood from (B.1). Integrating out $\mathbf{B}$ and $\boldsymbol{\Sigma}$ we can sample from the posterior:

$$\pi_{\mathbf{R}}(\mathbf{r} | \mathbf{Y}) \propto \pi_{\mathbf{R}}(\mathbf{r}) |\mathbf{K}_{\boldsymbol{\Theta}\boldsymbol{\Theta}}|^{-\frac{q}{2}} |\mathbf{H}^\top \mathbf{K}_{\boldsymbol{\Theta}\boldsymbol{\Theta}} \mathbf{H}|^{-\frac{q}{2}} |\mathbf{Y}^\top \mathbf{G} \mathbf{Y}|^{-\frac{n-m}{2}} \tag{B.9}$$

where n is the number of training points, m is the number of regression functions and $\mathbf{G} = \mathbf{K}^{-1} - \mathbf{K}^{-1} \mathbf{H} (\mathbf{H}^\top \mathbf{K}_{\boldsymbol{\Theta}\boldsymbol{\Theta}}^{-1} \mathbf{H})^{-1} \mathbf{H}^\top \mathbf{K}_{\boldsymbol{\Theta}\boldsymbol{\Theta}}^{-1}$. Further details on the MOGP can be found in Conti et al. (2009).

B.2 Accounting for uncertainty in the loss function

Let us assume that the trained GP emulator has a predictive mean, $\mu(\cdot)$, and predictive variance, $\nu(\cdot)$. The predictive distribution of output j, obtained from a GP trained on outputs $\boldsymbol{y}^{(j)}$ has the following form:

$$p(\tilde{y}^{(j)} | \boldsymbol{\theta}, \boldsymbol{y}_j) = \mathcal{N}\Big(\tilde{y}^{(j)} | \mu_j(\boldsymbol{\theta}), \nu_j(\boldsymbol{\theta})\Big) \tag{B.10}$$

Inclusion of the predictive uncertainty will be considered for the case of univariate GPs used to emulate the outputs of the simulator. As discussed above, the standard approach to parameter estimation would use the predictive mean as the emulator:

$$L(\boldsymbol{\theta}) = \sum_{j=1}^{m} (y_j^* - \mu_j(\boldsymbol{\theta}))^2. \tag{B.11}$$

However, this loss function ignores the uncertainty associated with our prediction at $\boldsymbol{\theta}$.

For the inclusion of the predictive variance, we can compute the marginal likelihood by integrating over the emulator outputs:

$$p(\boldsymbol{y}^* | \boldsymbol{\theta}, \sigma^2) \;=\; \int p(\boldsymbol{y}^* | \eta, \sigma^2) p(\eta | \boldsymbol{\theta}) d\eta \;=\; \prod_{i=0}^{m} \int p(y_j^* | \eta_j, \sigma^2) p(\eta_i | \boldsymbol{\theta}) d\eta_i \tag{B.12}$$

where conditional independence between the outputs has been assumed. The two probability distributions under the integral are given by (4.3) and (B.10), respectively. The integral in (B.12) is thus a standard Gaussian integral, which has the closed-form solution

(see e.g. Bishop (2006), Section 2.3.3):

$$p(\boldsymbol{y}^*|\boldsymbol{\theta}, \sigma^2) = \prod_{j=0}^{m} N\Big(y_j^*|\mu_j(\boldsymbol{\theta}), \sigma^2 + \nu_j(\boldsymbol{\theta})\Big). \tag{B.13}$$

This gives:

$$\log p(\boldsymbol{y}|\boldsymbol{\theta}, \sigma^2) = -\sum_{j=1}^{m} \frac{\Big(y_j^* - \mu_j(\boldsymbol{\theta})\Big)^2}{2[\sigma^2 + \nu_j(\boldsymbol{\theta})]} - \frac{1}{2}\log(2\pi[\sigma_m^2 + \nu_j(\boldsymbol{\theta})]) \tag{B.14}$$

which takes account of the entire GP posterior. The issue with such an approach is that we increase the computational costs of the parameter estimation. As discussed in Appendix A.1, the predictive mean of a GP requires a computational cost of $\mathcal{O}(n)$, where n is the size of the training set. The predictive variance incurs a computational cost of $\mathcal{O}(n^2)$, which is particularly troublesome when dealing with large datasets. At the same time, we would expect that the information provided by the predictive variance decreases as the training set grows, meaning that (B.14) should only really be used in situations where the training data are sparse.

B.2.1 Nonstationary GP model

A fair exploration of the loss functions with and without predictive variance requires that the emulator accurately accounts for our uncertainty about the function. With a GP model, this uncertainty is linked to the choice of the kernel function. Usually, we assume that this function is stationary, which essentially means that the level of variation in the function is constant over the training input support. For many functions, this assumption is an incorrect one, but given enough training data the effects of a suboptimal kernel choice are mitigated by the information fed in from the data. In cases where our data are sparse, which will be the basis of the loss function comparison in this chapter, we can consider more sophisticated kernels that allow for more flexible function properties. One such kernel that allows for an input dependent lengthscale function was presented in Section 3.4.2. In particular, the kernel from (3.46) is of interest in this study, where lengthscale functions are trained for each individual input dimension as a function of all the other inputs.

Rather than using a GP to model the lengthscale function, a neural network will be used with one hidden layer of 50 nodes. This means that the lengthscale for input

dimension k is defined by the following function:

$$\mathbf{h} = \tanh(\boldsymbol{\theta}\mathbf{W}_1 + \mathbf{b}_1) \tag{B.15}$$

$$\lambda_k = \text{softplus}(\mathbf{h}\mathbf{W}_2 + \mathbf{b}_2) \tag{B.16}$$

where $\mathbf{b}_1$ and $\mathbf{b}_2$ have fixed $\mathcal{N}(0, 5)$ priors. For $\mathbf{W}_1$ and $\mathbf{W}_2$, hierarchical priors are used. Letting $\mathbf{w}_1^{(j)}$ be the jth row of weights in $\mathbf{W}_1$ (in other words the weights out of input j) these were as follows:

$$\mathbf{W}_2 \sim \mathcal{N}(0, \sigma_2^2), \sigma_2^2 \sim \mathcal{IG}(0.001, 0.001)$$
$$\mathbf{w}_1^{(j)} \sim \mathcal{N}(0, \sigma_j^2), \sigma_j^2 \sim \mathcal{IG}(0.001, 0.001)$$

which allows for automatic relevance determination.

Sampling of the weights and variance parameters is performed using a HMC within Gibbs scheme. In the Gibbs step, the weight variances are sampled and then all other parameters in the GP model are sampled using HMC. The noise variance and signal variance could have been moved to the Gibbs step, which would probably improve efficiency of sampling. Convergence of the HMC sampler was only checked visually in traceplots of the lengthscale function (convergence of network weights is widely known to be difficult to achieve).

B.2.2 Comparing the loss functions

A comparison of the parameter estimation when using both objectives is found in Table B.1. We could question why the results are so good given a training set of only 100

Table B.1: Comparison of the loss function with and without GP uncertainty. Including uncertainty gives us a small improvement for small dataset sizes.

	MSE quantiles MSE loss NTR=100				MSE quantiles uncertain loss NTR=100		
	0.25	0.5	0.75		0.25	0.5	0.75
θ_1	5.5×10^{-5}	2.2×10^{-4}	1.1×10^{-3}	θ_1	5.1×10^{-5}	1.8×10^{-4}	6.9×10^{-4}
θ_2	3.6×10^{-5}	2.4×10^{-4}	1.3×10^{-3}	θ_2	3.2×10^{-5}	1.8×10^{-4}	1.0×10^{-3}
θ_3	1×10^{-3}	9.9×10^{-3}	3.3×10^{-2}	θ_3	8.1×10^{-4}	6.8×10^{-3}	3.5×10^{-2}
θ_4	1.4×10^{-3}	1.2×10^{-2}	8.0×10^{-2}	θ_4	1.3×10^{-3}	1.1×10^{-2}	6.0×10^{-2}

points. The answer is in the nonstationarity of the function. As we see in Figure B.1, predictions for the test set, which has been generated uniformly over the entire input space, are very good for nearly all points because most of them are generated from the stiff region of space. The error bars in this plot allow us to see the increase in the predictive variance caused by the changing function variation over the input space. A more thorough explo-

Figure B.1: Assessing the non-stationary GP model, trained on a dataset of size 100, on an out of sample test set.

ration of the loss functions would consider the problem over varying training set sizes. We would expect that the benefit of including uncertainty reduces as the training set size increases, but a comparison for large training sets is very computationally expensive.

Appendix C

Appendix for Chapter 5

C.1 A weighted autoencoder: task specific dimensionality reduction

For the purpose of emulation, we are most interested in the dimension reduction procedure that allows us to best represent the strains of the LV at end of diastole. It turns out that the locations of the strain measurements all lie between the mid ventricle and the basal plane, which is highlighted in Figure C.1. It is postulated that better reconstruction is necessary in this region of the LV than near the apical region. In the main text, it was discussed that the optimal dimension reduction approach would take the simulator into account when reducing the dimension of the LV geometry, but due to computational costs of the simulator, this is not possible. As a compromise, we can introduce weights to different regions of the LV that upweight particular nodes of the LV geometry in the loss function, and use the simulator to optimise these weights. Effectively, this turns the problem of optimising all the parameters of our neural network based on the simulator to a problem of optimising only one weight parameter that downweights the apical region. To assess the viability of this approach, a weight is assigned to the nodes of the LV geometry that are not highlighted in Figure C.1. This downweights these nodes in the loss function for training the autoencoder for dimension reduction, meaning that the trained network should be tailored more towards accurate representation of the region where strains are measured. In Figure C.2 the strain errors are obtained by running the reconstructed left ventricle (obtained in a LOOCV training scheme) in the simulator model. The label 0.3 corresponds to a weighting of 0.3 applied to the apical points in the loss function of the autoencoder therefore reducing their influence on the low dimensional mapping. We see in this plot that adjusting this weight parameter does not greatly affect the accuracy of the resulting forward simulations.

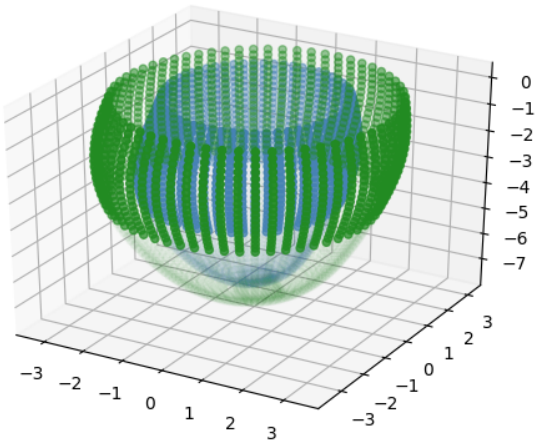


Figure C.1: Strains at end diastole are only measured in the upper portion of the LV, indicated by the less transparent nodes in the above plot.

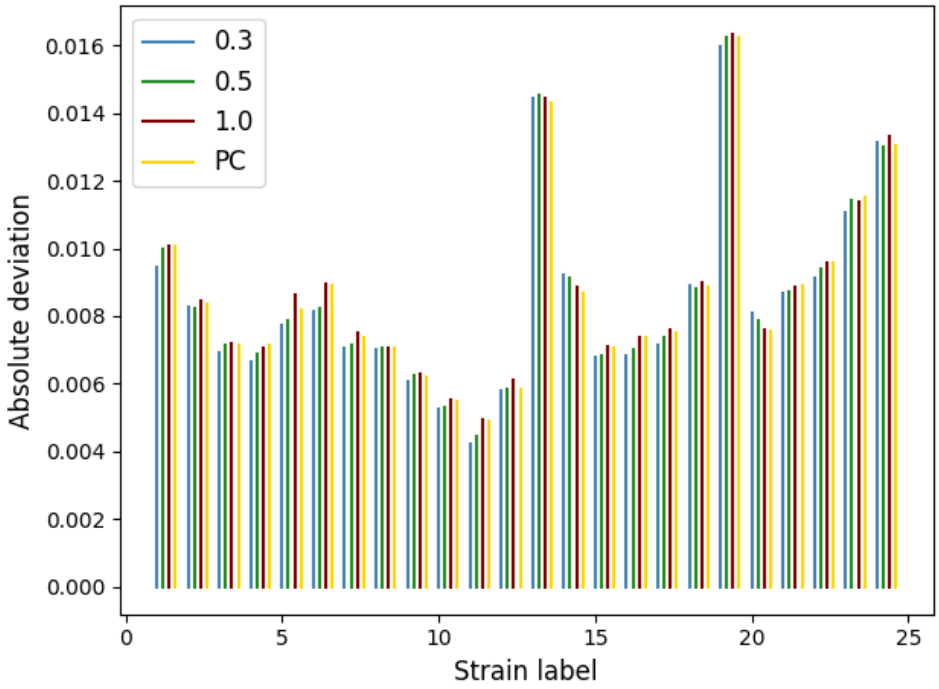


Figure C.2: Simulator strain errors for each weighting compared with the ground truth simulation. There is not a large difference in simulator error when we downweight the apical region.

C.2 Comparing different hidden layer sizes

The hidden layer size of the autoencoder was assumed to be 100 for all experiments in Chapter 5. Other hidden layer sizes were assessed in the model, but were shown to perform similarly, or worse, than a hidden layer of size 100 as we see in Table C.1.

Table C.1: Comparing different hidden layer sizes (H) in the neural network from Figure 5.4. For the nonlinear activation function, the leaky ReLU was used.

H	Median MSE
20	8.2×10^{-3}
50	8.1×10^{-3}
100	8.0×10^{-3}
200	8.1×10^{-3}

Appendix D

Appendix for Chapter 7

D.1 Removing α from the Klotz function

Recall the functional form proposed by Klotz et al. (2006):

$$\tilde{V} \;=\; \frac{V - V_0}{V_{30} - V_0} \quad P \;=\; \alpha \exp\left(\beta \tilde{V}\right) \tag{D.1}$$

Nondimensionalising (D.1) with the definitions

$$\tilde{P} \;=\; \frac{P}{30\text{mmHg}}; \qquad \tilde{\alpha} \;=\; \frac{\alpha}{30\text{mmHg}} \tag{D.2}$$

gives:

$$\tilde{P} \;=\; \tilde{\alpha} \exp\left(\beta \tilde{V}\right) \quad \Rightarrow \quad \tilde{V} \;=\; \frac{1}{\beta} \log\left(\frac{\tilde{P}}{\tilde{\alpha}}\right) \tag{D.3}$$

From (D.1) and (D.2) we have

$$\tilde{V}_{30} \;=\; 1, \qquad \tilde{P}_{30} \;=\; 1 \tag{D.4}$$

which implies, from (D.3):

$$\tilde{\alpha} \;=\; \exp(-\beta) \tag{D.5}$$

$$\tilde{V}_{20} \;=\; \frac{1}{\beta} \log\left(\frac{2}{3}\right) + 1 \tag{D.6}$$

where $\tilde{P}_{20} = 2/3$ has been used, from (D.2).

D.2 Delta method

Suppose we have random variable $\theta \sim \mathcal{N}(\mu, \sigma^2)$. The delta method says that a function of this random variable, $g(\theta)$, is approximately normally distribution with expected value and variance given by:

$$\begin{aligned} \mathbb{E}[g(\theta)] &\approx g(\mu) \\ \text{var}[g(\theta)] &\approx \left[\frac{\partial g}{\partial \theta}\right]^2_{\mu} \sigma^2 \end{aligned}$$

D.3 Gauss error propagation model

The Gauss error propagation model can be used in the multivariate case. For $\boldsymbol{\theta} = (\theta_1, ..., \theta_d)$, $\theta_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, the expected values and variance can be expressed as:

$$\begin{aligned} \mathbb{E}[g(\boldsymbol{\theta})] &\approx g(\boldsymbol{\mu}) \\ \text{var}[g(\boldsymbol{\theta})] &\approx \sum_{i=1}^{d} \left[\frac{\partial g}{\partial \theta_i}\right]^2_{\boldsymbol{\mu}} \sigma_i^2 \end{aligned} \tag{D.7}$$

This model ignores correlations between the variables, which can be accounted for through the multivariate delta method as outlined in Appendix D.6.

D.4 Derivation of (7.12)

We have built two emulators for V_{20} and V_{30}:

$$\begin{aligned} V_{20} &\sim N\left(.\middle|\hat{V}_{20}(\boldsymbol{\theta}, \tilde{\mathbf{h}}), \hat{\sigma}^2_{20}(\boldsymbol{\theta}, \tilde{\mathbf{h}})\right) \\ V_{30} &\sim N\left(.\middle|\hat{V}_{30}(\boldsymbol{\theta}, \tilde{\mathbf{h}}), \hat{\sigma}^2_{30}(\boldsymbol{\theta}, \tilde{\mathbf{h}})\right) \end{aligned} \tag{D.8}$$

where $\hat{V}_{20}(\boldsymbol{\theta}, \tilde{\mathbf{h}})$ and $\hat{V}_{30}(\boldsymbol{\theta}, \tilde{\mathbf{h}})$ are the parameter-dependent means of the two emulators, and $\hat{\sigma}^2_{20}(\boldsymbol{\theta}, \tilde{\mathbf{h}})$, $\hat{\sigma}^2_{30}(\boldsymbol{\theta}, \tilde{\mathbf{h}})$ are the corresponding variances. From (7.1), this gives us an alternative distribution of $\tilde{V}_{20}$, which we denote by $\pi_e(\tilde{V}_{20})$ (where the subscript e stands for "emulator"). To derive an approximate expression for $\pi_e(\tilde{V}_{20})$, we apply the delta method to the standardised volume

$$\tilde{V}_{20} = \frac{V_{20} - V_0}{V_{30} - V_0} \tag{D.9}$$

with derivatives

$$\frac{\partial \tilde{V}_{20}}{\partial V_{20}} = \frac{1}{V_{30} - V_0}, \qquad \frac{\partial \tilde{V}_{20}}{\partial V_{30}} = -\frac{V_{20} - V_0}{(V_{30} - V_0)^2}.$$

This gives:

$$\pi_e(\tilde{V}_{20}) \;=\; N(.|\mu, \sigma^2) \tag{D.10}$$

where

$$\mu \;=\; \frac{\hat{V}_{20}(\boldsymbol{\theta}, \tilde{\mathbf{h}}) - V_0}{\hat{V}_{30}(\boldsymbol{\theta}, \tilde{\mathbf{h}}) - V_0}, \tag{D.11}$$

$$\begin{aligned}\sigma^2 \;&=\; \left(\frac{\partial \tilde{V}_{20}}{\partial V_{20}}\right)^2 \hat{\sigma}^2_{20}(\boldsymbol{\theta}, \tilde{\mathbf{h}}) + \left(\frac{\partial \tilde{V}_{20}}{\partial V_{30}}\right)^2 \hat{\sigma}^2_{30}(\boldsymbol{\theta}, \tilde{\mathbf{h}}) \\ &=\; \left(\frac{1}{\hat{V}_{30}(\boldsymbol{\theta}, \tilde{\mathbf{h}}) - V_0}\right)^2 \hat{\sigma}^2_{20}(\boldsymbol{\theta}, \tilde{\mathbf{h}}) + \left(\frac{\hat{V}_{20}(\boldsymbol{\theta}, \tilde{\mathbf{h}}) - V_0}{(\hat{V}_{30}(\boldsymbol{\theta}, \tilde{\mathbf{h}}) - V_0)^2}\right)^2 \hat{\sigma}^2_{30}(\boldsymbol{\theta}, \tilde{\mathbf{h}})\end{aligned} \tag{D.12}$$

D.5 Details of test generation with MCMC

D.5.1 Overall generation procedure

As stated in the main text, we fix $\theta_1 = \theta_4 = 1$ for test data generation. Given that we now work in the two dimensional $\theta_2 - \theta_3$ space, we choose to build a series of high fidelity emulators for each of the 6 test LV geometries for the sole purpose of test data generation. This is made possible by the fact that no input LV geometry component is required since we build six separate emulators, each of them specific to one of the selected 6 LV geometries. These emulators—referred to as *high fidelity emulators*–allow us to efficiently sample the test data based on highly accurate emulated Klotz likelihood functions. However, these would not be available in the clinical setting and so we make no use of them during parameter inference. For the empirical Klotz prior case, we require emulators of volume at pressures 8 mmHg and 30 mmHg. Generation of data for the non-empirical Klotz curve requires several pressure-volume simulations, allowing us to replicate the upper region of the pressure-volume curve. We use pressures 16, 20 and 25 mmHg, with the volume at pressure 30 mmHg required for the normalisation.

From the MCMC samples of the material properties, we select 5 samples as test points for the emulator comparison, chosen at equally spaced points along the MCMC trajectory for a minimum degree of autocorrelation. We then run the expensive simulator to generate the test outputs. No fictitious noise is added since our sole concern here is whether the introduction of Klotz information reduces parameter inference errors that result from our LV geometry approximation. Once the test data have been generated, the high fidelity emulators are abandoned for the multi geometry emulator and generalised Klotz emulators outlined in Section 7.3.1 in the main text and described in Table 7.2 in the main text.

D.5.2 Non-empirical Klotz prior data generation

Recall that the Klotz curve introduces a relationship between normalised volume, $\tilde{V}$, and pressure P. In order to fairly test the ability of the non-empirical Klotz prior to reduce uncertainty and bias in material property estimates, we require test data that themselves satisfy the assumption of the Klotz curve. Letting $\mathbf{V}$ denote the vector of normalised volumes at pressures 16, 20 and 25 mmHg, we can adopt the likelihood:

$$p(\mathbf{V}|\boldsymbol{\theta}) = p(\tilde{V}_{16}|\boldsymbol{\theta})p(\tilde{V}_{20}|\boldsymbol{\theta})p(\tilde{V}_{25}|\boldsymbol{\theta}) \tag{D.13}$$

$$= \mathcal{N}(\tilde{V}_{16}|\mu_K^{16}, \sigma_K^2)\mathcal{N}(\tilde{V}_{20}|\mu_K^{20}, \sigma_K^2)\mathcal{N}(\tilde{V}_{25}|\mu_K^{25}, \sigma_K^2), \tag{D.14}$$

where $\tilde{V}_m$ denotes the normalised volume at pressure P_m. Assuming a non-informative $\mathcal{U}(0.1, 5)$ prior on each of the material properties we can now sample from the posterior distribution of the material properties, conditional on the normalised volume data. In the above, μ_K^m is the mean normalised Klotz volume at pressure m and $\tilde{V}_m$ is the normalised Klotz volume at pressure P_m. σ_K^2 is the Klotz variance (7.10) in the zero model mismatch case and is obtained from an estimate of the heteroscedastic noise for the model mismatch case. We fix θ_1 and θ_4, leaving a two dimensional parameter vector, $\boldsymbol{\theta}$. Making explicit the parameters that vary in the likelihood function, we express the likelihood as

$$p(\mathbf{V}|\theta_2, \theta_3) = p(\tilde{V}_{16}|\theta_2, \theta_3)p(\tilde{V}_{20}|\theta_2, \theta_3)p(\tilde{V}_{25}|\theta_2, \theta_3). \tag{D.15}$$

Instead of running MCMC using the actual simulator in the likelihood evaluations, it is now more computationally efficient to construct a highly accurate emulator for the different volumes for each of the test cases. These emulations are constructed using a series of 100 simulations run over the two dimensional $\theta_2 - \theta_3$ parameter subspace. This means that evaluation of the likelihood function in (D.15) requires the evaluation of 4 different volume emulators (at pressures 16, 20, 25 and 30 mmHg with V_{30} required for the denominator of the normalisation). We sample from the posterior distribution of the material properties using Delayed rejection adaptive Metropolis (Haario et al., 2006) to obtain a sample of size 10000 from the posterior distribution of the Klotz curve. The volumes at these parameter samples are shown superimposed on the Klotz curve in Figure D.1 for the case with model mismatch term included and Figure D.2 without. In both figures, the dashed line indicates the mean Klotz curve normalised volume at pressures 16 (yellow), 20 (green) and 25 (red).

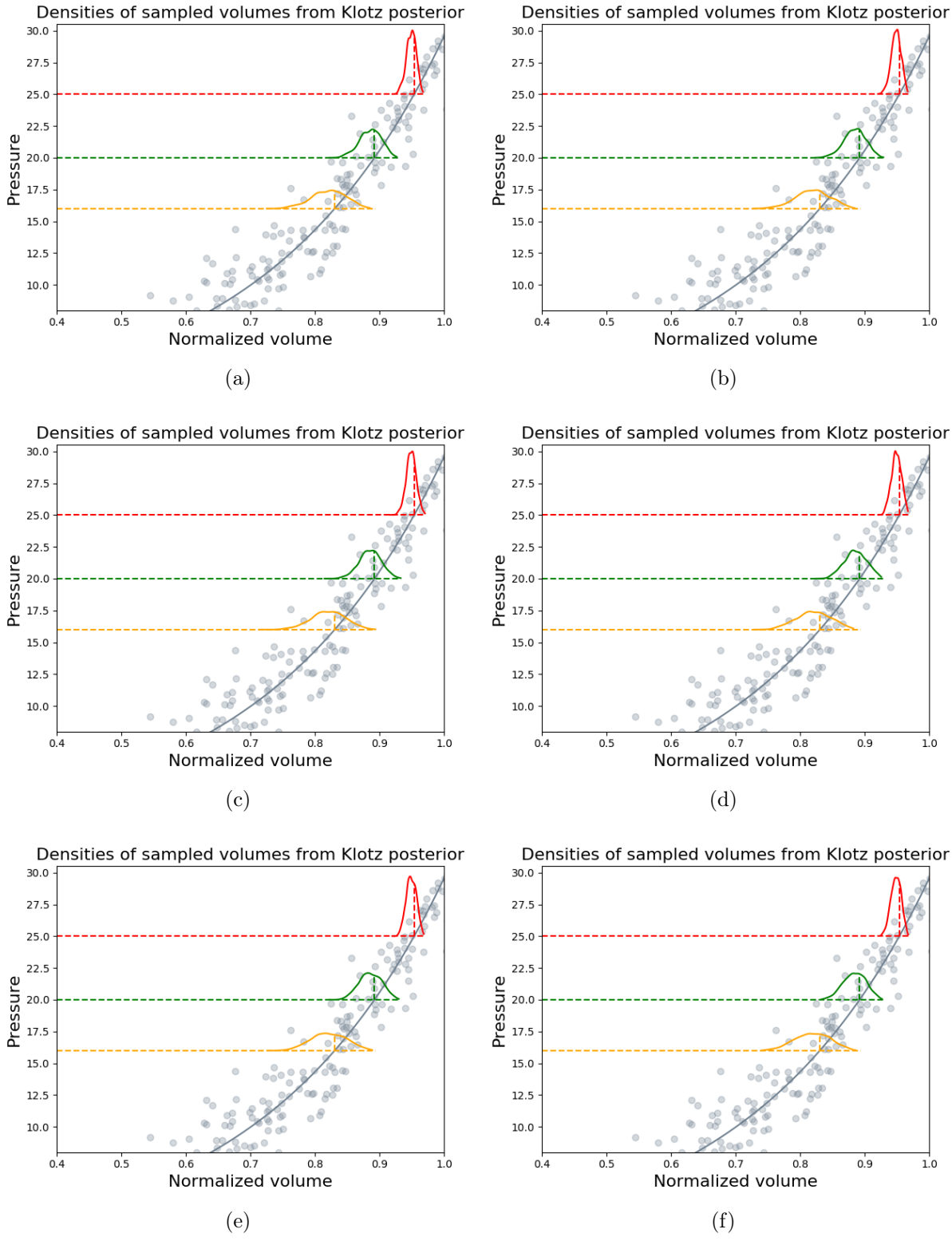


Figure D.1: Densities of normalised volumes obtained from MCMC samples for test points of the nonempirical Klotz prior with model mismatch included. Each plot corresponds to a different test case.

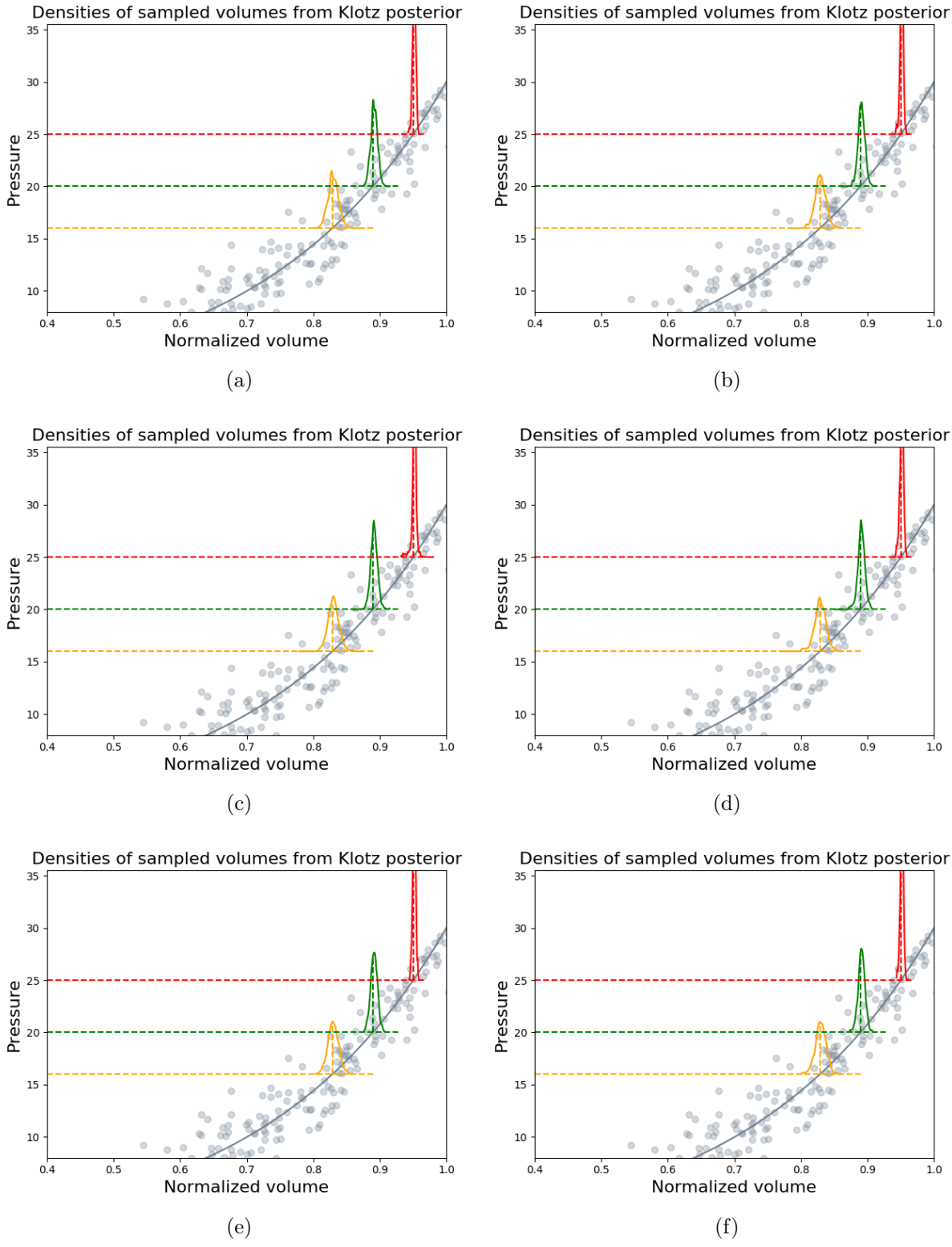


Figure D.2: Densities of normalised volumes obtained from MCMC samples for test points of the nonempirical Klotz prior with no model mismatch. These are each for different test LV geometries.

D.5.3 Empirical Klotz prior data generation

Recall the unnormalised form of the Klotz function, used to derive the empirical Klotz prior:

$$\tilde{V_p} = \frac{\log\left(\frac{p}{\alpha}\right)}{\beta} + \epsilon_p \Rightarrow V_{30} = \frac{V_p - V_0}{\frac{\log\left(\frac{p}{\alpha}\right)}{\beta} + \epsilon_p} + V_0. \tag{D.16}$$

We have a high fidelity emulator for V_{30} which gives, for some material parameters presented at the input, a distribution $\pi_{GP}(V_{30})$ with its associated mean $\mu_{30}(\boldsymbol{\theta}, \mathcal{H})$ and variance $\sigma^2_{30}(\boldsymbol{\theta}, \mathcal{H})$. We also have a high fidelity V_8 emulator with mean $\mu_8(\boldsymbol{\theta}, \mathcal{H})$ and variance $\sigma^2_8(\boldsymbol{\theta}, \mathcal{H})$, which can be plugged into (D.16). Together with the presence of uncertainty in α and β, this provides a distribution on V_{30}, $\pi_{KL}(V_{30})$. The mean $\mu_{KL}(\boldsymbol{\theta}, \mathcal{H})$ and variance $\sigma^2_{KL}(\boldsymbol{\theta}, \mathcal{H})$ are given as follows:

$$\mu_{KL}(\boldsymbol{\theta}, \mathcal{H}) = \frac{V_8 - V_0}{\frac{\log\left(\frac{8}{\alpha}\right)}{\beta} + \epsilon_8} + V_0 \Bigg|_{\alpha=\mu_\alpha, \beta=\mu_\beta, \epsilon_8=0, V_8=\mu_8(\boldsymbol{\theta},\mathcal{H})}, \tag{D.17}$$

$$\sigma^2_{KL}(\boldsymbol{\theta}, \mathcal{H}) = \frac{\partial V_{30}}{\partial \alpha}^2 \sigma^2_\alpha + \frac{\partial V_{30}}{\partial \beta}^2 \sigma^2_\beta + \frac{\partial V_{30}}{\partial \epsilon}^2 \sigma^2_\epsilon + \frac{\partial V_{30}}{\partial V_8}^2 \sigma^2_8(\boldsymbol{\theta}, \mathcal{H}). \tag{D.18}$$

Now, consider the integral of the product of these two densities on V_{30}:

$$\begin{aligned}\int \pi_{KL}(V_{30})\pi_{GP}(V_{30})dV_{30} &= \int \mathcal{N}(V_{30}|\mu_{KL}(\boldsymbol{\theta}, \mathcal{H}), \sigma^2_{KL}(\boldsymbol{\theta}, \mathcal{H}))\mathcal{N}(V_{30}|\mu_{30}(\boldsymbol{\theta}, \mathcal{H}), \sigma^2_{30}(\boldsymbol{\theta}, \mathcal{H}))dV_{30} \\ &= \mathcal{N}(\mu_{30}(\boldsymbol{\theta}, \mathcal{H})|\mu_{KL}(\boldsymbol{\theta}, \mathcal{H}), \sigma^2_{KL}(\boldsymbol{\theta}, \mathcal{H}) + \sigma^2_{30}(\boldsymbol{\theta}, \mathcal{H})).\end{aligned}$$

This is equivalent to the following likelihood function:

$$p(y|\boldsymbol{\theta}) = \mathcal{N}(\mu_{30}(\boldsymbol{\theta}, \mathcal{H}) - \mu_{KL}(\boldsymbol{\theta}, \mathcal{H})\big|0, \sigma^2(\boldsymbol{\theta}, \mathcal{H}) + \sigma^2_{KL}(\boldsymbol{\theta}, \mathcal{H})), \tag{D.19}$$

where y is the difference between the two V_{30} predictions. This can be combined with a non-informative $\mathcal{U}(0.1, 5)$ prior on the material properties to obtain samples satisfying the empirical Klotz prior. For the case with model mismatch, the variance of the $\pi_{KL}(V_{30})$ distribution includes a Klotz heteroscedastic noise variance term.

D.6 Comparing Gauss error model with the first order delta method for the empirical Klotz prior

The Gauss propagation model considers only variances in the function variance calculation. In the multivariate case (such as the empirical Klotz in (7.22)), we could also consider the

multivariate Delta method. The updated variance calculation from (D.7) is:

$$\text{var}[g(\boldsymbol{\theta})] \approx \sum_{i=1}^{d} \left[\frac{\partial g}{\partial \theta_i}\right]_{\boldsymbol{\mu}}^2 \sigma_i^2 + \sum_{i=1}^{d}\sum_{j\neq i} \frac{\partial g}{\partial \theta_i}\frac{\partial g}{\partial \theta_j}\sigma_{ij}^2$$

The multivariate delta method was applied to each test case to compare with the Gauss method. Detailed distributions for each individual test case are found in Figure D.3 and Figure D.4 while summaries based on the IQR of the univariate posterior distributions are found in Figure D.5. We see here that the multivariate delta method consistently gives a slightly smaller IQR in the marginal posterior of θ_3 and θ_2, but the change is small. Importantly, the smaller IQR suggests that the multivariate delta method will reduce the uncertainty in the parameters relative to the non informative prior case.

D.7 Real data results

Here the results for the real data examples are provided by way of stress–stretch curve samples. Figure D.6 provides inference results for the empirical Klotz prior without model mismatch and Figure D.7 shows results when we include the model mismatch term.

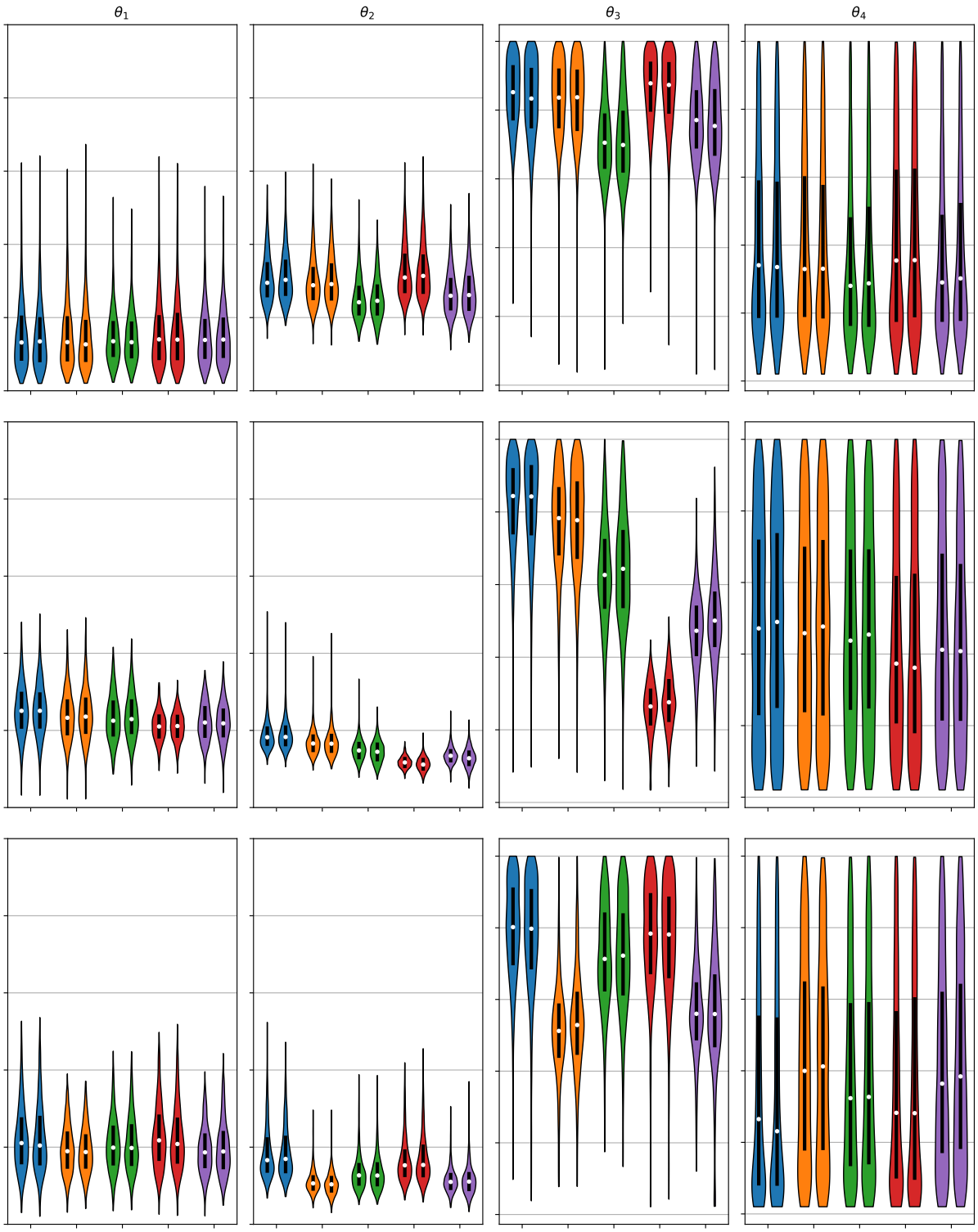


Figure D.3: Comparing the marginal posterior distributions when using the multivariate delta method and the Gauss method. Each row is a different LV geometry (G1,G2 and G3), with each pair of violin plots in a subplot showing the comparison for a particular test case. In each pair the first plot is the multivariate delta result and the second is the Gauss error method. While there are differences between the distributions, these are very minor.

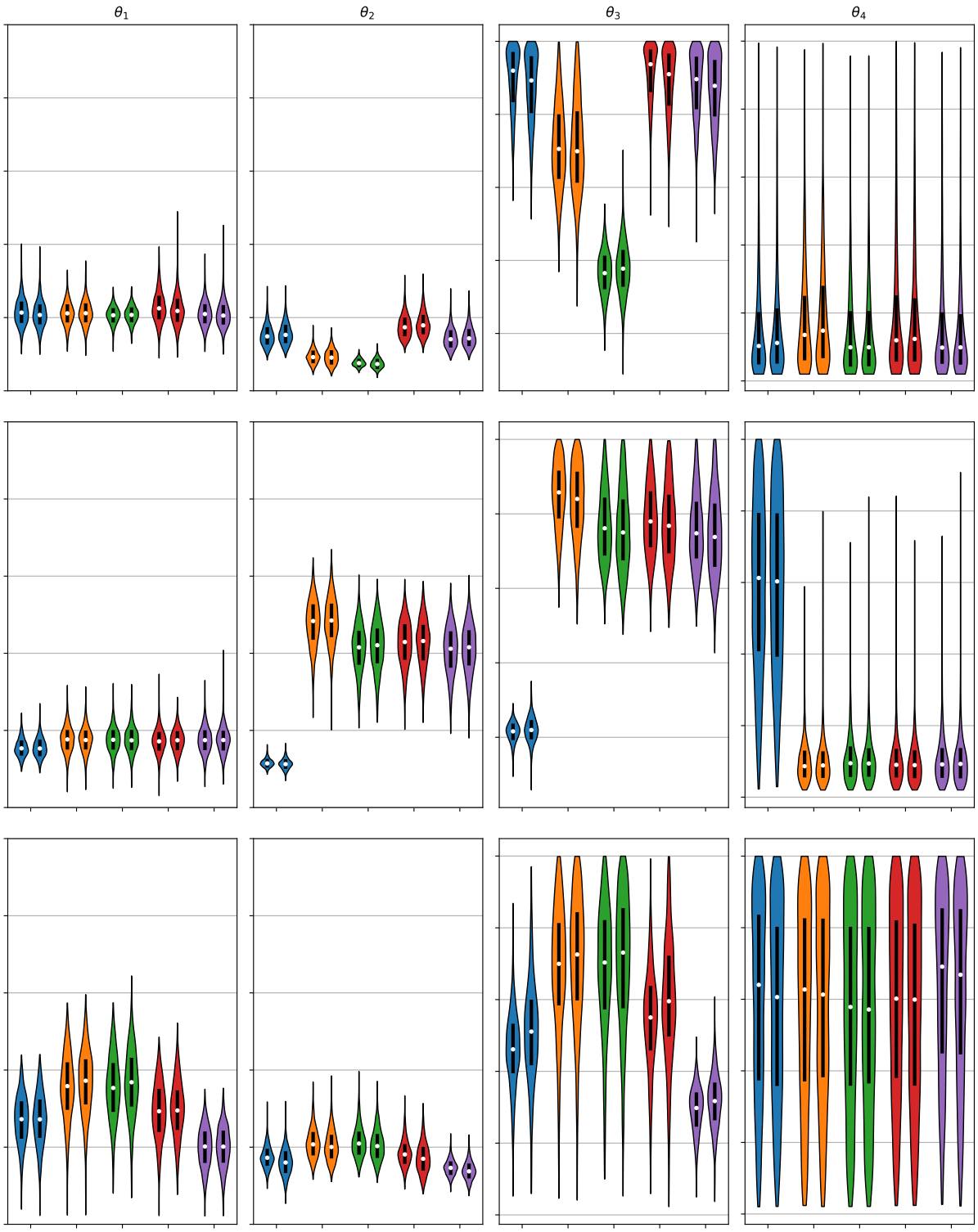


Figure D.4: Comparing the marginal posterior distributions when using the multivariate delta method and the Gauss method. Each row is a different LV geometry (G4,G5,G6), with each pair of violin plots in a subplot showing the comparison for a particular test case. In each pair the first plot is the multivariate delta result and the second is the Gauss error method. While there are differences between the distributions, these are very minor.

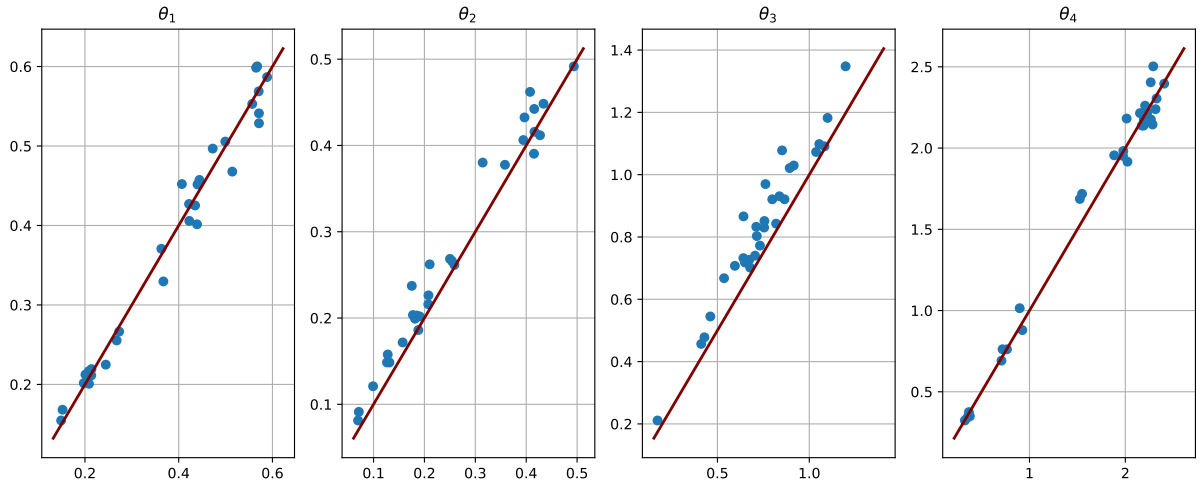


Figure D.5: Comparing the IQR of the marginal posterior distributions for the 30 test cases. The horizontal axis shows the results with the multivariate delta method while the vertical axis shows the Gauss method. As expected, the multivariate delta method gives slightly larger IQR for θ_3 but for all other parameters there is no consistent pattern in IQRs.

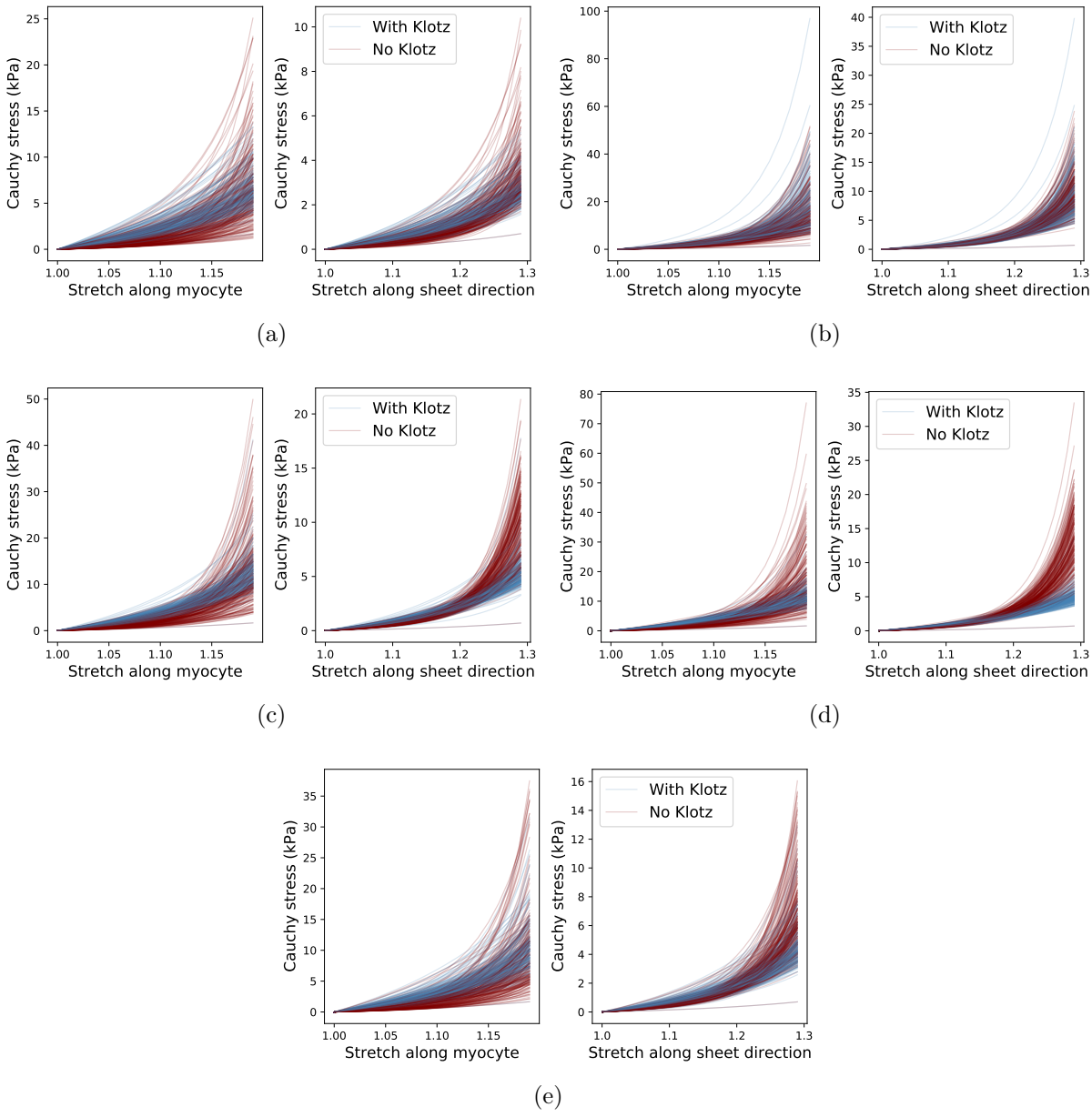


Figure D.6: Inferring material properties based on measured volume and strain data from 5 different healthy volunteers adopting the empirical Klotz prior without model mismatch.

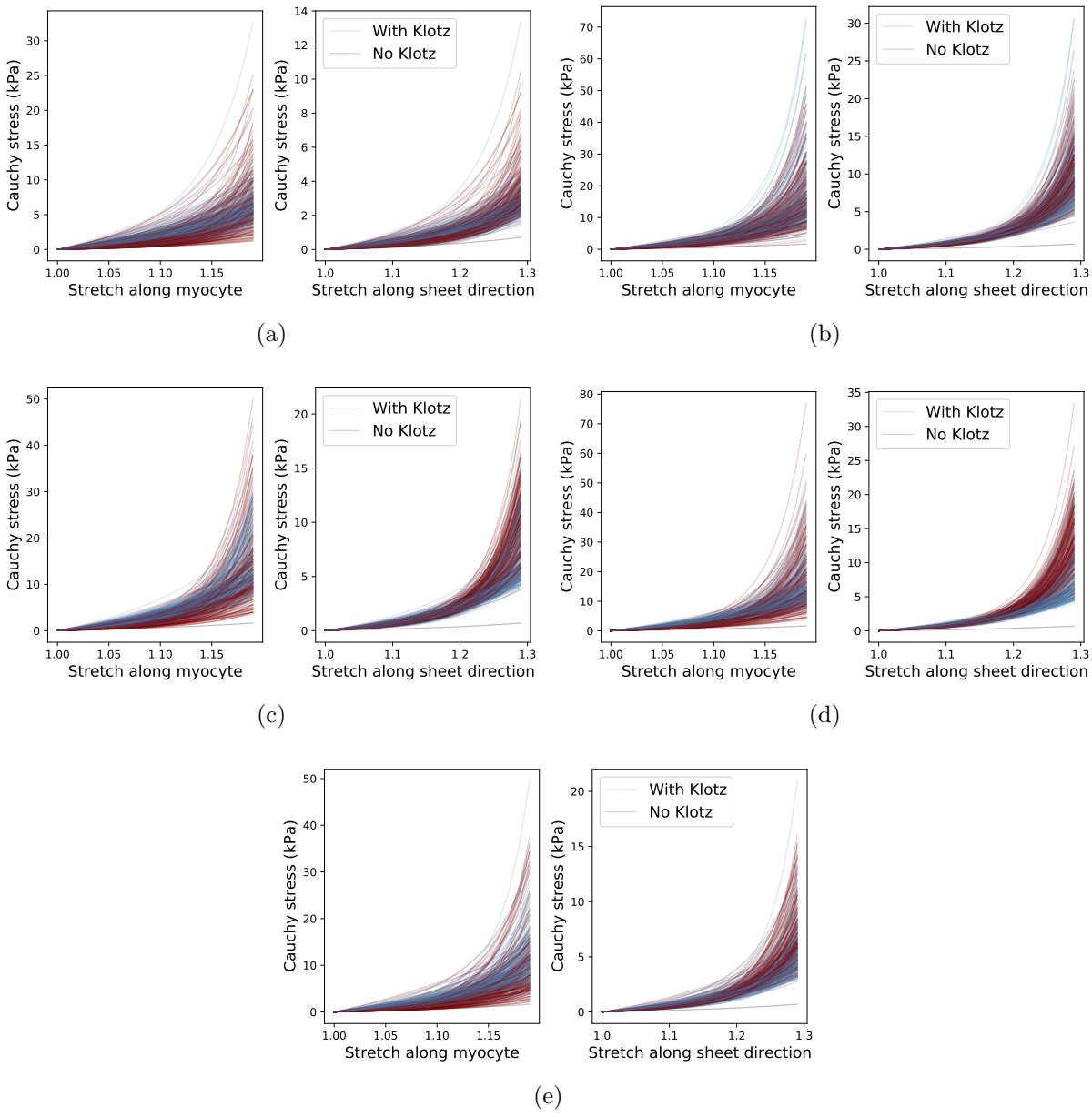


Figure D.7: Inferring material properties based on measured volume and strain data from 5 different healthy volunteers, adopting the empirical Klotz prior with model mismatch.

Appendix E

Appendix for Chapter 9

E.1 Further results of operator effect in the geometry domain

The final reconstructed geometries from the three different operators are found in Figures E.1 and E.2, where the LV geometry of the third operator is compared, in turn, with operators 1 and 2. These were obtained using the procedure outlined in Section 5.2.1.

We can consider the volume of the LV cavity at early diastole in Table E.1. The final row of the table provides a measure of the uncertainty using the approximation of the standard deviation as 1.4826 times the median absolute deviation (MAD). In the table, the maximum volume for each case is highlighted and we see that in all but one example, operator 3 produces LV geometries with the largest volume. These results suggest that a standard deviation of 5 ml is appropriate for the noise on the end diastolic volume, especially since the segmentation of the LV geometry at end diastole is subject to lower errors than early diastole as a result of the clearer boundaries, in the CMR scans, of the inflated LV.

Table E.1: Comparing the volumes of the segmented LV geometries across different operators. The final row provides an approximation of the standard deviation, but this is based on only 3 measurements.

Operator	G1	G2	G3	G4	G5	G6
1	82.0	105.1	85.5	76.9	114.0	**86.6**
2	78.9	97.7	92.2	75.1	114.8	83.4
3	**88.6**	**112.9**	**97.3**	**79.5**	**119.0**	85.9
1.4826×MAD	5.3191	7.5850	6.0877	2.3023	3.0307	1.8453

E.2 Full model calibration results

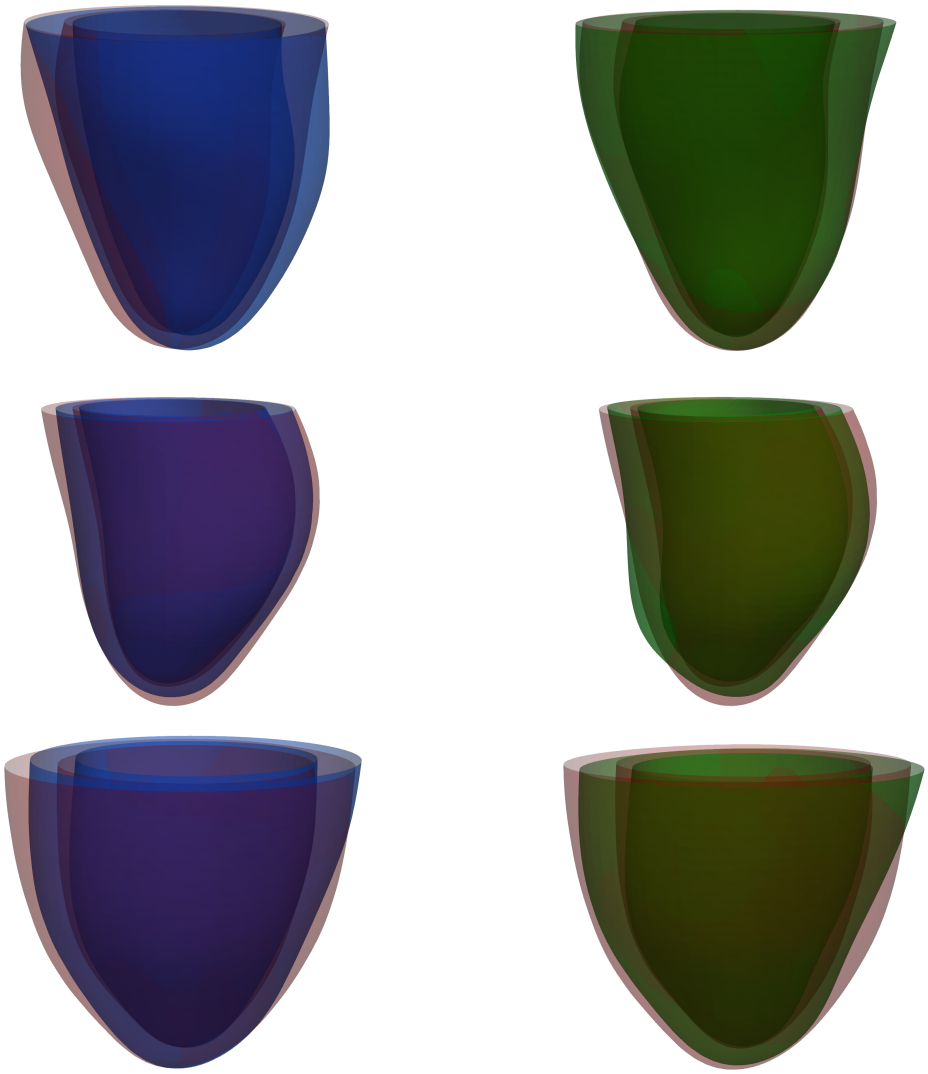


Figure E.1: Comparing the reconstruction of the first 3 LVs. operator 1 vs operator 3 (left) and operator 2 vs operator 3 (right).

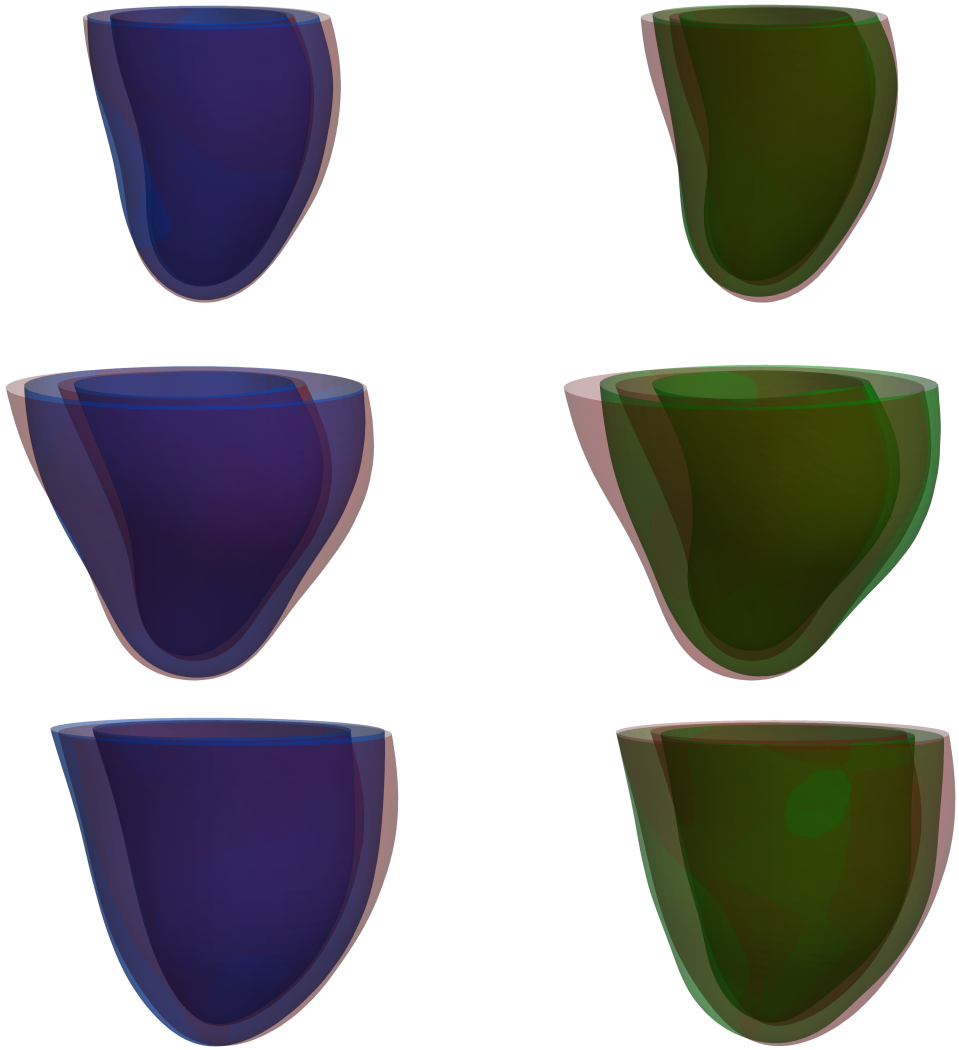


Figure E.2: Comparing the reconstruction of the final 3 LVs. operator 1 vs operator 3 (left) and operator 2 vs operator 3 (right).

Table E.2: The effect of operator on parameter estimation. The results in each table are produced by generating synthetic data using the LV geometry segmented by the main operator and optimising the parameters of the HO law using the simulator with the test geometry. The ground truth parameter values were: $a = 0.60$, $b = 1.30$, $a_\mathrm{f} = 1.00$, $b_\mathrm{f} = 3.60$ and values in the table highlighted in bold differ from these by greater than 0.1.

Main operator: 1					
ID	Operator	a	b	a_f	b_f
G1	2	0.55	1.28	0.93	3.58
	3	0.90	1.50	0.62	3.57
	5PC	0.77	1.21	0.91	3.57
	10PC	0.61	1.30	1.00	3.59
G2	2	0.45	0.98	1.26	3.61
	3	0.51	1.25	1.07	3.87
	5PC	0.60	1.30	1.00	3.60
	10PC	0.62	1.30	1.03	3.61
G3	2	0.60	1.30	0.99	3.59
	3	0.59	1.30	0.98	3.59
	5PC	0.57	1.29	0.96	3.59
	10PC	0.61	1.30	0.96	3.60
G4	2	0.52	1.26	0.91	3.57
	3	0.58	1.30	0.97	3.59
	5PC	0.64	1.33	0.93	3.58
	10PC	0.60	1.30	1.01	3.61
G5	2	0.47	1.26	0.87	3.55
	3	0.56	1.29	0.96	3.59
	5PC	0.60	1.31	1.02	3.59
	10PC	0.61	1.31	1.01	3.61
G6	2	0.62	1.30	1.02	3.61
	3	0.61	1.30	1.01	3.60
	5PC	0.68	1.54	0.82	3.76
	10PC	0.60	1.31	0.99	3.60

Table E.3: The effect of operator on parameter estimation. The results in each table are produced by generating synthetic data using the LV geometry segmented by the main operator and optimising the parameters of the HO law using the simulator with the test geometry. The ground truth parameter values were: $a = 0.60$, $b = 1.30$, $a_\mathrm{f} = 1.00$, $b_\mathrm{f} = 3.60$ and values in the table highlighted in bold differ from these by greater than 0.1.

Main operator: 2					
ID	Operator	a	b	a_f	b_f
G1	1	0.65	1.32	1.07	3.63
	3	0.60	1.29	0.97	3.60
	5PC	0.62	1.31	1.02	3.61
	10PC	0.62	1.29	1.01	3.60
G2	1	0.88	1.51	0.78	3.52
	3	0.62	1.35	1.01	3.63
	5PC	0.64	1.36	1.02	3.60
	10PC	0.63	1.32	1.03	3.62
G3	1	0.60	1.30	1.01	3.61
	3	0.62	1.34	0.94	3.65
	5PC	0.59	1.30	0.97	3.59
	10PC	0.62	1.32	0.95	3.62
G4	1	0.67	1.33	1.11	3.66
	3	0.65	1.32	1.08	3.64
	5PC	0.58	1.32	0.99	3.58
	10PC	0.53	1.27	1.00	3.59
G5	1	0.34	1.72	1.45	3.85
	3	0.68	1.33	1.11	3.64
	5PC	0.68	1.34	1.07	3.63
	10PC	0.91	1.53	0.74	3.42
G6	1	0.59	1.30	0.98	3.59
	3	0.56	1.30	1.00	3.61
	5PC	0.59	1.29	0.99	3.59
	10PC	0.57	1.30	0.98	3.60

Table E.4: The effect of operator on parameter estimation. The results in each table are produced by generating synthetic data using the LV geometry segmented by the main operator and optimising the parameters of the HO law using the simulator with the test geometry. The ground truth parameter values were: $a = 0.60$, $b = 1.30$, $a_\mathrm{f} = 1.00$, $b_\mathrm{f} = 3.60$ and values in the table highlighted in bold differ from these by greater than 0.1.

Main operator: 3					
ID	Operator	a	b	a_f	b_f
G1	1	0.30	1.19	1.35	3.72
	2	0.60	1.30	1.02	3.61
	5PC	0.60	1.30	1.04	3.58
	10PC	0.57	1.30	0.98	3.60
G2	1	0.56	1.30	0.95	3.59
	2	0.59	1.29	0.96	3.60
	5PC	0.57	1.31	0.97	3.61
	10PC	0.57	1.32	1.02	3.60
G3	1	0.60	1.30	1.02	3.61
	2	0.42	1.34	1.09	3.72
	5PC	0.58	1.28	0.92	3.56
	10PC	0.57	1.29	0.96	3.60
G4	1	0.62	1.31	1.02	3.61
	2	0.55	1.26	0.92	3.56
	5PC	0.59	1.30	0.96	3.60
	10PC	0.58	1.29	1.00	3.59
G5	1	0.64	1.31	1.04	3.61
	2	0.83	1.48	0.64	3.10
	5PC	0.62	1.31	1.04	3.63
	10PC	0.60	1.30	1.00	3.60
G6	1	0.60	1.29	0.98	3.59
	2	0.63	1.35	0.99	3.58
	5PC	0.58	1.29	0.98	3.59
	10PC	0.59	1.30	0.95	3.63

Appendix F

Emulation with in clinic correction

F.1 Bayesian last layer

Recall the in clinic emulation task: equipped with a set of training LV geometries and their corresponding simulations, learn to approximate the simulator function for a new left ventricle geometry. We anticipate that the simulator model does not change drastically depending on the LV geometry, motivating the idea of a quick in clinic correction based on only a few simulations. For instance, we know that no matter the geometry, the simulator model has a standard sigmoidal shape as a result of the H-O model. The method proposed here proposes to learn a set of basis functions that allows us to represent the simulator for a wide range of training LV geometries. Given a set of simulations obtained using a test LV geometry (called the *high fidelity simulations*), these basis functions can be fitted to the new data and used as an emulator.

Let the simulator for training LV geometry i be denoted by $\zeta^{(i)}(\boldsymbol{\theta})$ where the dependence on the LV geometry is removed because for each individual LV geometry, this component is fixed. We can generate several ensembles of training data:

$$\mathcal{D} = \{D_i\}_{i=1}^{m}, \text{ where } D_i = \{\boldsymbol{\theta}_j^{(i)}, \boldsymbol{\zeta}^{(i)}(\boldsymbol{\theta}_j^{(i)})\}_{j=1}^{n} \tag{F.1}$$

where the design in material parameter space is unique for each LV geometry. Suppose that we have a new LV geometry for which we wish to make predictions from the simulator, $\boldsymbol{\zeta}^*(\boldsymbol{\theta})$. The proposed scheme trains a neural network to learn the simulator functions of the training LVs, training in a way that allows for the model to be fitted to a new simulator function given some high fidelity simulations obtained online.

Essentially, the proposed scheme treats the problem as one of transfer learning. Here, we transfer knowledge gained about the function from the set of training LV geometries to a new unseen LV geometry. To do so, the neural network in Figure F.1 is trained on $\mathcal{D}$. During training, the first $k-1$ layers of the network are the same for each batch of

data, but the final layer of weights (highlighted in red) is unique to each D_i. By training in this manner, the first $n-1$ layers learn a flexible set of basis functions that allow us to fit each of the n separate simulator functions.

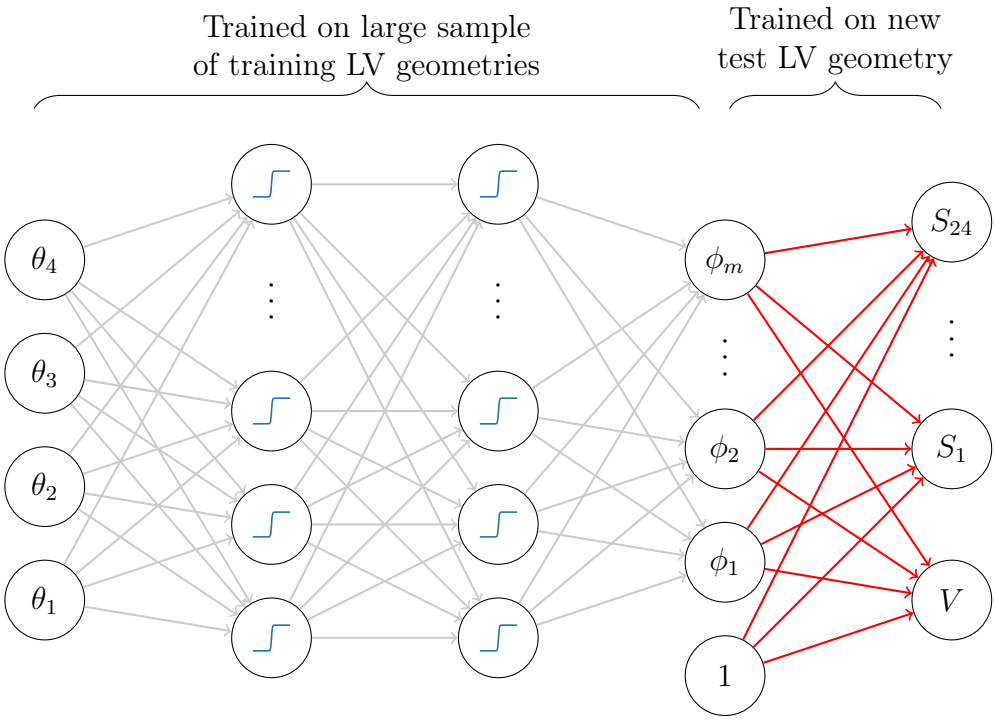


Figure F.1: The neural linear model or Bayesian last layer model is a neural network where the parameters in the first n-1 layers (black arrows above) are optimised and we account for uncertainty in the parameters of the final layer. If we train all weights up to the final hidden layer using simulations from training LV geometries, we learn a set of basis functions that allow us to infer the function for a new LV geometry by inferring the weights in only the final layer of the network. Activations in the first $n-1$ hidden layers are tanh functions to allow for non linear basis functions. However, the functions in the final hidden layer can be linear. Note that each hidden layer contains a bias parameter but this has only been shown for the final layer (the node labelled 1).

Placing a Gaussian prior on the weights in the final layer, these can be integrated out of the model in the same way as a standard Bayesian linear regression model. For a new test simulator, $\boldsymbol{\zeta}^*(\boldsymbol{\theta})$, we wish to tune the basis functions to the new simulator. Given a set of high fidelity simulations $\{\boldsymbol{\theta}_i, \boldsymbol{\zeta}^*(\boldsymbol{\theta}_i)\}_{i=1}^q$, the hyperparameters of the Gaussian priors are sampled and the predictive distribution is found in closed form. For the exact form of the marginal likelihood, and the predictive distribution of the model, see Ober and Rasmussen (2019). This approach is conceptually the same as the neural linear model or Bayesian last layer (BLL) model (Ober and Rasmussen, 2019; Watson et al., 2021; Thakur et al., 2020), which has received increasing interest in the literature in recent years. In particular, the model in Figure F.1 is similar to that of Thakur et al. (2020) where the authors introduced a set of auxiliary functions that the NLM targets during training, learning a flexible suite

of functions that are based on a priori knowledge of the function of interest. In the current model, the training cases allow us to learn a set of appropriate basis functions. Note that the BLL is essentially a GP model with the kernel defined by the trained basis functions (Rasmussen and Williams, 2006, Section 2.1.2).

F.2 Result

In this section, the BLL idea will first of all be applied to a simple toy problem. The BLL model will then be applied to the unseen LV geometries and compared with the multi fidelity model.

F.2.1 Applying to a toy problem

In this section, the method is applied to a simple 1D toy problem. The function of interest is of the form:

$$\zeta(x) = \sin(bx)\exp(ax) \tag{F.2}$$

For training, 30 different a, b combinations were sampled from $\mathrm{u}(-0.6, -0.2)$ and $\mathrm{u}(0.5, 1.1)$ priors respectively. For each pair of values, 30 training simulations were obtained, and the functions are plotted in grey in the bottom left of Figure F.2, labelled as training functions in the legend. The target functions takes values $a = -0.3, b = 1$. Using the neural network from Figure F.1, a network with two hidden layers of size 100 and 20 basis functions was trained on the training data. The final trained basis functions are visualised in the bottom right plot of Figure F.2 as grey curves. We see that where the function is more varied (to the left of the horizontal axis) the basis functions are also more varied. Using 4 simulations from the function of interest, as indicated by red points in Figure F.2, an emulator is fitted that fits the true function far better than the standard GP model in the top row.

F.2.2 Applying to the LV simulator

Of interest is the improvement gained using the BLL model compared with a standard 5PC generalised emulator. Three different test cases where the generalised emulator performs poorly were considered for comparison here. The predictions of a subset of test points are shown in Figure F.3, where the 5PC emulator has been used and the BLL with varying numbers of training simulations. The simulations for the BLL were generated in log-Uniform space, with a filter than filtered out simulations with circumferential strains of magnitude greater than 0.35. Values greater than this were deemed to be unrealistic given

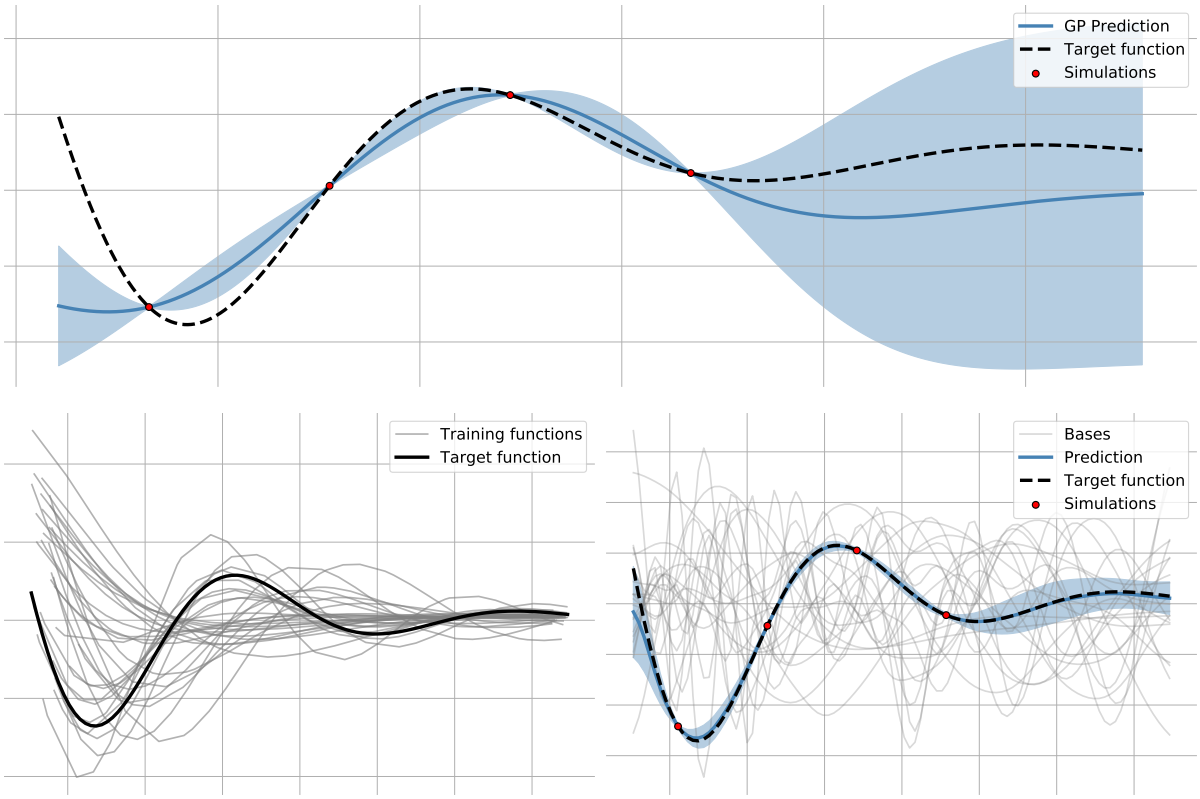


Figure F.2: Applying the Transferred Bayesian last layer network to a toy problem. Top row: standard Gaussian process with squared exponential kernel. The prior does not include enough information to allow a close fit to the simulator. In the bottom row, the basis functions are trained on functions similar to the one we are ultimately interested in (these training functions are represented by grey curves in the plot on the left). This allows us to learn the function with access to few function evaluations, using a set of learnt basis functions (indicated by grey curves on the right).

the values seen in measured datasets. In total, 80 forward simulations were used for each training LV geometry, with 120 LV geometries used in total. Error bars are provided for the BLL, since the Bayesian approach allows us to quantify the uncertainty in the function. With only 20 simulations, the BLL vastly outperforms the 5PC emulator. In Figures F.4 to F.6, the prediction error is compared across individual model outputs (on the log scale), with the bars showing the median absolute error and error bars providing the 0.25 and 0.75 quantiles. The BLL universally outperforms the generalised emulator for all outputs across the test cases, but the performance does appear to hit a plateau. This is as expected given that the basis functions are fixed beforehand and limit the capacity of the model.

F.3 Next steps

As mentioned in the main text, this study is very preliminary. Nonetheless, the results for the three LVs considered here are very encouraging, suggesting that with as little as 20 simulations we can achieve very good accuracy for these test cases that were hand-picked due to their lack of compatibility with the generalised emulator model. There are various directions that this work could be continued in the future. One of these is to consider the design; in the current work the simulations were taken from a log-Uniform design but a more carefully selected design could give better performance. An alternative model could be constructed using Gaussian processes. While the construction of a model like the one considered here would be challenging with Gaussian processes, a multi-fidelity GP model has been considered. This was constructed as follows: take the generalisable emulator with 5 PCs in the input space as a low fidelity model and, using simulations obtained from the exact LV geometry train a multi fidelity GP model like the one proposed by Perdikaris et al. (2017) (this is essentially the same as the model proposed by Kennedy and O'Hagan (2000) but with a non-linear transformation from low to high fidelity). In some preliminary simulations, this was shown to perform worse than the BLL for small numbers of high fidelity simulations but the unlimited expressiveness of the GP means that the predictive error does not reach a plateau as quickly as the BLL.

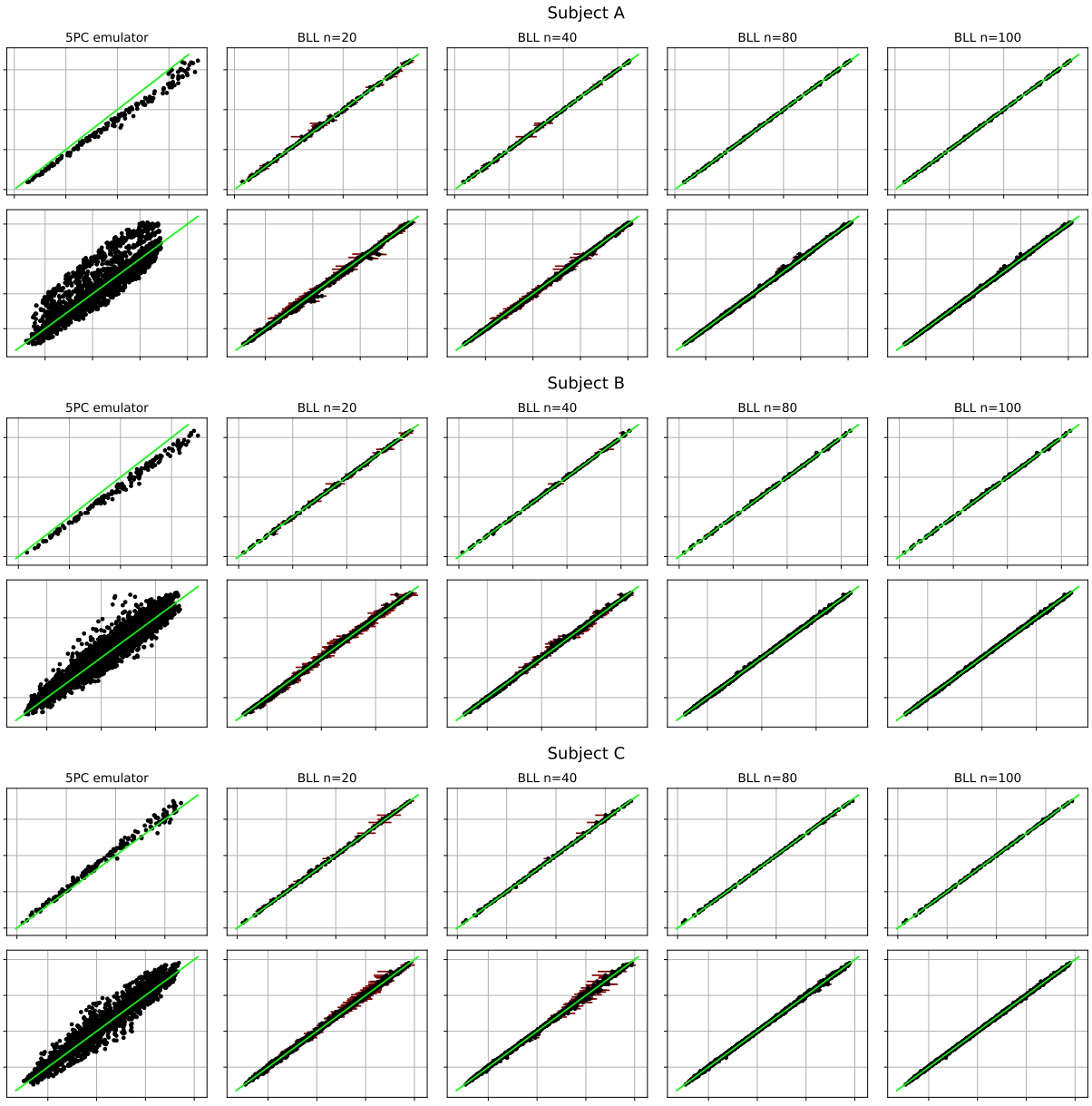


Figure F.3: Comparing the BLL with the generalised emulator that uses five PCs for three cases where the generalised emulator performs poorly. Each pair of rows give results for a separate subject (as titled) with the volume predictions in the top row and the strain predictions in the bottom row, plotted against the ground truth simulations.

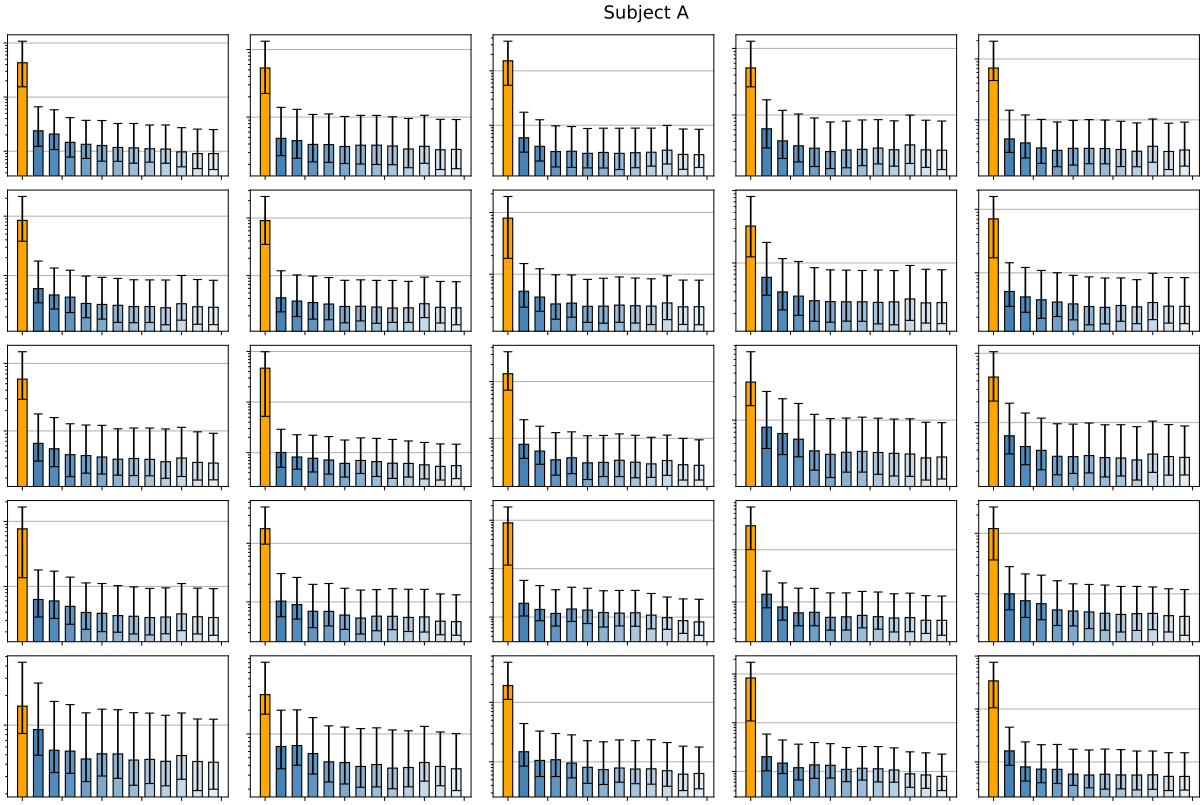


Figure F.4: Errors in each output for subject A. The (orange) leftmost bar in each subplot shows the median error of the generalised emulator, the second to last bars give the results of the BLL model for increasing dataset sizes: (20,30,40,50,60,70,80,90,100,200,300,400). Results are in log scale.

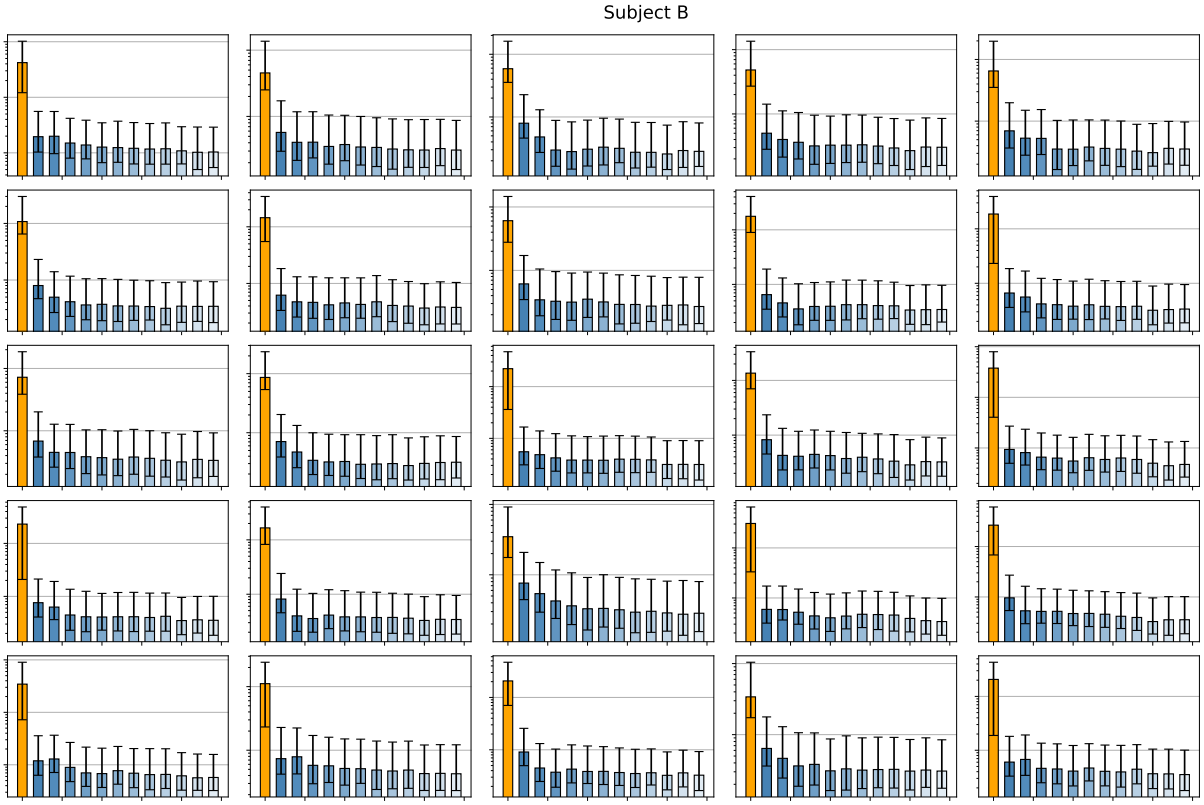


Figure F.5: Errors in each output for subject B. The (orange) leftmost bar in each subplot shows the median error of the generalised emulator, the second to last bars give the results of the BLL model for increasing dataset sizes: (20,30,40,50,60,70,80,90,100,200,300,400). Results are in log scale.

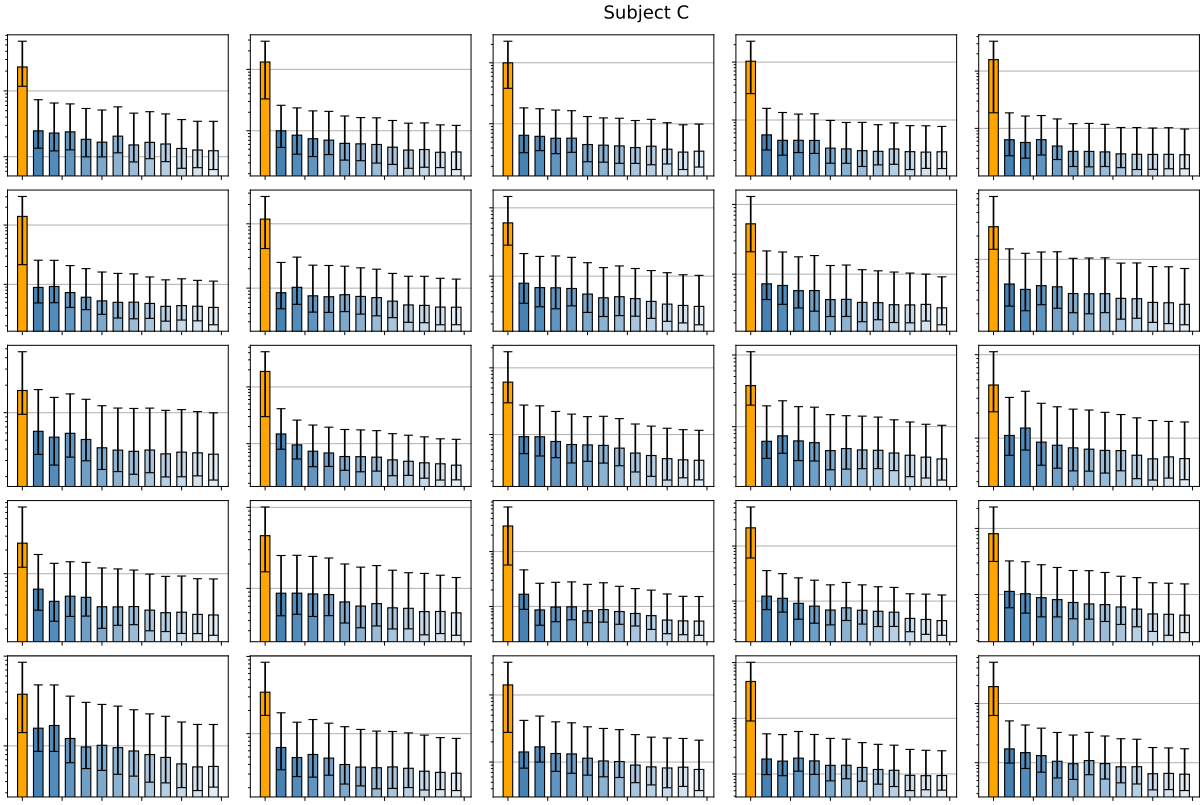


Figure F.6: Errors in each output for subject C. The (orange) leftmost bar in each subplot shows the median error of the generalised emulator, the second to last bars give the results of the BLL model for increasing dataset sizes: (20,30,40,50,60,70,80,90,100,200,300,400). Results are in log scale.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Al'Aref, S., Singh, G., Baskaran, L., and Metaxas, D. (2020). *Machine Learning in Cardiovascular Medicine*. Academic Press.

Alber, M. S., Tepole, A. B., Cannon, W. R., De, S., Dura-Bernal, S., Garikipati, K. C., Karniadakis, G. E., Lytton, W. W., Perdikaris, P., Petzold, L. R., and Kuhl, E. (2019). Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. *NPJ Digital Medicine*, 2.

Alvarez, M. and Lawrence, N. D. (2009). Sparse convolved gaussian processes for multi-output regression. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 21*, pages 57–64. Curran Associates, Inc.

Andrianakis, I., Vernon, I. R., McCreesh, N., McKinley, T. J., Oakley, J. E., Nsubuga, R. N., Goldstein, M., and White, R. G. (2015). Bayesian history matching of complex infectious disease models using emulation: A tutorial and a case study on hiv in uganda. *PLOS Computational Biology*, 11(1):1–18.

Asner, L., Hadjicharalambous, M., Chabiniok, R., Peressutti, D., Sammut, E., Wong, J., Carr-White, G., Chowienczyk, P., Lee, J., King, A., Smith, N., Razavi, R., and Nordsletten, D. (2016). Estimation of passive and active properties in the human heart using 3d tagged mri. *Biomechanics and Modeling in Mechanobiology*, 15.

Avazmohammadi, R., Soares, J. S., Li, D. S., Raut, S. S., Gorman, R. C., and Sacks, M. S.

(2019). A contemporary look at biomechanical models of myocardium. *Annual Review of Biomedical Engineering*, 21(1):417–442. PMID: 31167105.

Azevedo, P. S., Polegato, B., Minicucci, M., Paiva, S., and Zornoff, L. (2016). Cardiac remodeling: Concepts, clinical impact, pathophysiological mechanisms and pharmacologic treatment. *Arquivos Brasileiros de Cardiologia*, 106:62 – 69.

Bai, W., Sinclair, M., Tarroni, G., Oktay, O., Rajchl, M., Vaillant, G., Lee, A., Aung, N., Lukaschuk, E., Sanghvi, M., Zemrak, F., Fung, K., Paiva, J., Carapella, V., Kim, Y., Suzuki, H., Kainz, B., Matthews, P., Petersen, S., and Rueckert, D. (2018). Automated cardiovascular magnetic resonance image analysis with fully convolutional networks. *Journal of Cardiovascular Magnetic Resonance*, 20.

Baillargeon, B., Rebelo, N., Fox, D. D., Taylor, R. L., and Kuhl, E. (2014). The Living Heart Project: A robust and integrative simulator for human heart function. *European Journal of Mechanics - A/Solids*, 48:38–47.

Baker, R., Peña, J., Jayamohan, J., and Jérusalem, A. (2018). Mechanistic models versus machine learning, a fight worth fighting for the biological community? *Biology Letters*, 14.

Barbarotta, L. and Bovendeerd, P. H. M. (2021). A computational approach on sensitivity of left ventricular wall strains to fiber orientation. In Ennis, D. B., Perotti, L. E., and Wang, V. Y., editors, *Functional Imaging and Modeling of the Heart*, volume 12738, pages 296–304. Springer International Publishing.

Bauer, M., van der Wilk, M., and Rasmussen, C. E. (2016). Understanding probabilistic sparse gaussian process approximations. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 1533–1541. Curran Associates, Inc.

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. (2017). Automatic differentiation in machine learning: A survey. *J. Mach. Learn. Res.*, 18(1):5595âĂŞ5637.

Bayer, J., Blake, R., Plank, G., and Trayanova, N. (2012). A novel rule-based algorithm for assigning myocardial fiber orientation to computational heart models. *Annals of Biomedical Engineering*, 40:2243–2254.

Beck, J. and Guillas, S. (2016). Sequential design with mutual information for computer experiments (mice): Emulation of a tsunami model. *SIAM/ASA J. Uncertain. Quantification*, 4:739–766.

Betancourt, M. (2017). A conceptual introduction to hamiltonian monte carlo. *arXiv: Methodology*.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., USA.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Bottou, L. (1991). Stochastic gradient learning in neural networks.

Bouchard, R. J., Gault, J. H., and Ross, J. (1971). Evaluation of pulmonary arterial end-diastolic pressure as an estimate of left ventricular end-diastolic pressure in patients with normal and abnormal left ventricular performance. *Circulation*, 44(6):1072–1079.

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011). *Handbook of Markov Chain Monte Carlo*. CRC press.

Brynjarsdóttir, J. and O'Hagan, A. (2014). Learning about physical parameters: the importance of model discrepancy. *Inverse Problems*, 30(11):114007.

Buoso, S., Joyce, T., and Kozerke, S. (2021). Personalising left-ventricular biophysical models of the heart using parametric physics-informed neural networks. *Medical Image Analysis*, 71:102066.

Cai, L., Ren, L., Wang, Y., Xie, W., Zhu, G., and Gao, H. (2021). Surrogate models based on machine learning methods for parameter estimation of left ventricular myocardium. *Royal Society Open Science*, 8(1):201121.

Calandra, R., Peters, J., Rasmussen, C. E., and Deisenroth, M. P. (2016). Manifold gaussian processes for regression. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 3338–3345.

Calderhead, B., Girolami, M., and Lawrence, N. D. (2008). Accelerating Bayesian inference over nonlinear differential equations with Gaussian processes. *Neural Information Processing Systems (NIPS)*, 22.

Campos, J. O., Sundnes, J., dos Santos, R. W., and Rocha, B. M. (2019). Effects of left ventricle wall thickness uncertainties on cardiac mechanics. *Biomechanics and Modeling in Mechanobiology*, 18(5):1415–1427.

Campos, J. O., Sundnes, J., dos Santos, R. W., and Rocha, B. M. (2020). Uncertainty quantification and sensitivity analysis of left ventricular function during the full cardiac cycle. *Philosophical Transactions A*, 378:20190381.

Caruana, R., Lawrence, S., and Giles, C. L. (2000). Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *NIPS*.

Chabiniok, R., Wang, V. Y., Hadjicharalambous, M., Asner, L., Lee, J., Sermesant, M., Kuhl, E., Young, A. A., Moireau, P., Nash, M. P., Chapelle, D., and Nordsletten, D. A. (2016). Multiphysics and multiscale modelling, data-model fusion and integration of organ physiology in the clinic: ventricular cardiac mechanics. In *Interface Focus*.

Chib, S. and Greenberg, E. (1995). Understanding the metropolis-hastings algorithm. *The American Statistician*, 49(4):327–335.

Cho, Y. and Saul, L. K. (2009). Kernel methods for deep learning. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, NIPSâĂŹ09, page 342âĂŞ350, Red Hook, NY, USA. Curran Associates Inc.

Clayton, R. H., Aboelkassem, Y., Cantwell, C. D., Corrado, C., Delhaas, T., Huberts, W., Lei, C. L., Ni, H., Panfilov, A. V., Roney, C., and dos Santos, R. W. (2020). An audit of uncertainty in multi-scale cardiac electrophysiology models. *Philosophical Transactions of the Royal Society A*, 378(2173).

Colebank, M. J., Qureshi, M. U., and Olufsen, M. S. (2019). Sensitivity analysis and uncertainty quantification of 1âĂŘd models of pulmonary hemodynamics in mice under control and hypertensive conditions. *International Journal for Numerical Methods in Biomedical Engineering*, page e242.

Conrad, P. R., Marzouk, Y. M., Pillai, N. S., and Smith, A. (2016). Accelerating asymptotically exact mcmc for computationally intensive models via local approximations. *Journal of the American Statistical Association*, 111(516):1591–1607.

Conti, S., Gosling, J. P., Oakley, J. E., and O'Hagan, A. (2009). Gaussian process emulation of dynamic computer codes. *Biometrika*, 96(3):663–676.

Conti, S. and O'Hagan, A. (2010). Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, 140(3):640 – 651.

Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1988). A Bayesian Approach to the Design and Analysis of Computer Experiments. Technical report, Oak Ridge National Laboratory.

Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991). Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association*, 86(416):953–963.

Dabiri, Y., Van der Velden, A., Sack, K. L., Choy, J. S., Kassab, G. S., and Guccione, J. M. (2019). Prediction of left ventricular mechanics using machine learning. *Frontiers in Physics*, 7:117.

Dalton, D., Lazarus, A., and Husmeier, D. (2020). Comparative evaluation of different emulators for cardiac mechanics.

Dalton, D., Lazarus, A., Rabbani, A., Gao, H., and Husmeier, D. (2021). Graph neural network emulation of cardiac mechanics. *Conference: 3rd International Conference on Statistics: Theory and Applications (ICSTA'21)*, 1.

Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2933–2941, Cambridge, MA, USA. MIT Press.

Davies, V., Noè, U., Lazarus, A., Gao, H., Macdonald, B., Berry, C., Luo, X., and Husmeier, D. (2019). Fast parameter inference in a biomechanical model of the left ventricle by using statistical emulation. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 68(5):1555–1576.

Dette, H. and Pepelyshev, A. (2010). Generalized latin hypercube design for computer experiments. *Technometrics*, 52(4):421–429.

Di Achille, P., Harouni, A., Khamzin, S., Solovyova, O., Rice, J. J., and Gurev, V. (2018). Gaussian process regressions for inverse problems and parameter searches in models of ventricular mechanics. *Frontiers in Physiology*, 9:1002.

Dokos, S., Smaill, B. H., Young, A. A., and LeGrice, I. J. (2002). Shear properties of passive ventricular myocardium. *American Journal of Physiology-Heart and Circulatory Physiology*, 283(6):H2650–H2659.

Dondelinger, F., Filippone, M., Rogers, S., and Husmeier, D. (2013). ODE parameter inference using adaptive gradient matching with Gaussian processes. *Proceedings of Machine Learning Research*, 31:216–228.

Dumesnil, J. G. and Shoucri, R. M. (1982). Effect of the geometry of the left ventricle on the calculation of ejection fraction. *Circulation*, 65(1):91–98.

Eck, V. G., Donders, W. P., Sturdy, J., Feinberg, J., Delhaas, T., Hellevik, L. R., and Huberts, W. (2016). A guide to uncertainty quantification and sensitivity analysis for cardiovascular applications. *International Journal for Numerical Methods in Biomedical Engineering*, page e02755.

Fang, K., Li, R., and Sudjianto, A. (2006). *Design and modeling for computer experiments*. Chapman & Hall/CRC.

Farrar, G., Suinesiaputra, A., Gilbert, K., Perry, J. C., Hegde, S., Marsden, A., Young, A. A., Omens, J. H., and McCulloch, A. D. (2016). Atlas-based ventricular shape analysis for understanding congenital heart disease. *Progress in Pediatric Cardiology*, 43:61–69. Proceedings of 2015 Innovations in Pediatric Heart Failure Symposium.

Fort, S., Hu, H., and Lakshminarayanan, B. (2019). Deep ensembles: A loss landscape perspective. *ArXiv*, abs/1912.02757.

Friedland, G. W. (2009). Discovery of the function of the heart and circulation of blood. *Cardiovascular Journal of Africa*, 20:160 – 160.

Friel, N. and Pettitt, A. N. (2008). Marginal likelihood estimation via power posteriors. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 70(3):589–607.

Gao, H., Aderhold, A., Mangion, K., Luo, X., Husmeier, D., and Berry, C. (2017). Changes and classification in myocardial contractile function in the left ventricle following acute myocardial infarction. *Journal of The Royal Society Interface*, 14(132):20170203.

Gao, H., Li, W. G., Cai, L., Berry, C., and Luo, X. Y. (2015). Parameter estimation in a holzapfel–ogden law for healthy myocardium. *Journal of Engineering Mathematics*, 95(1):231–248.

Gao, H., Wang, H., Berry, C., Luo, X., and Griffith, B. (2014). Quasi-static image-based immersed boundary-finite element model of left ventricle under diastolic loading. *International Journal for Numerical Methods in Biomedical Engineering*, 30.

Gardner, P., Lord, C., and Barthorpe, R. J. (2020). Bayesian history matching for structural dynamics applications. *Mechanical Systems and Signal Processing*, 143:106828.

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., and Rubin, D. (2013). *Bayesian Data Analysis, Third Edition*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statist. Sci.*, 7(4):457–472.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741.

Genet, M., Lee, L. C., Nguyen, R., Haraldsson, H., Acevedo-Bolton, G., Zhang, Z., Ge, L., Ordovas, K., Kozerke, S., and Guccione, J. M. (2014a). Distribution of normal human left ventricular myofiber stress at end diastole and end systole: a target for in silico design of heart failure treatments. *Journal of applied physiology*, 117(2):142–152.

Genet, M., Lee, L. C., Nguyen, R., Haraldsson, H., Acevedo-Bolton, G., Zhang, Z., Ge, L., Ordovas, K., Kozerke, S., and Guccione, J. M. (2014b). Distribution of normal human left ventricular myofiber stress at end diastole and end systole: a target for in silico design of heart failure treatments. *Journal of Applied Physiology*, 117(2):142–152.

Gibbs, M. N. (1997). *Bayesian Gaussian Processes for Regression and Classification*. PhD thesis, Department of Physics, University of Cambridge.

Goldstein, M. and Rougier, J. (2009). Reified bayesian modelling and inference for physical systems. *Journal of Statistical Planning and Inference*, 139(3):1221 – 1239.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

Gramacy, R. B. (2020). *Surrogates: Gaussian Process Modeling, Design and Optimization for the Applied Sciences*. Chapman Hall/CRC, Boca Raton, Florida.

Gramacy, R. B. and Apley, D. W. (2015). Local gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics*, 24(2):561–578.

Graves, A. (2011). Practical variational inference for neural networks. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Guan, D., Ahmad, F., Theobald, P., Soe, S., Luo, X., and Gao, H. (2019). On the AIC-based model reduction for the general HolzapfelâĂŞOgden myocardial constitutive law. *Biomechanics and Modeling in Mechanobiology*, 18(4):1213–1232.

Guan, D., Zhuan, X., Holmes, W., Luo, X., and Gao, H. (2021). Modelling of fibre dispersion and its effects on cardiac mechanics from diastole to systole. *Journal of Engineering Mathematics*, 128.

Haario, H., Laine, M., Mira, A., and Saksman, E. (2006). Dram: Efficient adaptive mcmc. *Statistics and Computing*, 16:339–354.

Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223 – 242.

Hadjicharalambous, M., Asner, L., Chabiniok, R., Sammut, E., Wong, J., Peressutti, D., Kerfoot, E., King, A., Lee, J., Razavi, R., Smith, N., Carr-White, G., and Nordsletten, D. (2017). Non-invasive model-based assessment of passive left-ventricular myocardial stiffness in healthy subjects and in patients with non-ischemic dilated cardiomyopathy. *Annals of Biomedical Engineering*, 45(3):605–618.

Hadjicharalambous, M., Chabiniok, R., Asner, L., Sammut, E., Wong, J., Carr-White, G., Lee, J., Razavi, R., Smith, N., and Nordsletten, D. (2015). Analysis of passive cardiac constitutive laws for parameter estimation using 3d tagged mri. *Biomechanics and Modeling in Mechanobiology*, 14(4):807–28.

Hadjicharalambous, M., Stoeck, C. T., Weisskopf, M., Cesarovic, N., Ioannou, E., Vavourakis, V., and Nordsletten, D. A. (2021). Investigating the reference domain influence in personalised models of cardiac mechanics: Effect of unloaded geometry on cardiac biomechanics. *Biomechanics and Modeling in Mechanobiology*, 20(4):1579–1597.

Hannun, A. Y., Rajpurkar, P., Haghpanahi, M., Tison, G. H., Bourn, C., Turakhia, M. P., and Ng, A. (2019). Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature Medicine*, 25:65–69.

Hasaballa, A. I., Gamage, T. P. B., Wang, V. Y., Zhao, D., Mauger, C. A., Gilbert, K., Wang, Z. J., Freytag, B., Cao, J. J., Young, A. A., and Nash, M. P. (2021). Sensitivity of myocardial stiffness estimates to inter-observer variability in LV geometric modelling. In Ennis, D. B., Perotti, L. E., and Wang, V. Y., editors, *Functional Imaging and Modeling of the Heart - 11th International Conference, FIMH 2021, Stanford, CA, USA, June 21-25, 2021, Proceedings*, volume 12738 of *Lecture Notes in Computer Science*, pages 287–295. Springer.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109.

Heinonen, M., Mannerström, H., Rousu, J., Kaski, S., and Lähdesmäki, H. (2016). Non-stationary gaussian process regression with hamiltonian monte carlo. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, JMLR: Workshop and Conference Proceedings, pages 732–740. JMLR.

Herman, J. and Usher, W. (2017). SALib: An open-source python library for sensitivity analysis. *The Journal of Open Source Software*, 2(9).

Heusinkveld, M. H. G., Quicken, S., Holtackers, R. J., Huberts, W., Reesink, K. D., Delhaas, T., and Spronck, B. (2018). Uncertainty quantification and sensitivity analysis of an arterial wall mechanics model for evaluation of vascular drug therapies. *Biomechanics and Modeling in Mechanobiology volume*, 17:55–69.

Hoffman, M. D. and Gelman, A. (2014). The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(47):1593–1623.

Holzapfel, G. A. (2000). Nonlinear solid mechanics: A continuum approach for engineering science. *Meccanica*, 37:489–490.

Holzapfel, G. A. and Ogden, R. W. (2009). Constitutive modelling of passive myocardium: a structurally based framework for material characterization. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 367(1902):3445–3475.

Holzapfel, G. A., Ogden, R. W., and Sherifova, S. (2019). On fibre dispersion modelling of soft biological tissues: a review. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 475(2224):20180736.

Hong, B., Moulton, M., and Secomb, T. (2019). Modeling left ventricular dynamics with characteristic deformation modes. *Biomechanics and Modeling in Mechanobiology*, 18.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4:251–257.

Hudson, S. and Pettit, S. (2020). What is 'normal' left ventricular ejection fraction? *Heart*, 106(18):1445–1446.

Hurtado, D. E., Castro, S., and Madrid, P. (2017). Uncertainty quantification of 2 models of cardiac electromechanics: Uncertainty quantification of cardiac electromechanics. *International Journal for Numerical Methods in Biomedical Engineering*, 33(12):e2894.

Huyghe, J. M., Arts, T., van Campen, D. H., and Reneman, R. S. (1992). Porous medium finite element model of the beating left ventricle. *American Journal of Physiology-Heart and Circulatory Physiology*, 262(4):H1256–H1267.

Hyvarinen, A., Karhunen, J., and Oja, E. (2002). *Independent Components Analysis*. Wiley.

Jarrett, A. M., Liu, Y., Cogan, N. G., and Hussaini, M. Y. (2015). Global sensitivity analysis used to interpret biological experimental results. *Journal of Mathematical Biology*, 71:151–170.

Kallhovd, S., Sundnes, J., and Wall, S. T. (2019). Sensitivity of stress and strain calculations to passive material parameters in cardiac mechanical models using unloaded geometries. *Computer Methods in Biomechanics and Biomedical Engineering*, 22(6):664–675.

Kennedy, M. C. and O'Hagan, A. (2000). Predicting the Output from a Complex Computer Code When Fast Approximations Are Available. *Biometrika*, pages 1–13.

Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464.

Keskar, N., Nocedal, J., Tang, P., Mudigere, D., and Smelyanskiy, M. (2017). On large-batch training for deep learning: Generalization gap and sharp minima. 5th International Conference on Learning Representations, ICLR 2017 ; Conference date: 24-04-2017 Through 26-04-2017.

Khalique, Z., Ferreira, P. F., Scott, A. D., Nielles-Vallespin, S., Firmin, D. N., and Pennell, D. J. (2020). Diffusion tensor cardiovascular magnetic resonance imaging: a clinical perspective. *JACC: Cardiovascular Imaging*, 13(5):1235–1255.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Klotz, S., Hay, I., Dickstein, M. L., Yi, G.-H., Wang, J., Maurer, M. S., Kass, D. A., and Burkhoff, D. (2006). Single-beat estimation of end-diastolic pressure-volume relationship: a novel method with potential for noninvasive application. *American Journal of Physiology-Heart and Circulatory Physiology*, 291(1):H403–H412.

Krishnamurthy, A., Villongco, C. T., Chuang, J., Frank, L. R., Nigam, V., Belezzuoli, E., Stark, P., Krummen, D. E., Narayan, S., Omens, J. H., McCulloch, A. D., and Kerckhoffs, R. C. (2013a). Patient-specific models of cardiac biomechanics. *Journal of Computational Physics*, 244:4–21.

Krishnamurthy, A., Villongco, C. T., Chuang, J., Frank, L. R., Nigam, V., Belezzuoli, E., Stark, P., Krummen, D. E., Narayan, S., Omens, J. H., McCulloch, A. D., and Kerckhoffs, R. C. (2013b). Patient-specific models of cardiac biomechanics. *Journal of Computational Physics*, 244:4 – 21. Multi-scale Modeling and Simulation of Biological Systems.

Krogh, A. and Hertz, J. (1992). A simple weight decay can improve generalization. In Moody, J., Hanson, S., and Lippmann, R. P., editors, *Advances in Neural Information Processing Systems*, volume 4. Morgan-Kaufmann.

Kuss, M. and Rasmussen, C. E. (2005). Assessing approximate inference for binary gaussian process classification. *Journal of Machine Learning Research*, 6(57):1679–1704.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International*

Conference on Neural Information Processing Systems, NIPS'17, page 6405âĂŞ6416, Red Hook, NY, USA. Curran Associates Inc.

Land, S., Gurev, V., Arens, S., Augustin, C. M., Baron, L., Blake, R., Bradley, C., Castro, S., Crozier, A., Favino, M., Fastl, T. E., Fritz, T., Gao, H., Gizzi, A., Griffith, B. E., Hurtado, D. E., Krause, R., Luo, X., Nash, M. P., Pezzuto, S., Plank, G., Rossi, S., Ruprecht, D., Seemann, G., Smith, N. P., Sundnes, J., Rice, J. J., Trayanova, N., Wang, D., Jenny Wang, Z., and Niederer, S. A. (2015). Verification of cardiac mechanics software: benchmark problems and solutions for testing active and passive material behaviour. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2184):20150641.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. (2018). Deep neural networks as gaussian processes. *International Conference on Learning Representations*.

Lei, C. L., Ghosh, S., Whittaker, D. G., Aboelkassem, Y., Beattie, K. A., Cantwell, C. D., Delhaas, T., Houston, C., Novaes, G. M., Panfilov, A. V., Pathmanathan, P., Riabiz, M., dos Santos, R. W., Walmsley, J., Worden, K., Mirams, G. R., and Wilkinson, R. D. (2020). Considering discrepancy when calibrating a mechanistic electrophysiology model. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 378(2173):20190349.

Levrero-Florencio, F., Margara, F., Zacur, E., Bueno-Orovio, A., Wang, Z., Santiago, A., Aguado-Sierra, J., Houzeaux, G., Grau, V., Kay, D., Vázquez, M., Ruiz-Baier, R., and Rodriguez, B. (2020). Sensitivity analysis of a strongly-coupled human-based electromechanical cardiac model: Effect of mechanical parameters on physiologically relevant biomarkers. *Computer Methods in Applied Mechanics and Engineering*, 361:112762.

Li, W., Gao, H., Mangion, K., Berry, C., and Luo, X. (2020a). Apparent growth tensor of left ventricular post myocardial infarction–in human first natural history study. *Computers in Biology and Medicine*, 129:104168.

Li, W., Lazarus, A., Gao, H., Martinez-Naharro, A., Fontana, M., Hawkins, P., Biswas, S., Janiczek, R., Cox, J., Berry, C., Husmeier, D., and Luo, X. (2020b). Analysis of cardiac amyloidosis progression using model-based markers. *Frontiers in Physiology*, 11:324.

Liang, F. and Wong, W. H. (2001). Real-parameter evolutionary monte carlo with applications to bayesian mixture models. *Journal of the American Statistical Association*, 96(454):653–666.

Liu, Y., Wen, H., Gorman, R. C., Pilla, J. J., Gorman, J. H., Buckberg, G., Teague, S. D., and Kassab, G. S. (2009). Reconstruction of myocardial tissue motion and strain

fields from displacement-encoded mr imaging. *American Journal of Physiology-Heart and Circulatory Physiology*, 297(3):H1151–H1162. PMID: 19561315.

Lombaert, H., Peyrat, J.-M., Croisille, P., Rapacchi, S., Fanton, L., Clarysse, P., Delingette, H., and Ayache, N. (2011). Statistical analysis of the human cardiac fiber architecture from dt-mri. In *International Conference on Functional Imaging and Modeling of the Heart*, pages 171–179. Springer.

MacKay, D. J. (1992a). *Bayesian Methods For Adaptive Models*. PhD thesis, California Institute of Technology.

MacKay, D. J. C. (1992b). Information-based objective functions for active data selection. *Neural Comput.*, 4(4):590âĂŞ604.

Mangion, K., Gao, H., McComb, C., Carrick, D., Clerfond, G., Zhong, X., Luo, X., Haig, C., and Berry, C. (2016a). A novel method for estimating myocardial strain: Assessment of deformation tracking against reference magnetic resonance methods in healthy volunteers. *Scientific Reports*, 6:38774.

Mangion, K., Gao, H., McComb, C., Carrick, D., Clerfond, G., Zhong, X., Luo, X., Haig, C. E., and Berry, C. (2016b). A Novel Method for Estimating Myocardial Strain: Assessment of Deformation Tracking Against Reference Magnetic Resonance Methods in Healthy Volunteers. *Scientific Reports*, 6(1):38774.

Mann, H. B. and Whitney, D. R. (1947). On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics*, 18(1):50 – 60.

Marlevi, D., Balmus, M., Hessenthaler, A., Viola, F., Fovargue, D., Vecchi, A. d., Lamata, P., Burris, N. S., Pagani, F. D., Engvall, J., Edelman, E. R., Ebbers, T., and Nordsletten, D. A. (2021). Non-invasive estimation of relative pressure for intracardiac flows using virtual work-energy. *Medical Image Analysis*, 68:101948.

Marquis, A. D., Arnold, A., Dean-Bernhoft, C., Carlson, B. E., and Olufsen, M. S. (2017). Practical identifiability and uncertainty quantification of a pulsatile cardiovascular model. *Mathematical Biosciences*, page 304.

Maso Talou, G. D., Babarenda Gamage, T. P., Sagar, M., and Nash, M. P. (2020). Deep learning over reduced intrinsic domains for efficient mechanics of the left ventricle. *Frontiers in Physics*, 8:30.

Masters, D. and Luschi, C. (2018). Revisiting small batch training for deep neural networks. *ArXiv*, abs/1804.07612.

Matthews, A. G. d. G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., and Hensman, J. (2017). GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6.

McClarren, R. (2018). *Uncertainty Quantification and Predictive Computational Science: A Foundation for Physical Scientists and Engineers*.

Mcculloch, W. and Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:127–147.

Melis, A., Clayton, R. H., and Marzo, A. (2017). Bayesian sensitivity analysis of a 1D vascular model with Gaussian process emulators. *International Journal for Numerical Methods in Biomedical Engineering*, 33:e2882.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092.

Metropolis, N. and Ulam, S. (1949). The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341.

Mira, A. (2001). On metropolis-hastings algorithms with delayed rejection. *Metron*, 59.

Mirams, G. R., Niederer, S. A., and Clayton, R. H. (2020). The fickle heart: uncertainty quantification in cardiac and cardiovascular modelling and simulation. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 378(2173):20200119.

Mirams, G. R., Pathmanathan, P., Gray, R. A., Challenor, P., and Clayton, R. H. (2016). Uncertainty and variability in computational and mathematical models of cardiac physiology. *Journal of Physiology*, pages 6833–6847.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., USA, 1 edition.

Monnahan, C. C., Thorson, J. T., and Branch, T. A. (2017). Faster estimation of bayesian models in ecology using hamiltonian monte carlo. *Methods in Ecology and Evolution*, 8(3):339–348.

Morio, J. (2011). Global and local sensitivity analysis methods for a physical system. *European Journal of Physics*, 32(6):1577.

Moss, R. L. and Fitzsimons, D. P. (2002). Frank-starling relationship. *Circulation Research*, 90(1):11–13.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer, New York. ISBN 0-387-94724-8.

Niederer, S. A., Campbell, K. S., and Campbell, S. G. (2019). A short history of the development of mathematical models of cardiac mechanics. *Journal of Molecular and Cellular Cardiology*, 127:11–19.

Niederer, S. A. and Smith, N. P. (2016). Using physiologically based models for clinical translation: predictive modelling, data interpretation or something in-between? *The Journal of Physiology*, 594(23):6849–6863.

Noè, U., Lazarus, A., Gao, H., Davies, V., Macdonald, B., Mangion, K., Berry, C., Luo, X., and Husmeier, D. (2019). Gaussian process emulation to accelerate parameter estimation in a mechanical model of the left ventricle: a critical step towards clinical end-user relevance. *Journal of the Royal Society Interface*, 16.

Nordsletten, D. A., Niederer, S. A., Nash, M. P., Hunter, P. J., and Smith, N. P. (2011). Coupling multi-physics models to cardiac mechanics. *Progress in biophysics and molecular biology*, 104 1-3:77–88.

Oakley, J. E. and O'Hagan, A. (2004). Probabilistic sensitivity analysis of complex models: a bayesian approach. *Journal of The Royal Statistical Society Series B-statistical Methodology*, 66:751–769.

Ober, S. W. and Rasmussen, C. E. (2019). Benchmarking the neural linear model for regression. In *Advances in Approximate Bayesian Inference (AABI)*.

Osnes, H. and Sundnes, J. (2012). Uncertainty analysis of ventricular mechanics using the probabilistic collocation method. *IEEE Transaction on Biomechanical Engineering*, 59:2171âĂŞ2179.

Padonou, E. and Roustant, O. (2016). Polar gaussian processes and experimental designs in circular domains. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):1014–1033.

Pagani, S. and Manzoni, A. (2021). Enabling forward uncertainty quantification and sensitivity analysis in cardiac electrophysiology by reduced order modeling and machine learning. *International Journal for Numerical Methods in Biomedical Engineering*, 37:e3450.

Palit, A., Bhudia, S., Arvanitis, T., Turley, G., and Williams, M. (2018a). In vivo estimation of passive biomechanical properties of human myocardium. *Medical & Biological Engineering & Computing*, 56:1–17.

Palit, A., Bhudia, S., Arvanitis, T., Turley, G., and Williams, M. (2018b). In vivo estimation of passive biomechanical properties of human myocardium. *Medical & Biological Engineering & Computing*, 56:1–17.

Palit, A., Bhudia, S., Arvanitis, T., Turley, G. A., and Williams, M. (2015). Computational modelling of left-ventricular diastolic mechanics: effect of fibre orientation and right-ventricle topology. *Journal of biomechanics*, 48 4:604–12.

Pearce, T., Leibfried, F., and Brintrup, A. (2020). Uncertainty in neural networks: Approximately bayesian ensembling. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 234–244, Online. PMLR.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Peirlinck, M., Costabal, F., Sack, K., Choy, J., Kassab, G., Guccione, J., Beule, M., Segers, P., and Kuhl, E. (2019). Using machine learning to characterize heart failure across the scales. *Biomechanics and Modeling in Mechanobiology*, 18.

Peirlinck, M., Costabal, F. S., Yao, J., Guccione, J., Tripathy, S., Wang, Y., Ozturk, D., Segars, P., Morrison, T., Levine, S., and Kuhl, E. (2021). Precision medicine in human heart modeling. *Biomechanics and Modeling in Mechanobiology*, 20:803 – 831.

Perdikaris, P., Raissi, M., Damianou, A., Lawrence, N. D., and Karniadakis, G. E. (2017). Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2198):20160751.

Plumlee, M. (2017). Bayesian calibration of inexact computer models. *Journal of the American Statistical Association*, 112(519):1274–1285.

Quarteroni, A., Lassila, T., Rossi, S., and Ruiz-Baier, R. (2017). Integrated heart–coupling multiscale and multiphysics models for the simulation of the cardiac function. *Computer Methods in Applied Mechanics and Engineering*, 314:345–407. Copyright (c) 2016 Elsevier B. V. All rights reserved. This is an author produced version of a paper published in Computer Methods in Applied Mechanics and Engineering. Uploaded in accordance with the publisher's self-archiving policy.

Quer, G., Arnaout, R., Henne, M., and Arnaout, R. (2021). Machine learning and the future of cardiovascular care: Jacc state-of-the-art review. *Journal of the American College of Cardiology*, 77(3):300–313.

Quiñonero Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate gaussian process regression. *J. Mach. Learn. Res.*, 6:1939âĂŞ1959.

Rasmussen, C. (2003). Gaussian processes to speed up hybrid monte carlo for expensive bayesian integrals. pages 651–659. Max-Planck-Gesellschaft.

Rasmussen, C. E. and Williams, K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA.

Requeima, J., Tebbutt, W., Bruinsma, W., and Turner, R. E. (2019). The gaussian process autoregressive regression model (gpar). In *AISTATS*, pages 1860–1869.

Robert, C. P. (1994). *The Bayesian choice: a decision-theoretic motivation*. Springer-Verlag.

Robert, C. P. and Casella, G. (2005). *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Robert, C. P. and Casella, G. (2009). *Introducing Monte Carlo Methods with R (Use R)*. Springer-Verlag, Berlin, Heidelberg, 1st edition.

Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N., and Aigrain, S. (2013). Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110550.

Rodero, C., Strocchi, M., Marciniak, M., Longobardi, S., Whitaker, J., OâĂŹNeill, M. D., Gillette, K., Augustin, C., Plank, G., Vigmond, E. J., Lamata, P., and Niederer, S. A. (2021). Linking statistical shape models and simulated function in the healthy adult human heart. *PLOS Computational Biology*, 17(4):e1008851.

Rodriguez-Cantano, R., Sundnes, J., and Rognes, M. E. (2018). Uncertainty in cardiac myofibre orientation and stiffnesses dominate the variability of left ventricle deformation response. *International Journal for Numerical Methods in Biomedical Engineering*, (35):e3178.

Rojas, R. (1996). *Neural Networks: A Systematic Introduction*. Springer-Verlag, Berlin, Heidelberg.

Romaszko, L., Borowska, A., Lazarus, A., Dalton, D., Berry, C., Luo, X., Husmeier, D., and Gao, H. (2021). Neural network-based left ventricle geometry prediction from cmr images with application in biomechanics. *Artificial Intelligence in Medicine*, 119:102140.

Romaszko, L., Lazarus, A., Gao, H., Borowska, A., Luo, X., and Husmeier, D. (2019). Massive dimensionality reduction for the left ventricular mesh. *International Conference on Statistics: Theory and Applications (ICSTA)*, 1(24).

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747.

Sahli Costabal, F., Choy, J., Sack, K., Guccione, J., Kassab, G., and Kuhl, E. (2019). Multiscale characterization of heart failure. *Acta Biomaterialia*, 86:66–76.

Sahli Costabal, F., Yang, Y., Perdikaris, P., Hurtado, D. E., and Kuhl, E. (2020). Physics-informed neural networks for cardiac activation mapping. *Frontiers in Physics*, 8:42.

Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., and Tarantola, S. (2010). Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index. *Computer Physics Communications*, 181(2):259–270.

Santner, T. J., B., W., and W., N. (2018). *The Design and Analysis of Computer Experiments, Second Edition*. Springer-Verlag.

Scholkopf, B., Smola, A., and Muller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299âĂŤ1319.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. (2016). Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.

Smith, R. C. (2014). *Uncertainty Quantification: Theory, Implementation, and Applications*. SIAM, Philadelphia, PA.

Smith, S. and Le, Q. V. (2018). A bayesian perspective on generalization and stochastic gradient descent. In *ICLR*.

Snelson, E. and Ghahramani, Z. (2006). Sparse gaussian processes using pseudo-inputs. In Weiss, Y., Schölkopf, B., and Platt, J. C., editors, *Advances in Neural Information Processing Systems 18*, pages 1257–1264. MIT Press.

Sobol, I. (1967). On the distribution of points in a cube and the approximate evaluation of integrals. *Ussr Computational Mathematics and Mathematical Physics*, 7:86–112.

Sobol, I. M. (2001). Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulations*, 55:271–280.

Sommer, G., Schriefl, A. J., AndrÃď, M., Sacherer, M., Viertler, C., Wolinski, H., and Holzapfel, G. A. (2015). Biomechanical properties and microstructure of human ventricular myocardium. *Acta Biomaterialia*, 24:172–192.

Staib, M., Reddi, S. J., Kale, S., Kumar, S., and Sra, S. (2019). Escaping saddle points with adaptive gradient methods. In *ICML*.

Suinesiaputra, A., Ablin, P., Albà, X., Alessandrini, M., Allen, J., Bai, W., Çimen, S., Claes, P., Cowan, B. R., D'hooge, J., Duchateau, N., Ehrhardt, J., Frangi, A. F., Gooya, A., Grau, V., Lekadir, K., Lu, A., Mukhopadhyay, A., Oksuz, I., Parajuli, N., Pennec, X., Pereañez, M., Pinto, C., Piras, P., Rohé, M.-M., Rueckert, D., Säring, D., Sermesant, M., Siddiqi, K., Tabassian, M., Teresi, L., Tsaftaris, S. A., Wilms, M., Young, A. A., Zhang, X., and Medrano-Gracia, P. (2018). Statistical shape modeling of the left ventricle: Myocardial infarct classification challenge. *IEEE Journal of Biomedical and Health Informatics*, 22(2):503–515.

Thakur, S., Lorsung, C., Yacoby, Y., Doshi-Velez, F., and Pan, W. (2020). Learned uncertainty-aware (luna) bases for bayesian regression using multi-headed auxiliary networks. *ICML Workshop on Uncertainty in Deep Learning*, 2:1–18.

Titsias, M. (2009). Variational learning of inducing variables in sparse gaussian processes. In van Dyk, D. and Welling, M., editors, *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 567–574, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA. PMLR.

Toussaint, N., Stoeck, C. T., Schaeffter, T., Kozerke, S., Sermesant, M., and Batchelor, P. G. (2013). In vivo human cardiac fibre architecture estimation using shape-based diffusion tensor processing. *Medical image analysis*, 17(8):1243–1255.

Tran, G. T., Oliver, K. I. C., Sóbester, A., Toal, D. J. J., Holden, P. B., Marsh, R., Challenor, P., and Edwards, N. R. (2016). Building a traceable climate model hierarchy with multi-level emulators. *Advances in Statistical Climatology, Meteorology and Oceanography*, 2(1):17–37.

Tsuchida, R., Roosta-Khorasani, F., and Gallagher, M. (2018). Invariance of Weight Distributions in Rectified MLPs. In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5002–5011. PMLR.

Ugray, Z., Lasdon, L., Plummer, J., Glover, F., Kelly, J., and Martí, R. (2007). Scatter Search and Local NLP Solvers: A Multistart Framework for Global Optimization. *INFORMS Journal on Computing*, 19(3):328–340.

van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *JMLR*, 9:2579–2605.

Volodina, V. and Challenor, P. (2021). The importance of uncertainty quantification in model reproducibility. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2197):20200071.

Voorhees, A. and Han, H.-C. (2015). Biomechanics of cardiac function. *Comprehensive Physiology*, 5 4:1623–44.

Wakefield, J. (2013). *Bayesian and Frequentist Regression Methods*. Springer.

Wang, H. M., Gao, H., Luo, X. Y., Berry, C., Griffith, B. E., Ogden, R. W., and Wang, T. J. (2013a). Structure based finite strain modelling of the human left ventricle in diastole. *International journal for numerical methods in biomedical engineering*, 29(1):83–103.

Wang, Z., Mohamed, S., and de Freitas, N. (2013b). Adaptive hamiltonian and riemann manifold monte carlo samplers. *30th International Conference on Machine Learning, ICML 2013*.

Wang, Z. J., Wang, V. Y., Babarenda Gamage, T. P., Rajagopal, V., Cao, J. J., Nielsen, P. M. F., Bradley, C. P., Young, A. A., and Nash, M. P. (2020). Efficient estimation of load-free left ventricular geometry and passive myocardial properties using principal component analysis. *International Journal for Numerical Methods in Biomedical Engineering*, 36(3):e3313.

Watson, J., Lin, J. A., Klink, P., and Peters, J. (2021). Neural linear models with functional gaussian process priors. In *Third Symposium on Advances in Approximate Bayesian Inference*.

Williams, C. K. I. (1997). Computing with infinite networks. In Mozer, M. C., Jordan, M. I., and Petsche, T., editors, *Advances in Neural Information Processing Systems 9*, pages 295–301. MIT Press.

Wilson, D. and Martinez, T. (2003). The general inefficiency of batch training for gradient descent learning. *Neural networks : the official journal of the International Neural Network Society*, 16 10:1429–51.

Wood, A. T. (1995). When is a truncated covariance function on the line a covariance function on the circle? *Statistics & Probability Letters*, 24(2):157–164.

Zammit-Mangion, A., Ng, T. L. J., Vu, Q., and Filippone, M. (2021). Deep compositional spatial models. *Journal of the American Statistical Association*, 0(0):1–22.

Zhang, B., Cole, D. A., and Gramacy, R. B. (2019). Distance-distributed design for gaussian process surrogates. *Technometrics*, 0(0):1–13.

Zhang, Y., Wang, V. Y., Morgan, A. E., Kim, J., Tafreshi, R., Wallace, A. W., Guccione, J. M., Weinsaft, J. W., Ge, L., and Ratcliffe, M. B. (2021). Finite-element based optimization of left ventricular passive stiffness in normal volunteers and patients after myocardial infarction: Utility of an inverse deformation gradient calculation of regional diastolic strain. *Journal of the Mechanical Behavior of Biomedical Materials*, 119:104431.