Bayesian Parameter Estimation and Uncertainty Quantification in Fluid-Dynamics Models of the Pulmonary Circulation System

Luiza-Mihaela Paun

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

School of Engineering
College of Science and Engineering
University of Glasgow

September 2020

To my family

Abstract

The past few decades have witnessed an explosive synergy between physics and the life sciences. In particular, physical modelling in medicine and physiology is a topical research area. The present work focuses on the inverse problem, more specifically on the parameter inference and uncertainty quantification in a 1D fluid-dynamics model for quantitative physiology: the pulmonary blood circulation. The particular application is pulmonary hypertension, requiring an analysis of the blood pressure, whose measurement in the pulmonary system can only be obtained invasively for patients. The ultimate goal is to develop a non-invasive disease diagnostication method. This could be accomplished by combining non-invasively obtained haemodynamic data (blood flow measured with MRI) with imaging data (CT scans of the lung structure), to be used in conjunction with mathematical and statistical modelling. This will provide a decision-making support mechanism in the clinic, ultimately aiding in personalised medicine.

This thesis adopts a Bayesian approach to uncertainty quantification in physiological models, allowing to assess the credibility of these models. The danger with using overly confident models is that they could produce biased predictions, ultimately leading to the wrong disease diagnosis and treatment. Inference of unknown and immeasurable parameters of several 1D fluid-dynamics models, expressed through partial differential equations, is performed with Markov Chain Monte Carlo. These parameters act as bio-indicators for the disease, e.g. vessel wall stiffness, which is high in pulmonary hypertension patients. In addition, the uncertainty in the model form and the data measurement process (jointly called model mismatch) is captured, and the model mismatch is represented with Gaussian Processes. Given that the mathematical model is not a perfect representation of the reality, and that the data measurement process is prone to errors, this introduces an extra layer of uncertainty. If unaccounted for, the result is biased and overly confident parameter estimates and model predictions. Yet another source of uncertainty modelled in this study is the variability in the vessel network geometry, connectivity and size, which is shown to introduce variability in the model predictions, and must be accounted for. The uncertainty in the model parameters, model form, data measurement process and vessel network propagates through to the model predictions, which is also quantified.

Lastly, this thesis is concerned with accelerating the computational efficiency of the statistical inference procedure, aiming to make the methods suitable for use in the clinic. Statistical emulation is used in conjunction with a series of efficient Hamiltonian Monte Carlo algorithms, particularly adapted to computationally expensive models. A comparative evaluation study is carried out to identify the algorithm giving the best trade-off between accuracy and efficiency on a set of representative benchmark differential equation models.

Contents

List of Tables

List of Figures

Acknowledgements

The end of my PhD studies has officially come. It has been a beautiful, yet challenging road to travel. I could not have come this far without the support I have received from several people.

First, my sincere appreciation goes to my main PhD supervisor, Prof. Dirk Husmeier, for whom I have high respect, and thanks to whom this journey has been a beautiful one. Dirk, you are the exact definition of a mentor, one who has motivated me so much, who has generously shared your knowledge with me, who has turned me into the best PhD version I could be. For that, and so much more, I thank you very much.

I would like to also express my thanks to my second supervisor, Prof. Nick Hill and my collaborators, Prof. Mette Olufsen, Dr. Umar Qureshi and Mitchel Colebank. Thank you for the many informative discussions we have had, for all the knowledge you have passed on to me, and for all the help you have offered me.

I would not be writing this if it was not for my family who, although not without sacrifice, have supported me to pursue my studies in Glasgow. Mother, you have been my number one supporter, you have firmly believed in me, and motivated me – your determination, ambition and perseverance have truly inspired me. Although words can never describe my true feelings, I am saying a big thank you! My deepest thanks also go to my father, my brother and my grandparents. Thank you for your constant support, encouragements, prayers and for the unconditional love. I truly could not have asked for a more supportive family.

Liuyang, you are my other half, one who has become part of my family. Glasgow with you is home in my heart. You have unselfishly devoted so much time into caring for me, supporting me, helping me with the maths, and have challenged me to many highly informative discussions. I can never thank you enough. You are my solid rock and our bond is unbreakable.

I also want to thank my friends: Aurora, Maria, Maali, Andrey, Vlad, to name a few. You have made this journey much more enjoyable.

Lastly, I would like to acknowledge the University of Glasgow for accepting me to study here 9 years ago, and for the many opportunities and resources I benefited from throughout the years. I am delighted to call myself an alumna of this university.

Declaration of authorship

This thesis reports my contributions to an interdisciplinary collaborative research project with mathematical and physiological modellers in Mette Olufsen's group at North Carolina State University (NCSU). In general, my contributions have been on the statistical inference side; below I give a more specific overview. My contribution has also been clearly highlighted within the appropriate chapter.

Chapters 1 and 2 contain background information, as well as known and relevant mathematical and statistical methodology. The rest of the chapters contain my original work jointly with my collaborators. The first part of Chapter 3 is my own work, and has been published jointly with my two supervisors: Dirk Husmeier and Nicholas Hill and with collaborators from Mette Olufsen's group (Mitchel Colebank, Umar Qureshi, Mette Olufsen and Mansoor Haider) in a special issue of Statistica Neerlandica, please see [141]. The second part of Chapter 3 presents my contribution to an article by Qureshi and collaborators from NCSU and Glasgow University published in Biomechanics and Modeling in Mechanobiology, please see [151]. Chapter 4 describes my own work, submitted to Journal of the Royal Society Interface, jointly with collaborators from NCSU and Glasgow University. The majority of the work presented in Chapter 5 is based on Colebank's contribution, and my contribution is clearly outlined in the appropriate place within the chapter. This chapter is based on our joint publication [38] in Journal of the Royal Society Interface jointly with collaborators from NCSU and Glasgow University. Chapter 6 is my work exclusively (supervised by Husmeier), and is based on material from two papers: one which has been submitted to International Journal for Numerical Methods in Biomedical Engineering, and a second paper in preparation, both jointly with Husmeier. No part of this thesis has been previously submitted for a degree at this or any other university.

Nomenclature

Abbreviations:

AIC: Akaike information criterion

AM: adaptive Metropolis

ARMA: autoregressive moving average

BIC: Bayesian information criterion

cdf: cumulative distribution function

CFD: computational fluid-dynamics

CT: computed tomography

DA: delayed acceptance

DIC: Deviance information criterion

DR: delayed rejection

DRAM: delayed rejection adaptive Metropolis

ESS: effective sample size

GP: Gaussian Process

HMC: Hamiltonian Monte Carlo

iid: independent and identically distributed

KDE: kernel density estimation

LDMC: Lagrangian Dynamical Monte Carlo

MC: Monte Carlo

MCMC: Markov Chain Monte Carlo

M-H: Metropolis-Hastings

MPA: main pulmonary artery

MPSRF: multivariate potential scale reduction factor

MRI: magnetic resonance imaging

NUTS: No U-turn Sampler

ODE: ordinary differential equation

PDE: partial differential equation

pdf: probability distribution function

PH: pulmonary hypertension

RMHMC: Riemann Manifold Hamiltonian Monte Carlo

RSS: residual sum-of-squares

UQ: uncertainty quantification

WAIC: Watanabe Akaike information criterion

1D: one-dimensional

Mathematical notation:

x: scalar value

$\mathbf{x}$: vector

$\mathbf{X}$: matrix

Chapter 1

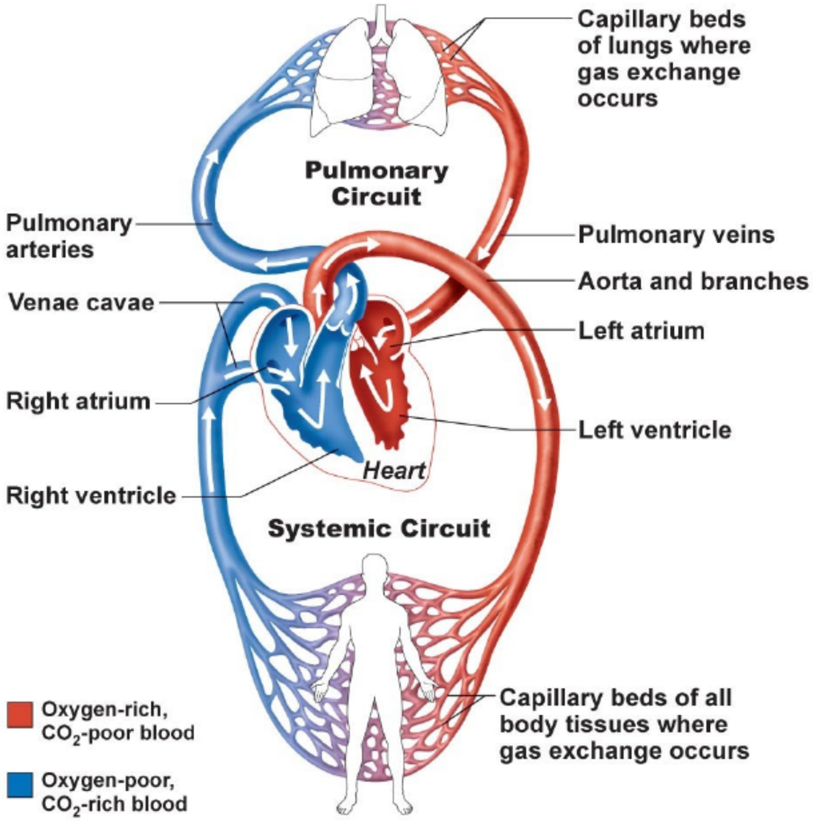
Introduction

1.1 Cardiovascular circulation: insight into physiology

The cardiovascular circulation is composed of the systemic and the pulmonary circulation, and they both connect to the heart (see Figure 1.1) ensuring continuous blood flow across the entire body. The systemic circulation transports the oxygenated blood from the left ventricle to the rest of the body (organs, tissues, muscles receive oxygen O_2) through the systemic arteries and the blood returns to the right atrium through the systemic veins. The pulmonary circulation transports the de-oxygenated blood from the right ventricle to the lungs through the pulmonary arteries and capillaries, where CO_2 is removed and the blood receives oxygen, and back to the left atrium through the pulmonary veins; a new cycle, called cardiac cycle then starts. During a cardiac cycle the heart contracts (systolic phase) and relaxes (diastolic phase). During contraction blood is pumped by the heart in a pulsatile manner, i.e. waves of blood flow and pressure get propagated along the arterial circulation due to the pulsatile rhythm of the heart and the elasticity of the vessel wall. The form of this pulse wave through the systemic and pulmonary vessels is influenced by multiple factors, such as properties of the blood (e.g. viscosity, density), or of the arterial walls (e.g. stiffness). Thus, an assessment of the health of the circulation haemodynamics requires an analysis of the pulse waveforms' characteristics (e.g. speed, magnitude, shape) [150].

Pulmonary hypertension (PH) is characterised by elevated blood pressure in the pulmonary circulation (mean pressure larger than 25mmHg in the main pulmonary artery), including the pulmonary arteries, capillaries and veins [192]. Causes for PH include left heart disease, lung disease, hypoxia, pulmonary embolism [181]. PH often occurs with remodelling of pulmonary arteries, e.g. increased vessel wall stiffness (and decreased compliance), and vessel wall thickening (narrowing of the vessels), and if left untreated, PH eventually leads to right ventricle failure [192].

Currently the diagnosis of PH is performed based on a series of medical tests including invasive right-heart catheterization for pulmonary blood pressure measurement, non-invasive mag-


Capillary beds of lungs where gas exchange occurs
Pulmonary Circuit
Pulmonary arteries
Pulmonary veins
Aorta and branches
Venae cavae
Left atrium
Right atrium
Left ventricle
Heart
Right ventricle
Systemic Circuit
Capillary beds of all body tissues where gas exchange occurs
🟥 Oxygen-rich, CO$_2$-poor blood
🟦 Oxygen-poor, CO$_2$-rich blood

© 2011 Pearson Education, Inc.

Figure 1.1: Depiction of the cardiovascular blood circulation. Source: `https://www.studocu.com`.

netic resonance imaging for pulmonary blood flow measurement, and non-invasive computed topography (CT) imaging of the heart and lungs [110]. Diagnostic protocols interpret each data source independently to make an ultimate decision about the disease classification and severity [63].

1.2 Pulmonary circulation: computational modelling

1.2.1 Mathematical modelling

Computational haemodynamics models are emerging as powerful tools to increase knowledge of cardiovascular disease progression for treatment [187] and surgery purposes [127, 191]. Computational modelling of the blood flow allows an in-depth understanding of the underlying disease mechanism, providing important metrics about the blood flow, which could not be obtained from in-vivo experiments [98], e.g. pulmonary blood flow and pressure predictions in vessels beyond the point where measurements can be taken, pulmonary blood pressure prediction to eliminate the invasive measurement procedures, relation between blood pressure and material properties of the vessels which cannot be measured in-vivo (e.g. vessel wall stiffness). Computational haemodynamic models can be combined with imaging and haemodynamic data with the ultimate goal of achieving personalised medicine, allowing patient-specific care and treatment.

1.2.2 Statistical modelling

Before using the mathematical models for decision-making in the clinic, the credibility of the models must be rigorously tested by analysing and incorporating all sources of uncertainty (Figure 1.2) into the analysis in a statistical modelling framework. Uncertainty can be caused by variability, e.g. in model inputs (material parameters: vessel wall stiffness, or boundary condition parameters: Windkessel parameters, both of which are inferred from data, or arterial network geometry, size and connectivity, which are found through image segmentation), in model form/structure (i.e. model discrepancy, inability of the model to faithfully capture the real-system), in measurement process (resulting in noisy experimental data), in simulator output (e.g. numerical errors due to tolerances, grid resolution, convergence, time steps in equation solving, or emulator approximations), in residuals, or the uncertainty can be caused by lack of information, e.g. insufficient amount of experimental data [124]. These uncertainties are dependent on each other, for example quantifying uncertainty of parameters of a mathematical model which has been mis-specified without accounting for the model discrepancy leads to biased parameter inference results. All these types of uncertainty (summarised in Figure 1.2) thus combine and lead to uncertainty in the model output (Figure 1.3), which is ultimately interpreted for diagnosis and treatment. Therefore, performing uncertainty quantification (UQ) is vital, and can be performed by assigning a probability density function (pdf) to the unknown quantities (rather than fixed values). UQ is of two types: inverse UQ and forward UQ (see Figure 1.4). Inverse UQ deals with estimating pdfs of the model inputs, also called model parameters, based on the measured data, e.g. inferring vessel wall stiffness from pulmonary blood pressure. Inverse UQ can be achieved by sampling from the parameter posterior distribution. Posterior sampling also enables performing model selection, i.e. which model, out of a number of possible mathematical models, is most likely under the data? Forward UQ is concerned with deriving pdfs of the data based on the model inputs, e.g. propagate uncertainty in model inputs through to model outputs – see Figure 1.3. If the output is a time series, e.g. pulmonary blood pressure throughout an entire cardiac cycle, then the output at each time point is assigned a pdf. A related concept is sensitivity analysis [38], which quantifies how sensitive model outputs are to changing model inputs, and is of two types: local and global, with the latter type requiring the pdf of the model inputs. Sensitivity analysis can identify model parameters to which the output is most sensitive to, hence these parameters should be inferred from the data (or, if possible, measured in the experiment), while parameters for which the sensitivity is low could be assigned fixed values (e.g. based on literature knowledge).

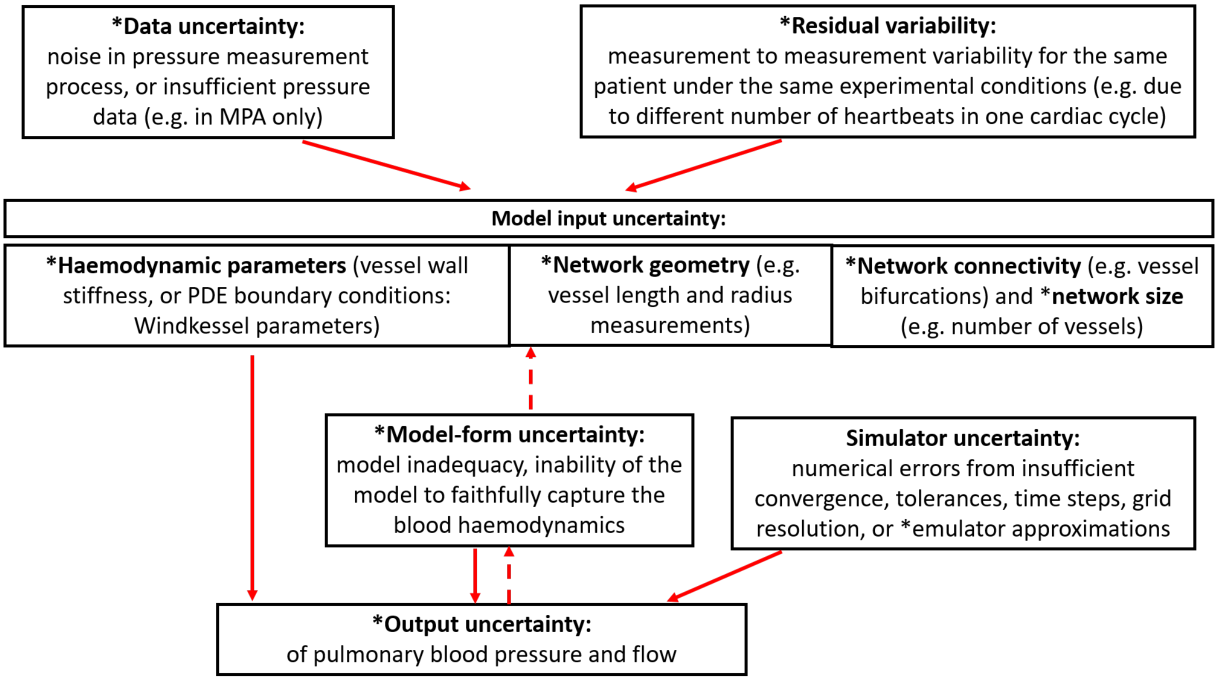

*Data uncertainty:
noise in pressure measurement process, or insufficient pressure data (e.g. in MPA only)
*Residual variability:
measurement to measurement variability for the same patient under the same experimental conditions (e.g. due to different number of heartbeats in one cardiac cycle)
Model input uncertainty:
*Haemodynamic parameters (vessel wall stiffness, or PDE boundary conditions: Windkessel parameters)
*Network geometry (e.g. vessel length and radius measurements)
*Network connectivity (e.g. vessel bifurcations) and *network size (e.g. number of vessels)
*Model-form uncertainty:
model inadequacy, inability of the model to faithfully capture the blood haemodynamics
Simulator uncertainty:
numerical errors from insufficient convergence, tolerances, time steps, grid resolution, or *emulator approximations
*Output uncertainty:
of pulmonary blood pressure and flow

Figure 1.2: Illustration of sources of uncertainty for the computational model describing the pulmonary circulation: different sources of uncertainty (from data, residual, input, model-form, simulator) combine to produce output uncertainty. Continuous dashes indicate the direct dependence between types of uncertainty (e.g. the noise model chosen influences the model parameters estimated), while dashed arrows give the indirect dependence (e.g. ignoring the model-form uncertainty biases parameter estimates). The sources of uncertainty modelled in the analysis presented in this thesis are indicated by asterisks. Legend: MPA - main pulmonary artery, PDE - partial differential equations. Figure adapted from [124].

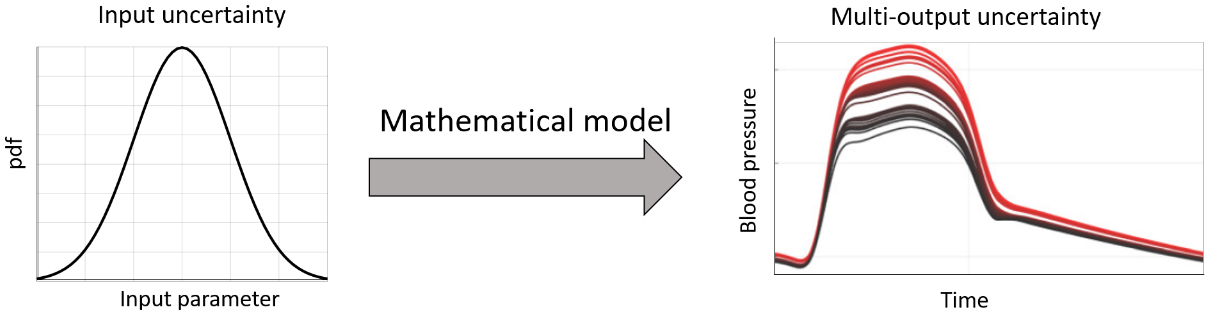


Figure 1.3: Example of 1D model input uncertainty characterised by a probability density function (pdf). Input values are sampled from the pdf. Each input value sampled is inserted into the mathematical model to produce a multi-output (pulmonary blood pressure output during a cardiac cycle). The ensemble of multi-outputs illustrated with colors ranging from black to red give a measure of output uncertainty. Figure adapted from [124].

1.3 Work overview

The current work focuses on the inverse UQ problem, and adopts a Bayesian approach to quantifying the uncertainty of the parameters of a 1D fluid-dynamics model of the pulmonary blood

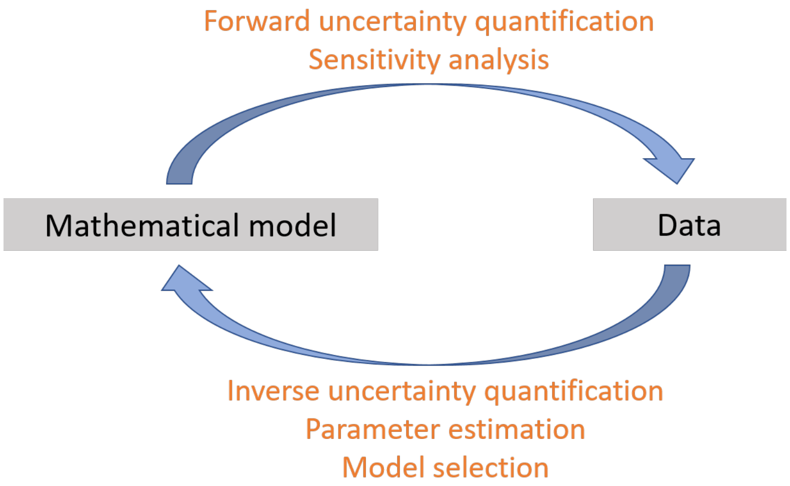


Figure 1.4: Discriminating between forward and inverse uncertainty quantification (UQ) -related concepts. Forward UQ deals with deriving probability distributions for the data based on varying the mathematical model parameters; sensitivity analysis identifies which parameters the model output is most sensitive to. Inverse UQ is concerned with inferring model parameter distributions from the data; parameter estimation with no UQ is also possible, and model selection selects the best mathematical model consistent with the data.

circulation described through coupled non-linear partial differential equations (PDEs). The emphasis is on the inference of haemodynamic parameters (e.g. vessel wall stiffness and Windkessel parameters, respectively) from measured blood pressure data in a fixed pulmonary arterial network (coming from one single image segmentation). This study also investigates the variability of the arterial network parameters obtained from segmentation of the CT scan of a lung, i.e. network geometry (vessel radius and length), network size (e.g. number of pulmonary vessels) and network connectivity (e.g. location of the pulmonary vessel bifurcations or trifurcations) based on multiple image segmentations. Moreover, the mis-specification of the mathematical model and of the error (noise) model is incorporated into the analysis, and the effect of ignoring the model mismatch is analysed. Additionally, discrimination between competing mathematical models (e.g. with different vessel wall properties, or vessel wall stiffness assumptions) is carried out, with the purpose of identifying the model that is best supported by the measured data.

Although covered at a much lesser extent, forward UQ is also performed, by propagating parameter and model-form uncertainty through to the model output of blood flow and pressure.

In addition, this study places great emphasis on computational efficiency, since Bayesian methods are computationally expensive due to repeated numerical integrations of the PDEs. The use of computationally affordable approximations (emulators) to the mathematical model are adopted, and an extensive comparison of Bayesian algorithms coupled with emulation is performed in the context of the 1D fluid-dynamics model to identify the most efficient and accurate method. The algorithm comparison is extended to toy problems described by ordinary differential equations (ODEs) for method accuracy checking (i.e. can the ground-truth parameter values be correctly inferred?) and method robustness checking (can the methods perform well across problems of different complexity?).

This thesis is organised into seven chapters.

Chapter 1 gives a brief physiological insight into the cardiovascular circulation, with a focus on pulmonary hypertension. It also introduces the motivation behind using mathematical modelling to describe the pulmonary circulation. Additionally, it emphasizes the need for rigorous testing of the validity of the mathematical models by incorporating all sources of uncertainty into the analysis in a statistical modelling framework.

Chapter 2 reviews relevant methodology used throughout this thesis, such as Bayesian methods (Markov Chain Monte Carlo, MCMC), Gaussian Processes and model selection criteria. In addition, it describes the mathematical model describing the pulmonary blood circulation based on which the statistical inference is performed.

Chapter 3 describes a Bayesian approach to estimation and UQ of parameters characterising the haemodynamic equations of the mathematical model described in Chapter 2.

Chapter 4 adopts a Bayesian approach to investigate the importance of modelling the model form uncertainty. Additionally, model selection is performed to choose the mathematical model which is best supported by the measured data out of a number of models making different stiffness and vessel wall elasticity assumptions.

Chapter 5 investigates the uncertainty of model predictions to changes in network geometry, size and connectivity based on multiple image segmentations.

Chapter 6 focuses on methodological aspects of emulation-accelerated MCMC methods as a viable parameter estimation and UQ tool in computationally expensive models described by non-linear differential equations (DEs). Proofs of convergence for these novel methods are provided and results from an empirical method comparison on several DEs systems are shown.

Chapter 7 concludes with a discussion of the work presented in this thesis and offers avenues for improvement.

Chapter 2

Materials and Methods

This chapter first introduces the experimental data, i.e. the pulmonary blood flow and pressure measurements (Section 2.1) based on which the statistical analysis was carried out. Section 2.2 offers a description of the mathematical models providing haemodynamic predictions. These predictions are compared to the measurements by using Bayesian statistical methods based on MCMC, described in Section 2.3. Several concepts related to MCMC are reviewed, such as random-walk and gradient-based MCMC type algorithms, as well as MCMC convergence, efficiency and consistency diagnostics. Additionally, the concept of Gaussian Processes is introduced in Section 2.3, which will be jointly used with MCMC in Chapters 4 and 6. Lastly, model selection criteria are reviewed, which will be used to discriminate between competing mathematical models in Chapters 3 and 4.

2.1 Experimental data

This study compares model predictions to measured MPA blood pressure data from a control mouse lung (Figure 2.1). The experimental protocols used to extract the haemodynamic and image data are summarised in our recent study [151], and a more detailed experimental protocol is found in [188] and [202]. Here a brief overview of the data used in this study is provided.

A 3D segmentation of the vessel geometry was obtained from micro-CT images of excised mice lungs as described in detail by Vanderpool et al. [202]. After the image data were segmented, a 1D directional graph was obtained using the Vascular Modeling ToolKit (VMTK, `vmtk.org`, [5]) and custom algorithms presented in [38]. The analysis in Chapter 3 assumed non-zero radial tapering, which was inferred from data. For reasons discussed in our previous study [38], as well as Chapters 3 and 5, the radial tapering was assumed negligible in Chapters 4, 5 and 6.

Dynamic pressure and flow waves were measured in-vivo in the MPA. Pressure was measured using a 1.0-F pressure-tip catheter (Millar Instruments, Houston, TX) and recorded on a haemodynamic workstation (Cardiovascular Engineering, Norwood, MA) at 5 kHz. MPA flow velocity was simultaneously measured during catheterisation on the same workstation via ultrasound (Visualsonics, Toronto, Ontario, CA) at a rate of 30 MHz [188]. The haemodynamic data include wave forms averaged over 20 cardiac cycles, using a fiducial point from simultaneously recorded ECG data. The raw data over 20 heart beats were not available. Consequently, it was not possible to estimate heart beat-to-beat variability in the pressure and flow waveforms.

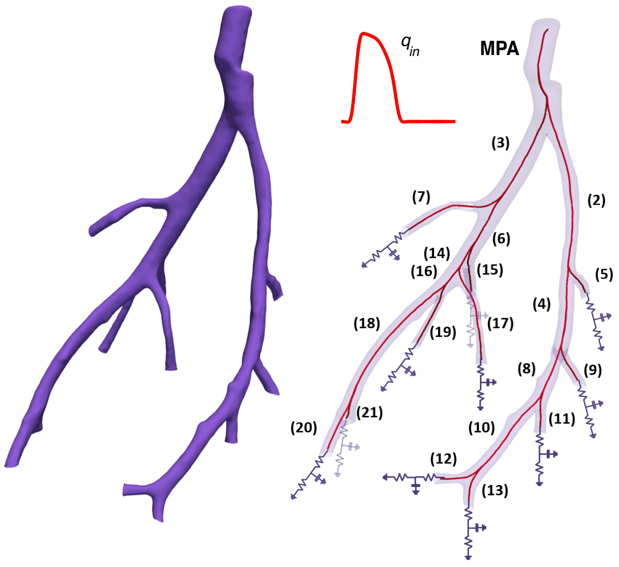


Figure 2.1: 3D smoothed segmented network from a micro-CT image of a healthy mouse lung (left) and the directional graph of the same network with vessel numbers attached (right). At the network inlet a flow waveform taken from measurements is specified, and at the outlet of each terminal vessel three-element Windkessel model with two resistors and a capacitor are attached.

2.2 Mathematical modelling

2.2.1 Fluid-dynamics model

This study uses a 1D fluid-dynamics model developed by my collaborators, Qureshi et al. [151], to simulate pressure, flow, and cross-sectional area in the pulmonary arterial network shown in Figure 2.1. This section summarises the model; the reader is referred to the study [151] in which it was proposed for details. The 1D model is derived under the assumptions that blood is incompressible and that the flow is Newtonian, laminar and axisymmetric, and has no swirl. Under these assumptions, the Navier−Stokes equations describing conservation of mass and momentum reduce to

$$\begin{aligned} \frac{\partial A}{\partial t} + \frac{\partial q}{\partial x} &= 0, \\ \frac{\partial q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{q^2}{A}\right) + \frac{A}{\rho}\frac{\partial p}{\partial x} &= -\frac{2\pi\mu r}{\rho\delta}\frac{q}{A}, \end{aligned} \tag{2.1}$$

where x (cm) and t (s) denote the distance (along each vessel) and time, p (mmHg) denotes the blood pressure, q (ml/s) denotes volumetric blood flow, and A (cm^2) denotes the cross-sectional area. $A(x,t) = \pi r(x,t)^2$, where $r(x,t)$ (cm) is the vessel radius. $\rho = 1.055$ g/ml is the blood density and $\mu = 0.049$ g/(cm s) is the blood viscosity, assumed constant. Lastly, a Stokes boundary layer was applied with a linearly decreasing boundary layer thickness, parameterised by $\delta = \sqrt{\mu T/2\pi\rho}$ (cm), where T (s) is the length of the cardiac cycle.

2.2.2 Vessel wall elasticity model

To close the system of equations, a constitutive equation relating pressure and area was added. Two types of wall models were investigated:

- Linear elastic wall model [167] in two forms:

  1. Linear A
  $$p = \frac{4}{3}\frac{Eh}{r_0}\left(1 - \sqrt{\frac{A_0}{A}}\right), \tag{2.2}$$
  2. Linear B
  $$p = \frac{4}{3}\frac{Eh}{r_0}\left(\sqrt{\frac{A}{A_0}} - 1\right), \tag{2.3}$$

  where $\frac{Eh}{r_0}$ (mmHg) denotes the vessel stiffness defined in terms of Young's modulus E (mmHg) in the circumferential direction, the wall thickness h (cm), and the reference vessel radius r_0 (cm) corresponding to the cross-sectional area A_0 ($A_0 = \pi r_0^2$). The reference vessel radius r_0 is obtained from the image segmentation process, see Chapter 5 for details.

- Non-linear elastic wall model [151] given by
  $$p = p_1 \tan\left[\frac{\pi}{\gamma}\left(\frac{A}{A_0} - 1\right)\right], \tag{2.4}$$

  where $p_1 > 0$ (mmHg) is the pressure that achieves the half-maximum compliance (equivalent of parameter $\frac{Eh}{r_0}$ for the linear model), and $\gamma > 0$ (dimensionless) is a scaling parameter specifying the maximal lumen area A_∞ for $p \to \infty$.

In both wall model formulations, as originally proposed by Olufsen [139] using data from the systemic circulation, the stiffness, i.e. $\frac{Eh}{r_0}$ or p_1 is assumed to be expressed by

$$s(f_1, f_2, f_3) = f_1 \exp(f_2 r_0) + f_3, \tag{2.5}$$

where, f_1 (mmHg), f_2 (cm^{-1}), $f_2 \leq 0$, and f_3 (mmHg).

2.2.3 Boundary conditions

At the inlet to the network (shown in Fig. 2.1), a flow taken from measurements was specified. At each junction it was assumed that flow is conserved and pressure is continuous, i.e.

$$\begin{aligned} p_{\mathrm{p}}(L_{\mathrm{p}},t) &= p_{\mathrm{d}_1}(0,t) = p_{\mathrm{d}_2}(0,t), \quad \forall t \geq 0 \\ q_{\mathrm{p}}(L_{\mathrm{p}},t) &= \sum_{i=1}^{2} q_{\mathrm{d}_i}(0,t), \quad \forall t \geq 0 \end{aligned} \tag{2.6}$$

where p denotes the parent vessel, d_1 and d_2 are the daughter vessels, and L_{p} (cm) is the length of the parent vessel. Finally, at the terminal vessels, pressure and flow were related via a 3-element Windkessel model (represented by a RCR circuit), and the relation can be expressed either in frequency or time domain, as follows:

- Frequency domain:

$$Z(L,\omega) = \frac{P(L,\omega)}{Q(L,\omega)} = R_1 + \frac{R_2}{1+i\omega CR_2} \implies p(L,t) = \frac{1}{T}\int_0^T q(L,t-\tau)z(L,\tau)d\tau, \tag{2.7}$$

- Time domain:

$$\frac{dp(L,t)}{dt} - R_1\frac{dq(L,t)}{dt} = q(L,t)\left(\frac{R_1+R_2}{R_2C}\right) - \frac{p(L,t)}{R_2C}, \tag{2.8}$$

where $Z(L,\omega)$ is the impedance, $P(L,\omega)$ and $Q(L,\omega)$ are the pressure and flow in frequency domain, T (s) is the length of the cardiac cycle, $\omega = \frac{2\pi}{T}$ is the angular frequency, R_1 and R_2 (mmHg s/ml) are the two resistances (proximal and distal), C (ml/mmHg) is the capacitance, and $1+i\omega CR_2$ is a complex number. Equations (2.7) and (2.8) are equivalent.

2.2.4 Model parameters

The haemodynamics modelling parameters include those describing the vascular structure (vessel radius, length, and stiffness), the fluid dynamics (including viscosity, density, and the boundary layer thickness), and the inflow and outflow boundary conditions. The inflow, viscosity, density, and boundary layer thickness were assumed known (see Section 2.2.1). Parameters specifying the vessel stiffness, radius, length, and Windkessel outflow boundary conditions were allowed to vary. The vessel radius and length were inferred from multiple segmentations of a CT-scan image of a mouse lung. The vessel stiffness was estimated from the MPA pressure data. The three Windkessel elements (R_1, R_2, C) vary across the different terminal arteries. Nominal (initial) values for these parameters were computed using the junction conditions and Poiseuille's flow as described in detail in our previous studies [38, 151], and briefly summarised here. The nominal total compliance C_0^j for every terminal vessel j can be determined from the diastolic pressure decay [151], i.e.

$$C_0^j = \frac{\tau}{R_\mathrm{T}^j}, \tag{2.9}$$

where τ is a constant pressure diastolic decay time (computed based on the MPA data) and R_T^j is the total vascular resistance for the j^th vessel, calculated as:

$$R_\mathrm{T}^j = \frac{\bar{p}}{\bar{q}^j}, \tag{2.10}$$

where $\bar{p}$ is the mean MPA pressure and $\bar{q}^j$ is the mean flow in the j^th terminal vessel. To find the mean flow distribution relationship (hence find $\bar{q}^j$), the junction conditions in eq (2.6) can be used together with Poiseuille's law [37, 151] to give:

$$\bar{q}_{\mathrm{d}_1} = \bar{q}_\mathrm{p}\frac{\psi_{\mathrm{d}_1}}{\psi_{\mathrm{d}_1}+\psi_{\mathrm{d}_2}}, \quad \bar{q}_{\mathrm{d}_2} = \bar{q}_\mathrm{p}\frac{\psi_{\mathrm{d}_2}}{\psi_{\mathrm{d}_1}+\psi_{\mathrm{d}_2}}, \tag{2.11}$$

where $\psi_i = \frac{r_i^4}{l_i}$, where r and l are the vessel radius and length, and i stands for d_1 or d_2. Eq (2.11) is repeatedly applied from the MPA down the tree, to the terminal vessels.

After R_T^j in eq (2.10) is found, the nominal (initial) resistance values for each terminal vessel are computed by setting

$$R_{01}^j = 0.2R_\mathrm{T}^j, \quad R_{02}^j = 0.8R_\mathrm{T}^j. \tag{2.12}$$

Once nominal resistance and compliance estimates are found for every terminal vessel, global scaling factors r_1, r_2, c for these estimates were introduced, and they were kept constant across all 11 terminal arteries (since estimating 33 Windkessel parameters from MPA data would most likely lead to parameter non-identifiability). Two alternative scaling methods were used:

$$R_1^j = (1-0.5r_1)R_{01}^j, \quad R_2^j = (1-0.5r_2)R_{02}^j, \quad C^j = (1-0.5c)C_0^j, \tag{2.13}$$

and

$$R_1^j = r_1R_{01}^j, \quad R_2^j = r_2R_{02}^j, \quad C^j = cC_0^j. \tag{2.14}$$

The three Windkessel scaling factors r_1, r_2, c were inferred from the MPA pressure data.

2.2.5 Overview of models

An overview of the models considered throughout this thesis is given in Table 2.1.

Chapter	Model	Windkessel scaling
3 (first part)$^{(*)}$ & 6$^{(*)}$	Linear A (eq (2.2))	eq (2.13)
5$^{(**)}$	Linear A (eq (2.2))	eq (2.14)
3 (second part)$^{(*)}$ & 4$^{(**)}$	Linear B (eq (2.3))	eq (2.14)
3 (second part)$^{(*)}$ & 4$^{(**)}$	Non-linear (eq (2.4))	eq (2.14)

Table 2.1: Summary of models used in the chapters of this thesis. The linear A model was used in early work, while the linear B model was used in later stages of the work, as numerical instabilities were encountered with the linear A model. (*) signifies that the Windkessel equations were solved in frequency domain (see eq (2.7)), while (**) stands for the time domain (see eq (2.8)).

2.3 Statistical methods

2.3.1 Approach to Bayesian inference in ODEs/PDEs

An ODE describes the dynamics of a system state for any 1D variable, which in this thesis is taken to be time, and takes the form:

$$\frac{d\mathbf{m}(t)}{dt} = f(\mathbf{m}(t), \theta), \tag{2.15}$$

where $\mathbf{m}$ represents the time-dependent state of the system, t is time, θ are some unknown parameters, and $f(.)$ is a potentially non-linear vector-valued function.

In contrast, a PDE may describe the dynamics of a state for a 2D variable, which in this thesis is taken to be time t and space x:

$$\frac{\partial \mathbf{m}(t,x)}{\partial t} = f_t(\mathbf{m}(t,x), \theta), \quad \frac{\partial \mathbf{m}(t,x)}{\partial x} = f_x(\mathbf{m}(t,x), \theta), \tag{2.16}$$

Noisy measurements (assumed for now to be independent and identically distributed, additive Gaussian) may be recorded at certain time points (and locations). For simplicity, the time and location indices are suppressed and the solution (output) from the ODEs/PDEs denoted as $\mathbf{m}(\theta)$, since this solution depends on the unknown parameters θ:

$$\mathbf{y} = \mathbf{m}(\theta) + \varepsilon, \quad \varepsilon \sim \mathscr{MVN}(\mathbf{0}, \sigma^2\mathbf{I}), \tag{2.17}$$

which can be equivalently expressed as

$$\mathbf{y}|\theta \sim \mathscr{MVN}(\mathbf{m}(\theta), \sigma^2\mathbf{I}), \tag{2.18}$$

where $\mathscr{MVN}$ stands for the multivariate normal distribution. This defines the data likelihood, and can be further expressed as:

$$p(\mathbf{y}|\theta,\sigma^2) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{\sum_{i=1}^n (y_i - m_i(\theta))^2}{2\sigma^2}\right) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{\mathscr{S}(\theta)}{2\sigma^2}\right), \tag{2.19}$$

where $\mathbf{m}(\theta) = (m_1(\theta),\ldots,m_n(\theta))$ is the vector of predictions from the ODEs/PDEs, $\mathbf{y} = (y_1,\ldots,y_n)$ is the vector of measurements, and n is the number of data points.

Bayes Theorem can be used to combine the data likelihood with the prior distribution, $p(\theta)$ (encoding knowledge about the parameters prior to seeing the data) to form the posterior distribution, $p(\theta|\mathbf{y})$:

$$p(\theta|\mathbf{y}) = \frac{p(\mathbf{y}|\theta)p(\theta)}{\int_\theta p(\mathbf{y}|\theta)p(\theta)d\theta}, \tag{2.20}$$

However, the integral in the denominator, which is simply a normalisation constant (obtained based on all possible values of the parameters), denoted by $C = \int_\theta p(\mathbf{y}|\theta)p(\theta)d\theta$ here, is usually intractable due to being multidimensional or for models with complicated forms for the posterior distribution[1]. Hence, the exact posterior distribution cannot be explicitly obtained, and numerical schemes based on Markov Chain Monte Carlo (MCMC) sampling may be used to approximate the posterior distribution:

$$p(\theta|\mathbf{y}) = C^{-1}p(\mathbf{y}|\theta)p(\theta) \propto p(\mathbf{y}|\theta)p(\theta), \tag{2.21}$$

thus for the MCMC sampling only the terms $p(\mathbf{y}|\theta)$ and $p(\theta)$ are needed, while the normalisation term can be ignored. The samples drawn converge in distribution to the normalised posterior distribution nevertheless, and an explanation of this is given below. In MCMC a Markov chain is set up that has as stationary distribution the (normalised) posterior distribution that one would like to sample from. If the Markov chain satisfies detailed balance with respect to a distribution, then that distribution is its stationary distribution. By defining by $T(.)$ the transition probability of the chain, and $g(\theta) = p(\mathbf{y}|\theta)p(\theta)$ one can immediately see that the following equivalence holds:

$$g(\theta)T(\theta^*|\theta) = g(\theta^*)T(\theta|\theta^*) \iff p(\theta|\mathbf{y})T(\theta^*|\theta) = p(\theta^*|\mathbf{y})T(\theta|\theta^*) \tag{2.22}$$

Hence the Markov chain with transition probability $T(.)$ has the normalised distribution $p(.)$ as a stationary distribution.

For non-linear differential equations, a second layer of intractability further complicates the analysis, i.e. the non-linear differential equations do not have an analytical solution, hence numerical schemes (e.g. Runge-Kutta [163] for ODEs, or Lax-Wendroff scheme [113] for PDEs) must be employed, which may result into numerical errors.

[1] For simple models with a low number of parameters and conjugate priors, the normalisation constant can be easily obtained, e.g. a Beta-Binomial model, see Section 5.3.2 in [128].

MCMC can be based on a random-walk, and can also be gradient-based. Both types are summarised below.

2.3.2 Random-walk MCMC

Metropolis-Hastings

In a Metropolis-Hastings (M-H) algorithm [83], given the current point, θ, a new point θ^* is generated from a proposal density $q(\theta^*|\theta)$. By denoting $a \wedge b = \min(a,b)$, the proposed point is accepted with the following acceptance probability:

$$\alpha(\theta^*|\theta) = 1 \wedge \frac{p(\theta^*|\mathbf{y})q(\theta|\theta^*)}{p(\theta|\mathbf{y})q(\theta^*|\theta)}, \tag{2.23}$$

where $p(\theta|\mathbf{y})$ is the posterior distribution of θ given the data $\mathbf{y}$, expressed as:

$$p(\theta|\mathbf{y}) = \frac{p(\theta)p(\mathbf{y}|\theta)}{p(\mathbf{y})}, \tag{2.24}$$

where $p(\theta)$ is the prior distribution, $p(\mathbf{y}|\theta)$ is the data likelihood, and $p(\mathbf{y}) = \int p(\theta)p(\mathbf{y}|\theta)d\theta$ is the marginal likelihood.

If the new point is accepted, it becomes the current point and the algorithm proposes a new point. If a symmetric proposal distribution is used, e.g. $q(\theta^*|\theta^{k-1}) = \mathscr{MVN}(\theta^{k-1},\mathbf{V})$, then the ratio of the proposal distributions is 1.

The size of the step taken to move from the current to the proposed point (i.e. defined through the covariance matrix of the proposal distribution) controls the efficiency of the algorithm. A poor proposal distribution results in high rejection rate.

DR

To overcome the difficulty of choosing an optimal step size, upon rejection of θ^*, instead of retaining θ^{k-1}, one can make a second attempt to move [193]. The 2nd stage proposal can be generated from a different distribution [78], e.g. $\mathscr{MVN}(\theta^{k-1},\beta^{\mathrm{T}}\beta\mathbf{V})$, where β is a scaling matrix used to alter the step size of the sampler. Studies in [78, 81] suggest reducing the variance upon rejection of a proposal, as this can result in estimators with a lower asymptotic variance.

The 2nd stage proposal is accepted with the following probability:

$$\alpha_2(\theta^{*(2)}|\theta^*,\theta^{k-1}) = 1 \wedge \frac{p(\theta^{*(2)}|\mathbf{y})q_1(\theta^*|\theta^{*(2)})q_2(\theta^{k-1}|\theta^*,\theta^{*(2)})[1-\alpha_1(\theta^*|\theta^{*(2)})]}{p(\theta^{k-1}|\mathbf{y})q_1(\theta^*|\theta^{k-1})q_2(\theta^{*(2)}|\theta^*,\theta^{k-1})[1-\alpha_1(\theta^*|\theta^{k-1})]}. \tag{2.25}$$

The acceptance probability in eq (2.25) is calculated in a way that ensures detailed balance is preserved, see A.1 for a proof.

The Delayed Rejection (DR) algorithm may have a higher acceptance rate than M-H, which implies higher efficiency and smaller asymptotic variance of the estimates compared to the M-H algorithm [123].

AM

The Adaptive Metropolis (AM) [82] brings an improvement to the M-H algorithm by adapting the proposal covariance matrix based on past chain samples. This enables learning the shape of the target distribution, while ensuring that detailed balance is satisfied (ergodicity is ensured by assuming diminishing adaptation and that the target is bounded from above and has bounded support) [82]. The proposal covariance matrix in the AM algorithm is Gaussian centred at the current point. The covariance matrix is adapted at given intervals, after possibly some non-adaptation time t_{ad}, as follows:

$$\mathbf{V}_k = \begin{cases} \mathbf{V}_0, & \text{if } k \leq t_{\text{ad}} \\ s_d \text{Cov}(\theta^0, \dots, \theta^{k-1}) + \delta \mathbf{I}_d, & \text{if } k > t_{\text{ad}}, \end{cases} \tag{2.26}$$

where $\mathbf{V}_0$ is the initial proposal covariance matrix, s_d is a parameter that depends on the dimension d of the target ($s_d = 2.38^2/d$ has been shown to be optimal for Gaussian targets in terms of mixing [82]), $\delta > 0$ is a small constant (e.g. 10^{-9}) that ensures that $\mathbf{V}_k$ does not become singular, $\theta^0, \dots \theta^{k-1}$ are the past chain samples, and their covariance is given by

$$\text{Cov}(\theta^0, \dots \theta^{k-1}) = \frac{1}{k-1} \left(\sum_{i=0}^{k-1} \theta_i \theta_i^{\text{T}} - k \bar{\theta}_{k-1} \bar{\theta}_{k-1}^{\text{T}} \right), \tag{2.27}$$

where $\bar{\theta}_{k-1} = \frac{1}{k} \sum_{i=0}^{k-1} \theta_i$ and θ_i is a $d \times 1$ vector.

DRAM

The DR and AM algorithms can be married to produce the DRAM algorithm, which satisfies detailed balance [81]. In the first stage of DRAM, the proposal distribution is $\mathscr{MVN}(\theta^{k-1}, \mathbf{V}_k)$, where the proposal covariance matrix $\mathbf{V}_k$ is calculated according to AM (eq (2.26)). In the 2nd stage of DRAM, the proposal distribution is $\mathscr{MVN}(\theta^{k-1}, \beta^{\text{T}} \beta \mathbf{V}_k)$, i.e. the proposal covariance matrix is a scaled version of the covariance matrix in the first stage.

DRAM is superior to M-H in terms of efficiency due to local adaptation based on rejected proposals within each MCMC iteration in the DR step and global adaptation based on all previously accepted samples in the AM step [81].

2.3.3 Gradient-based MCMC

HMC

HMC [132] is a powerful MCMC scheme which suppresses the random walk behaviour of Metropolis-Hastings MCMC by introducing an auxiliary variable, the 'momentum' variable, $\mathbf{r} \in \mathbb{R}^{d\times 1}$, with density $p(\mathbf{r}) = \mathscr{MVN}(\mathbf{r}|\mathbf{0},\mathbf{M})$, which guides the search towards high posterior density regions. HMC simulates Hamiltonian dynamics by using gradient information from the log target density.

Defining H: Hamiltonian, $\mathbf{r}$: 'momentum' vector, and θ : 'position' vector (representing the parameters that need inferring), the Hamiltonian dynamic equations are:

$$\frac{d\theta_i}{dt} = \frac{\partial H}{\partial r_i}, \quad \frac{dr_i}{dt} = -\frac{\partial H}{\partial \theta_i}. \tag{2.28}$$

The Hamiltonian dynamics are:

- reversible – the mapping from the current to the proposed point is reversed by negating the time derivative equations (2.28), thus, the dynamics leave the target distribution invariant.
- leave the Hamiltonian invariant, i.e. $H(\theta^*,\mathbf{r}^*) = H(\theta,\mathbf{r})$,

$$\frac{dH}{dt} = \sum_{i=1}^{d}\left(\frac{\partial H}{\partial r_i}\frac{dr_i}{dt} + \frac{\partial H}{\partial \theta_i}\frac{d\theta_i}{dt}\right) = \sum_{i=1}^{d}\left(\frac{\partial H}{\partial r_i}\left(-\frac{\partial H}{\partial \theta_i}\right) + \frac{\partial H}{\partial \theta_i}\frac{\partial H}{\partial r_i}\right) = 0. \tag{2.29}$$

- preserve volume in phase space, hence the determinant of the Jacobian matrix is 1, and no adjustment is needed in the M-H acceptance probability.

Further defining $\mathbf{M}$: 'mass matrix' for the 'momentum' (its covariance matrix), E: 'potential energy', K: 'kinetic energy',

$$E(\theta) = -(\log p(\theta) + \log p(\mathbf{y}|\theta)); \quad K(\mathbf{r}) = \frac{\mathbf{r}^T\mathbf{M}^{-1}\mathbf{r}}{2}; \quad H(\theta,\mathbf{r}) = E(\theta) + K(\mathbf{r}), \tag{2.30}$$

where $\log p(\theta)$ and $\log p(\mathbf{y}|\theta)$ are the log prior distribution of the parameters and the log data likelihood.

Note that the terms 'mass matrix', 'momentum', as well as 'kinetic' and 'potential energy' are used for their mathematical equivalence to the corresponding terms in Hamiltonian mechanics.

In statistical terms, the joint distribution is

$$p(\theta,\mathbf{r}|\mathbf{y}) = p(\theta|\mathbf{r},\mathbf{y})p(\mathbf{r}) = p(\theta|\mathbf{y})\mathscr{MVN}(\mathbf{r}|\mathbf{0},\mathbf{M}) \propto \exp(-E(\theta))\exp(-K(\mathbf{r})) \propto \exp(-H(\theta,\mathbf{r})). \tag{2.31}$$

Thus, by denoting the log posterior of the parameters that need inferring θ as $\log p(\theta|\mathbf{y})$, the negative log auxiliary joint distribution is given by:

$$H(\theta,\mathbf{r}|\mathbf{y}) = -\log p(\theta|\mathbf{y}) + \frac{1}{2}\log((2\pi)^d|\mathbf{M}|) + \frac{\mathbf{r}^T\mathbf{M}^{-1}\mathbf{r}}{2} + \log Z, \tag{2.32}$$

where Z is a normalising constant and $|\mathbf{M}|$ is the determinant of the 'mass matrix' $\mathbf{M}$.

The Hamiltonian dynamics (eq (2.28)) are numerically integrated for a specified fictitious time. The Leapfrog integrator [132] is used, in which time is discretised using a small step size, $\varepsilon > 0$, and the trajectory is run for a number of leapfrog steps, L.

$$\mathbf{r}\left(t+\frac{\varepsilon}{2}\right) = \mathbf{r}(t) + \frac{\varepsilon}{2}\frac{d\mathbf{r}}{dt}, \tag{2.33}$$

$$\theta\left(t+\varepsilon\right) = \theta(t) + \varepsilon\frac{d\theta}{dt}, \tag{2.34}$$

$$\mathbf{r}(t+\varepsilon) = \mathbf{r}\left(t+\frac{\varepsilon}{2}\right) + \frac{\varepsilon}{2}\frac{d\mathbf{r}}{dt}, \tag{2.35}$$

where using equations (2.28) and (2.32), we have:

$$\frac{d\theta}{dt} = \mathbf{M}^{-1}\mathbf{r}, \quad \frac{d\mathbf{r}}{dt} = \frac{d\log p(\theta|\mathbf{y})}{d\theta}. \tag{2.36}$$

The Leapfrog integrator ensures reversibility (by negation of the time derivative eqns) and unit determinant of the Jacobian matrix (since $(\mathbf{r},\theta) \to \left(\mathbf{r} + \frac{\varepsilon}{2}\frac{d\log p(\theta|\mathbf{y})}{d\theta}, \theta\right)$ and $(\mathbf{r},\theta) \to (\mathbf{r}, \theta + \varepsilon\mathbf{M}^{-1}\mathbf{r})$).

However, the numerical integration induces an error, which implies that $H(\theta^*,\mathbf{r}^*)$ no longer equals $H(\theta,\mathbf{r})$. This bias is corrected by the M-H accept/reject step, which ensures convergence to the correct target distribution.

In HMC, at the end of each leapfrog trajectory (defined as a segment between two subsequent acceptance steps), a new point is proposed and accepted with probability

$$\alpha(\theta^*,\mathbf{r}^*|\theta,\mathbf{r}) = 1 \wedge \exp(-H(\theta^*,\mathbf{r}^*|\mathbf{y}) + H(\theta,\mathbf{r}|\mathbf{y}). \tag{2.37}$$

If the error from the numerical integration of the Hamiltonian dynamics in eq (2.28) is small, then the acceptance probability will be high. If the new point is rejected, the current point is kept. The next trajectory is then simulated, and each trajectory starts with the resampling of the 'momentum' variables from their marginal distribution $\mathscr{MVN}(\mathbf{r}|\mathbf{0},\mathbf{M})$, to allow properly being integrated out from the joint distribution, i.e.

$$p(\theta|\mathbf{y}) = \int p(\theta|\mathbf{r},\mathbf{y})p(\mathbf{r})d\mathbf{r}. \tag{2.38}$$

The HMC algorithm produces an ergodic, time reversible Markov chain which satisfies detailed balance (see proof in Section A.2 in the Appendix) and whose stationary distribution is the marginal distribution $p(\theta|\mathbf{y})$.

In the classical HMC algorithm, $\mathbf{M}$ is kept fixed at the identity matrix, and the HMC hyperparameters, ε and L, are hand-tuned to get an acceptance rate $> 65\%$ [132] and a large effective sample size. Throughout this thesis the hyperparameters are optimised using Bayesian optimisation, see Section 2.3.3 for details.

Below extensions to the HMC algorithm are summarised, and they are aimed at improving the algorithm's performance by automatic tuning of the HMC hyperparameters or of the mass matrix $\mathbf{M}$, to which the algorithm is known to be highly sensitive.

RMHMC

Riemann Manifold HMC (RMHMC) [74] is an improved version of HMC, as it exploits the Riemannian geometry of the parameter space. RMHMC sets $\mathbf{M}$ based on the curvature of the (approximate) target distribution. $\mathbf{M}$ is the metric tensor of the Riemann space and is calculated based on the expected Fisher information matrix plus the negative Hessian log prior:

$$M_{i,j} = \mathbb{E}_{\mathbf{y}|\theta}\left[-\frac{\partial^2 \log p(\mathbf{y},\theta)}{\partial\theta_i\partial\theta_j}\right] = \mathbb{E}_{\mathbf{y}|\theta}\left[-\frac{\partial^2 \log p(\mathbf{y}|\theta)}{\partial\theta_i\partial\theta_j}\right] + \left(-\frac{\partial^2 \log p(\theta)}{\partial\theta_i\partial\theta_j}\right). \tag{2.39}$$

The first term in the equality above is the expected Fisher information matrix, and the second term is the negative Hessian log prior. The expected Fisher information matrix can be taken to be the covariance of the vectors $\frac{\partial \log p(\mathbf{y}|\theta)}{\partial\theta}$:

$$\begin{aligned} &\mathrm{Cov}\left(\frac{\partial \log p(\mathbf{y}|\theta)}{\partial\theta_i}, \frac{\partial \log p(\mathbf{y}|\theta)}{\partial\theta_j}\right) \\ &= \mathbb{E}_{\mathbf{y}|\theta}\left[\frac{\partial \log p(\mathbf{y}|\theta)}{\partial\theta_i}\frac{\partial \log p(\mathbf{y}|\theta)}{\partial\theta_j}\right] = \mathbb{E}_{\mathbf{y}|\theta}\left[-\frac{\partial^2 \log p(\mathbf{y}|\theta)}{\partial\theta_i\partial\theta_j}\right]. \end{aligned} \tag{2.40}$$

The mass matrix $\mathbf{M}$ changes within every trajectory to adapt to the target density curvature ($\mathbf{M} \rightarrow \mathbf{M}(\theta)$ in equation (2.32)), and eqns (2.39) and (2.40) ensure that the covariance matrix $\mathbf{M}$ is always positive definite. In the Appendix, Section A.6, the equality in eq (2.40) is proven to hold.

The joint distribution in equation (2.31) is no longer factorisable. An implicit integrator is used (the Generalised Leapfrog algorithm), as proposals generated from the Leapfrog integrator no longer satisfy detailed balance in HMC: $\mathbf{M}(\theta(t)) \neq \mathbf{M}(\theta(t+\varepsilon))$. The integrator ensures reversibility and unit determinant of the Jacobian matrix. The implicit integrator incurs high numerical costs. The resulting RMHMC algorithm can be proved to satisfy detailed balance, see proof in SectionA.3 in the Appendix. In the classical RMHMC, ε and L are fixed, however in the current work they are optimised with Bayesian optimisation, see details in Section 2.3.3.

LDMC

To overcome the increased computational costs in RMHMC associated with iteratively solving the equations of the implicit integrator, Lagrangian dynamics can be used instead of Hamiltonian dynamics. This leads to the Lagrangian Dynamical Monte Carlo algorithm - LDMC [109]. LDMC uses an explicit geometric integrator that replaces the 'momentum' variable in RMHMC by 'velocity' (used in analogy with classical mechanics), which improves the computational efficiency. However, the volume in phase space is no longer preserved, hence the Jacobian transformation is needed to adjust the acceptance probability to ensure detailed balance [109], and a proof can be found in Section A.4 in the Appendix. In LDMC, $\mathbf{M}$ is adjusted to the curvature of the posterior distribution at every step throughout the trajectory, and ε and L are traditionally kept fixed, however in this work they are optimised with Bayesian optimisation, see Section 2.3.3 for details.

NUTS

The No U-turn sampler (NUTS), proposed in [87], chooses L recursively by moving in parameter space until the HMC trajectory starts to double back and retrace its steps. This is achieved via a tree doubling process which implicitly builds a binary tree whose leaf nodes correspond to the position-momentum states. The points collected along the way are sampled in a way that ensures detailed balance (see proof in SectionA.5 in the Appendix). The algorithm adapts ε in the burn-in phase by means of a stochastic optimisation algorithm (the primal-dual averaging). $\mathbf{M}$ is kept fixed at the identity matrix.

Bayesian optimisation for hyperparameter tuning

Throughout this work Bayesian optimisation [125, 208] is utilised to tune the hyperparameters of HMC, RMHMC and LDMC (the step size ε and number of steps L). ε and L are optimised by maximising an objective function, which, following [208], is taken to be the expected squared jumping distance (ESJD) normalised by the number of leapfrog steps:

$$\frac{\mathbb{E}^{\varepsilon,L}_{p(.)}||\theta^{(t+1)} - \theta^{(t)}||^2}{\sqrt{L}}. \tag{2.41}$$

In a random walk the average distance travelled in space is proportional to $\sqrt{L}$ as the algorithm moves backward and forward, while in HMC the average distance is proportional to L as the algorithm moves in one direction following the gradient of the log posterior distribution. Thus, the normalisation term in ESJD could be anywhere between $\sqrt{L}$ and L. This study follows [208] and takes $\sqrt{L}$ to allow for the possibility that the HMC algorithm has not properly suppressed the random-walk behaviour.

The normalised ESJD contains an intractable expectation with respect to the target distribution $p(.)$ which is approximated by an empirical estimator based on the MCMC posterior samples. The idea of emulation and Bayesian optimisation [175] is adopted; the normalised ESJD is maximised by constructing a surrogate objective function using Gaussian Processes (reviewed in Section 2.3.7), called acquisition function. The problem is turned into the maximisation of this computationally cheap acquisition function. Following [208], the Upper Confidence Bound is taken as the acquisition function.

2.3.4 MCMC convergence diagnostics

Several MCMC convergence diagnostics are available to test whether the Markov chains resulting from running MCMC algorithms have converged in distribution to the posterior distribution. In this chapter a few such convergence tests are reviewed, with an emphasis on those used throughout this work.

Gelman Rubin test

Gelman-Rubin test [69] assesses convergence of a scalar (univariate) parameter for a number of chains run in parallel. The test computes the potential scale reduction factor (PSRF) based on the within and between variance of the posterior samples. Suppose m chains are run in parallel, and θ_i^k is the scalar parameter drawn at the k^{th} iteration, $k = 1, ...N$ from the i^{th} chain, $i = 1, ...m$. The within and between chain variance can be computed as follows:

$$W = \frac{1}{m(N-1)} \sum_{i=1}^{m} \sum_{k=1}^{N} (\theta_i^k - \bar{\theta}_{i.})^2 \tag{2.42}$$

and

$$B = \frac{N}{m-1} \sum_{i=1}^{m} (\bar{\theta}_{i.} - \bar{\theta}_{..})^2 \tag{2.43}$$

PSRF is given by

$$\hat{R}_{\text{PSRF}} = \frac{N-1}{N} + \frac{m+1}{m} \frac{B}{W} \frac{1}{N}. \tag{2.44}$$

If the chains have converged to the stationary distribution, then $\hat{R}_{\text{PSRF}} \to 1$ for $N \to \infty$ and $W \approx B$.

Brooks Gelman Rubin test

Brooks Gelman-Rubin test [18] is a multivariate extension to the Gelman-Rubin test [69], assessing convergence of the parameters simultaneously for a number of chains run in parallel. It computes the multivariate potential scale reduction factor (MPSRF), which is an approximate estimate of the maximum upper bound of the univariate SRF for every parameter. Suppose m chains are run in parallel, and θ_i^k is the vector of parameters drawn at the k^{th} iteration, $k = 1, \ldots N$ from the i^{th} chain, $i = 1, \ldots m$. The within and between covariance matrices are given by

$$\mathbf{W} = \frac{1}{m(N-1)} \sum_{i=1}^{m} \sum_{k=1}^{N} (\theta_i^k - \bar{\theta}_{i.})(\theta_i^k - \bar{\theta}_{i.})^{\mathrm{T}} \tag{2.45}$$

and

$$\mathbf{B} = \frac{N}{m-1} \sum_{i=1}^{m} (\bar{\theta}_{i.} - \bar{\theta}_{..})(\bar{\theta}_{i.} - \bar{\theta}_{..})^{\mathrm{T}} \tag{2.46}$$

MPSRF is given by:

$$\hat{R}_{\text{MPSRF}} = \frac{N-1}{N} + \frac{m+1}{m}\lambda_1, \tag{2.47}$$

where λ_1 is the largest eigenvalue of the symmetric and positive definite matrix $\mathbf{W}^{-1}\mathbf{B}/N$.

If the chains have converged to the stationary distribution, then $\hat{R} \to 1$ for $N \to \infty$, $\mathbf{W} \approx \mathbf{B}$ and $\lambda_1 \to 0$.

If **W** is singular, MPSRF in eq (2.47) is incalculable. If solely **W** is singular, this means that one or more of the parameters have not moved away from the starting values, pointing to problems with the sampler. However, if both **W** and **B** are singular, it indicates that the problem is ill-posed, e.g. some of the parameters are very highly correlated.

In deciding how many independent chains to run and how to select the starting parameter values for each chain, the following can be noted. Cowles et al. [46] advise to run 10 independent chains if the posterior distribution is unimodal, and more for multi-modal distributions. Generally, as proposed in the study by Gelman and Rubin [69], an optimisation, mode-finding algorithm may be used to find regions of high density, and the starting values can be generated by sampling from a mixture of t-distributions centred at these modes.

Geweke convergence test

The Geweke convergence test [70] performs hypothesis testing to formally test for equality of the mean of two sub-chains (typically the first 10% and last 50% of the iterations within the chain). A p-value < 0.05 implies rejecting the null hypothesis of equal means in favour of the null hypothesis of different means, which suggests non-convergence of the sampler.

2.3.5 Sampler efficiency (mixing)

The sampler efficiency is defined as the ratio of the variance under iid sampling of the estimator (sample mean: $\bar{\theta} = \frac{1}{N}\sum_{i=1}^{N} \theta_i$, with N: number of MCMC samples), to the variance of the estimator under MCMC sampling [68], i.e.

$$\mathrm{Eff}(\bar{\theta}) = \frac{\mathrm{Var}_{\pi}(\bar{\theta})}{\mathrm{Var}_{p(.)}(\bar{\theta})} = \frac{1}{1+2\sum_{l=1}^{\infty}\rho_l}, \tag{2.48}$$

where Var_{π} denotes the variance under independent sampling, $\mathrm{Var}_{p(.)}$ denotes the variance under MCMC sampling, and $\rho_l = \frac{1}{N-l}\sum_{l=1}^{N-1}\theta_l\theta_{N-l}$ measures the autocorrelation between the chain samples at lag l. Highly efficient samplers give an efficiency close to 1, while less efficient samplers have high autocorrelations, thus larger variance of the estimator:

$$\mathrm{Var}_{p(.)}(\bar{\theta}) = \mathrm{Var}_{\pi}(\bar{\theta}) + 2\sum_{i=2}^{\infty}\mathrm{Cov}_{\pi}(\theta_1, \theta_i), \tag{2.49}$$

where Cov_{π} denotes the covariance under the stationary distribution.

The effective sample size ESS [95] can be calculated, which gives an estimate of the number of independent samples out of the total number of MCMC samples, and is defined as:

$$\mathrm{ESS} = \frac{N}{1+2\sum_{l=1}^{\infty}\rho_l}. \tag{2.50}$$

The denominator of eq (2.50), $\tau = 1+2\sum_{l=1}^{\infty}\rho_l^i$ is called the integrated autocorrelation time (IACT) [34]. ESS estimates in this thesis are calculated based on the monotone positive sequence estimator of Geyer [72], and low values can indicate inefficiency, hence poor mixing or lack of convergence of the sampler. For a d-dimensional parameter space, ESS can be defined per parameter $i, i = 1, \ldots d$:

$$\mathrm{ESS}^i = \frac{N}{1+2\sum_{l=1}^{\infty}\rho_l^i}. \tag{2.51}$$

2.3.6 Geweke consistency test: convergence to the correct posterior distribution

The mathematical and coding correctness of the MCMC samplers, i.e. convergence to the correct posterior distribution is assessed using the Geweke consistency test [71], as described below:

S.1 Draw samples from the prior distribution: $\theta^{(i)} \sim p(\theta)$ for $i = 1, \ldots M$, where $M \to \infty$

S.2 For every prior sample $\theta^{(i)}$, generate a noisy data set $D^{(i)}$, $\theta^{(i)} \to D^{(i)}$

S.3 For every data set $D^{(i)}$, run an MCMC simulation to obtain the corresponding posterior distribution, $p(\theta|D^{(i)}) \leftarrow \mathrm{MCMC}\left(p(\theta), p(D^{(i)}|\theta)\right)$

S.4 The mathematical correctness of the sampler is verified by checking if the following holds:

$$\mathbb{E}_{\mathbf{D}}(p(\theta|\mathbf{D})) = \int p(\theta|\mathbf{D})p(\mathbf{D})d\mathbf{D} = \int p(\theta)p(\mathbf{D}|\theta)d\mathbf{D} = p(\theta)\int p(\mathbf{D}|\theta)d\mathbf{D} = p(\theta)1 = p(\theta). \tag{2.52}$$

In practice S.4 is checked by comparing the prior distribution to the ensemble of posterior distributions thus obtained by following steps S.1-S.3. This can be achieved by investigating the quantiles of the prior distribution and of the ensemble of posterior distributions via a QQ plot. The sampler is considered mathematically correct if the two distributions agree, i.e. points lie on the equality line in the QQ plot, and thus eq (2.52) holds. Note that for every data set $D^{(i)}$, a number of MCMC posterior samples are obtained, and for the QQ plot, the last chain posterior sample can be used.

2.3.7 Gaussian Processes

This section gives a brief overview of Gaussian Processes (GPs). For a detailed introduction to GPs, the reader is referred to [157].

GPs for regression

Definition: A stochastic process $f = f(\mathbf{x})_{\mathbf{x}\in\mathscr{X}}$ is defined as a Gaussian process (GP) if the random variables $\mathbf{f} = (f(\mathbf{x}_1), \dots f(\mathbf{x}_n))$ are jointly normal for any inputs $\mathbf{x}_i \in \mathbb{R}^{d\times 1}$, with $i = 1, \dots n$, $\mathbf{f} \sim \mathscr{MVN}(\mathbf{m}, \mathbf{K})$, where $\mathbf{m} = (m(\mathbf{x}_1), \dots, m(\mathbf{x}_n))$ is the mean n-vector and $\mathbf{K} = [k(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^{n}$ is the $n \times n$ variance-covariance matrix of $\mathbf{f}$. In GP models [157, 54], inputs $\mathbf{X} = [\mathbf{x}_1, \dots \mathbf{x}_n]^{\mathrm{T}}$ ($\mathbf{X}$ is an $n \times d$ matrix) are mapped into outputs $\mathbf{y} = (y_1, \dots y_n)$ ($\mathbf{y}$ is an n-vector) by means of latent noiseless functions $\mathbf{f}$. A GP prior is placed on the distribution of these functions as a way to account for the uncertainty in the functional form. Assuming iid Gaussian noisy observations $\mathbf{y}$:

$$\begin{aligned} \mathbf{y}|\mathbf{f} &\sim \mathscr{MVN}(\mathbf{f}, \sigma^2\mathbf{I}), \\ \mathbf{f}(\mathbf{X})|\gamma &\sim \mathscr{GP}(\mathbf{m}(\mathbf{X}), \mathbf{K}|\gamma), \\ \gamma, \sigma^2 &\sim p(\gamma)p(\sigma^2), \end{aligned} \tag{2.53}$$

where γ contains the covariance function (kernel) hyperparameters, and σ^2 is the observation noise variance.

Covariance functions (kernels): The covariance function $k(\mathbf{x}, \mathbf{x}'|\gamma)$ gives the smoothness and variability of the latent functions. Several covariance functions are available, stationary (e.g. squared exponential, Matèrn class, periodic) or non-stationary (e.g. neural network), see Chapter 4 in Rasmussen's book [157] for a review.

The squared exponential kernel has the form:

$$k(\mathbf{x}, \mathbf{x}^{'}|\gamma) = \sigma_m^2 \exp\left(-\frac{||\mathbf{x}-\mathbf{x}^{'}||^2}{2l^2}\right), \tag{2.54}$$

where $\gamma = (\sigma_m^2, l)$, with σ_m^2 being the marginal variance of the function or response variable,

equivalent to the signal variance, and l being the lengthscale, which controls the flexibility of the function on the x-axis (the explanatory or input variable).
If there is more than one input variable (i.e. $d \geq 2$), each variable can have its own lengthscale (automatic relevance determination, ARD) [157], and eq (2.54) turns into

$$k(\mathbf{x},\mathbf{x}^{'}|\gamma) = \sigma_m^2 \exp\left(-0.5\sum_{j=1}^{d}\frac{(x_j - x_j^{'})^2}{l_j^2}\right). \tag{2.55}$$

The Matèrn class kernel takes the form:

$$k(\mathbf{x},\mathbf{x}^{'}|\gamma) = \sigma_m^2 \frac{2^{1-\nu}}{\Gamma(\nu)}(r\sqrt{2\nu})^{\nu} K_{\nu}(r\sqrt{2\nu}), \quad r = \sqrt{\sum_{j=1}^{d}\frac{(x_j - x_j^{'})^2}{l_j^2}}, \tag{2.56}$$

where K_ν is a modified Bessel function, and the parameter ν controls the smoothness of the GP functions. Examples of Matèrn kernels include:

- Matèrn 3/2:
$$k(r) = \sigma_m^2(1+\sqrt{3}r)\exp(-\sqrt{3}r), \tag{2.57}$$

- Matèrn 5/2:
$$k(r) = \sigma_m^2\left(1+\sqrt{5}r+\frac{5r^2}{3}\right)\exp(-\sqrt{5}r). \tag{2.58}$$

Additionally, the neural network kernel has the form:

$$k(\mathbf{x},\mathbf{x}^{'}|\gamma) = \frac{2}{\pi}\arcsin\left(\frac{2\hat{\mathbf{x}}^{\mathsf{T}}\Sigma\hat{\mathbf{x}}^{'}}{\sqrt{(1+2\hat{\mathbf{x}}^{\mathsf{T}}\Sigma\hat{\mathbf{x}})(1+2\hat{\mathbf{x}}^{'\mathsf{T}}\Sigma\hat{\mathbf{x}}^{'})}}\right), \tag{2.59}$$

where $\hat{\mathbf{x}} = (1 \quad \mathbf{x})^{\mathrm{T}}$ is an augmented vector (T superscript denotes the transposition of the vector), and Σ is a diagonal matrix containing the kernel hyperparameters, $\Sigma = \mathrm{diag}(b,w)$, and $\gamma = (b,w)$ with b controlling the amount of offset of the functions from the origin and w controlling the scaling on the x-axis (equivalent of the inverse lengthscale in stationary kernels).

The kernel hyperparameters γ and the observation noise variance σ^2 can be placed a prior on and integrated out from the joint posterior distribution $p(\mathbf{f},\gamma,\sigma^2|\mathbf{X},\mathbf{y})$, or they can be found in a so-called empirical Bayes approach by maximisation of the log marginal likelihood, see eqns (2.64) and (2.79).

Kernel differentiability: Differentiation is a linear operator, hence the derivative of a GP is another GP with the following covariance function

$$\mathrm{Cov}\left(\frac{\partial f(\mathbf{x})}{\partial x_i}, \frac{\partial f(\mathbf{x}^{'})}{\partial x_j^{'}}\right) = \frac{\partial^2 k(\mathbf{x},\mathbf{x}^{'})}{\partial x_i \partial x_j^{'}}. \tag{2.60}$$

Thus, the covariance function of a process $\dfrac{\partial f(\mathbf{x})}{\partial x_i}$ is given by $\dfrac{\partial^2 k(\mathbf{x},\mathbf{x}^{'})}{\partial x_i \partial x_j^{'}}$. For a stationary process $f(\mathbf{x})$, if the $2M^{\text{th}}$ order partial derivative $\dfrac{\partial^{2M} k(\mathbf{x},\mathbf{x}^{'})}{\partial x_1 \ldots \partial x_M \partial x_1^{'} \ldots \partial x_M^{'}}$ exists and is finite for $\mathbf{x} = \mathbf{x}^{'}$, i.e. at $||\mathbf{x}-\mathbf{x}^{'}|| = \mathbf{0}$, then the M^{th} order partial derivative $\dfrac{\partial^M f(\mathbf{x})}{\partial x_1 \ldots \partial x_M}$ exists for all $\mathbf{x} \in \mathbb{R}^{d\times 1}$, and $f(\mathbf{x})$ is said to be M^{th} order mean square (MS) differentiable. Thus, it is the properties of the kernel k around $\mathbf{0}$ that determines the MS differentiability, thus smoothness of a stationary process $f(\mathbf{x})$.

The GP defined by a squared exponential kernel (eq (2.54)) is infinitely differentiable in a MS sense with respect to $\mathbf{x}$ [157] (thus highly smooth), since by denoting $\mathbf{q} = \mathbf{x}-\mathbf{x}^{'}$, one can check that

$$\lim_{\mathbf{q}\to\mathbf{0}} \frac{\partial^{2M} k(\mathbf{x},\mathbf{x}^{'})}{\partial q_1 \ldots \partial q_{2M}} \tag{2.61}$$

exists for $M \to \infty$, implying that, by using the chain rule, the expression

$$\frac{\partial^{2M} k(\mathbf{x},\mathbf{x}^{'})}{\partial x_1 \ldots \partial x_M \partial x_1^{'} \ldots \partial x_M^{'}} = (-1)^M \frac{\partial^{2M} k(\mathbf{x},\mathbf{x}^{'})}{\partial q_1 \ldots \partial q_{2M}} \tag{2.62}$$

exists for $\mathbf{x} = \mathbf{x}^{'}$.

The GP defined by Matèrn kernels is $(\lceil \nu \rceil - 1)$ MS differentiable with respect to $\mathbf{x}$, e.g. the process is once differentiable with respect to x_i for Matèrn 3/2 (defined in eq (2.57)) and twice differentiable with respect to x_i, x_j for Matèrn 5/2 (defined in eq (2.58)). This can be proved by showing that the limit in eq (2.61) for $M > 1$ does not exist in the case of Matèrn 3/2 and for $M > 2$ in the case of Matèrn 5/2 by making use of the right-hand side of eq (2.56) and applying the chain rule of standard calculus.

Posterior distribution The posterior distribution of the latent functions can be constructed by using Bayes theorem:

$$p(\mathbf{f}|\mathbf{X},\mathbf{y},\gamma,\sigma^2) = \frac{p(\mathbf{f}|\mathbf{X},\gamma)p(\mathbf{y}|\mathbf{f},\sigma^2)}{\int p(\mathbf{f}|\mathbf{X},\gamma)p(\mathbf{y}|\mathbf{f},\sigma^2)d\mathbf{f}} = \frac{p(\mathbf{f}|\mathbf{X},\gamma)p(\mathbf{y}|\mathbf{f},\sigma^2)}{p(\mathbf{y}|\mathbf{X},\gamma,\sigma^2)}. \tag{2.63}$$

Zero mean GP: Typically, the mean of the GP is set to zero ($\mathbf{m}(\mathbf{X}) = \mathbf{0}$), i.e. the data are standardised to zero mean. In this case, the log marginal likelihood is found by integrating out the latent functions $\mathbf{f}$:

$$\log p(\mathbf{y}|\mathbf{X},\gamma,\sigma^2) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{K}+\sigma^2\mathbf{I}| - \frac{1}{2}\mathbf{y}^T(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{y}. \tag{2.64}$$

The aim of constructing a probabilistic model using GPs is prediction at unseen input values. The predictive distribution of a new function, $\tilde{f} = f(\tilde{\mathbf{x}})$, is available in closed form, i.e. it is

Gaussian [157]:

$$\tilde{f}|\tilde{\mathbf{x}},\mathbf{X},\mathbf{y},\gamma,\sigma^2 \sim \mathscr{N}(m_\mathrm{p}(\tilde{\mathbf{x}}),k_\mathrm{p}(\tilde{\mathbf{x}},\tilde{\mathbf{x}}^{'}|\gamma)), \tag{2.65}$$

$$m_\mathrm{p}(\tilde{\mathbf{x}}) = k(\tilde{\mathbf{x}},\mathbf{X}|\gamma)(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{y}, \tag{2.66}$$

$$k_\mathrm{p}(\tilde{\mathbf{x}},\tilde{\mathbf{x}}^{'}|\gamma) = k(\tilde{\mathbf{x}},\tilde{\mathbf{x}}^{'}|\gamma) - k(\tilde{\mathbf{x}},\mathbf{X}|\gamma)(\mathbf{K}+\sigma^2\mathbf{I})^{-1}k(\mathbf{X},\tilde{\mathbf{x}}^{'}|\gamma), \tag{2.67}$$

where $k(\tilde{\mathbf{x}},\mathbf{X}|\gamma)$ is a vector valued kernel function, $k(\tilde{\mathbf{x}},\mathbf{X}) : \mathbb{R}^{d\times 1}\times\mathbb{R}^{d\times n}\rightarrow\mathbb{R}_{+}^{1\times n}$, $k(\mathbf{X},\tilde{\mathbf{x}}|\gamma)$ is a vector valued kernel function, $k(\mathbf{X},\tilde{\mathbf{x}}) : \mathbb{R}^{d\times n}\times\mathbb{R}^{d\times 1}\rightarrow\mathbb{R}_{+}^{n\times 1}$, and the hyperparameters γ and σ^2 have been obtained based on the training data $(\mathbf{X},\mathbf{y})$. Additionally, the predictive distribution of a new observation $\tilde{y}$ can be obtained by integrating out $\tilde{f}$:

$$p(\tilde{y}|\tilde{\mathbf{x}},\mathbf{X},\gamma,\sigma^2) = \int p(\tilde{f}|\tilde{\mathbf{x}},\mathbf{X},\mathbf{y},\gamma,\sigma^2)p(\tilde{y}|\tilde{f},\sigma^2)d\tilde{f}, \tag{2.68}$$

thus,

$$p(\tilde{y}|\tilde{\mathbf{x}},\mathbf{X},\gamma,\sigma^2) \sim \mathscr{N}(m_\mathrm{p}(\tilde{\mathbf{x}}),k_\mathrm{p}(\tilde{\mathbf{x}},\tilde{\mathbf{x}}^{'}|\gamma)+\sigma^2), \tag{2.69}$$

where m_p and $k_\mathrm{p}(\tilde{\mathbf{x}},\tilde{\mathbf{x}}^{'}|\gamma)$ are given by eqns (2.66) and (2.67).

Mean functions GP: In certain scenarios a zero mean assumption may not be appropriate (e.g. when the behaviour of the model outside the data range becomes important, more details on this can be found in Chapter 6). Instead a mean function may be used, and the idea is that a GP with zero mean is placed on the difference between the observations and the mean function.

A mean function can take the form of a weighted sum of basis functions:

$$\mathbf{m}(\mathbf{x}) = \mathbf{h}(\mathbf{x})^\mathrm{T}\beta, \tag{2.70}$$

where $\mathbf{h}(\mathbf{x})$ are the basis functions for the input vector $\mathbf{x}$ and β are the weights. The model becomes:

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{h}(\mathbf{x})^\mathrm{T}\beta + g(\mathbf{x}), \\ g(\mathbf{x}) &\sim \mathscr{GP}(\mathbf{0},\mathbf{K}). \end{aligned} \tag{2.71}$$

By placing a Gaussian prior on the weights:

$$\beta \sim \mathscr{MVN}(\mathbf{b},\mathbf{B}), \tag{2.72}$$

they can be integrated out from the model to obtain a new GP prior:

$$\mathbf{f}(\mathbf{x}) \sim \mathscr{GP}(\mathbf{h}(\mathbf{x})^\mathrm{T}\mathbf{b}, k(\mathbf{x},\mathbf{x}') + \mathbf{h}(\mathbf{x})^\mathrm{T}\mathbf{B}\mathbf{h}(\mathbf{x}')). \tag{2.73}$$

The new predictive equations build upon the zero mean GP predictive equations (see eqns (2.66) and (2.67)) [157] as follows:

$$
\begin{aligned}
m^*_{\mathrm{p}}(\tilde{\mathbf{x}}) &= m_{\mathrm{p}}(\tilde{\mathbf{x}}) + \mathbf{R}^{\mathrm{T}}\bar{\beta}, && (2.74)\\
k^*_{\mathrm{p}}(\tilde{\mathbf{x}},\tilde{\mathbf{x}}^{'}|\gamma) &= k_{\mathrm{p}}(\tilde{\mathbf{x}},\tilde{\mathbf{x}}^{'}|\gamma) + \mathbf{R}^{\mathrm{T}}(\mathbf{B}^{-1} + \mathbf{H}(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{H}^{\mathrm{T}})^{-1}\mathbf{R}, && (2.75)
\end{aligned}
$$

where

$$
\begin{aligned}
\bar{\beta} &= (\mathbf{B}^{-1} + \mathbf{H}(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{H}^{\mathrm{T}})^{-1}(\mathbf{B}^{-1}\mathbf{b} + \mathbf{H}(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{y}), && (2.76)\\
\mathbf{R} &= \mathbf{h}(\tilde{\mathbf{x}}) - \mathbf{H}(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{k}(\mathbf{X},\tilde{\mathbf{x}}), && (2.77)\\
\mathbf{H}^{\mathrm{T}} &= \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \mathbf{h}(\mathbf{x}_2) \\ \vdots \\ \mathbf{h}(\mathbf{x}_n) \end{bmatrix}. && (2.78)
\end{aligned}
$$

The log marginal likelihood becomes:

$$
\log p(\mathbf{y}|\mathbf{X},\mathbf{b},\mathbf{B},\sigma^2) = -\frac{1}{2}\mathbf{L}^{\mathrm{T}}\mathbf{P}^{-1}\mathbf{L} - \frac{1}{2}\log\det(\mathbf{K}+\sigma^2\mathbf{I}) - \frac{1}{2}\log\det(\mathbf{B}) - \frac{1}{2}\log\det(\mathbf{Q}) - \frac{n}{2}\log 2\pi, \quad (2.79)
$$

$$
\mathbf{L} = \mathbf{H}^{\mathrm{T}}\mathbf{b} - \mathbf{y}, \quad \mathbf{P} = (\mathbf{K}+\sigma^2\mathbf{I}) + \mathbf{H}^{\mathrm{T}}\mathbf{B}\mathbf{H} \quad \mathbf{Q} = \mathbf{B}^{-1} + \mathbf{H}(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{H}^{\mathrm{T}}. \quad (2.80)
$$

GPs for classification

GPs can also be applied in classification problems, where the response is a binary variable, $\lambda = (\lambda_1, \ldots \lambda_n)$, with $\lambda_i \in \{0,1\}$, associated with inputs $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$, where 0: failure and 1: success (see Chapter 6 for an example). A GP classification model places a GP prior over the distribution of noiseless latent functions **f**, as follows:

$$
\begin{aligned}
\lambda_i|f(\mathbf{x}_i) &\sim \text{Bernoulli}(p(\lambda_i = 1|f(\mathbf{x}_i))),\\
\mathbf{f}(\mathbf{X})|\gamma &\sim \mathscr{GP}(\mathbf{m}(\mathbf{X}),\mathbf{K}|\gamma), && (2.81)\\
\gamma &\sim p(\gamma),
\end{aligned}
$$

where γ contains the kernel hyperparameters.

The binary observations (class labels) λ are drawn from a Bernoulli distribution with a success probability $p(\lambda_i = 1|f(\mathbf{x}_i))$, and likelihood shown in eq. (2.83). The success probability is related to the function $f(\mathbf{x}_i)$ via the sigmoid function, $\text{sig}(f(\mathbf{x}_i)) = (1+\exp(-f(\mathbf{x}_i)))^{-1}$, which transforms the probability into the unit interval [0,1], as shown in eq. (2.82):

$$p(\lambda_i = 1|f(\mathbf{x}_i)) = \mathrm{sig}(f(\mathbf{x}_i)); \quad p(\lambda_i = 0|f(\mathbf{x}_i)) = 1 - p(\lambda_i = 1|f(\mathbf{x}_i)), \tag{2.82}$$

$$\text{Likelihood}: p(\lambda_i|f(\mathbf{x}_i)) = (\mathrm{sig}(f(\mathbf{x}_i)))^{\lambda_i}(1 - \ \mathrm{sig}(f(\mathbf{x}_i)))^{(1-\lambda_i)} \text{ for a binary outcome } \lambda_i \in \{0,1\}. \tag{2.83}$$

The latent functions $\mathbf{f}$ can be integrated out from the conditional posterior distribution given the hyperparameters γ to obtain the marginal likelihood (eq. (2.84)), which has no closed-form solution:

$$p(\lambda|\mathbf{X},\gamma) = \int p(\lambda,\mathbf{f}|\mathbf{X},\gamma)d\mathbf{f} = \int p(\lambda|\mathbf{f})p(\mathbf{f}|\mathbf{X},\gamma)d\mathbf{f}. \tag{2.84}$$

With a non-Gaussian likelihood, MCMC [88] can be run, or the conditional posterior distribution can be approximated by a Gaussian form using the Laplace approximation [214], variational inference [134] or expectation propagation (EP) [105].

2.3.8 Model selection

This thesis also presents work on model discrimination between competing models - see Chapters 3 and 4. A more complex model is not necessarily favoured although it provides a better fit to the data, which can be quantified using the log predictive density. Penalising for parameter complexity (too many parameters) is a necessary adjustment to prevent from choosing an overparameterised model that overfits the existing data and is not generalisable to future data. Model selection can be performed based on various criteria, which are summarised below.

AICc and BIC

AIC(c) [1, 23] and BIC [172] can be used for model selection in a maximum likelihood framework. These criteria use the maximum log likelihood (or minimum residual sum-of-squares) as a measure of goodness of fit, and overfitting is avoided by applying a penalty based on the model complexity (i.e. number of model parameters, or effective number of parameters, where the latter may depend on the nature of the model or the data). AIC applies a lower penalty than BIC, thus the latter chooses simpler models (with a lower number of parameters). AICc is a modification of AIC [1] by adding a correction for finite sample sizes, hence applying a higher penalty. AIC(c) and BIC are defined as:

$$\begin{aligned}
\mathrm{AIC} &= -2\log(\mathbf{y}|\hat{\theta}_{\mathrm{MLE}}) + 2d, \\
\mathrm{AICc} &= -2\log(\mathbf{y}|\hat{\theta}_{\mathrm{MLE}}) + 2d + \frac{2d(d+1)}{n-d-1}, \\
\mathrm{BIC} &= -2\log(\mathbf{y}|\hat{\theta}_{\mathrm{MLE}}) + d\log n,
\end{aligned} \tag{2.85}$$

where $\log(\mathbf{y}|\hat{\theta}_{\text{MLE}})$ is the maximum log likelihood (i.e. log likelihood for the maximum likelihood estimate, MLE), d is the number of parameters in the model and n is the total number of observations.
The model which gives the lowest AIC(c) or BIC score is the favoured model.

DIC

DIC [185] is a partially Bayesian version of AIC, where the maximum likelihood estimate $\hat{\theta}_{\text{MLE}}$ is replaced by the posterior mean $\hat{\theta}_{\text{Bayes}} = \mathbb{E}(\theta|\mathbf{y})$ and the penalisation is data-driven. DIC is defined as

$$\text{DIC} = -2\log p(\mathbf{y}|\hat{\theta}_{\text{Bayes}}) + 2p_{\text{DIC}}, \tag{2.86}$$

$$p_{\text{DIC}} = 2\left(\log p(\mathbf{y}|\hat{\theta}_{\text{Bayes}}) - \mathbb{E}_{\theta|\mathbf{y}}(\log p(\mathbf{y}|\theta))\right), \tag{2.87}$$

where $\mathbb{E}_{\theta|\mathbf{y}}(\log(\mathbf{y}|\theta))$ is the expectation of the log posterior predictive density, which in practice is calculated by replacing the expectation with the average over the posterior draws [67], as follows:

$$\text{computed } p_{\text{DIC}} = 2\left(\log p(\mathbf{y}|\hat{\theta}_{\text{Bayes}}) - \frac{1}{S}\sum_{s=1}^{S}\log p(\mathbf{y}|\theta^s)\right), \tag{2.88}$$

where the sum is over the set of parameters $\theta^s, s = 1, \dots S$, that have been drawn approximately from the posterior distribution with MCMC (i.e. θ^s is the s^{th} MCMC sample).

The posterior mean of θ, $\mathbb{E}(\theta|\mathbf{y})$ can also produce the maximum log predictive density if it is also the mode of the posterior distribution. p_{DIC} reduces to d for linear models with uniform prior distributions [185], and $p(\mathbf{y}|\hat{\theta}_{\text{Bayes}}) = p(\mathbf{y}|\hat{\theta}_{\text{MLE}})$, making DIC equivalent to AIC [67].
The model which gives the lowest DIC score is the favoured model.

WAIC

WAIC [209] is fully Bayesian in the sense that it is calculated using the whole posterior distribution – see eq. (2.89). It takes the log posterior predictive density (first term) and adjusts for overfitting by adding a correction for the effective number of parameters (second term) [67],

$$\text{WAIC} = -2\sum_{i=1}^{n}\log\int p(y_i|\theta)p(\theta|\mathbf{y})d\theta + 2\sum_{i=1}^{n}\text{var}_{\theta|\mathbf{y}}(\log p(y_i|\theta)), \tag{2.89}$$

which in practice is computed as follows:

$$\text{computed } WAIC = -2\sum_{i=1}^{n}\log\left(\frac{1}{S}\sum_{s=1}^{S}p(y_i|\theta^s)\right) + 2\sum_{i=1}^{n}V_{s=1}^{S}(\log p(y_i|\theta^s)), \tag{2.90}$$

where $\theta^s, s = 1,...S$, are posterior samples from $p(\theta|\mathbf{y})$, and the following holds: $V_{s=1}^S a_s = \frac{1}{S-1}\sum_{s=1}^S (a_s - \bar{a})^2$.
The model which gives the lowest WAIC score is the favoured model.

AIC(c), DIC and WAIC are adjusted in-sample predictive accuracy measures. Given that they are calculated on the data on which the model is trained, the adjustment is necessary to avoid biased estimates of the predictive accuracy. However they can only provide unbiased results in expectation (i.e. scores based on the estimator of the discrepancy between the true and predictive data distribution are computed for a large number of data sets generated under the same conditions, and the average score is unbiased, see e.g. [28]); this is not guaranteed to hold for an individual specific data set. In the absence of multiple data sets, unbiased predictive accuracy of AIC(c), DIC and WAIC can only be obtained asymptotically, for number of observations, $n \to \infty$ in one single data set. Under some conditions, AIC and DIC have been shown to be asymptotically equal to leave one out cross-validation [179, 186], while WAIC is asymptotically equivalent to Bayesian leave-one-out cross-validation (LOO-CV) [209].

While AIC(c), DIC and WAIC are predictive measures, BIC (and WBIC [210]) are explanatory measures. BIC was constructed as a method of approximating the marginal probability density of the data $\mathbf{y}$ under a model M, $p(\mathbf{y}|M)$ to be used in calculating the posterior probability of the model M:

$$p(M|\mathbf{y}) \propto p(\mathbf{y}|M)p(M), \tag{2.91}$$

required for model comparison via Bayes factors in a discrete model comparison setting [204].

AIC and BIC tend to be unreliable for small sample data sets. DIC is not suitable for singular posterior models, including mixture or hierarchical models (i.e. models for which the Fisher information matrix plus the negative second-order derivative of the log prior is not strictly positive definite, hence is semi-positive definite, thus singular[2]) or multimodal models (e.g. for a bimodal posterior distribution the posterior mean would lie between the modes) [209]. WAIC has better small-sample behaviour than AIC and unlike DIC, works for singular models. Compared to AIC(c), BIC and DIC, WAIC averages over the posterior distribution instead of conditioning on a point estimate, thus the parameter uncertainty is naturally incorporated.

In this study AIC(c), BIC, DIC and WAIC criteria are used. The marginal likelihood method [13] performing model selection by comparing the posterior probabilities of every model given the data (see eq (2.91)) was not considered as it entails high computational complexity when numerically stable procedures are used, such as thermodynamic integration [62, 111], and numerical instability of computationally affordable schemes like the harmonic mean estimator [154]. However, the advantage of the marginal likelihood is that the method is exact, in the sense of not relying on the asymptotics. Moreover, WBIC was not used as it would require running

[2] In that case the matrix has zero eigenvalues, hence its determinant, given by the product of its eigenvalues, is zero, making the matrix singular.

additional MCMC simulations for an annealed likelihood, incurring additional computational overheads (see [210] for details). In addition, Bayesian LOO-CV was not used as it incurs high computational costs. Bayesian LOO-CV proceeds by removing one data point at a time from the data set of n points, running MCMC on the model with $n-1$ observations and calculating the log predictive density of the held-out data point conditional on the resulting posterior samples (the model with a higher LOO-CV is preferred):

$$\sum_{i=1}^{n} \log p(y_i|\mathbf{y}_{-i}) = \sum_{i=1}^{n} \log \int p(y_i|\theta)p(\theta|\mathbf{y}_{-i})d\theta = \sum_{i=1}^{n} \log\left(\frac{1}{S}\sum_{s=1}^{S} p(y_i|\theta^s)\right), \tag{2.92}$$

where the second equality shows how Bayesian LOO-CV is computed in practice, and θ^s is the s^{th} posterior sample drawn approximately from $p(\theta|\mathbf{y}_{-i})$. Bayesian LOO-CV does not rely on asymptotics, and by partitioning the data, it avoids overfitting, however, it is computed based on one data set, thus only providing unbiased results in expectation [59].

All of these measures help compare between models, however they do not assess the quality of the models, i.e. out of a set of candidate models, the best model chosen does not necessarily provide a faithful description to the data (model checking diagnostics can be employed for that purpose, e.g. based on residuals for frequentist methods, or posterior predictive checks for Bayesian approaches, see Chapter 6 in [66]).

Chapter 3

Uncertainty of haemodynamic parameters

This study performs inference of parameters describing the haemodynamic equations of a 1D fluid-dynamics model of the pulmonary blood circulation. The fluid-dynamics model takes selected parameter values and aims to mimic the behaviour of the pulmonary haemodynamics under normal physiological and pathological (hypoxia) conditions. The Delayed Rejection Adaptive Metropolis (DRAM) algorithm is jointly used with constraint non-linear optimisation to learn the parameter values and quantify the uncertainty in the parameter estimates. To accommodate for different magnitudes of the parameter values, an improved parameter scaling technique is introduced in the DRAM algorithm. In addition, model selection using different information criteria, including the Watanabe Akaike Information Criteria is performed to discriminate between a model including or excluding a vessel tapering factor. Differences in parameters and model predictions between the control and hypoxic mice are investigated. The results from the analysis assuming independent and identical (iid) measurement errors is presented in the first part of this chapter. In the second part, correlated measurement errors are incorporated into a model selection analysis to discriminate between a linear and non-linear wall model.

Note: This chapter is adapted based on two papers: one study by Paun et al. [141] (first part) and a study by Qureshi et al. [151] (second part), on which I am the third author. In the second part of the chapter I present those results which I obtained and for which I take full responsibility, for the full set of results, the reader is referred to the original paper [151].

3.1 Introduction

The cardiovascular circulation is composed of the systemic circulation and the pulmonary circulation. Extensive work has been done to model the systemic circulation [178], however pulmonary hypertension is one of the leading causes of right heart failure [136]. The current work focuses on predicting the observed haemodynamic behaviour in the pulmonary circulation under normal and pathological conditions (hypoxia). Hypoxia is a pathological condition in which the body tissues are not sufficiently well oxygenated, leading to pulmonary hypertension.

Pulmonary hypertension (PH) is characterised by high mean blood pressure in the lungs (above 25 mmHg in the main pulmonary artery). PH leads to vascular remodelling, including stiffening, thickening, constriction of the small and large pulmonary arteries and microvascular rarefaction (microvascular rarefaction is a pathological condition in which there are fewer capillaries per unit volume of body tissue [206]). A reliable predictive model for the pulmonary haemodynamics can assist clinicians in diagnosing and treating PH by offering additional information which is otherwise immeasurable, e.g. blood flow and pressure predictions beyond the location where it can be measured. In addition, it helps reduce the number of invasive procedures for the patients as, currently, pulmonary pressure is measured invasively via right-heart catheterisation [188]. The model is tested in the context of mice data, however the work can be extended to human data.

Inferring key parameters for disease diagnosis and treatment planning is an essential, yet challenging step in predicting the observed haemodynamics. One such parameter is the arterial stiffness, which is significantly higher for patients having PH. However, such parameters cannot be measured in-vivo, which creates the need for them to be learnt indirectly from the observed pulmonary blood flow and pressure. This poses several challenges due to a number of reasons: e.g. the non-linear partial differential equations (PDEs) by which the mathematical model is defined need to be numerically integrated a large number of times (tens of thousands of times), using a multi-scale and multi-components mathematical model results in a potentially large number of parameters being estimated, and the data are noisy and insufficient.

In this chapter statistical techniques for inference of haemodynamic parameters (vessel wall stiffness and boundary conditions parameters) of a 1D fluid-dynamics model of the pulmonary circulation are implemented to predict blood pressure and flow in a fixed arterial network coming from one single image segmentation, and zero vessel tapering, as well as non-zero tapering is assumed. A few restrictive assumptions about the parameters characterising the vessels' geometry or the boundary conditions in the PDEs allowed reducing the parameter dimension from a 55D problem to 5D (details offered in Section 3.2.3). Additionally, to combat the high computational costs of the inference procedure, non-linear constraint optimisation was first employed to find the maximum likelihood estimates. Subsequently, the uncertainty in these estimates was quantified by approximately sampling parameters from their posterior distribution using the DRAM algorithm and the Adaptive Metropolis (AM) algorithm using the Matlab MCMC toolbox [107]. Pseudocode is provided showing how to deal with different parameter magnitudes by incorporating a parameter scaling technique in the MCMC Matlab toolbox [107] (details in Section 3.3). The importance of incorporating the parameter scaling in the inference procedure is emphasized, and the consequences of not allowing for scaling are shown. This code is tested on real data coming from a healthy and a hypoxic mouse.

In addition, this study investigates whether vessel tapering is consistent with the measured data by performing model selection using different information criteria, including the Watanabe

Akaike Information Criteria [209].

The analysis was initially performed assuming independent and identically distributed errors, and the assumption was subsequently relaxed to incorporate the correlation of the measurement errors.

3.2 Materials and Methods

3.2.1 Experimental data

The experimental data are described in Section 2.1.

3.2.2 Mathematical model

The mathematical model used in this chapter is a linear wall model, as indicated in Table 2.1 and described in Section 2.2.

3.2.3 Parameterisation

The coupled model contains a total of 100 parameters used to determine pressure and flow in the network, and they are associated with the vessel fluid dynamics (24 parameters), the geometry (43 parameters), and the outflow boundary conditions (33 parameters). Of these, 45 parameters describing physical properties are fixed (e.g. vessel blood and viscosity which are measured thus known, or geometry radii and lengths extracted from the lung micro-CT image). The remaining 55 parameters characterise vessel stiffness (21 parameters - one per vessel), outflow boundary conditions (33 parameters, 3 parameters per terminal vessel), and a tapering factor.

To reduce the dimension of the parameter space, it was assumed that the vessel stiffness is constant throughout the network [103] (reducing the dimension by 20). Second, as described in [151] and in Section 2.2.4, the nominal (i.e. initial) outflow boundary condition were predicted for each vessel from estimates of total resistance and compliance and a constant scaling factor was used to adjust the terminal vessel-specific resistances and capacitances in the Windkessel models (reducing the dimension by 30). The reduction of the parameter space could have been performed with sensitivity analysis [36], which identifies the parameters that affect the output the most, and constitutes future work. To account for the physiologically observed tapering in large arteries, a tapering factor ξ was introduced via a linear relation between top and bottom vessel radii:

$$r_{\mathrm{B}} = r_{\mathrm{T}}(1 - 0.5\xi), \tag{3.1}$$

where r_{T} and r_{B} are the inlet and outlet radii of the vessels.

The above assumptions allowed reducing the dimension of the free parameters from 55 to 5. The parameters to be estimated included: the vessel stiffness s – see eq (2.5), where to impose constant stiffness, f_1 was set to 0 (i.e. parameter f_3 only was inferred), scaling factors r_1, r_2, c and the tapering factor ξ.

Two network configurations were analysed:

$$\begin{aligned} &\text{Network with straight vessels (4D model):} \quad && \theta = \{s, r_1, r_2, c\} \\ &\text{Network with tapered vessels (5D model):} \quad && \theta = \{s, r_1, r_2, c, \xi\} \end{aligned} \tag{3.2}$$

The univariate ranges for the θ parameter vectors were chosen to be biologically meaningful: $s \in [10^4, 10^6], r_1, r_2, c \in [-3, 2.5], \xi \in [0, 0.5]$. The range for s is in Torr units (not mmHg).

3.2.4 Statistical methods

Statistical model

The statistical model is defined as

$$\begin{aligned} y_i^q &= m^q(\mathbf{x}; \theta) + \varepsilon_i^q, \quad \varepsilon_i^q \sim \mathscr{N}(0, \sigma_{1i}^2), \\ y_i^p &= m^p(\mathbf{x}; \theta) + \varepsilon_i^p, \quad \varepsilon_i^p \sim \mathscr{N}(0, \sigma_{2i}^2), \end{aligned} \tag{3.3}$$

where

- $\mathbf{y}^q = [y_1^q, y_2^q, ..., y_n^q]$ are the noisy MPA-measured flow data (a time series of $n = 1024$ points), and $\mathbf{y}^p = [y_1^p, y_2^p, ... y_n^p]$ are the noisy MPA-measured pressure data,
- $\mathbf{m}^q(.)$ is an $n \times 1$ vector containing the MPA-predicted flow output from numerically solving the PDEs, and $m^p(.)$ is an $n \times 1$ vector containing the MPA-predicted pressure,
- θ is a d-dimensional parameter vector to be inferred from the haemodynamic data (see eq (3.2)).
- $\mathbf{x}$ denote other input variables (e.g. inflow into the MPA, or network properties such as vessel radius and length measurements coming from the image segmentation),
- ε_i are the measurement errors, which initially were assumed to be additive iid Gaussian. They have a different variance for flow and pressure due to different measurement techniques (i.e. flow is measured with ultra-sound, and pressure is measured with right-heart catheterisation).

In this study, the MPA-measured flow is used as inflow boundary condition for the PDEs, and the MPA-measured pressure is used for parameter inference.

Parameter estimation and uncertainty quantification

This study employs MCMC methods, in particular the DRAM algorithm [81] to sample parameters approximately from their posterior distribution; these methods are reviewed in Chapter 2. Since MCMC used for non-linear PDEs problems is too slow to converge when started from random points in parameter space (MCMC with emulation is explored in later chapters, see Chapter 6), point estimates were first obtained using the maximum likelihood approach, and these estimates were used as initial values in the DRAM algorithm.

Non-linear constraint optimisation was used to find the global or local optimum of the objective function, chosen to be the residual sum of squares (RSS), which measures the deviation of the simulated signal from the measured signal and is given by:

$$\mathscr{S}(\theta) = (\mathbf{y} - \mathbf{m}(\theta))^2 = \sum_{i=1}^{n} (y_i - m_i(\theta))^2, \tag{3.4}$$

where for simplicity the index $\mathbf{x}$ and the superscript p from $m(.)$ in eq (3.3) have been dropped.

Non-linear constraint optimisation was chosen since there was evidence of the RSS function being unimodal (see Section 3.5.1) and because the objective function is non-linear in the (bounded) parameters. The optimisation minimised RSS by solving the problem defined by eq (3.5)):

$$\begin{aligned} &\text{minimize} && \mathscr{S}(\theta), && \\ &\text{subject to} && g(\theta) = 0, && \text{equality constraints,} \\ & && h(\theta) \leq 0, && \text{inequality constraints.} \end{aligned} \tag{3.5}$$

where functions $g(\theta)$ and $h(\theta)$ define the equality and inequality relations between the parameters.

To run DRAM, the likelihood, prior, proposal and posterior distribution must be first defined.

Data likelihood:

$$y_i|\theta \sim \mathscr{N}(m_i(\theta), \sigma^2), \tag{3.6}$$

i.e.

$$p(\mathbf{y}|\theta, \sigma^2) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{\sum_{i=1}^n (y_i - m_i(\theta))^2}{2\sigma^2}\right) \tag{3.7}$$

If θ is outside the biological boundaries or the PDEs produce no output due to biologically unrealistic parameter combinations, RSS is assigned a large value, i.e. 10^{10}, leading to a zero value likelihood for the particular θ parameter vector.

Prior distribution:

$$\theta \sim \text{Truncated } \mathscr{MVN}(\mu, \mathbf{T}), \tag{3.8}$$

i.e.

$$p(\theta) = \det(2\pi\mathbf{T})^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\theta-\mu)'\mathbf{T}^{-1}(\theta-\mu)\right), \tag{3.9}$$

where $\mathbf{T}$ is a positive definite covariance matrix, chosen to be diagonal (i.e. independent priors) with high variance entries (10^{10}), making the prior non-informative between the biological ranges due to lack of prior knowledge,

$$\sigma^2 \sim \mathscr{I}\mathscr{G}(a,b), \tag{3.10}$$

i.e.

$$p(\sigma^2) = \frac{b^a}{\Gamma(a)}(\sigma^2)^{-a-1} \exp\left(-\frac{b}{\sigma^2}\right), \tag{3.11}$$

where $\mathscr{I}\mathscr{G}$ stands for the Inverse-Gamma distribution. The default settings in the DRAM MCMC toolbox [107] were used, i.e. shape and scale parameters: $a = \frac{n_s}{2} > 0$, $b = \frac{n_s\gamma_s^2}{2} > 0$, with γ_s^2 being the prior value for σ^2 and n_s the prior accuracy for γ_s^2 – a large value of n_s means high certainty in the prior, i.e. the prior dominates, and so the posterior samples are close to γ_s^2. This prior was chosen for conjugacy reasons (i.e. for a Gaussian likelihood and Inverse-Gamma prior, the conditional posterior distribution is also Inverse-Gamma), allowing to draw posterior samples from its full conditional distribution in a Gibbs step. A weakly informative prior for σ^2 was assumed by setting $n_s = 1$, and γ_s^2 was taken to be the initial noise variance σ^2 chosen by the user (the defaults in the toolbox).

Proposal distribution:

$$q(\theta^*,\theta^{k-1}) = \text{Truncated } \mathscr{M}\mathscr{V}\mathscr{N}(\theta^{k-1},\mathbf{V}), \tag{3.12}$$

where the proposal covariance matrix $\mathbf{V}$ is obtained as explained in the paragraph discussing DRAM in Section 2.3.2. For θ^* inside boundaries:

$$q(\theta^*,\theta^{k-1}) = \det(2\pi\mathbf{V})^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\theta^*-\theta^{k-1})'\mathbf{V}^{-1}(\theta^*-\theta^{k-1})\right), \tag{3.13}$$

Posterior distribution:

$$p(\theta,\sigma^2|\mathbf{y}) \propto p(\mathbf{y}|\theta,\sigma^2)p(\theta)p(\sigma^2), \tag{3.14}$$

which yields:

$$\begin{aligned} p(\theta,\sigma^2|\mathbf{y}) \propto (\sigma^2)^{-\frac{n}{2}-a-1} \exp\left(-\frac{0.5\sum_{i=1}^n (y_i - m_i(\theta))^2 + b}{\sigma^2}\right) \times \\ \exp\left(-0.5(\theta-\mu)'\mathbf{T}^{-1}(\theta-\mu)\right). \end{aligned} \tag{3.15}$$

DRAM was used to sample the parameters θ in a Metropolis-Hastings step since the marginal posterior distribution for θ is not available in closed form. The θ parameters were then fixed to the sample value drawn and the noise variance σ^2 was sampled from an Inverse-Gamma distribution in a Gibbs step:

$$\sigma^2|\mathbf{y},\theta \sim \mathscr{IG}\left(0.5n+a, 0.5\sum_{i=1}^{n}(y_i - m_i(\theta))^2 + b\right). \tag{3.16}$$

Hence, when going from the prior to the posterior for σ^2, the distribution is modified to include the contribution from the data $0.5\sum_{i=1}^{n}(y_i - m_i(\theta))^2$, and the accuracy increases when data become available ($0.5n$ is added to the prior shape parameter - the more data points, the higher the accuracy).

3.3 Methodological contribution

This work builds on and further improves on existing literature by allowing for a novel parameter scaling technique in the DRAM algorithm with informative priors for the parameters.

A pseudocode of the DRAM algorithm [81, 107] can be found on pages 175–176 of the book by Smith [183]. In practice, the algorithm is implemented by making use of the Matlab DRAM toolbox [107]. However, the toolbox does not offer the option to scale parameters if they differ by orders of magnitude, and so the code had to be appropriately modified. The pseudocode for DRAM with an informative prior, together with the parameter scaling adjustment is novel and can be found in Algorithms 1a-1d. Algorithms 8.8 and 8.10 in [183] were therefore modified to allow using an informative prior and the parameter scaling was improved. The improvement concerns the way that the acceptance probabilities are calculated. More precisely, the jump densities should be calculated for the transformed parameters (see step 12 in Algorithm 1d). Using the original parameters can cause the ratio of the jump densities become very large if one or more parameters have high magnitude values. Occasionally, in the second DR try, the proposed value for the high magnitude parameter may be much smaller than the current value. Hence, the numerator becomes much larger than the denominator in the jump ratio in eq (2.25), which encourages wrongly accepting the proposals. The jump ratio is

$$\frac{q_1(\theta^*|\theta^{*(2)})q_2(\theta^{k-1}|\theta^*,\theta^{*(2)})}{q_1(\theta^*|\theta^{k-1})q_2(\theta^{*(2)}|\theta^*,\theta^{k-1})}. \tag{3.17}$$

In simplifying terms, for a symmetric Gaussian proposal density, the log jump ratio is proportional to:

$$-\left(\theta^* - \theta^{*(2)}\right)^2 - \left(\theta^{k-1} - \theta^{*(2)}\right)^2 + \left(\theta^* - \theta^{k-1}\right)^2 + \left(\theta^{*(2)} - \theta^{k-1}\right)^2 \tag{3.18}$$

$$= -\left(\theta^{*(2)} - \theta^*\right)^2 + \left(\theta^{k-1} - \theta^*\right)^2, \tag{3.19}$$

hence if $\theta^{*(2)} << \theta^{k-1}$, the jump ratio in the acceptance probability in eq (2.25) is dominated by a large magnitude value of θ^{k-1}, thus the ratio becomes very large, encouraging to falsely accept $\theta^{*(2)}$. This however does not represent a problem for the first stage acceptance probabilities when a symmetric proposal distribution is used, as the proposals cancel out in the ratio. Hence, the correction only affects the DRAM and DR algorithms, while no correction is needed for the AM algorithm.

Algorithm 1a Delayed Rejection Adaptive Metropolis Algorithm (DRAM) with informative prior (fixed hyperparameters)

```
Initialise design parameters: M: number of chain iterations, $\beta$: scaling factor for proposal covariance matrix in DR, $n_{\mathrm{dr}}$: number of DR tries, $t_{\mathrm{ad}}$: adaptation interval, $b_{\mathrm{s}}$: burn-in scale, $b_{\mathrm{t}}$: burn-in time, $\gamma_{\mathrm{s}}^2$: prior for $\sigma^2$ (error variance), $n_{\mathrm{s}}$: prior accuracy for $\gamma_{\mathrm{s}}^2$
Compute $\theta^0 = \arg\min_\theta \sum_{i=1}^n [y_i - m_i(\theta)]^2$, where $a_j \leq \theta_j \leq b_j, j = 1...d$, $d$: number of parameters, using non-linear constraint optimisation with 20 overdispersed starting values generated from a Sobol sequence
Initialise $\sigma_0^2 = \frac{\mathscr{S}_0}{n}$, where $\mathscr{S}_0$: optimised residual-of-squares value, $n$: number of observations
Assume that the prior distribution for $\theta$ is a Multivariate Normal distribution (independent priors $\theta$s), $\theta \sim \mathscr{MVN}(\mu, \mathbf{T})$ (i.e. $\mathbf{T}$ diagonal)
Set initial $\mathscr{S}_{\mathrm{prior}}^0 = \sum_{j=1}^d \frac{(\theta_j^0 - \mu_j)^2}{t_j^2}$, where $t_j$: $j^{\mathrm{th}}$ diagonal term in matrix $\mathbf{T}$
Either set the initial covariance matrix $\mathbf{V}_0$ to be the Hessian matrix from optimisation or set $\mathbf{V}_0$ via monitoring of acceptance rate (i.e. decrease variance if acceptance rate too low, e.g. below 5%, and increase variance if acceptance rate too high, e.g. above 95%)
Set $\mathbf{R}_0$ = chol($\mathbf{V}_0$), where "chol" stands for the Choleski factorisation, as an efficient numerical solution of $\mathbf{V}_0 = \mathbf{R}_0^{\mathrm{T}}\mathbf{R}_0$
for k = 1, ... M do
    Sample $\mathbf{u}_k \sim \mathscr{MVN}(\mathbf{0}, \mathbf{I})$, where $\mathbf{I}$: identity matrix
    Construct candidate from Gaussian proposal: $\theta^* = \theta^{k-1} + \mathbf{R}_{k-1} u_k$
    if $a_j \leq \theta_j \leq b_j$ for any $j$ then                ▷ outside boundaries
        Jump to DR Algorithm 1c for iteration $k$
    else
        Calculate $\mathscr{S}_{\theta^*} = \sum_{i=1}^n [y_i - m_i(\theta^*)]^2, \quad \mathscr{S}_{\mathrm{pri}}^* = \sum_{j=1}^d \frac{(\theta^*{}_j - \mu_j)^2}{t_j^2}$
    end if
    Calculate the acceptance probability,

        $\alpha_1(\theta^* | \theta^{k-1}) = \min\left(1, \exp\left[-0.5\left(\frac{\mathscr{S}_{\theta^*} - \mathscr{S}_{\theta^{k-1}}}{\sigma_{k-1}^2} + \mathscr{S}_{\mathrm{pri}}^* - \mathscr{S}_{\mathrm{pri}}^{k-1}\right)\right]\right)$

    Sample $u_{\alpha_1} \sim \mathscr{U}(0,1)$
    if $u_{\alpha_1} < \alpha_1$ then
        $\theta^k = \theta^*, \quad \mathscr{S}_{\theta^k} = \mathscr{S}_{\theta^*}, \quad \mathscr{S}_{\mathrm{pri}}^k = \mathscr{S}_{\mathrm{pri}}^*$
    else
        Enter DR Algorithm 1c for iteration $k$
    end if
    Update $\sigma_k^2 | \theta^k \sim \mathscr{IG}\left(\frac{n_{\mathrm{s}} + n}{2}, \frac{n_{\mathrm{s}}\gamma_{\mathrm{s}}^2 + \mathscr{S}_{\theta^k}}{2}\right)$
    Enter AM Algorithm 1b for iteration $k$
end for
```

Algorithm 1b Adaptive Metropolis Algorithm (AM)

```
if $k < b_\mathrm{t}$ then                    ▷ during burn-in time, no adaptation, just scaling $\mathbf{R}$
    $\mathbf{R}_k \leftarrow \mathbf{R}_{k-1}/b_\mathrm{s}$,                ▷ decrease $\mathbf{R}$ if rejection rate $> 0.95$
    $\mathbf{R}_k \leftarrow b_\mathrm{s}\mathbf{R}_{k-1}$,                ▷ increase $\mathbf{R}$ if rejection rate $< 0.05$
else
    if $\mathrm{mod}(k, t_\mathrm{ad}) = 1$ then
        Set $\mathbf{V}_k = s_d \mathrm{Cov}(\theta^0, \theta^1, ...\theta^k) + \delta \mathbf{I}_d$ and use eq (2.27), $s_d = 2.38^2/d$, $\delta = 10^{-9}$, $\mathbf{R}_k = \mathrm{chol}(\mathbf{V}_k)$
    else
        $\mathbf{V}_k = \mathbf{V}_{k-1}$, $\mathbf{R}_k = \mathbf{R}_{k-1}$
    end if
end if
Output $\mathbf{R}_k$
```

Algorithm 1c Delayed Rejection Algorithm (DR) for $n_\mathrm{dr} = 2$, i.e. for 2nd stage proposals

```
Sample $\mathbf{u}_k \sim \mathscr{MVN}(\mathbf{0}, \mathbf{I})$
Construct 2nd stage candidate from Gaussian proposal: $\theta^{*(2)} = \theta^{k-1} + \beta \mathbf{R}_{k-1} u_k$
if $a_j \leq \theta_j \leq b_j$ for any j then
    $\theta^k = \theta^{k-1}$, $\mathscr{S}_{\theta^k} = \mathscr{S}_{\theta^{k-1}}$, $\mathscr{S}^k_\mathrm{pri} = \mathscr{S}^{k-1}_\mathrm{pri}$
    and jump to AM Algorithm 1b
else
    $\mathscr{S}_{\theta^{*(2)}} = \sum_{i=1}^n [y_i - m_i(\theta^{*(2)})]^2$, $\mathscr{S}^{*(2)}_\mathrm{pri} = \sum_{j=1}^d \frac{(\theta^{*(2)}_j - \mu_j)^2}{t_j^2}$
end if
Calculate the acceptance probability,
    $$\alpha_2(\theta^{*(2)}|\theta^*, \theta^{k-1}) = \min\left(1, \frac{p(\theta^{*(2)}|\mathbf{y}) q_1(\theta^*|\theta^{*(2)}) q_2(\theta^{k-1}|\theta^*, \theta^{*(2)})[1 - \alpha_1(\theta^*|\theta^{*(2)})]}{p(\theta^{k-1}|\mathbf{y}) q_1(\theta^*|\theta^{k-1}) q_2(\theta^{*(2)}|\theta^*, \theta^{k-1})[1 - \alpha_1(\theta^*|\theta^{k-1})]}\right)$$
Sample $u_{\alpha_2} \sim \mathscr{U}(0,1)$
if $u_{\alpha_2} < \alpha_2$ then,
    $\theta^k = \theta^{*(2)}$, $\mathscr{S}_{\theta^k} = \mathscr{S}_{\theta^{*(2)}}$, $\mathscr{S}^k_\mathrm{pri} = \mathscr{S}^{*(2)}_\mathrm{pri}$
else
    $\theta^k = \theta^{k-1}$, $\mathscr{S}_{\theta^k} = \mathscr{S}_{\theta^{k-1}}$, $\mathscr{S}^k_\mathrm{pri} = \mathscr{S}^{k-1}_\mathrm{pri}$
end if
```

Algorithm 1d Delayed Rejection Adaptive Metropolis (DRAM) with scaled parameters

```
Denote $\boldsymbol{\theta}_s = \boldsymbol{\theta}./\mathbf{s}$, where $\mathbf{s}$ are the scaling factors
In Algorithm 1a:
DRAM 2: Compute $\boldsymbol{\theta}^0 = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} [y_i - m_i(\boldsymbol{\theta}. \times \mathbf{s})]^2$
DRAM 10: Construct candidate from Gaussian proposal:
        $$\boldsymbol{\theta}^*{}_s = \boldsymbol{\theta}_s^{k-1} + \mathbf{R}_{k-1} u_k$$
    and set
        $$\boldsymbol{\theta}^* = \boldsymbol{\theta}^*{}_s. \times \mathbf{s}$$
DRAM 14: Set
        $$\mathscr{S}_{\text{pri}}^* = \sum_{j=1}^{d} \frac{(\theta^*{}_j - \mu_j)^2}{t_j^2} + \log(\det(\mathbf{J}))$$
    where $\mathbf{J}$ is the Jacobian matrix of the parameter transformation. If the parameters are assumed independent and $\mathbf{s}$ is a vector of constants, then $\log(\det(\mathbf{J})) = \sum_{j=1}^{d} \log(s_j)$
DRAM 16: Calculate the acceptance probability in the scaled parameter space $\alpha_1(\boldsymbol{\theta}^*{}_s | \boldsymbol{\theta}_s^{k-1})$
**Note**: if a non-informative prior is used and $\mathbf{s}$ is a vector of constants, then only the likelihood contribution is left in the acceptance probability, i.e. $\min\left(1, \exp\left(-0.5\left[\frac{\mathscr{S}_{\boldsymbol{\theta}^*{}_s} - \mathscr{S}_{\boldsymbol{\theta}_s^{k-1}}}{\sigma_{k-1}^2}\right]\right)\right)$. In the likelihood calculations, the unscaled parameters are always used, whereas the prior and proposal are calculated in transformed parameter space.
DRAM 19: In addition, set $\boldsymbol{\theta}_s^k = \boldsymbol{\theta}^*{}_s$
In Algorithm 1c:
DR 2: Construct 2nd stage candidate from Gaussian proposal:
        $$\boldsymbol{\theta}^*{}_s^{(2)} = \boldsymbol{\theta}_s^{k-1} + \beta \mathbf{R}_{k-1} u_k,$$
    and set
        $$\boldsymbol{\theta}^{*(2)} = \boldsymbol{\theta}^*{}_s^{(2)}. \times \mathbf{s}$$
DR 7: Set
        $$\mathscr{S}_{\text{pri}}^{*(2)} = \sum_{j=1}^{d} \frac{(\theta_j^{(2)} - \mu_j)^2}{t_j^2} + \log(\det(\mathbf{J}))$$
DR 9: Calculate the acceptance probability in the scaled parameter space, $\alpha_2(\boldsymbol{\theta}^*{}_s^{(2)} | \boldsymbol{\theta}^*{}_s, \boldsymbol{\theta}_s^{k-1})$.
DR 12 or DR 14: In addition, set $\boldsymbol{\theta}_s^k = \boldsymbol{\theta}^*{}_s^{(2)}$ or $\boldsymbol{\theta}_s^k = \boldsymbol{\theta}_s^{k-1}$
```

Note: If a non-informative prior is used, then $S_{\text{pri}} = 0$ in Algorithms 1a, 1c and 1d.

3.4 Simulations

3.4.1 Software

The statistical methods were implemented in *Matlab* (Mathworks, Natick, MA) and simulations were run on a RedHat Enterprise Linux 6 machine with Intel(R) Xeon(R) CPU E5-2680 v2 2.80GHz and 32GB RAM. The DRAM algorithm and the convergence diagnostics made use of functions from the Matlab MCMC toolbox [107]. The PDEs of the 1D fluid-dynamics model in Section 2.2 were numerically integrated using a two step Lax-Wendroff scheme [113] implemented in C++ by Olufsen et. al. [139, 151]. The two-step Lax-Wendroff scheme used is second order accurate in space and time. To ensure an appropriate spatial and temporal resolution was used, several discretizations in space and time were tested, ensuring that the Courant-Fredrich-Levy (CFL) condition

$$\frac{\Delta t}{\Delta x} \leq |u \pm c|^{-1} \tag{3.20}$$

where u is the fluid velocity and c is the wave speed, which is dependent on the wall model used (see [152] for details). The analysis revealed that $\Delta x = 0.025$ (mm) and $\Delta t = 1.34 \times 10^{-5}$ (s) provide sufficient resolution. In addition, the PDE system was solved over multiple periods until a converged, steady-state solution was reached (pressure difference between cycles was less than 0.001 mmHg). A typical model evaluation requires 20s elapsed time on the hardware mentioned above.

3.4.2 Method implementation details

Optimisation

To perform non-linear constraint optimisation, the sequential quadratic programming (SQP) algorithm [11] was used, and was iterated until it satisfied the convergence criterion $||\theta^k - \theta^{k-1}||< 10^{-11}$. The SQP algorithm was started from 20 different parameter vector values, which were uniformly drawn from a Sobol sequence to ensure a good coverage of the multidimensional parameter space [16].

AM

The length of the adaptation interval for the AM algorithm was chosen to ensure good mixing, leading to optimal acceptance rate, e.g. 23%, see Chapter 12 in Gelman's book [66]. The value used was 1000. This value produced better mixing and higher acceptance rate by up to 5% than AM run with other adaptation values, e.g. 100 or 500.

DR

Two tries were attempted for the DR algorithm, i.e. if the first stage proposal was rejected, a second proposal was made. Two attempts were enough to double the acceptance rate (from 20% to 40%) at the cost of an extra PDE evaluation at every MCMC iteration.

3.4.3 Parameter transformation

To improve numerical stability and reduce round-off errors, the original parameters θ_i were scaled in the order of one when used in the SQP and DRAM algorithms: $\theta_i \in [l_i, u_i] \rightarrow \frac{\theta_i}{s_i} \in [-1,1]$, where s_i is a scaling factor which ensures $\frac{\theta_i}{s_i} \in [-1,1]$. The transformed parameters were then mapped back via the inverse transformation into the original domain $\theta_i \in [l_i, u_i]$ before being inserted into the PDEs.

3.5 Results

This analysis was performed on two mice, a control and a hypoxic mouse. For each mouse, two models were compared: a 5D and a 4D model (see eq (3.2)), including or excluding the tapering factor.

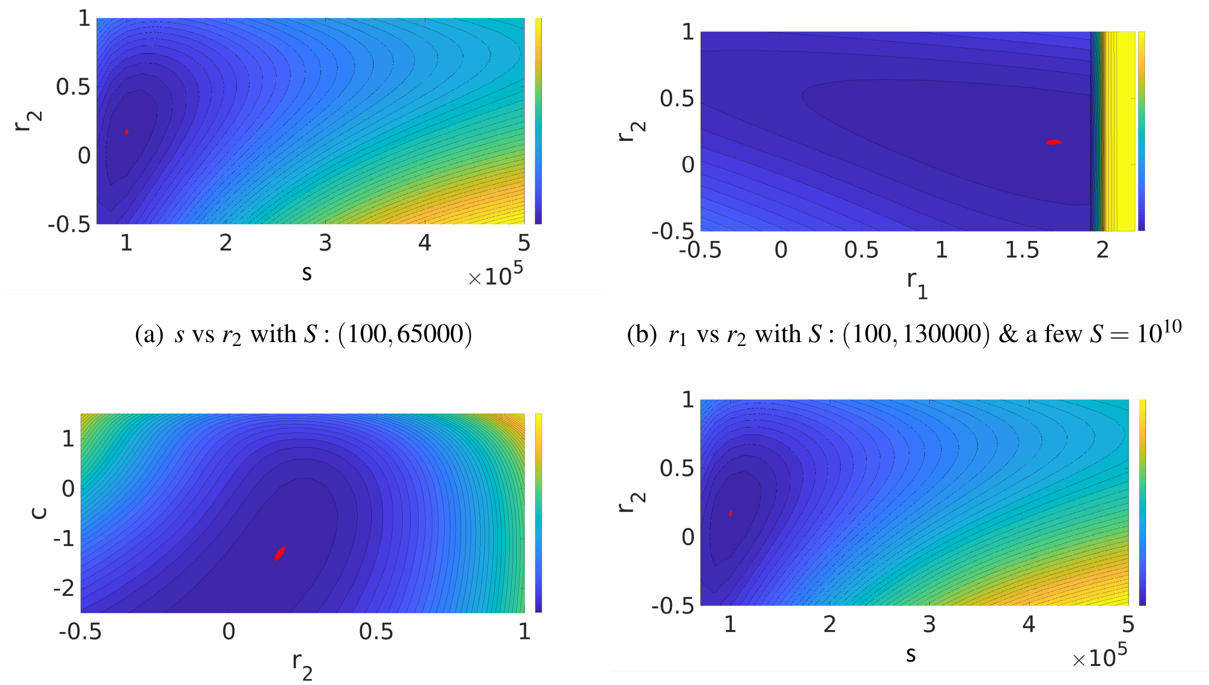
3.5.1 Exploration results

The analysis was started by an exploration of the objective function, the residual sum-of-squares (see eq (3.4)), calculated for the MPA-measured pressure. Figures 3.1 and 3.2 illustrate RSS in 2D, i.e. two parameters were varied at a time while the other parameters were kept fixed, for the 4D and 5D models for the control mouse. RSS exhibits unimodality and it is highly skewed in 2D (similar behaviour is obtained for the hypoxic mouse, plots not shown here). In some of the figures, there is a sudden jump to a region with RSS values of 10^{10} (yellow patch). This region marks the parameter domain that produces no PDE output because the physical assumptions of the mathematical model were violated.

3.5.2 Optimisation results

The unimodality in RSS motivates applying the non-linear constraint optimization to find the set of parameters that minimise RSS. Results indicate that regardless of the starting value, the optimisation algorithm reaches the same parameter values, suggesting that RSS is unimodal in 4D or 5D. The algorithm converges to the parameter estimates summarised in Table 3.1.

Table 3.1 shows that for both 4D and 5D models, the hypoxic mouse has a larger vessel stiffness, the resistance adjustment factors are larger and the compliance adjustment factor is



(a) s vs r_2 with $S:(100,65000)$

(b) r_1 vs r_2 with $S:(100,130000)$ & a few $S=10^{10}$

(c) r_2 vs c with $S:(100,23000)$

(d) s vs r_2 with $S:(100,130000)$

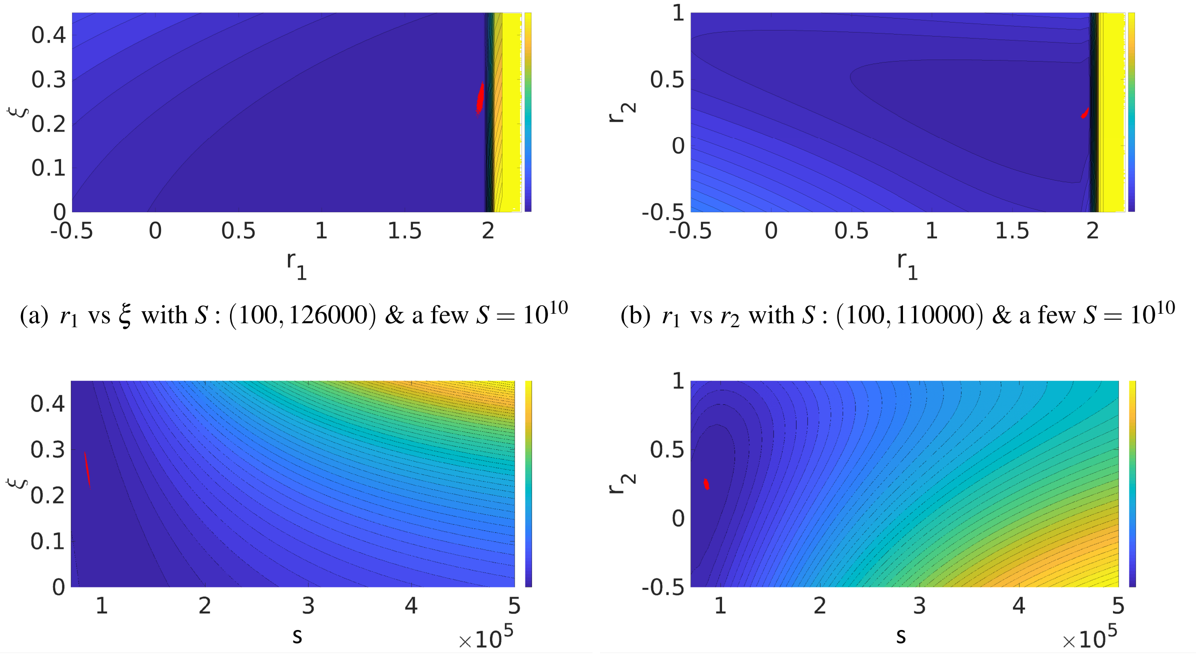
Figure 3.1: Contour plots of the objective function (residual sum-of-squares) in 2D, decreasing in the direction towards the inner dark blue slices for the 4D model corresponding to a control mouse. The yellow patches in the (b) panel figure show a region with RSS values of 10^{10}. The red trajectories in the innermost slice indicate parameter samples from the posterior distribution obtained using the Delayed Rejection Adaptive Metropolis algorithm. For graphical visibility, the innermost slice was not further resolved with additional contour lines. Here s is the vessel stiffness parameter (expressed in g/cm/s^2 units) and r_1, r_2, c are the Windkessel adjustment parameters (dimensionless). Figure adapted from our study in [141].

Mouse type	Model type	s	r_1	r_2	c	ξ	RSS
Control	4D	100398	1.69	0.17	-1.30	0	126.5
Hypoxic	4D	243645	0.52	0.13	-0.24	0	80.3
Control	5D	85529	1.96	0.25	-2.03	0.26	104.5
Hypoxic	5D	228383	0.59	0.14	-0.22	0.09	80

Table 3.1: Optimisation results for the 4D and 5D models (see eq (3.2)) for a control and hypoxic mouse. Legend: s: vessel stiffness (expressed in g/cm/s^2 units), r_1, r_2, c: resistances and compliance adjustments (dimensionless), ξ: vessel tapering factor (dimensionless), RSS: residual sum-of-squares.

smaller for the control mouse compared to the hypoxic, implying that the control mouse has a smaller resistance and higher compliance than the hypoxic mouse (see eq (2.13)). In addition, the quantitative fit to the control mouse data is better when the tapering factor is included (since RSS is smaller for the 5D model than for the 4D model), and the quantitative fits for the two models are very similar for the hypoxic mouse.

Next the parameter estimates in Table 3.1 were used to predict the pressure signal which is



(a) r_1 vs ξ with $S:(100,126000)$ & a few $S=10^{10}$

(b) r_1 vs r_2 with $S:(100,110000)$ & a few $S=10^{10}$

(c) s vs ξ with $S:(100,165000)$

(d) s vs r_2 with $S:(100,115000)$

Figure 3.2: Contour plots of the objective function (residual sum-of-squares) in 2D, decreasing in the direction towards the inner dark blue slices for the 5D model corresponding to a control mouse. The yellow patches in the top two figures (panels (a) and (b)) show a region with RSS values of 10^{10}. The red trajectories in the innermost slice indicate parameter samples from the posterior distribution obtained using the Delayed Rejection Adaptive Metropolis algorithm. For graphical visibility, the innermost slice was not further resolved with additional contour lines. Here s is the vessel stiffness parameter (expressed in g/cm/s^2 units) and r_1, r_2, c are the Windkessel adjustment parameters (dimensionless), and ξ: vessel tapering factor (dimensionless). Figure adapted from our study in [141].

compared to the measured pressure for both mice and models, see Figure 3.3. When comparing between the 4D and 5D models, qualitatively the signals look very similar. However, when comparing between mice, the overall model prediction appears to be better for the hypoxic mouse. In the case of the healthy mouse, the simulated pressure closely follows the measured pressure except near the peak, where an offset is registered. In addition, the systolic and diastolic pressures are larger for the hypoxic mouse than the control mouse, and the diastolic pressure decay is steeper for the hypoxic mouse, a consequence of higher vessel stiffness.

3.5.3 MCMC results

The uncertainty in the parameter estimates was quantified using the DRAM, AM and DR algorithms.

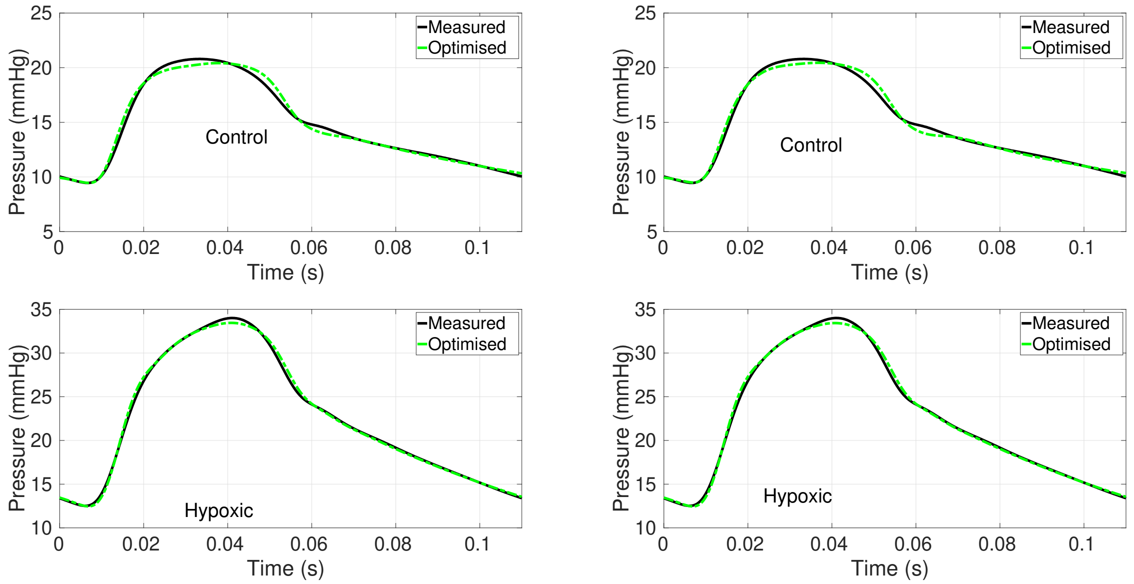


Figure 3.3: Comparison between measured pressure (solid black line) and pressure obtained from optimisation (dashed green line) for the 4D model (left column) and 5D model (right column) for a control and a hypoxic mouse. Figure taken from our study in [141].

AM

Table 3.2 summarises the AM results for every model and mouse type via the posterior mean and standard deviation calculated from the posterior samples. Similar observations as for the optimisation estimates can be made. Additionally, Figure 3.4 shows the uncertainty in the form of traceplots of posterior parameter samples obtained with the AM algorithm for the control mouse for the 5D model. The chains fluctuate around the optimised parameter values, which were the starting values, suggesting that the chains have reached the high posterior probability regions, and that the optimisation might have found the global optimum. A similar behaviour was registered for the 4D model, plots not shown here. There are no signs of multimodality in the objective function, which tallies with the optimisation results. Convergence appears to have been reached after roughly 13,000 iterations. The chain for RSS is steady around the initial optimised RSS value and the chain for σ^2 does not indicate non-convergence.

Additionally, the pairwise posterior correlations in Figure 3.5 indicate that the stiffness is (strongly) negatively correlated to the tapering factor, i.e. as the wall stiffness increases, the vessels become less tapered (top and bottom vessel radii become similar). Another example is the high correlation between the vessel stiffness and the compliance adjustment factor, i.e. as the stiffness increases, the compliance adjustment factor increases, hence the vessel compliance decreases (see eq (2.13)).

Figure 3.6 displays the traceplots for the 5D model parameter posterior samples for the hypoxic mouse. A pronounced skewness in the marginal distributions of the stiffness and tapering factor can be noticed, and they are strongly correlated. The biological range for the tapering

Mouse type	Model type	s	r_1	r_2	c	ξ	σ^2
Control	4D	100374 (391)	1.69 (0.01)	0.17 (0.005)	-1.30 (0.04)	0	0.12 (0.005)
Hypoxic	4D	243653 (1180)	0.52 (0.01)	0.13 (0.003)	-0.24 (0.01)	0	0.08 (0.003)
Control	5D	85471 (658)	1.96 (0.006)	0.25 (0.01)	-2.03 (0.07)	0.26 (0.01)	0.10 (0.004)
Hypoxic	5D	231627 (3720)	0.57 (0.02)	0.14 (0.003)	-0.23 (0.01)	0.07 (0.02)	0.08 (0.003)

Table 3.2: Posterior mean and standard deviation in brackets obtained from parameter posterior samples drawn with the Adaptive Metropolis algorithm and noise variance drawn with Gibbs algorithm for every mouse and model type. Legend: s: vessel stiffness (expressed in g/cm/s^2 units), r_1, r_2, c: Windkessel resistances and compliance factors (dimensionless), ξ: vessel tapering factor (dimensionless), σ^2: noise variance.

factor is overly conservative. The posterior mean is different from the RSS-based optimum estimates (the two values are similar for the 4D model, plots not shown).

DR

Figure 3.7 shows 10,000 iterations of a chain produced using the DR algorithm for the control mouse, 5D model. Despite the acceptance rate which is close to the optimal target acceptance rate given in literature (23% [66]), the plots reveal non-convergence of the chains, and the parameter space is explored slowly, leading to bad mixing, as judged from visual inspection. DR seems to be affected by the strong correlations between the parameters. Therefore, the chains would have to be run for a longer time, implying that the global adaption from the AM algorithm is essential in reaching convergence in a reasonable time frame.

DRAM with parameter scaling

Next the DRAM algorithm was implemented for every mouse type and model type. The posterior means and standard deviations for samples drawn with DRAM are extremely similar to those obtained using the AM algorithm, so they are not shown here. This emphasizes that the sampling algorithm is not important so long as the sampler has converged and is sampling (approximately) from the posterior distribution; the algorithms used can make a difference from an efficiency point of view, but not from an accuracy point of view. For example, Figure 3.8 displays Markov chains obtained with DRAM for the control mouse, model 5D. The visual inspection indicates good mixing of the chains around the optimum parameter values (formal tests. e.g. Geweke test [70] investigating mixing are presented in Section 3.5.4). The DRAM algorithm has a higher acceptance rate compared to the AM algorithm (the rate has increased

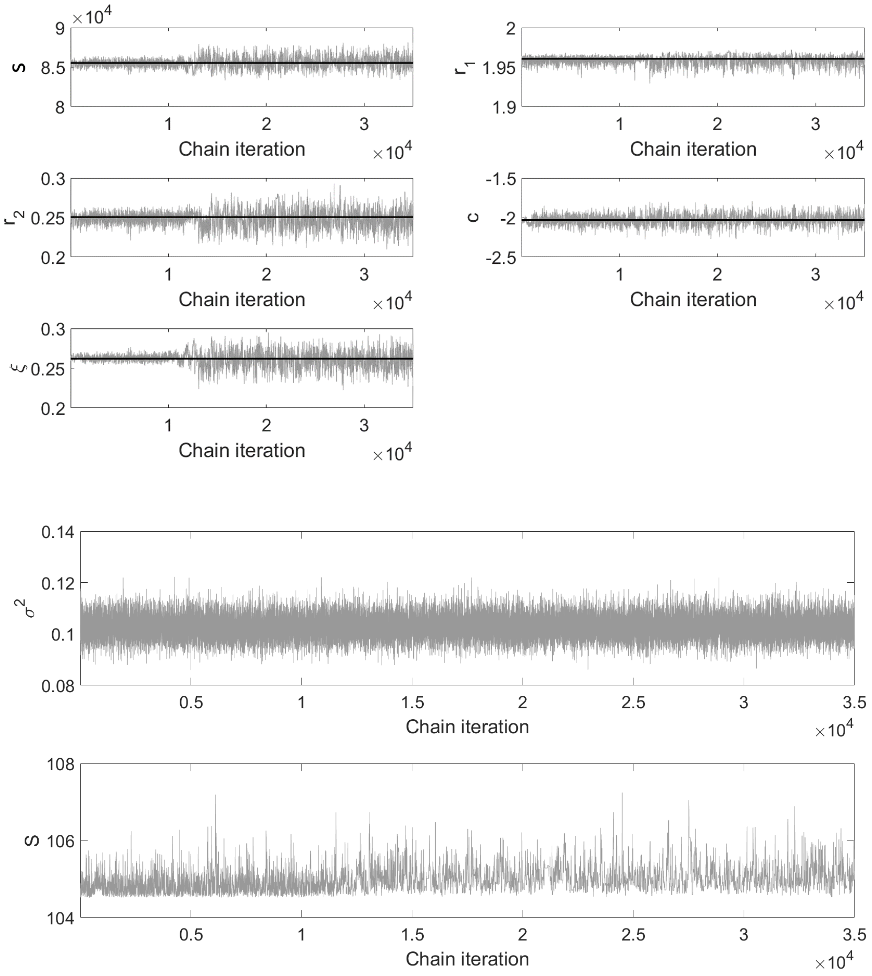


Figure 3.4: Markov chains traceplots for the parameters, residual sum-of-squares and noise variance obtained using the Adaptive Metropolis algorithm for the 5D model corresponding to a control mouse. Starting values for the algorithm were the optimised values, superimposed in black horizontal lines. Acceptance rate is 22% in a total of 36,000 MCMC iterations. Legend: s: vessel stiffness (expressed in g/cm/s^2 units), r_1, r_2, c: Windkessel resistances and compliance factors (dimensionless), ξ: vessel tapering factor (dimensionless), S: residual sum-of-squares, σ^2: noise variance. Figure taken from our study in [141].

from 20% up to 40%). This is not surprising considering the extra DR step in the DRAM algorithm, which upon rejection of the first proposal, proposes a second point at every iteration of the algorithm. The result is a higher acceptance rate, and thus, fewer iterations are needed to reach convergence when compared to AM. This does not necessarily incur lower computational time, since in every DRAM iteration, if a second point is proposed, the PDEs have to be solved twice within that iteration.

DRAM without parameter scaling

Figure 3.9 illustrates results from DRAM without allowing for parameter scaling in the proposal density. It is obvious that several proposals with extremely large RSS values are falsely ac-

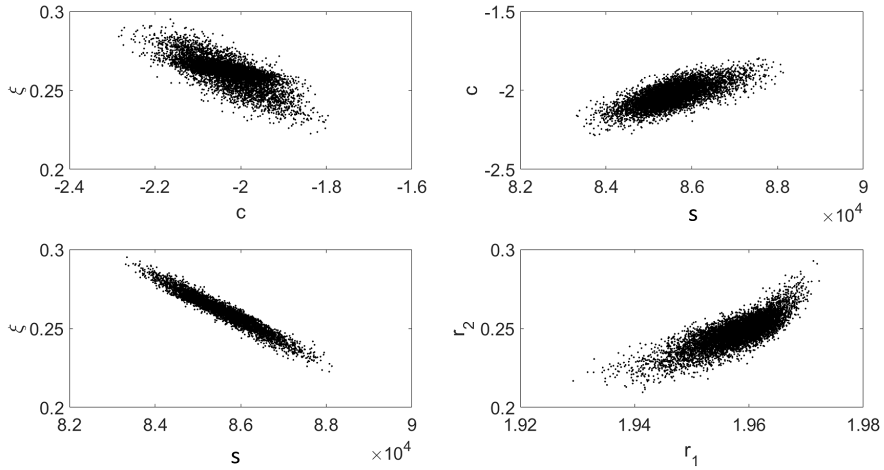


Figure 3.5: Pairwise posterior correlations for the parameter posterior samples obtained with the Adaptive Metropolis algorithm for the 5D model corresponding to a control mouse. Legend: s: vessel stiffness (expressed in g/cm/s^2 units), r_1, r_2, c: Windkessel resistances and compliance factors (dimensionless), ξ: vessel tapering factor (dimensionless). Figure taken from our study in [141].

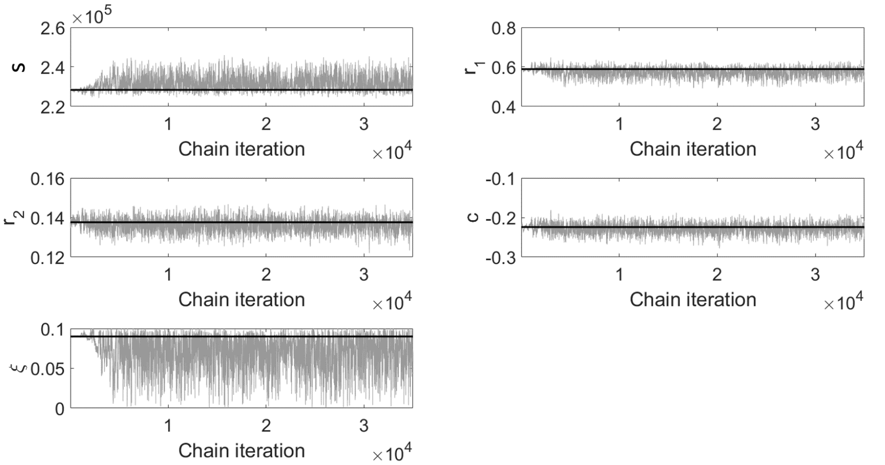


Figure 3.6: Markov chains traceplots for the parameters obtained using the Adaptive Metropolis algorithm for the 5D model corresponding to a hypoxic mouse. Starting values for the algorithm were the optimised values, superimposed in black horizontal lines. Acceptance rate is 22% in a total of 36,000 MCMC iterations. Legend: s: vessel stiffness (expressed in g/cm/s^2 units), r_1, r_2, c: Windkessel resistances and compliance factors (dimensionless), ξ: vessel tapering factor (dimensionless). Figure taken from our study in [141].

cepted, which is not the case for DRAM allowing for parameter scaling (Figure 3.8). Parameter vector values producing extremely high RSS values are falsely accepted because the ratio of the proposal distributions in unscaled space (see eq (3.17)) becomes large, making the acceptance rate 1.

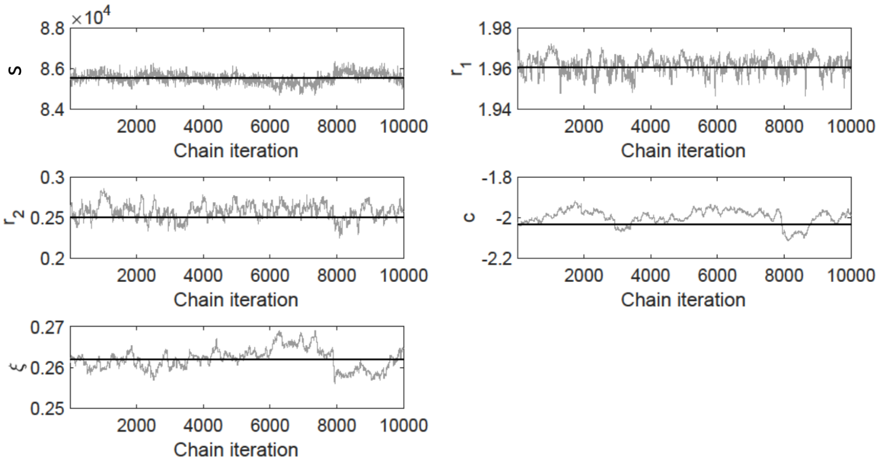


Figure 3.7: Markov chains traceplots for the parameters obtained using the Delayed Rejection algorithm for the 5D model corresponding to a control mouse. Starting values for the algorithm were the optimised values, superimposed in black horizontal lines. Acceptance rate is 22% in a total of 10,000 MCMC iterations. Legend: s: vessel stiffness (expressed in g/cm/s^2 units), r_1, r_2, c: Windkessel resistances and compliance factors (dimensionless), ξ: vessel tapering factor (dimensionless). Figure taken from our study in [141].

3.5.4 MCMC convergence and efficiency diagnostics

Visual inspection of the traceplots helped rule out algorithms producing sub-optimal mixing of the Markov chains (DR), and MCMC convergence and efficiency diagnostics (reviewed in Chapter 2) were implemented only for the DRAM and AM algorithms, which exhibited good mixing. Tables 3.3 and 3.4 show the results based on the Geweke test, the Brooks Gelman Rubin test, and the Effective Sample Size (ESS). For both DRAM and AM, the Geweke test registered a p-value of 0.99, well above 0.05 for all model and mouse types, suggesting that there was no evidence of a difference between the mean of the first 10% samples and last 50% samples of the Markov chain. The MPSRF was below 1.1 for all four cases. The estimates for ESS suggest that there is a great amount of correlation between the samples for both algorithms with ESS normalised by the number of MCMC iterations N being much smaller than 1 (1 would be obtained for independent sampling). Both algorithms, AM and DRAM register comparable efficiency, and the 5D model tends to have slightly lower ESS/N than the 4D model, possibly due to the strong correlation between the stiffness and tapering, retarding the convergence of the algorithm.

3.5.5 Model Selection

Model selection to discriminate between two competing models: a 4D model (without tapering), and a 5D model (with tapering) is employed. Table 3.5 shows results based on four criteria: AICc, BIC, DIC and WAIC. WAIC was calculated based on 1000 DRAM samples. Results

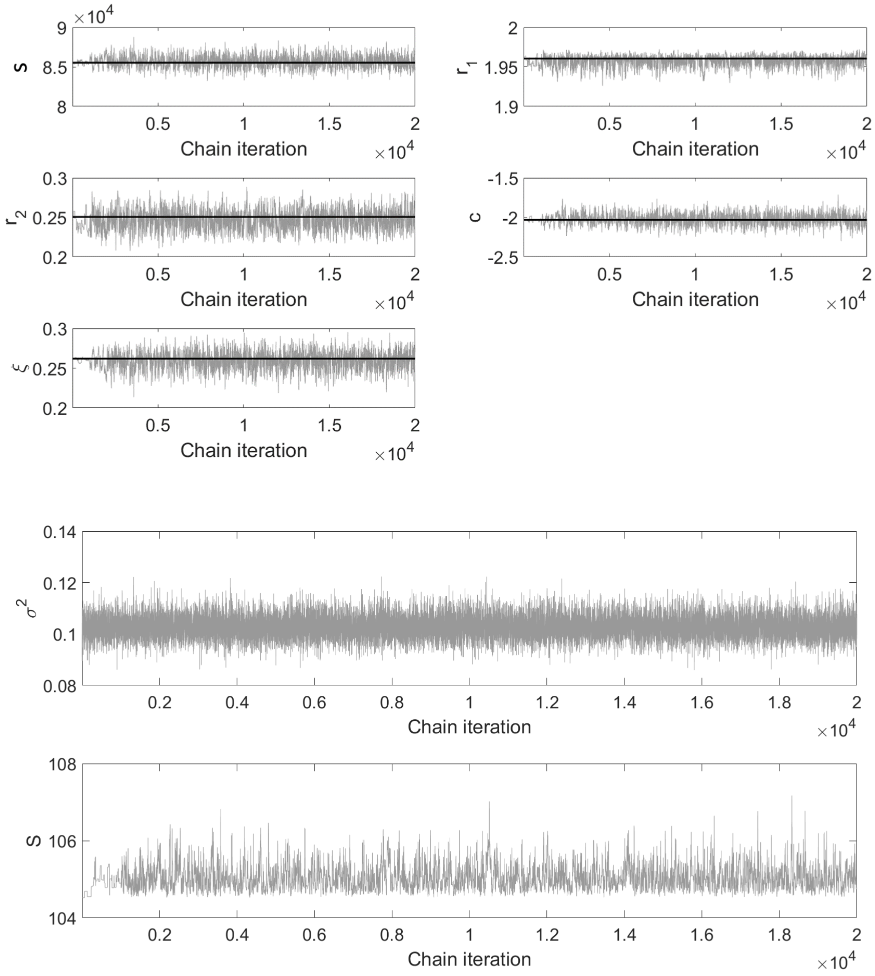


Figure 3.8: Markov chains traceplots for the parameters, residual sum-of-squares and noise variance obtained using the Delayed Rejection Adaptive Metropolis algorithm with parameter scaling for the 5D model corresponding to a control mouse. Starting values for the algorithm were the optimised values, superimposed in black horizontal lines. Acceptance rate is 39% in a total of 20,000 MCMC iterations. Legend: s: vessel stiffness (expressed in g/cm/s^2 units), r_1, r_2, c: Windkessel resistances and compliance factors (dimensionless), ξ: vessel tapering factor (dimensionless), S: residual sum-of-squares, σ^2: noise variance. Figure taken from our study in [141].

indicate that the scores are similar and fairly consistent across different criteria. For the control mouse, all the scores are lower for the 5D model compared to the 4D model, implying that in the case of the control mouse adding the tapering factor makes the model more appropriate for the data. On the other hand, for the hypoxic mouse, the scores are very similar across the two models, WAIC favours the 4D model, while AICc, BIC and DIC favour the 5D model. Considering the very small difference in RSS between the 4D and 5D models, the similarity of the scores and the fact that WAIC, which is the most reliable score (see Section 2.3.8) favours having four parameters, the 4D model is chosen as the better model for the hypoxic mouse data. Hence, there is evidence that the tapering factor is not significant for the hypoxic mouse, but the

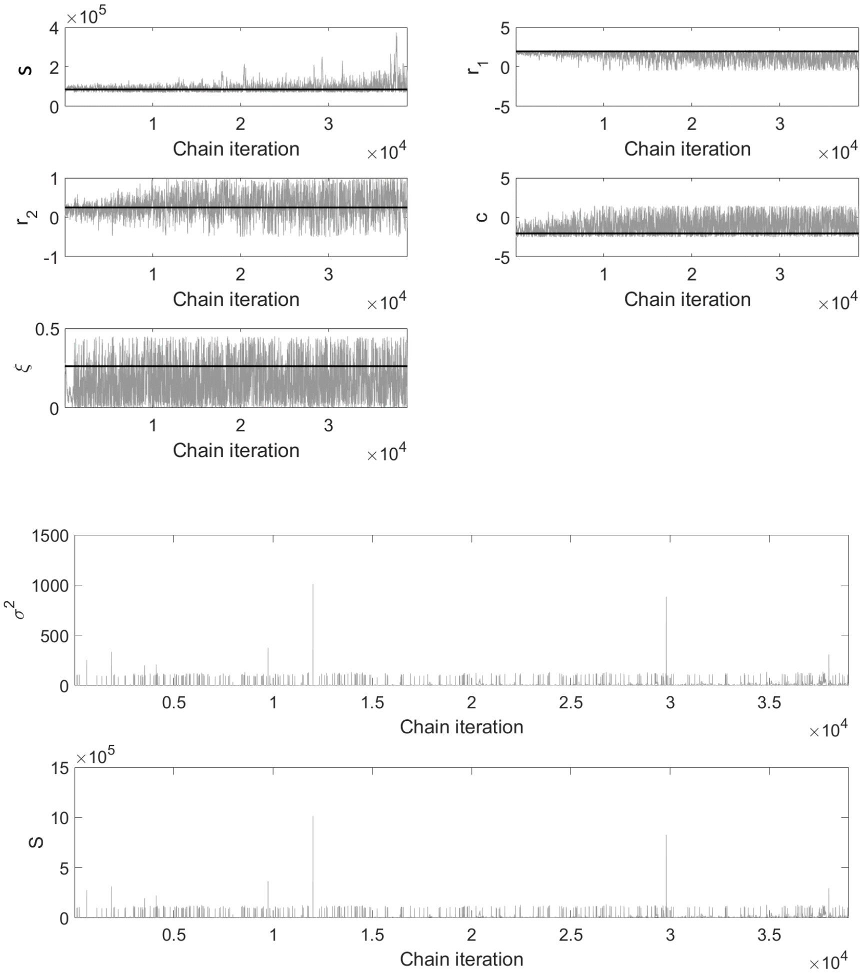


Figure 3.9: Displaying the effect of not allowing for parameter scaling: Markov chains traceplots for the parameters, residual sum-of-squares and noise variance obtained using the Delayed Rejection Adaptive Metropolis algorithm with no parameter scaling for the 5D model corresponding to a control mouse. Starting values for the algorithm are the optimised values, superimposed in black horizontal lines. Acceptance rate is 30% in a total of 40,000 MCMC iterations. Legend: s: vessel stiffness (expressed in g/cm/s^2 units), r_1, r_2, c: Windkessel resistances and compliance factors (dimensionless), ξ: vessel tapering factor (dimensionless), S: residual sum-of-squares, σ^2: noise variance. Figure taken from our study in [141].

opposite is true for the control mouse.

3.6 Discussion and Conclusions

This chapter performs inference and uncertainty quantification of haemodynamic parameters: vessel wall stiffness and boundary conditions (Windkessels) parameters of a 1D fluid-dynamics model of the pulmonary circulation in a fixed vessel network obtained from one image segmentation, and a zero vessel tapering, as well as a non-zero tapering is assumed, which is inferred from data. Unknown parameters were sampled using MCMC, namely Delayed Rejection Adap-

Mouse type	Model type	Geweke test (p-values)	MPSRF	ESS/N
Control	4D	all > 0.05	1.004	(0.07,0.07,0.07,0.06)
Hypoxic	4D	all > 0.05	1.006	(0.07,0.07,0.07,0.09)
Control	5D	all > 0.05	1.01	(0.05,0.03,0.04,0.05,0.05)
Hypoxic	5D	all > 0.05	1.03	(0.04,0.04,0.08,0.06,0.04)

Table 3.3: MCMC convergence diagnostics: Geweke test, Multivariate Potential Scale Reduction Factor (MPSRF), and sampler efficiency: Effective Sample Size (ESS) normalised by number of samples ($N = 20,000$) for all parameters, corresponding to results obtained using the Delayed Rejection Adaptive Metropolis algorithm for every mouse and model type.

Mouse type	Model type	Geweke test (p-values)	MPSRF	ESS/N
Control	4D	all > 0.05	1.01	(0.05,0.05,0.05,0.05)
Hypoxic	4D	all > 0.05	1.004	(0.06,0.06,0.06,0.06)
Control	5D	all > 0.05	1.04	(0.03,0.02,0.02,0.04,0.02)
Hypoxic	5D	all > 0.05	1.06	(0.02,0.03,0.05,0.04,0.02)

Table 3.4: MCMC convergence diagnostics: Geweke test, Multivariate Potential Scale Reduction Factor (MPSRF), and sampler efficiency: Effective Sample Size (ESS) normalised by number of samples ($N = 36,000$) for all parameters, corresponding to results obtained using the Adaptive Metropolis algorithm for every mouse and model type.

Mouse type	Model type	AICc (AIC)	BIC	DIC	WAIC
Control	4D	772	792	772	774
Control	5D	**579**	**603**	**578**	**580**
Hypoxic	4D	307	326	306	**304**
Hypoxic	5D	**301**	**325**	**300**	310

Table 3.5: Model selection scores (AICc, BIC, DIC, WAIC) for every mouse and model type. AICc is very similar to AIC (up to 4 dp), since the sample size is large (1024 data points). Lower scores (in bold) indicate the better model.

tive Metropolis algorithm and its variations Adaptive Metropolis and Delayed Rejection. To make the computations feasible, the MCMC algorithms were started from optimised parameter values, which were found with non-linear constraint optimisation. The methodology proposed builds on and further improves existing literature by combating the different parameter magnitudes through the use of a novel parameter scaling applied to real mouse data. Results indicated that the DRAM, DR and AM algorithms performed similarly in terms of accuracy, DRAM and AM had comparable performance in terms of efficiency, while visual inspection indicated that DR was inferior.

This study found that pulmonary hypertension is associated with stiffer and less compliant proximal and distal vasculature. In addition, model selection was performed to identify whether there was evidence of vessel tapering from data. The AICc, BIC, DIC and WAIC model selection criteria consistently chose the more complex model for the control mouse, i.e. tapering was consistent with the control mouse data. For the hypoxic mouse the scores were inconsistent, but given the difference between them was small and the most reliable score WAIC favoured the simpler model, the conclusion was that tapering was not significant for the hypoxic mouse data. A possible explanation for this could be that in hypoxia the vessel walls become stiff, hence there may not be a pronounced decrease in the radii along the vessels' length.

All these results are based on a mathematical model that looks more appropriate for the hypoxic mouse than for the control mouse. It was hypothesized that the slight model discrepancy is caused by: (i) the simplicity of the model specifying the elastic behaviour of the blood vessels and/or the boundary conditions, (ii) uncertainty of the geometry measurements which are not specific to a given mouse, (iii) a combination of (i) and (ii). This slight model mis-specification has the undesired effect of breaking the iid error Normality assumption. Therefore, to account for the correlated structure in the errors, a Markov model that incorporates the correlation structure is incorporated in subsequent analysis presented in this thesis.

Making a few restrictive assumptions about the parameters allowed us to decrease the parameter dimension considerably from a 55D problem to a 5D problem. Some of these assumptions are checked in the next chapter.

Finally, work in next chapters includes implementing MCMC methods which are more efficient than the random-walk DRAM algorithm, e.g. Hamiltonian Monte Carlo [132], which suppresses the random behaviour of the algorithm by using a momentum variable guiding the direction in which the proposals are made. To speed up simulations, HMC can be run on an objective function emulated using Gaussian Processes [157].

In the next part of this chapter, an analysis incorporating the correlation structure of the residuals is presented.

3.7 Introduction and methods

The first part of this chapter presents results based on a statistical analysis which made the assumption of iid residuals. However, a plot of the errors in time reveals that the iid assumption is invalid, see Figure 3.11. Therefore, the aim of this second part is show results based on incorporating the correlation structure of the errors into the analysis. The error correlation is a consequence of the model inadequacy (i.e. inability of the mathematical model to faithfully capture the measured haemodynamics, see the peak offset in Figure 3.3, potentially due to the simplicity of the model specifying the elasticity of the blood vessel walls, the simplifying Windkessel boundary conditions, fixed inflow, or the geometry measurements, which are not mouse-specific), of the nature of the data (i.e. haemodynamic measurements at the current time point within the heartbeat depend on the previous time points), and on the data post-processing (i.e. noise reduction/filtering techniques via data smoothing or averaging). The exact experimental protocols for data measurement are unknown, which makes a direct deconvolution impossible. A parametric assumption about the residuals is made, namely that they follow the multivariate normal distribution. Under this assumption, the statistical model is expressed as:

$$\mathbf{y} = \mathbf{m}(\theta) + \varepsilon, \quad \varepsilon \sim \mathscr{MVN}(\mathbf{0}, \mathbf{C}), \tag{3.21}$$

and the log likelihood is given by:

$$\begin{aligned} \log p(\mathbf{y}|\theta) &= -\frac{1}{2}\log(\det(2\pi\mathbf{C})) - \frac{1}{2}(\mathbf{y} - \mathbf{m}(\theta))^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{y} - \mathbf{m}(\theta)) \\ &= -\frac{1}{2}\log(\det(2\pi\mathbf{C})) - \frac{1}{2}\mathbf{r}(\theta)^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{r}(\theta), \end{aligned} \tag{3.22}$$

where $\mathbf{C}$ is the covariance matrix and $\mathbf{r}(\theta)$ is the vector of residuals as a function of θ, given by the difference between the measured data and the data simulated with the θ PDE parameter vector.

In this chapter the analysis focuses on comparing two wall models: a linear wall and a non-linear wall model, as indicated in Table 2.1 and described in Section 2.2, for the control mouse (hypoxic mouse data were not included as our study in Qureshi et al. [151] revealed hardly any difference in fit between the two models, see Figure 3.10). The model selection was performed with AICc and BIC scores. The data are described in Section 2.1.

The vessel tapering factor was set to zero despite having been found significant in the previous part of this chapter. Post-analysis found that inferring vessel tapering from the MPA-measured data can lead to unrealistic physiological characteristics (e.g. unrealistically high pulse pressure) in distal vessels. The tapering factor interacts differently with different wall models, hence it should be estimated from image data during the image segmentation process rather than from haemodynamic simulations; this is beyond the scope of the current study.

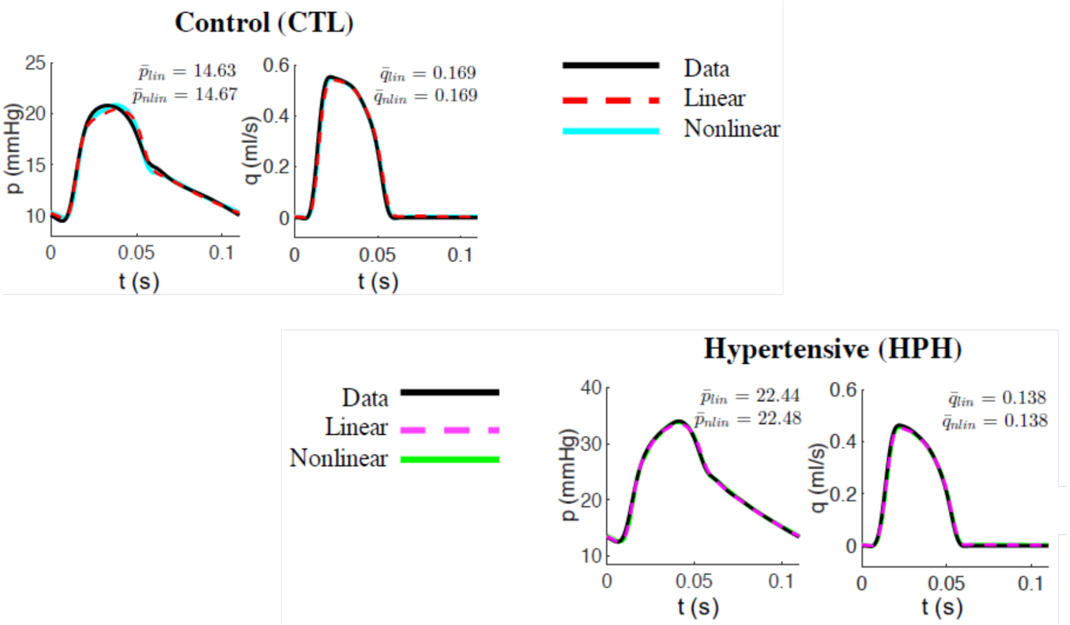


Figure 3.10: Measured pressure and flow data in the main pulmonary artery plotted alongside the optimised pressure waveform for the linear (Linear B) and non-linear wall models described in Chapter 2 for the control and hypoxic (hypertensive) mouse. Figure taken from our study in [151].

3.7.1 Covariance matrix estimation

To estimate the covariance matrix **C** in eq (3.22), several approaches were taken to fit the time series of residuals in Figure 3.10: an autoregressive moving average model ARMA(p,q) and Gaussian process (GP) models with several standard kernels: a squared exponential, a Matérn 5/2, a Matérn 3/2, a neural network kernel and a periodic kernel [157].

Linear interpolation

The time series has 1024 time points that come from linear interpolation of 552 original measurements. Linear interpolation can result in ill-conditioned or singular covariance matrix. If the columns of the matrix are merely similar, the matrix is ill-conditioned, while if a column is a linear combination of other columns, the matrix is singular. To avoid this, linear interpolation of the 552 original measurements was done for data reduction, to obtain 512 points (the numerical scheme integrates the Windkessel eq (2.7) in frequency domain using Fast Fourier transform and requires 2^m points).

For illustration, consider the simple case of six equidistant time points $T_0, T_1, T_2, T_3, T_4, T_5$ with associated measurements $f(T_0), f(T_1), f(T_2), f(T_3), f(T_4), f(T_5)$. To reduce this to five equidistant time points, points are spaced out at $\frac{6}{5} = 1.25$ units apart, instead of 1 unit apart, i.e. $T_0, T_{1.25}, T_{2.5}, T_{3.75}, T_5$. By linear interpolation: $f(T_{1.25}) = 0.75 \times f(T_1) + 0.25 \times f(T_2)$, $f(T_{2.5}) = 0.5 \times f(T_2) + 0.5 \times f(T_3)$, $f(T_{3.75}) = 0.25 \times f(T3) + 0.75 \times f(T_4)$ with $f(T_0)$ and $f(T_5)$ remaining unchanged. This can be generalised to downsampling 552 time points to 512.

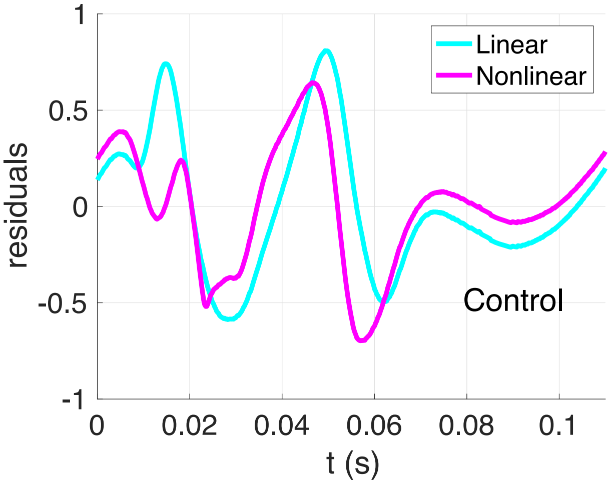


Figure 3.11: Residual time series in the main pulmonary artery, as given by the difference between the measured and the simulated pressure signal corresponding to the linear and nonlinear wall model for the control mouse. Figure taken from our study in [151].

ARMA model

The residuals in Figure 3.11 indicate that the residual time series is non-stationary. In order to fit an ARMA model, the non-stationary residual time series must be transformed into a stationary time series. This was done by taking 2nd order differences, where the order was chosen based on a formal statistical test, Kwiatkowski–Phillips–Schmidt–Shin (KPSS) [106], which tests for stationarity of the residual time series. Denoting by r_t the t^{th} element of the residuals vector, second order differences are calculated as: $B^2(r_t) = B(r_t - r_{t-1}) = (r_t - r_{t-1}) - (r_{t-1} - r_{t-2}) = r_t - 2r_{t-1} + r_{t-2}$, where B denotes the lag operator. The result of this is a stationary time series, as seen in the left panel of Figure 3.12. The middle panel shows the autocorrelation function (ACF) of the stationary time series, and the right panel displays the partial autocorrelation function (PACF). ACF and PACF figures help identify if the differenced residuals can be modelled using either AR or MA models. If the ACF is significantly different from zero for the first q lags only (for small q), then an MA(q) model is appropriate. If the PACF is significantly different from zero for the first p lags only (for small p), then an AR(p) model is appropriate. Figure 3.12 suggests that the process is neither an AR nor an MA model, and the two components need to be combined into a more complex ARMA model. The ARMA model was fitted to the twice differenced residuals, which is the equivalent of fitting an autoregressive integrated moving average model (ARIMA) model with difference parameter $d = 2$ to the original residuals.

An ARMA(p,q) process is the stationary solution to:

$$A_p(B)r_t = C_q(B)e_t, \tag{3.23}$$

where e_t is white Gaussian noise, and A_p and C_q are polynomials of order p and q, respectively.

An ARIMA(p,d,q) process is defined by:

$$A_p(B)(1-B)^d r_t = C_q(B)e_t. \tag{3.24}$$

In order to select the order $p\&q$ of the ARIMA model, a few $p\&q$ values were tried, the ARIMA(p,d,q) was fitted to the original residuals, and the BIC score was computed for each ARIMA(p,q) model. The model with the lowest BIC score was taken forward.

Computation of the covariance matrix was done following the standard procedure proposed by Box & Jenkins [15, 120]. The analysis was carried out in *R Studio*, and the correlation function at lag l, $\rho(l)$ was computed with the ARMAacf inbuilt function (or TacvfARMA) in the FitARMA package [121]. The residual variance was estimated as $\hat{\sigma}^2 = \frac{\sum_{i=1}^{n}(r_i-\bar{\mathbf{r}})^2}{n-1}$, and the covariance function was found by multiplying $\hat{\sigma}^2$ and $\hat{\rho}(l)$, where l is the lag, $l = 0,...,n-1$. The covariance matrix was computed based on the covariance function between any two residual points, as follows:

$$\mathbf{C} = \begin{pmatrix} \mathrm{Cov(r_1,r_1)} & \mathrm{Cov(r_1,r_2)} & \mathrm{Cov(r_1,r_3)} & ... & \mathrm{Cov(r_1,r_n)} \\ \mathrm{Cov(r_2,r_1)} & \mathrm{Cov(r_2,r_2)} & \mathrm{Cov(r_2,r_3)} & ... & \mathrm{Cov(r_2,r_n)} \\ . & . & . & ... & . \\ . & . & . & ... & . \\ . & . & . & ... & . \\ \mathrm{Cov(r_n,r_1)} & \mathrm{Cov(r_n,r_2)} & \mathrm{Cov(r_n,r_3)} & ... & \mathrm{Cov(r_n,r_n)} \end{pmatrix} \tag{3.25}$$

Hence,

$$\mathbf{C} = \sigma^2 \begin{pmatrix} \hat{\rho}(0) & \hat{\rho}(1) & \hat{\rho}(2) & ... & \hat{\rho}(n-1) \\ \hat{\rho}(1) & \hat{\rho}(0) & \hat{\rho}(1) & ... & \hat{\rho}(n-2) \\ . & . & . & ... & . \\ . & . & . & ... & . \\ . & . & . & ... & . \\ \hat{\rho}(n-1) & \hat{\rho}(n-2) & \hat{\rho}(n-3) & ... & \hat{\rho}(0), \end{pmatrix} \tag{3.26}$$

where $\sigma^2\hat{\rho}(0) = \hat{\sigma}^2$ is the estimated noise variance, and $\mathbf{C}$ is a symmetric matrix with constant diagonals under the assumption that the twice differenced residual time series is stationary.

GP models

Gaussian process (GP) models were also fitted to the time series of original residuals using a variety of standard kernels: a squared exponential, a Matérn 5/2, a Matérn 3/2, a neural network kernel and a periodic kernel [157] (*Matlab GPstuff* toolbox [203] was used). The hyperparameters of the GP covariance function were estimated from the residuals time series in Figure 3.11 by maximisation of the log marginal likelihood using standard optimisation algorithms (see [157] and Chapter 2). Using the optimum covariance function hyperparameters, the covariance matrix

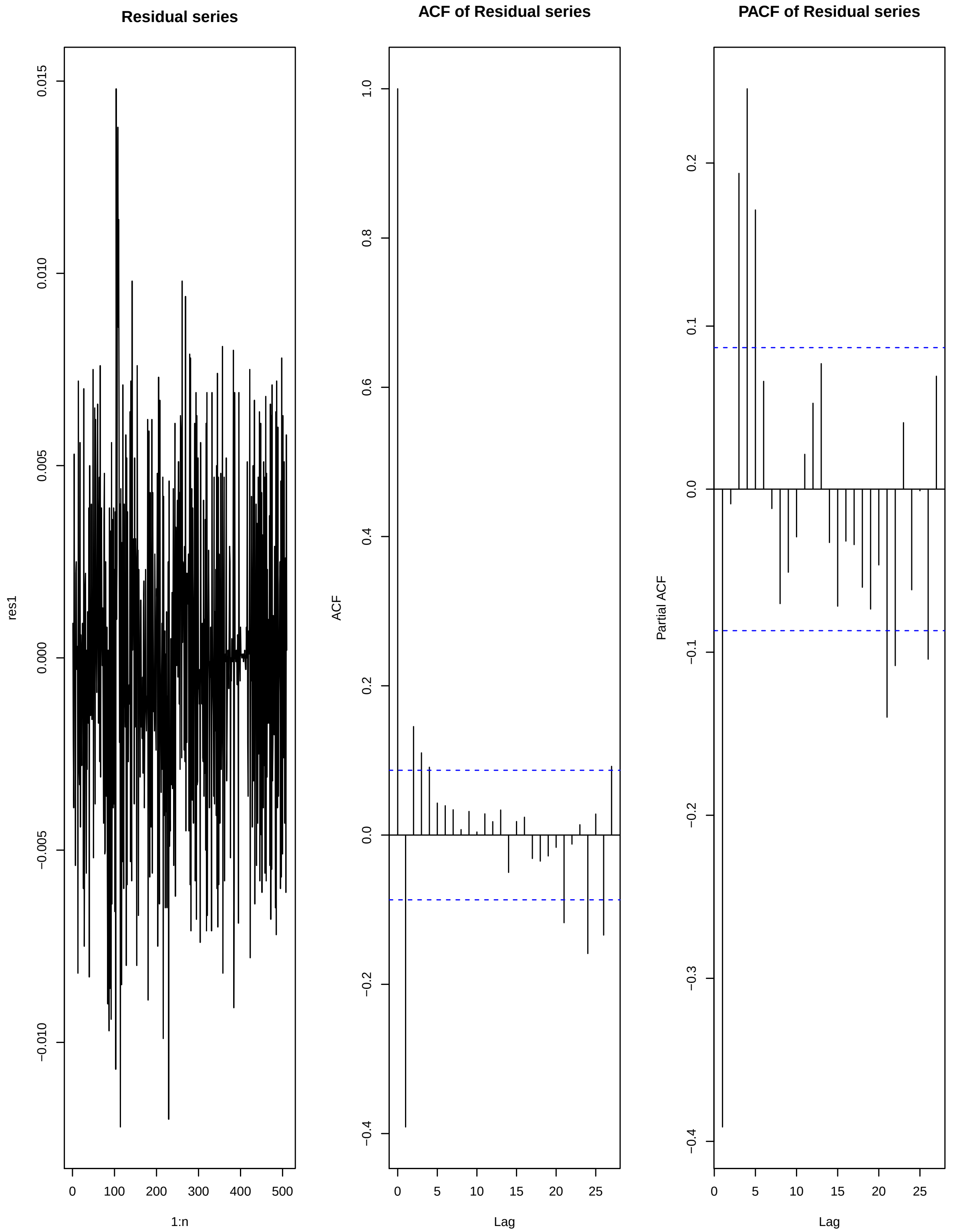


Figure 3.12: Twice differenced residual time series for the non-linear wall model - left, and the corresponding autocorrelation function (ACF) - middle and partial autocorrelation function (PACF) - right. Figures were obtained by using the *arma* function in *R Studio*.

can be obtained by integrating out the latent functions from the joint posterior distribution (see Chapter 2 for more details).

Since the correlation structure of the noise depends on the experimental protocol and is independent of the type of the mathematical model, the hyperparameter optimisation was carried out under the constraint that the covariance matrices for both wall models had to be the same[1]. Two approaches were taken. For the first, for each GP model the log marginal likelihood of each wall model was maximised separately, resulting in two covariance matrices that were averaged, and the average covariance matrix was the estimated covariance matrix. For the second approach, the log marginal likelihood for both wall models was jointly maximised, subject to the constraint of equal covariance matrices, yielding one single covariance matrix. Both approaches returned the same model ranking for the model selection, thus, for simplicity the first approach was applied to the ARMA model to obtain the covariance matrix.

3.7.2 Covariance matrix inversion

Obtaining the log likelihood in eq (3.22) requires inverting the covariance matrix, which was performed by Choleski decomposition in combination with the numerically more stable backslash operator. For a triangular matrix, the backslash operator employs a forward substitution algorithm. For example, suppose $\mathbf{L}$ is a lower triangular matrix with non-zero diagonal elements,

$$\mathbf{Lx} = \mathbf{b} \Leftrightarrow \mathbf{x} = \mathbf{L}^{-1}\mathbf{b} \Leftrightarrow \mathbf{x} = \mathbf{L}\backslash\mathbf{b}. \tag{3.27}$$

The forward substitution algorithm proceeds as follows:

$$\begin{bmatrix} L_{1,1} & 0 & \dots & 0 \\ L_{2,1} & L_{2,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L_{n,1} & L_{n,2} & \dots & L_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

the system can be solved using the forward substitution algorithm:

$$\begin{aligned} x_1 &= \frac{b_1}{L_{1,1}}, \\ x_2 &= \frac{b_2 - L_{2,1}x_1}{L_{2,2}}, \\ &\vdots \\ x_n &= \frac{b_n - L_{n,1}x_1 - L_{n,2}x_2 - \cdots - L_{n,n-1}x_{n-1}}{L_{n,n}}. \end{aligned} \tag{3.28}$$

[1] Early analysis made the assumption that the factor with the main contribution to the error correlation was independent of the wall model used, this assumption is relaxed in Chapter 4.

The covariance matrix $\mathbf{C}$ can be expressed as

$$\mathbf{C} = \mathbf{L}\mathbf{L}^{\mathrm{T}}, \tag{3.29}$$

where $\mathbf{L}$ is a lower triangular matrix and inversion of $\mathbf{C}$ is given by:

$$\mathbf{C}^{-1} = (\mathbf{L}^{\mathrm{T}})^{-1}\mathbf{L}^{-1}. \tag{3.30}$$

Thus the computation of the Mahalanobis distance in eq (3.22) follows as

$$\mathbf{r}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{r} = (\mathbf{r}^{\mathrm{T}}/\mathbf{L}^{\mathrm{T}})(\mathbf{L}\backslash\mathbf{r}) = (\mathbf{L}\backslash\mathbf{r})^{\mathrm{T}}(\mathbf{L}\backslash\mathbf{r}), \tag{3.31}$$

where $\backslash$ represents the backslash operator and $/$ is the right matrix division operator.

For the $\log(\det(\mathbf{C}))$ calculation in eq (3.22), the following can be noted. If the covariance matrix $\mathbf{C}$ is high-dimensional, e.g. $n = 500$, and the matrix entries are between [0,1], naively calculating the determinant $\det(\mathbf{C})$ can be prone to numerical instabilities, i.e. numerical underflow leading to a zero determinant, and hence $\log(\det(\mathbf{C})) = -\infty$. This can easily be avoided by computing

$$\log(\det(\mathbf{C})) = 2\log(\det(\mathbf{L})) = 2(\log(L_{1,1}L_{2,2}...L_{n,n}) = 2(\log(L_{1,1}) + \log(L_{2,2}) + ...\log(L_{n,n})), \tag{3.32}$$

which is the sum of the log main diagonal entries (eigenvalues) of the lower triangular matrix $\mathbf{L}$.

3.7.3 Parameter optimisation

For model selection with AICc and BIC, the maximum likelihood estimates need calculating, see eq (2.85). This would require an iterative optimisation scheme, where for each parameter adaptation, the covariance matrix would have to be re-computed, and the procedure repeated until convergence. As this would lead to a substantial increase in the computational costs, the maximum likelihood parameters were approximated by the parameters that minimise the residual sum-of-squares error.

3.8 Results and Conclusions

An ARIMA(3,2,3) was fitted to the original residuals of the linear wall model residuals, and an ARIMA(3,2,5) to the residuals of the non-linear wall model. The AICc and BIC scores for the linear and non-linear wall model with covariance matrix obtained with the ARIMA and the five GP models are displayed in Table 3.6. All six models consistently indicate lower (better) AICc and BIC scores for the linear wall model compared to the non-linear wall model.

Model	Score	ARIMA	GP SqExp	GP Mat3/2	GP Mat5/2	GP NN	GP periodic
Linear	AICc	**-4418**	**-4260**	**-4304**	**-4396**	**-4436**	**-4252**
Non-linear	AICc	-4279	-4152	-4198	-4251	-4352	-4150
Linear	BIC	**-4401**	**-4243**	**-4288**	**-4380**	**-4420**	**-4236**
Non-linear	BIC	-4254	-4131	-4177	-4230	-4330	-4129

Table 3.6: AICc and BIC scores for the linear and non-linear wall model with the covariance matrices obtained using the ARIMA model and five Gaussian Process (GP) models with squared exponential, Matérn 3/2, Matérn 5/2, neural network and periodic covariance kernels. Lower AICc and BIC values (in bold) indicate the better wall model.

Therefore, based on the analysis carried out, the control mouse data seems to support the linear wall model better than the non-linear wall model. These findings are based on possibly sub-optimal parameter estimates since the maximum likelihood parameters were approximated by the parameters that minimise the residual sum-of-squares error. In addition, by employing an optimisation procedure the uncertainty in the PDE parameters and error covariance hyper-parameters is ignored. In the next chapters these issues are tackled by jointly sampling the haemodynamic and error parameters using MCMC, which allows calculating a more reliable model selection criteria, WAIC.

Chapter 4

Assessing model mismatch and model selection

This chapter emphasizes an often neglected, though important source of uncertainty: in the mathematical model form due to the discrepancy between the model and the reality, and in the measurements due to the wrong noise model (jointly called ‘model mismatch’). The model considered is a 1D fluid-dynamics model of the blood pulmonary circulation, whose haemodynamic parameters are estimated with Bayesian methods. This approach allows a natural quantification of the uncertainty in the parameter values, which is propagated through to the model (pulmonary blood flow and pressure) predictions. The focus is on investigating the impact of minimising the mean squared error between the measured and the predicted data (the conventional method) in the presence of model mismatch, which is represented with Gaussian Processes. Additionally, several vessel stiffness relations, as well as a linear and a non-linear wall model are considered, and the Watanabe Akaike Information Criterion is used to select the model that best predicts pulmonary haemodynamics.

Note: This chapter is based on the paper ‘Assessing model mismatch and model selection in a Bayesian uncertainty quantification analysis of a fluid-dynamics model of pulmonary blood circulation’ (submitted to J. R. Soc. Interface) on which I am the first author.

4.1 Introduction

Emanating from the right ventricle, the main pulmonary artery (MPA) bifurcates to the left and right pulmonary arteries, and continues to branch in rapid succession until reaching the capillary beds surrounding the alveoli. This intricate network transports blood to the lungs at a low pressure (8 to 30 mmHg). Clinical monitoring of pulmonary health includes analysis of blood pressure, measured invasively via catheterisation, imaging data, such as computed tomography (CT), and ventilation perfusion scans [97]. While such data are routinely analysed, they are not measured simultaneously nor integrated. To understand the interaction between the pulmonary circulation and the right ventricle in both healthy and disease environments it is essential to combine imaging data with haemodynamic data. To do so, a 1D fluid-dynamics model (described in Section 2.2) is used in combination with statistical inference, providing predictions of blood pressure, blood flow, and vessel area, along with parameter estimates and measures of uncertainty in model predictions. Haemodynamic predictions are computed in an arterial network model constructed from micro-CT images from a control mouse and compared to dynamic pressure data in the MPA.

Several previous studies, e.g. [35, 38, 114, 152, 151] have developed 1D fluid-dynamics models which predict pulmonary blood flow and pressure. A few of these studies [38, 151] aimed at devising subject-specific predictions requiring estimation of model parameters, which was performed by minimising the least squares error between the predictions and the measurements. While these studies provided valuable insight into the physiology, they did not analyse predictions in a statistical framework.

Similar to previous studies [38, 151], the 1D model analysed here has two types of parameters: specifying the vessel network (i.e. radius, length, and connectivity of arteries in the network), and the haemodynamics (i.e. pressure and flow). This study focuses on analysing parameters intrinsic to the haemodynamics, i.e. model predictions in a fixed network geometry obtained from one image segmentation are investigated.

The haemodynamic equations have three types of parameters specifying the characteristics of the blood, the vessel tissue properties, and the boundary conditions. The blood viscosity and density are measured [215], and therefore assumed known and constant. The vessel stiffness can only be measured ex-vivo via stress-strain tests, and while it is possible to determine a stiffness value in the large vessels, it is difficult to do so in smaller vessels. This makes it challenging to validate the hypothesis that the smaller pulmonary vessels are stiffer than the large vessels [12]. The final parameter types specify the boundary conditions at each terminal vessel, as finite resolution in the imaging data requires premature truncation of the arterial network. From the pressure and flow data, the total resistance and compliance in the system can be approximated, and by assuming a given flow distribution at each junction, initial boundary condition estimates at each terminal vessel [151] can be provided (see Section 2.2.4). Here, the focus is on analysing the influence that parameters describing the vessel stiffness and boundary conditions have on model predictions.

To understand the influence of these parameters, several sub-models are examined. Motivated by results in our previous studies [114, 151] two constitutive equations relating pressure and area via a linear and a non-linear wall model are compared. Furthermore, it is investigated if the vessel stiffness is constant over the entire network (as suggested by Dawson et al. [103]), if it depends on the radius (as suggested by Olufsen et al. [139]), or if the vessel stiffness should be estimated for each vessel separately. In addition, as suggested in our previous studies [38, 151], global scaling factors are introduced for the boundary condition parameters at the terminal arteries, which are estimated.

This model potentially gives rise to parameter correlations and non-identifiability problems, i.e. not all parameters can be estimated uniquely given the model and the available data. To tackle these issues, a Bayesian approach was adopted, with the aim to obtain the posterior distribution of the parameters. This is analytically intractable, so Markov Chain Monte Carlo (MCMC) was used to sample parameters approximately from the posterior distribution (with an asymptotic convergence guarantee). The MCMC samples were then used to estimate the uncertainty of model predictions throughout the arterial network.

While several recent studies have incorporated Bayesian parameter estimation (e.g. [38, 141, 151, 170, 171]) and uncertainty quantification (UQ) [55, 122], most studies ignore the model mismatch. The current study assumes that the model mismatch stems from two sources: 1) inadequate mathematical model (i.e., model discrepancy, since the mathematical model is not a perfect representation of the real system) [96] and 2) incorrect noise model (i.e., erroneously assuming independence when the errors are, in fact, correlated). A few studies have investigated the former, i.e. the assumption of independent and identically distributed (iid) errors. In an analysis of viscoelestic models, Valdez et al. [199] showed that by scaling the least squares error, the residual vector is close to iid. Yet this study only obtained point estimates and did not estimate uncertainty of model predictions. Konukoglu et al. [101] included an inhomogenous variance, informed by the authors' experience with the data, in an electrophysiology model, and the authors noticed that the noise model greatly influences the inference results. Additionally, Qureshi et al. [151] accounted for correlated errors in a point estimate analysis of a fluid-dynamics model. Another study [205] uses a multifidelity approach, and the model discrepancy between the high-fidelity model and the low-fidelity models is investigated in an application of computational fluid dynamics. In contrast, the current study uses solely high-fidelity models, and instead asks about model-form uncertainty, which is quantified from available data. The work in [216] enhanced fluid velocity predictability of a Reynolds-averaged Navier Stokes (RANS) model by including shear tensor model uncertainty via a Kalman update scheme. These results showed that model predictions can be improved by adding a discrepancy term, even when predicting quantities where data are unavailable for calibration. Furthermore, Whittaker et al. [211] and Mirams et al. [124] discuss model discrepancy in a review of cardiac model calibration. Finally, Lei et al. [115] explore model discrepancy in electrophysiology, which they model with GPs and ARMA models; the authors show through synthetic studies that ignoring the model-form uncertainty leads to biased predictions and uncertainty underestimation. Despite of this, most cardiovascular modelling studies do not account for model discrepancy [45, 93].

In this study, a Bayesian approach is adopted, which quantifies the model structure uncertainty by incorporating a data-driven error model. The measurement and model error is explicitly included and modelled using Gaussian Processes (GPs) [157] following Kennedy et al. [96], with the aim that biophysical parameters are not misadjusted to compensate for the model mismatch [52]. The proposed Bayesian inference framework jointly samples the mathematical model parameters and the error model parameters from the posterior distribution with MCMC. Thus, uncertainties associated with parameters, measurements and model discrepancy are all accounted for in the analysis [218]. An alternative approach accounting for the model uncertainty is Bayesian model averaging [86] or dynamic model averaging [153], which combines all candidate models; a weighted average of the posterior distribution of the parameters under each model is taken, where the weights are given by the model posterior probability. However, the resulting averaged model lacks physiological interpretation, i.e. focuses on predictive, rather than explanatory performance, hence this approach is not pursued in the current work.

Physiological and synthetic pressure data are utilised to examine the consequence of inferring parameters when suspected model mismatch is unaccounted for. Results show that ignoring model mismatch biases parameter estimates and underestimates uncertainty in parameter space and output space, and the proposed method allows to correct this bias. In addition, a synthetic study is carried out displaying the effect of using data from multiple vessels on the parameter inference and UQ.

Finally, model selection is performed to discriminate between two constitutive models (a linear and non-linear wall model), with a number of parameter constraints (related to the vessel stiffness). The model selection analysis is based on a recent information criterion, the Watanabe Akaike Information Criterion (WAIC) [209], which can be directly computed from the MCMC samples.

4.2 Data

4.2.1 Physiological Data

This study compares model predictions to measured MPA blood pressure data from a control mouse lung (Figure 2.1). The experimental data are summarised in Section 2.1.

4.2.2 Synthetic data

For the control mouse, synthetic data obtained from a forward simulation of the mathematical model were also used. The synthetic, error-free data in all 21 vessels were generated using a linear wall model with a radius-dependent exponential stiffness (see Table 4.1) with parameter values consistent with the physiological data. To these data non-stationary, additive Gaussian correlated errors were added, which were generated using the same error parameters for all the vessels, assuming that the pressure transducer provides measurements that are independent of the measurement location. The error correlation induces a model mismatch if the wrong error model is utilised. To make the synthetic data physiologically realistic, a signal-to-noise ratio of approximately 100 was used (see Figure 5 in [194]), while ensuring that the pressure monotonicity constraint was satisfied (i.e. that pressure decreases as it approaches the periphery).

4.3 Model

4.3.1 Fluid-dynamics model of the pulmonary circulation

The mathematical models used are a linear wall model and non-linear wall model, as indicated in 2.1 and described in Section 2.2.

4.3.2 Model parameters

The vessel geometry, including vessel connectivity, and vessel radius and length, was assumed fixed and known from micro-CT images, as obtained based on one image segmentation (this uncertainty is quantified in Chapter 5). The vessels were assumed straight, i.e. radial tapering is negligible. Additionally, the viscosity and density were assumed known and constant, leaving the vessel stiffness and outflow boundary condition parameters to be inferred.

As noted earlier, two constitutive models relating pressure and area are compared: a linear wall model – see eq (2.3) and a non-linear wall model – see eq (2.4). For the linear wall model one parameter Eh (Young's modulus times the wall thickness) was estimated, while for the non-linear model two parameters, p_1 and γ were estimated. For each model two scenarios were investigated, that the vessel stiffness is constant, i.e. Eh and p_1 are constant, and that they increase as vessel radii decrease. For the latter, the dependence of Eh and p_1 on r_0 is expressed in eq (2.5).

The parameters are constrained to lie within physiologically plausible bounds in a univariate sense, but the parameters' behaviour in the joint space is unknown prior to carrying out the statistical analysis. Table 4.2 shows the univariate parameter ranges.

4.3.3 Overview of models: physiological hypotheses and model mismatch scenarios

Table 4.1 outlines the different models considered in this work, which help explore several physiological hypotheses and model mismatch scenarios. By analysing these models, the aim is to test if for the physiological data described in Section 2.1, the wall model is linear or non-linear, if the vessels share or have independent vessel stiffness, or if the vessel stiffness depends on the radius. Furthermore, it is examined if it is important to account for the model mismatch.

Linear wall model	Constant stiffness and no model mismatch
	Constant stiffness and model mismatch
	Radius-dependent stiffness and no model mismatch
	Radius-dependent and model mismatch
	Vessel-specific stiffness and no model mismatch
	Vessel-specific stiffness and model mismatch
Non-linear wall model	Constant stiffness and model mismatch
	Radius-dependent stiffness and model mismatch

Table 4.1: Models investigated: two constitutive models relating pressure-area (linear & non-linear, as indicated in Table 2.1) with several stiffness relations: constant (common to all vessels), radius-dependent (via eq (2.5)), vessel-specific (in a Bayesian hierarchical model), and model and measurement error assumptions via including or ignoring model mismatch. For the non-linear wall model, the no model mismatch scenario was not considered based on conclusions drawn from the linear model, clearly supporting modelling the model mismatch. In addition, the vessel-specific stiffness scenario was not pursued due to the interaction between the parameters p_1 and γ in eq (2.4), requiring vessel-specific (p_1, γ). This would lead to a very large number of parameters being estimated, requiring extremely high computational efforts (simulations would most likely take months to complete).

4.4 Statistical methods

4.4.1 Data likelihood

Normally distributed errors were assumed, with two scenarios: iid and correlated errors. Under these assumptions, the likelihood function can be expressed as:

- Iid errors: $\mathbf{y} \sim \mathscr{MVN}(\mathbf{m}(\theta), \sigma^2\mathbf{I})$, i.e.

$$p(\mathbf{y}|\theta, \sigma^2) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{\sum_{i=1}^n (y_i - m_i(\theta))^2}{2\sigma^2}\right), \tag{4.1}$$

  where

$$\sum_{i=1}^n (y_i - m_i(\theta))^2 = (\mathbf{y} - \mathbf{m}(\theta))^{\mathrm{T}}(\mathbf{y} - \mathbf{m}(\theta)) \tag{4.2}$$

  is the Euclidean distance, and $\mathbf{m}(\theta)$ is the vector of temporal pressure predictions from the mathematical model, $\mathbf{y}$ is the vector of temporal pressure measurements, and n is the number of time points. Lastly, $\mathscr{MVN}$ stands for Multivariate Normal distribution.

- Correlated errors: $\mathbf{y} \sim \mathscr{MVN}(\mathbf{m}(\theta), \mathbf{C})$, i.e.

$$p(\mathbf{y}|\theta, \mathbf{C}) = \det(2\pi\mathbf{C})^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{m}(\theta))^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{y} - \mathbf{m}(\theta))\right), \tag{4.3}$$

where

$$(\mathbf{y}-\mathbf{m}(\theta))^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{y}-\mathbf{m}(\theta)) \tag{4.4}$$

is the Mahalanobis distance.

4.4.2 Using GPs for model mismatch

The model mismatch is incorporated into the UQ analysis following the approach by Kennedy et al. [96]. The model mismatch stems from two sources: correlated measurement errors (when the iid assumption is wrongly made) and model discrepancy between the real system and the mathematical model as per eq (4.5) [96],

$$\mathbf{y}(\mathbf{t}) = \mathbf{m}(\theta,\mathbf{t}) + \zeta(\mathbf{t}) + \varepsilon(\mathbf{t}), \tag{4.5}$$

where $\mathbf{y}(\mathbf{t}) = (y(t_1),\dots y(t_n))$ are the measurements at every time point in the vector $\mathbf{t}$, $\mathbf{m}(\theta,\mathbf{t})$ is the simulator output in time (i.e. the prediction from the mathematical model evaluated at time points $\mathbf{t}$ with parameters θ), $\zeta(\mathbf{t})$ is the model discrepancy function, and $\varepsilon(\mathbf{t})$ is the measurement errors vector. Note that there is a distinction between *model discrepancy* and *model mismatch* in the way described above, so these words are not used synonymously.

Possible causes for the measurement error correlation are: the nature of the data (i.e. the blood flow or pressure measurements at the current time point depend on measurements at previous time points), and smoothing and averaging of the data. Possible causes for the model discrepancy are: numerical errors (e.g., numerical integration of the PDEs), model assumptions (e.g., purely elastic vessel walls, or the 1D model simplification), pressure and flow waveforms that do not have a direct physical relationship (due to the data smoothing and averaging), by the strict periodicity assumption of the measured pressure and flow (eqns (2.7) and (2.8)), uncertainty of the network geometry (kept fixed), and inconsistency between network geometry and haemodynamic data (e.g., the network geometry and the blood flow data do not come from the same mouse).

Due to the limited data, the lack of prior knowledge on the model discrepancy function, the unknown smoothing and averaging technique applied to the raw data, and the unknown machine precision for data measurement, both contributions from the measurement error and model error (model discrepancy) are modelled with one single GP model. Thus, the model mismatch is defined as in eq (4.6)

$$\begin{aligned}\mathbf{y}(\mathbf{t}) &= \mathbf{m}(\theta,\mathbf{t}) + \Gamma(\mathbf{t}),\\ \Gamma(\mathbf{t}) &= \zeta(\mathbf{t}) + \varepsilon(\mathbf{t}),\end{aligned} \tag{4.6}$$

where $\Gamma(\mathbf{t})$ is the model mismatch, represented by residuals.

While Kennedy's formulation of eq (4.6) allows for model discrepancy (modelled with a GP), it makes the assumption of iid measurement errors, i.e. assumes that $\varepsilon \sim \mathscr{MVN}(\mathbf{0},\sigma_n^2\mathbf{I})$.

The formulation in this chapter generalises this by allowing correlated measurement errors, and the iid errors term are included into the GP (which helps obtain the limiting case of iid errors), as explained in Section 4.4.3.

To help with the GP set-up for the model mismatch (i.e. for the residuals), the model mismatch is further expressed in terms of latent functions **f** as

$$\begin{aligned} \Gamma(\mathbf{t}) &= \mathbf{f}(\mathbf{t}) + \mathbf{u}(\mathbf{t}), \\ \mathbf{u}(\mathbf{t}) &\sim \mathscr{MVN}(\mathbf{0}, \sigma_n^2 \mathbf{I}), \end{aligned} \tag{4.7}$$

where σ_n^2 is the noise variance of the residuals, assumed iid Gaussian.

Next, a GP prior is placed on the latent functions **f**, which following eq (2.53), is given by

$$\mathbf{f}(\mathbf{t})|\gamma \sim \mathscr{GP}(\mathbf{0}, \mathbf{K}|\gamma). \tag{4.8}$$

This GP captures the correlation stemming from both the model discrepancy $\zeta(\mathbf{t})$ and the measurement errors $\varepsilon(\mathbf{t})$ (when iid errors are wrongly assumed), defined in eq (4.6). In principle one could place a GP prior on the model discrepancy function ζ, i.e. $\zeta(\mathbf{t}) \sim \mathscr{GP}(\mathbf{0}, \mathbf{K_1}|\gamma_1)$, and another GP prior on the measurement errors, i.e. $\varepsilon(\mathbf{t}) \sim \mathscr{GP}(\mathbf{0}, \mathbf{K_2}|\gamma_2)$, but due to the limited pressure data, these two terms cannot be distinguished and separately modelled; instead, a GP prior is placed on the sum of the two terms, and in eq (4.8), $\mathbf{K} = \mathbf{K_1} + \mathbf{K_2}$, assuming that the correlation between the model mismatch processes (i.e. measurement error and model discrepancy) is zero.

4.4.3 Using GPs to obtain the pressure data likelihood

In the likelihood functions above (eqns (4.1) and (4.3)), the iid errors scenario corresponds to neglecting model mismatch, defined in eq (4.6), while the correlated errors scenario allows for model mismatch. **C** in eq (4.3) is the covariance matrix of the residuals, i.e. $\mathbf{C} = \mathbf{K} + \sigma_n^2 \mathbf{I}$. Eq (4.7) allows to obtain the limiting case of iid errors, i.e. no model mismatch, by constraining the kernel hyperparameters. For example, for the squared exponential kernel in eq (2.54), by making the lengthscale, $l \to 0$, $\mathbf{C} = (\sigma_m^2 + \sigma_n^2)\mathbf{I}$ is obtained, hence by setting $\sigma^2 = \sigma_m^2 + \sigma_n^2$, the iid errors scenario is obtained, which allows getting the pressure data likelihood under iid errors, see eq (4.1). This approach is however not numerically stable (due to obtaining $\frac{0}{0}$ in eq (2.54)). The issue can be alleviated by placing a prior with zero mass on 0 for the lengthscale, or sampling $\frac{1}{l}$, which in practice never approaches ∞. Another approach that does not pose such problems and is numerically stable is to set σ_m^2 to 0 in eq (2.54) (making the covariance matrix $\mathbf{K_1} + \mathbf{K_2}$ a zero matrix), which implies that $\sigma^2 = \sigma_n^2$. For the neural network kernel in eq (2.59), the same can be accomplished by setting $w = 0$ and $b = 0$.

In the current work, the neural network kernel with hyperparameters w and b, and noise variance of the residuals σ_n^2 is used for the GP model representing the residuals to capture the non-stationarity in the residual series (Figure 4.7).

4.4.4 Prior distributions

The following prior distributions for the biophysical and error parameters were chosen.

Biophysical parameters

Constant or radius-dependent stiffness models. For all models with constant or radius-dependent stiffness (Table 4.1), a rescaled Beta distribution for the biophysical parameters was used to ensure positive support within physiologically realistic ranges $([l_i, u_i])$ [103, 151], $\theta_i \sim \text{Rescaled Beta}(1,1), \quad l_i \leq \theta_i \leq u_i$, where $i = 1, \ldots d$, with d being the parameter dimensionality.

Vessel-specific stiffness in a Bayesian hierarchical model. Different pulmonary arteries may have different vessel wall stiffness values, but since all vessels have similar tissue composition, the parameters are related. A Bayesian hierarchical model [66] is needed to incorporate the prior notion that the vessel stiffnesses are similar, and this model provides a mechanism of information sharing among the vessel stiffness parameters. The dependence of these parameters can be captured by using a common "population" prior distribution, from which each stiffness parameter is sampled. Next, to allow the stiffness parameters to influence each other, a layer of priors for the hyperparameters for the population distribution is introduced. This construct enables the hyperparameters to be variable, ensuring a dependency between the stiffness parameters while the hyperparameters' uncertainty is naturally incorporated into the modelling procedure. The result is a Bayesian hierarchical model (shown in Figure 4.1), which tends to avoid overfitting the existing data by allowing information sharing between the stiffness parameters. This model subsumes two simpler models as limiting cases: the model where all vessels have the same stiffness (when the prior distribution of the vessel-specific stiffness parameter collapses to a delta spike), and the model of independent vessel-specific stiffness parameters without information sharing (when the prior distribution of the stiffness parameters is the uniform distribution).

When employing the hierarchical model, apriori, all parameters are assumed conditionally independent given m_s and σ_s^2, but marginally (after integrating out m_s, σ_s^2) dependent. Additionally, the parameters are assumed to come from a "population" Normal distribution, $\mathcal{N}(m_s, \sigma_s^2)$, and $m_s \sim \mathcal{N}(m^*, \sigma^{2*})$ and $\sigma_s^2 \sim \mathcal{IG}(\alpha^*, \beta^*)$ (where $\mathcal{IG}$ stands for the Inverse-Gamma distribution). The hyper-hyperparameters $m^*, \sigma^{2*}, \alpha^*, \beta^*$ take fixed values and are set such that there is roughly 90% prior probability that the stiffness parameters are within the physiologically plausible range. Since m_s and σ_s^2 are not fixed, the random variables $s_1, \ldots s_d$ are not d-separated (see Section 10.5.1 in [128]), which enables information coupling. The choice of the priors is motivated by conjugacy, and leads to closed-form posterior distributions (more

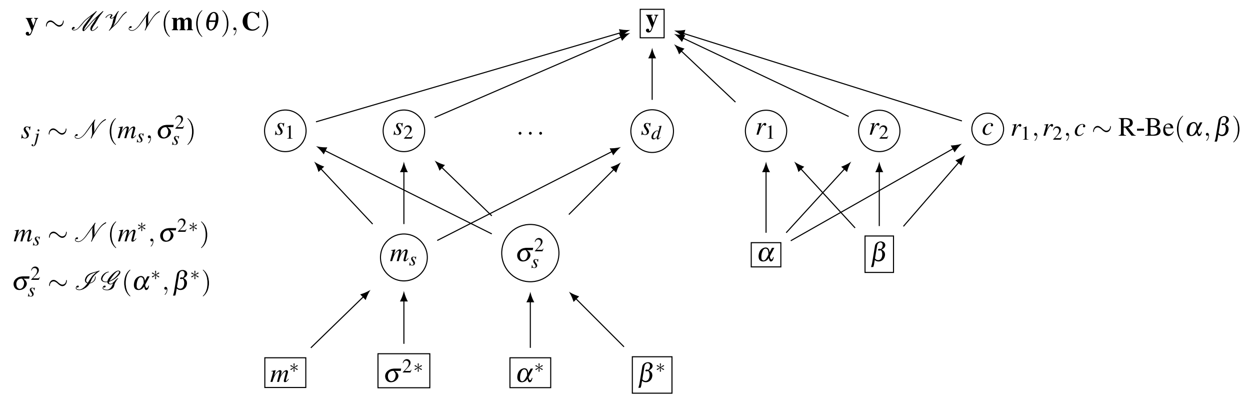

$\mathbf{y} \sim \mathscr{MVN}(\mathbf{m}(\theta), \mathbf{C})$
$\mathbf{y}$
$s_j \sim \mathscr{N}(m_s, \sigma_s^2)$
s_1
s_2
$\dots$
s_d
r_1
r_2
c
$r_1, r_2, c \sim$ R-Be(α, β)
$m_s \sim \mathscr{N}(m^*, \sigma^{2*})$
$\sigma_s^2 \sim \mathscr{IG}(\alpha^*, \beta^*)$
m_s
σ_s^2
α
β
m^*
σ^{2*}
α^*
β^*

Figure 4.1: Bayesian hierarchical model used for vessel-specific stiffness analysis. The data, denoted by $\mathbf{y}$, are assumed to follow a multivariate normal distribution $\mathscr{MVN}$ with mean $\mathbf{m}(\theta)$ and covariance matrix $\mathbf{C}$. If iid errors are assumed (i.e. model mismatch is ignored), $\mathbf{C}$ is a diagonal matrix, $\mathbf{C} = \sigma^2\mathbf{I}$ (where σ^2: error variance and $\mathbf{I}$: identity matrix), and if correlated errors are assumed (i.e. model mismatch is incorporated), $\mathbf{C}$ is a full matrix. The biophysical parameters, $\theta = (s_1, \dots s_d, r_1, r_2, c)$, and the hyperparameters, m_s, σ_s^2, are apriori drawn from the distributions indicated in the graphical model (where $\mathscr{N}$: Normal distribution, $\mathscr{IG}$: Inverse-Gamma, R-Be: rescaled Beta distribution). The circle represents variable quantities, which are inferred using MCMC, and the rectangle stands for fixed quantities. Inference in this model is analytically intractable, thus a Gibbs sampling scheme is employed, as showed in equations (4.11)-(4.12). A modification of this model, where an additional edge is introduced from σ_s^2 to m_s, allows these two parameters to be integrated out in closed form, potentially leading to a more efficient sampling scheme. See Section B.2 for details.

details in Section 4.4.5).

This model assumes exchangeability apriori (see Section 5.2 in [66]), so parameters $(s_1, \dots s_d)$ are exchangeable in their joint distribution, which means that swapping the i^{th} and j^{th} vessel stiffness leads to the same joint distribution. This is a limitation of the current work, as vessel stiffness may depend on the vessel radius. However, the independence of the stiffness from the radius is only assumed apriori, and is most likely overruled aposteriori if a radius dependence exists.

Under this model (Figure 4.1), the three Windkessel parameters r_1, r_2, c are assumed common to all the vessels, and to ensure positive support (within physiological ranges) for them, a rescaled Beta distribution was used.

Error parameters

For the error correlation analysis, the hyperparameters of the GP neural network kernel w, b, were given a log uniform distribution with the range being chosen based on maximising the profile log likelihood (see Section B.4 for more details). For the iid errors analysis, a conjugate weakly informative Inverse-Gamma prior was placed on the error variance: $\sigma^2 \sim \mathscr{IG}(a, b)$, with $a = 0.001$ and $b = 0.001$, leading to an $\mathscr{IG}$ posterior distribution.

4.4.5 Posterior inference with Bayesian methods

The posterior distribution is computed as

$$p(\theta,\phi|\mathbf{y}) \propto p(\mathbf{y}|\theta,\phi)p(\theta,\phi), \tag{4.9}$$

where θ are the biophysical parameters and ϕ are the error parameters.

This study pursues Bayesian inference based on sampling the biophysical parameters from their posterior distribution. This posterior is unavailable in closed form, hence MCMC techniques were employed, i.e. the Adaptive Metropolis (AM) algorithm [82], summarised in Section 2.3.2.

In the correlated errors analysis, the GP neural network hyperparameters were jointly sampled with the biophysical parameters from their joint posterior distribution using the AM algorithm. The noise variance of the residuals was set to a very small value (10^{-6}), as supported by preliminary analysis findings, to avoid numerical instabilities from the matrix inversion $\mathbf{C}^{-1}$ in eq (4.3).

In the iid errors analysis, the noise variance σ^2 was sampled in a Gibbs step [26] from the conditional posterior distribution, $p(\sigma^2|\theta,\mathbf{y})$, which is available in closed form:

$$p(\sigma^2|\theta,\mathbf{y}) = \mathscr{IG}\left(\frac{n}{2}+a, 0.5\sum_{i=1}^{n}(y_i - m_i(\theta))^2 + b\right). \tag{4.10}$$

The choice of the priors in the Bayesian hierarchical model leads to closed-form posterior distributions for the hyperparameters m_s, σ_s^2, see eqns (4.11) and (4.12), which can be sampled using the Gibbs algorithm [26]. Thus, posterior inference in the hierarchical model, summarised in Algorithm 1e, proceeds by iterative sampling from eqns (4.11), (4.12) and (4.13) (see Section B.3 for more details).

$$p(m_s|m^*,\sigma^{2*},\mathbf{s},\sigma_s^2) = \mathscr{N}\left(\frac{\frac{m^*}{\sigma^{2*}} + \frac{1}{\sigma_s^2}\sum_{i=1}^{d} s_i}{\frac{1}{\sigma^{2*}} + \frac{d}{\sigma_s^2}}, \left[\frac{1}{\sigma^{2*}} + \frac{d}{\sigma_s^2}\right]^{-1}\right), \tag{4.11}$$

$$p(\sigma_s^2|\alpha^*,\beta^*,\mathbf{s},m_s) = \mathscr{IG}\left(\alpha^* + \frac{d}{2}, \beta^* + 0.5\sum_{i=1}^{d}(s_i - m_s)^2\right), \tag{4.12}$$

$$p(s,r_1,r_2,c|m_s,\sigma_s^2,\mathbf{y},\alpha,\beta) \tag{4.13}$$

where the sampling of s, r_1, r_2, c cannot be done analytically and follows a Metropolis-Hastings within Gibbs scheme. Also, $d = |\mathbf{s}|$ is the cardinality of the stiffness parameter vector. A slight modification of this Bayesian hierarchical model, with an additional edge introduced in the graph of Figure 4.1 allows m_s and σ_s^2 to be integrated out in closed form, potentially leading to a more efficient sampler. The resulting equations are less intuitive though, and due to space restrictions, the details have been relegated to Section B.2.

The Bayesian hierarchical framework can reveal if vessel-specific stiffness parameters are needed: if $\sigma_s^2 \to 0$, then all stiffness parameters take values close to m_s with high probability, encouraging a common stiffness parameter, while $\sigma_s^2 \to \infty$ will be evidence that the stiffness parameters are statistically independent of each other, implying vessel-specific stiffness values.

4.4.6 Bayesian Model Selection: WAIC

WAIC, described in Section 2.3.8 was used for model selection.

4.5 Simulations

4.5.1 Code

The statistical methods were implemented in *Matlab* (Mathworks, Natick, MA) and simulations were run on a RedHat Enterprise Linux 6 machine with Intel(R) Xeon(R) CPU E5-2680 v2 2.80GHz and 32GB RAM. The simulated pressure waveforms were obtained by numerically solving the PDEs of the fluid-dynamics models in Section 2.2 using a two step Lax-Wendroff scheme [113] implemented in *C++* by Olufsen et al. [139, 151]. The GP models were implemented using the *GPstuff* toolbox [203] and the MCMC convergence dignostics using the MCMC toolbox [107].

4.5.2 Set-up

The simulations were split into two categories. The simulations in the first category use the measured MPA pressure data for all the models summarised in Section 4.3.3 and Table 4.2. Simulations in the second category use synthetic data generated in 1, 3 or 21 vessels from the linear wall model with radius-dependent stiffness and correlated errors created using a neural network kernel.

4.5.3 Computational efficiency

Given the high computational complexity from repeated numerical integrations of the PDEs in the Bayesian analysis, simulations were also set up with a focus on computational efficiency, i.e. MCMC with a GP surrogate (emulator) for the posterior distribution [140] on a few of the models (A, B, H), see Table 4.2. This approach, called the N-steps ahead Adaptive Metropolis with emulation, briefly described in Section B.1 (details in [140]), significantly speeds up computationally expensive simulations, which is essential if the analysis performed here is translated to analysis of a large quantity of human data. Models with a high number of parameters (D, E, F, G, I) used a numerical integration of the PDE model in every step of the MCMC simulation. As shown in Section 4.6, some of the stiffness parameters are unidentifiable, with posterior samples covering the entire prior parameter space. This implies that an emulator would need to cover a large region in the parameter space. While this could potentially be achieved by using a larger number of training points, it was decided to focus less on the development of the surrogate model, and more on the exploration of different mathematical models and hypotheses.

Algorithm 1e Adaptive Metropolis with Gibbs sampling for the hyperparameters of the hierarchical model

```
Define N: number of MCMC samples.
for i=1:N do
    Given $m_s^{(i-1)}$ and ${\sigma^2}_s^{(i-1)}$, use the AM algorithm to draw $(\mathbf{s}^{(i)}, r_1^{(i)}, r_2^{(i)}, c^{(i)})$ from the approximate posterior distribution derived from eqns (4.13) and (4.9) with the likelihood in Section 4.4.1 and priors in Section 4.4.4
    Given $\mathbf{s}^{(i)}$ and ${\sigma^2}_s^{(i-1)}$, draw $m_s^{(i)}$ in a Gibbs step [26] using the conditional distributions in eq (4.11)
    Given $\mathbf{s}^{(i)}$ and $m_s^{(i)}$, draw ${\sigma^2}_s^{(i)}$ in a Gibbs step [26] using the conditional distributions in eq (4.12)
end for
```

Model abbreviation	Model	No. of parameters	f_1 ($\times 10^3$)	f_2	f_3 ($\times 10^4$)	γ	r_1	r_2	c	Model mismatch	w ($\times 10^4$)	b	Emulator
A	linear	4	0	any	(2, 10)	-	(0.05, 2.50)	(0.05, 2.50)	(0.05, 2.50)	no	-	-	+
B	linear	6	0	any	(2, 10)	-	(0.05, 2.50)	(0.05, 2.50)	(0.05, 2.50)	yes	(1, 9)	(1, 500)	+
C	linear	6	0	any	(2, 10)	-	(0.05, 2.50)	(0.05, 2.50)	(0.05, 2.50)	yes	(1, 9)	(1, 500)	-
D	linear	6	(1, 10^4)	(-300, -50)	(3, 6)	-	(0.05, 2.50)	(0.05, 2.50)	(0.05, 2.50)	no	-	-	-
E	linear	8	(1, 10^4)	(-300, -50)	(3, 6)	-	(0.05, 2.50)	(0.050, 2.50)	(0.05, 2.5)	yes	(1, 9)	(1, 500)	-
F	linear	24	0	any	(2, 10)*	-	(0.05, 2.50)	(0.05, 2.50)	(0.05, 2.50)	no	-	-	-
G	linear	26	0	any	(2, 10)*	-	(0.05, 2.50)	(0.05, 2.50)	(0.05, 2.50)	yes	-	-	-
H	non-linear	7	0	any	(3, 50)	(1,2π)	(0.05, 2.50)	(0.05, 2.50)	(0.05, 2.50)	yes	(1, 9)	(1, 500)	+
I	non-linear	9	(5, 10^2)	(-200, 0)	(1, 50)	(1,2π)	(0.05, 2.50)	(0.05, 2.50)	(0.05, 2.50)	yes	(1, 9)	(1, 500)	-

Table 4.2: Models analysed for the measured data: the constitutive models (linear and non-linear, as indicated in Table 2.1) with model parameters $(f_1, f_2, f_3, \gamma, r_1, r_2, c)$ prior ranges. It is indicated whether the model mismatch is incorporated (by yes or no), and if it is, the hyperparameters w and b for the GP model mismatch are given. In the second column from the right, the symbol '+' indicates that the emulator was used to accelerate the MCMC simulations, while '-' indicates that the standard MCMC was used. The stiffness relation used is given in eq (2.5). Legend: * in column f_3 indicates that 90% prior probability has been placed on these bounds as part of a Bayesian hierarchical scheme (Figure 4.1) to infer 21 individual vessel stiffness parameters.

4.6 Results

4.6.1 Importance of correcting for model mismatch

Inference results based on MCMC are compared between the conventional method ignoring model mismatch and the proposed approach, which explicitly incorporates the model mismatch, defined in eq (4.6), with GPs. The results are shown for both synthetic and physiological data. Convergence of MCMC methods (described in Section 2.3.4) was tested using the Geweke test [70] (the p-values from the Z test were greater than 0.05) and the Brooks Multivariate Potential Scale Reduction Factor (MPSRF) [18] (ensuring that MPSRF ≤ 1.1).

Synthetic data

Using model E (Table 4.2) synthetic data with additive, correlated Gaussian errors were generated in the MPA (20 different data instantiations), which mimics the physiological data (as described in Section 4.2.2). Two MCMC simulations were than run: one which incorporates the model mismatch, and another simulation which does not. Parameter estimates obtained from these two simulations are compared to the ground truth parameter values in Table 4.3 using the relative sum of squared errors (SSE),

$$\sum_{i=1}^{d} \left(\frac{\theta_i - \hat{\theta}_i}{\theta_i} \right)^2, \tag{4.14}$$

which is the relative deviation in Euclidean space of the estimated values from the true parameter values. Table 4.3 also shows the median marginal and joint posterior density value of the true parameter vector, $\theta = (f_1, f_2, f_3, r_1, r_2, c)$, under the assumed model, as found from 20 synthetic realisations. To obtain the marginal posterior density of the true parameter vector, the kernel smoothing function estimate for univariate data with the optimal bandwidth for normal densities [14] was used. To check for consistency of the results, the joint posterior density was obtained in two ways: using the multivariate kernel density estimation with the bandwidth estimated with Silverman's rule [180], and using Chib's method (see Section 2.1 in [32]). The parameters were scaled to the same order of magnitude, as both methods were affected by having parameters with different orders of magnitude.

From Figure 4.2, showing the marginal posterior density values of the parameters for three of the 20 data sets, it can be noticed that with the standard method neglecting model mismatch, the ground truth parameter value lies in the tail of the posterior distribution for most cases investigated, and the uncertainty is underestimated. In contrast, with the proposed method allowing for model mismatch, the distribution contains the true parameter and the uncertainty is wider. For the complete set of results, see Table 4.3, which shows that neglecting model mismatch leads to a lower (better) relative SSE for the parameter estimates. However, as seen from Figure 4.2, a small SSE does not rule out the possibility of seriously underestimating the uncertainty. A better measure, to capture both estimation accuracy and UQ, is the marginal posterior density of the true parameters. Here it is obvious that the median marginal posterior density value of the true parameter with the standard method is substantially lower (worse) for the identifiable parameters f_3, r_1, r_2.

Allow for model mismatch	SSE	Median $p(\theta_i\|\mathbf{y})$	Median $p(\theta\|\mathbf{y})$
No	0.02	(9.8e-08, 0.004, 0, 0, 0, 0)	0, 0
Yes	0.03	(9.7e-08, 0.004, 0.0001, 4.36, 5.25, 1.30)	1037, 648

Table 4.3: Results obtained when allowing for or ignoring model mismatch, defined in eq (4.6), on synthetic data generated from model E in Table 4.2 with correlated errors. First row: standard approach ignoring model mismatch; second row: the proposed new method, where a GP mismatch model has been introduced. The relative sum of squared errors (SSE), as well as the median posterior distribution of the true parameter vector, $\theta = (f_1, f_2, f_3, r_1, r_2, c)$, under the assumed model are presented (median calculated from 20 data sets). Marginal and joint posteriors were obtained from the MCMC samples with kernel density estimation (first entry in last column) and Chib's method [32] (second entry in last column). Parameters were scaled to the same order of magnitude.

Physiological data

For the physiological data, Table 4.5 shows comparative results from the MCMC analysis for the nine models explored. Table 4.5 contains the median posterior density value (50th posterior quantile) for each of the models' parameters, and the associated 95% posterior credible interval obtained from the 2.5% and 97.5% quantiles of the MCMC posterior samples. WAIC scores, calculated from 1000 random MCMC samples, are compared to the Euclidean distance (eq (4.2)) obtained with the median posterior parameter value. A lower WAIC score indicates a better model.

Models incorporating the model mismatch, defined in eq (4.6), record a lower (better) WAIC and a higher (worse) Euclidean distance in output space compared to models which ignore it, implying that the former are better supported by the data, and that minimising the Euclidean distance, which is equivalent to minimising the mean squared error (MSE), is a sub-optimal inference procedure. The reason is that the MSE does not take the error correlation into account and does not penalise models for poor UQ. This is illustrated in Figure 4.3 showing that the posterior uncertainty in parameter space is much wider when allowing for model mismatch, which aligns with findings from the synthetic study. Moreover, parameters f_3 and r_1 have different posteriors depending on whether the model mismatch is incorporated (Figure 4.3). Additionally, Figure 4.4 displays the posterior uncertainty in output space for pressures in several vessels using the linear model with constant stiffness ignoring or correcting for the model mismatch (models A and B in Table 4.2). Alike the posterior uncertainty in parameter space, the posterior

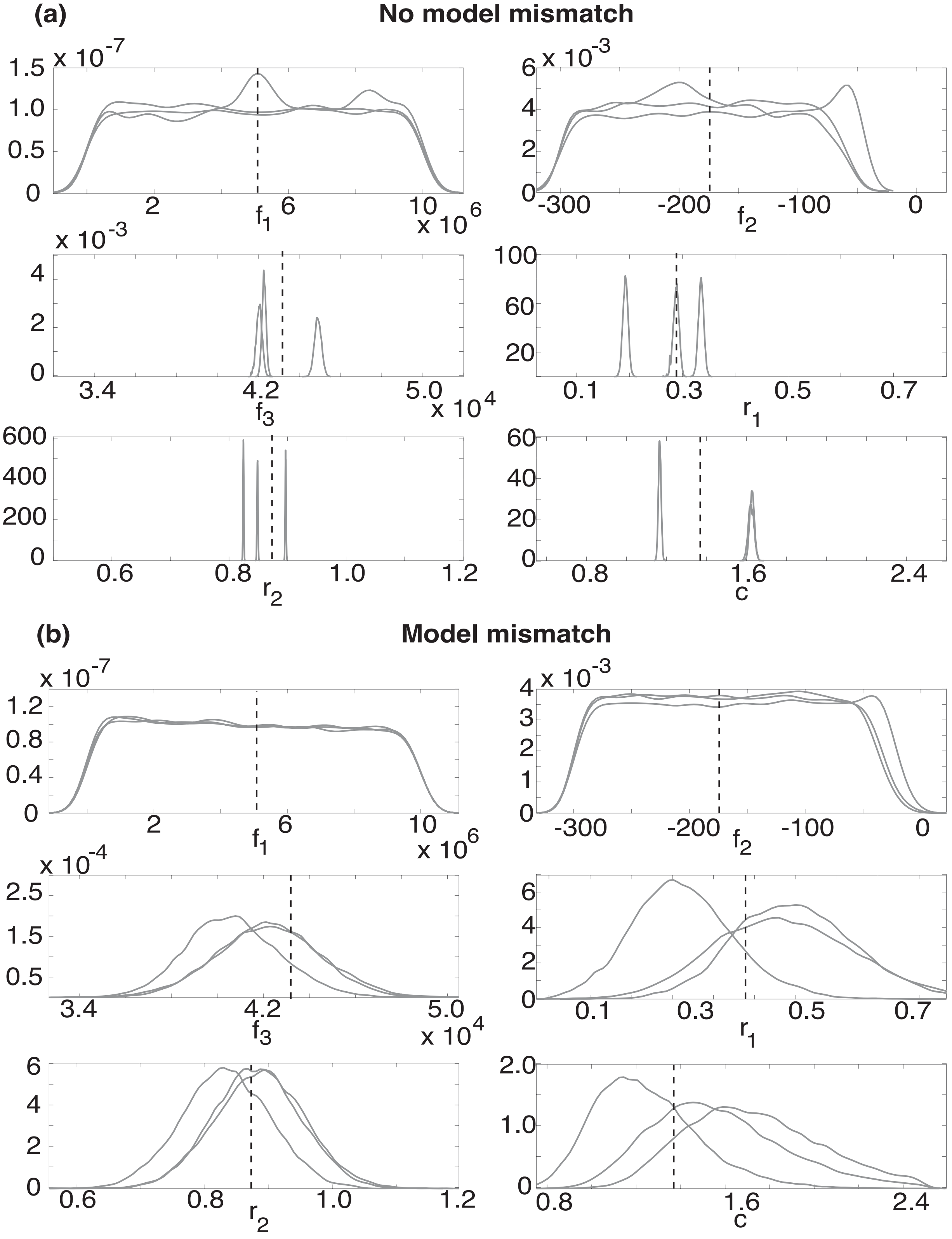


Figure 4.2: Inference results for three synthetic data sets generated from model E in Table 4.2 with correlated errors using (a) the standard method, which ignores model mismatch, defined in eq (4.6), and (b) a GP introduced to allow for model mismatch. Marginal posterior densities for the parameters of the exponential radius-dependent stiffness linear model $(f_1, f_2, f_3, r_1, r_2, c)$, where the stiffness is given by eq (2.5), are shown. The different density per parameter correspond to three random data sets out of 20 (for complete results, see Table 4.3). The black dashed vertical line marks the ground truth parameter values which generated these data.

uncertainty in output space is much wider when correcting for the model mismatch, which is also shown in Table B.1 in the Appendix containing the time-averaged 95% explanatory and predictive credible interval width for the pressure data from every model.

4.6.2 Vessel wall stiffness

Table 4.5 also shows a lower WAIC score for the linear wall model with vessel-specific stiffness relative to the other linear models, which assume constant or radius-dependent stiffness. The exponential radius-dependent stiffness model has the same WAIC score as the constant stiffness model, suggesting that the exponential radius dependence is not consistent with the data, as the f_1 and f_2 parameters are non-influential (their marginal posterior distributions are uniform on the prior range), see the left panel of Figure 4.5. In addition, the 21 stiffness model reveals that the median posterior stiffness values are somewhat similar for most of the 21 vessels (right panel of Figure 4.5). However, the stiffness becomes increasingly variable for small-radius vessels, which is also evident from the 95% credible interval width presented in Section B.6. This suggests that the Bayesian hierarchical model should allow for vessel-specific variance, thus the common variance σ_s^2 in Figure 4.1 should be replaced by a variance-covariance matrix. This method extension will lead to a substantial increase in the computational complexity due to higher parameter space dimension, and thus, it is subject to future work. Almost all the stiffness values are of 10^4 order, which is the regime where the MPA pressure is sensitive to changes in stiffness (see Figure 4.6). This figure depicts the the systolic, diastolic and pulse pressures (systolic minus diastolic pressure) against the constant stiffness f_3 for the linear wall model. It is clear that beyond a certain threshold (3×10^5 mmHg), the change in pressure is minimal, as the vessels become very rigid. The plot is produced with a set of Windkessel parameters consistent with the physiological data. In addition, a slight radius-dependent stiffness is needed for the non-linear model, as the term $f_2 r_0$ in the stiffness expression $f_1 \exp(f_2 r_0) + f_3$ is close to 0 (compare prior range in Table 4.2 to posterior uncertainty interval in Table 4.5).

4.6.3 Vessel wall model

Next the linear and non-linear wall models are compared. The WAIC scores in Table 4.5 indicate that the non-linear wall model outperforms the linear wall model. More specifically, results suggest that the non-linear model with radius-dependent stiffness is preferred, as it registers the lowest WAIC score.

Based on the WAIC scores in Table 4.5 the conclusion is that out of all the models investigated, the best linear wall model is that with vessel-specific stiffness, and the best non-linear model is that with an exponential radius-dependent stiffness, incorporating the model mismatch in both models. The non-linear model has not been applied with vessel-specific stiffness due to the interaction between the parameters p_1 and γ in eq (2.4), requiring vessel-specific (p_1, γ).

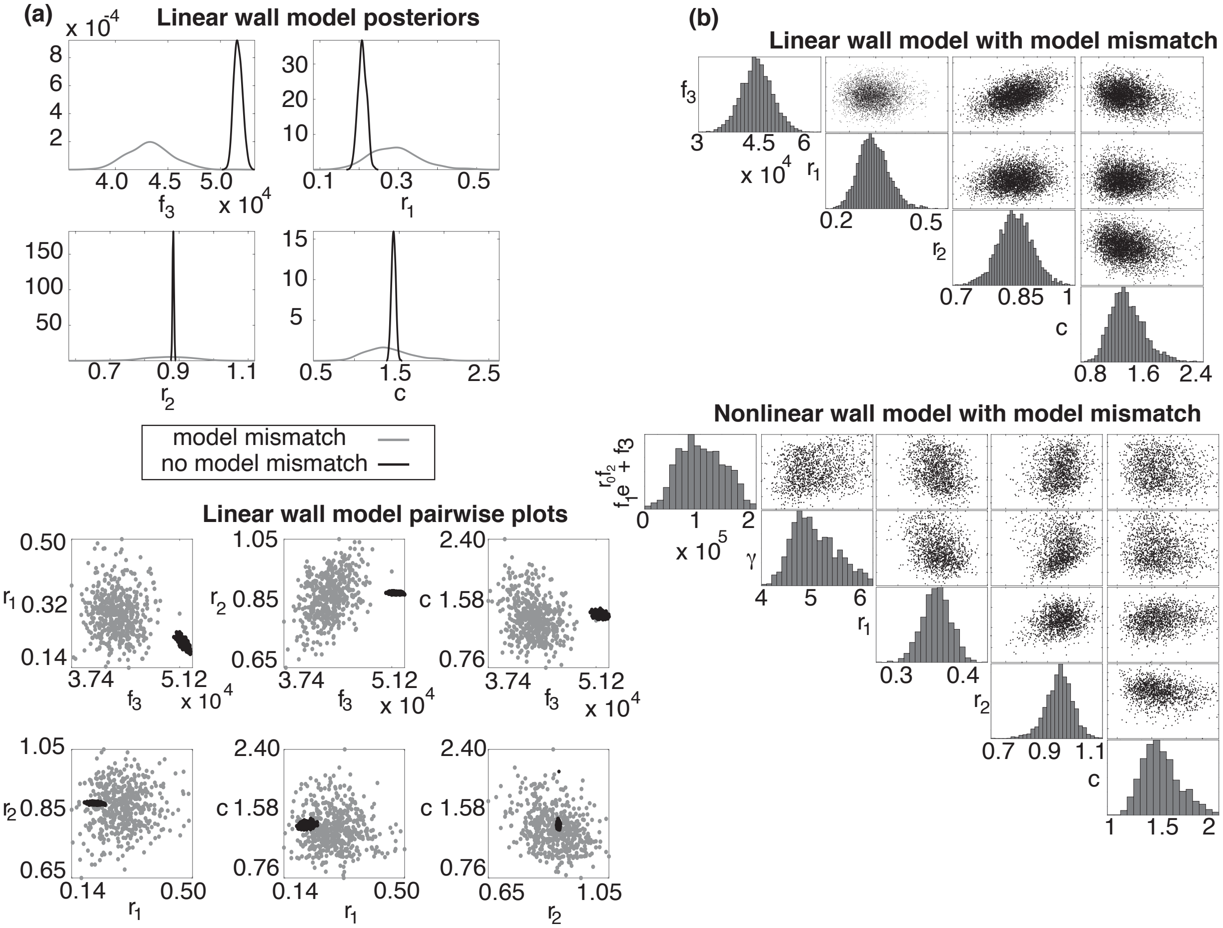


Figure 4.3: (a) Marginal posterior distributions (top) and pairwise scatterplots (bottom) of the posterior sample (obtained with MCMC) for the constant stiffness linear wall model with the standard method ignoring the model mismatch (eq (4.6)), (black) vs the proposed GP mismatch model (grey), i.e. models A and B in Table 4.2. (b) Pairwise scatterplots between the MCMC posterior parameter samples of the linear model with constant stiffness and model mismatch (top), i.e. model B in Table 4.2, and non-linear model with radius-dependent stiffness and model mismatch (bottom), i.e. model I in Table 4.2. For the non-linear model, $s(f_1, f_2, f_3)$ in eq 2.5 is expressed instead of individual parameters f_1, f_2, f_3 due to parameter identifiability issues – see Section 4.6 for a discussion on this. Here the distribution of $s(f_1, f_2, f_3)$ in eq (2.5) is shown for radius r_0 corresponding to vessel 1, the MPA, but the pattern of the distribution is similar for the other vessels.

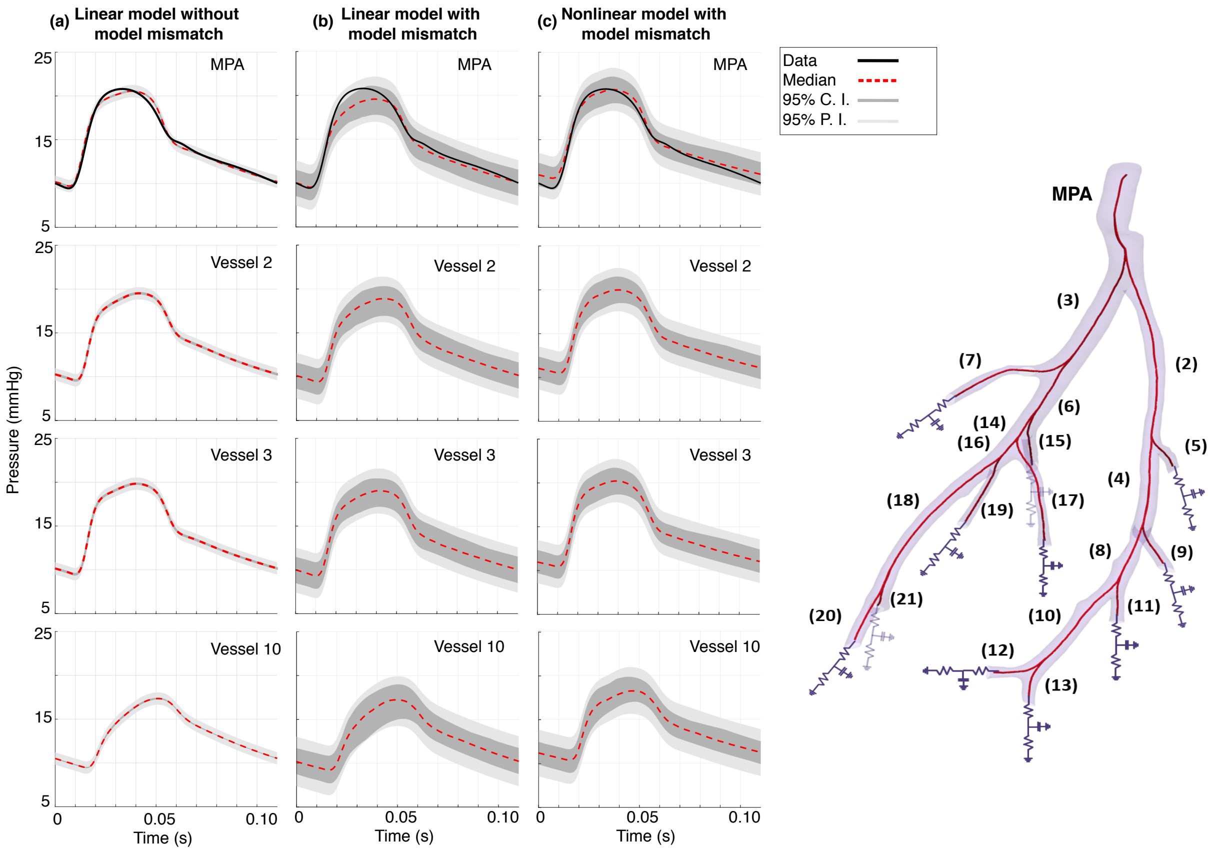


Figure 4.4: 95% credible intervals (C.I.) and prediction intervals (P.I.) for the pressure prediction from the linear model with one stiffness and ignoring model mismatch in eq (4.6) (left) (model A in Table 4.2), the linear model with one stiffness and model mismatch correction (centre) (model B in Table 4.2), and the non-linear model with radius-dependent stiffness and model mismatch correction (right) (model I in Table 4.2), obtained from MCMC posterior samples. The measured pressure data in the MPA and the median prediction are superimposed, and plots in three other vessels are shown (for all the other vessels see Figures B.5, B.6 and B.7).

This would lead to a very large number of parameters being estimated, requiring extremely high computational efforts (simulations would most likely take months to complete).

4.6.4 Model fits

The model fits for all the models analysed (Figures 4.7 and 4.8) are investigated. The first observation is that the median pressure predictions obtained using the MCMC-simulated posterior parameter values (Figure 4.7) are qualitatively similar for all nine models investigated. Pressure predictions in the MPA are compared between all models, and they all produce a waveform similar to the measured data. The best linear model (model G) fits the measured data better in the diastolic phase, but gives a peak shift in the systolic phase (Figure 4.4). On the other hand, the best non-linear model (model I) provides a better fit in the systolic phase, but has a slight discrepancy in diastole. Generally, the pressure increases more steeply in the systolic phase for the non-linear model compared to the linear model. Figure 4.7 also shows that the pressure

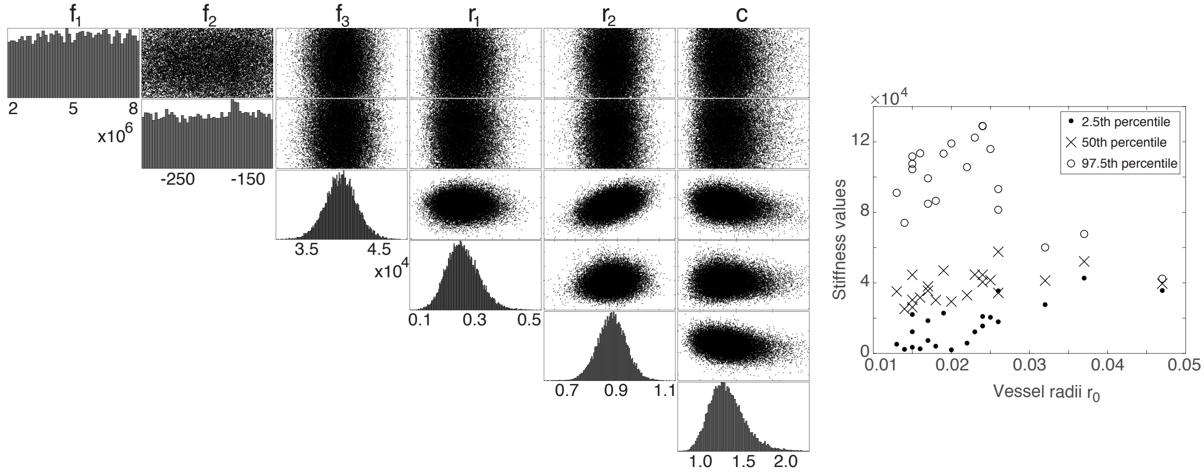


Figure 4.5: Left panel: MCMC results (marginal posterior densities and scatterplots) based on the linear wall model with exponential stiffness (eq (2.5)) and correcting for model mismatch (defined in eq (4.6)), for measured data. Right panel: MCMC results for 21 individual stiffness values corresponding to every vessel radius r_0 for the linear wall model and correcting for model mismatch. Here the posterior median value for each vessel stiffness parameter is shown, as well as the 2.5$^{\text{th}}$ and 97.5$^{\text{th}}$ quantiles.

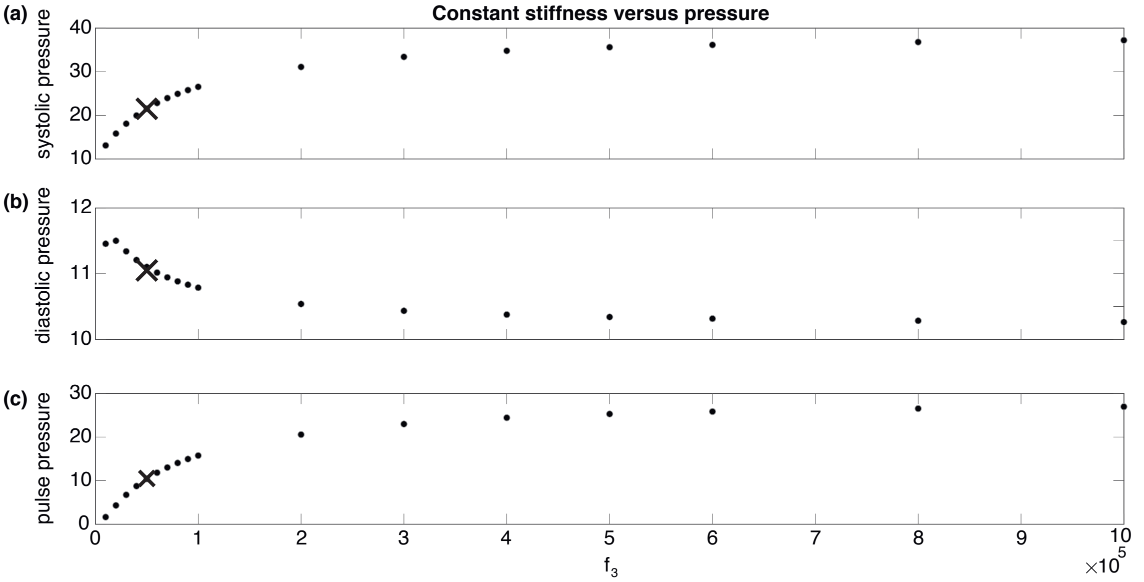


Figure 4.6: Relation between the radius-constant stiffness and the systolic, diastolic and pulse pressure for the linear wall model. Here f_3 is varied within the range $[10^4, 10^6]$, f_1 is set to 0, f_2 can take any value in the exponential radius-dependent expression in eq (2.5), and the Windkessel parameters r_1, r_2, c are kept fixed to 0.3, 0.97, 1.23, which are plausible values for the measured data. A similar trend is observed when the Windkessel parameters are fixed to other values, or when the non-linear wall model is used. The cross point marks the f_3 stiffness value estimated from the measured data using the constant stiffness linear model.

predicted with the non-linear model is slightly higher than that predicted with the linear model for all 21 vessels. In addition, the predictions obtained with the linear model with 21 individual stiffnesses while ignoring model mismatch (model F) provides fits similar to the other models in the proximal arteries, but predicts downstream pressure waves with slightly different shape and increased oscillatory behavior. This suggests that assuming a vessel-specific stiffness and ignoring the model mismatch provides poor extrapolation performance. When analysing the median flow predictions obtained from the parameter posteriors (Figure 4.8), it can be noticed that all are again very similar in shape, except for the outlier model F. An unequal flow distribution between the right and left side of the tree for all the models except model F is observed.

Figure 4.8 shows pressure-area relations obtained by using the posterior median parameter values. The best non-linear model (model I) consistently predicts larger areas than the best linear model (model G) for the large proximal vessels (see predictions in MPA and vessels 2 and 3), and the opposite trend is observed for most terminal vessels (see predictions in Figure B.4). Furthermore, the non-linear model with constant stiffness gives systematically larger area values than the non-linear model with radius-dependent stiffness except in vessels 1 and 3, which aligns with the former model having a smaller stiffness than the latter (Table 4.5). The linear model with 21 individual stiffnesses while ignoring model mismatch gives drastically different results than the other models for some of the vessels (e.g. vessels 4, 8, 12 in Figure B.4), which further indicates that this model can lead to drastic changes in downstream predictions.

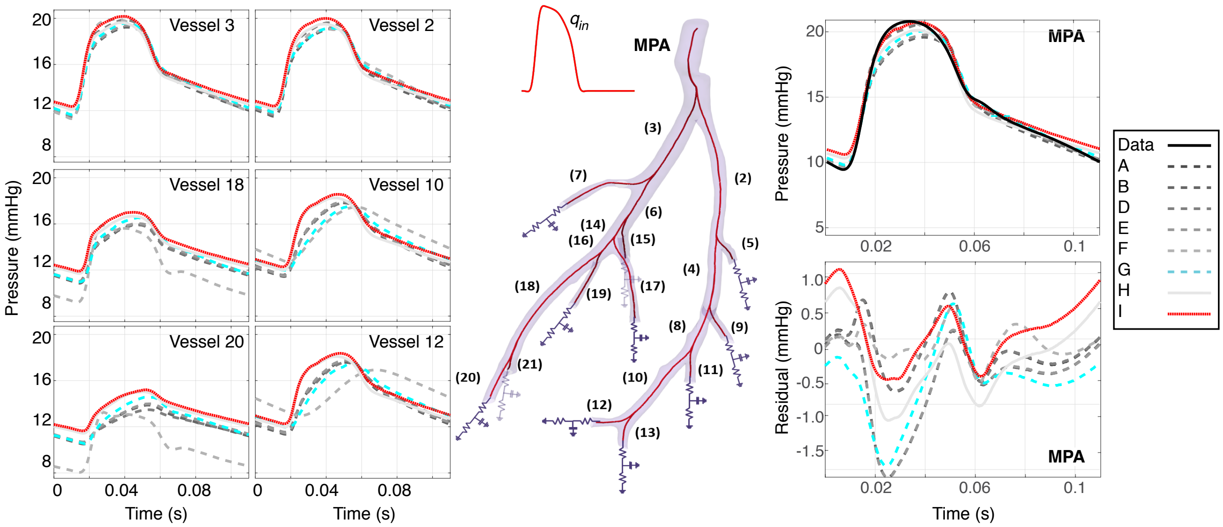


Figure 4.7: Pressure predictions obtained using the MCMC posterior samples for the parameters from all the models considered – see Section 4.3.3 and Tables 4.2 and 4.5 for a summary of the models, which are denoted by A-I in the figure legend. The median pressure signal for seven of the 21 blood vessels in time is shown (see Figure B.2 for all the other vessels). The measured pressure data in the MPA is superimposed (top right). Examples of pressure residuals, that is, the difference between the predicted and measured blood pressure, are shown in the bottom right panel.

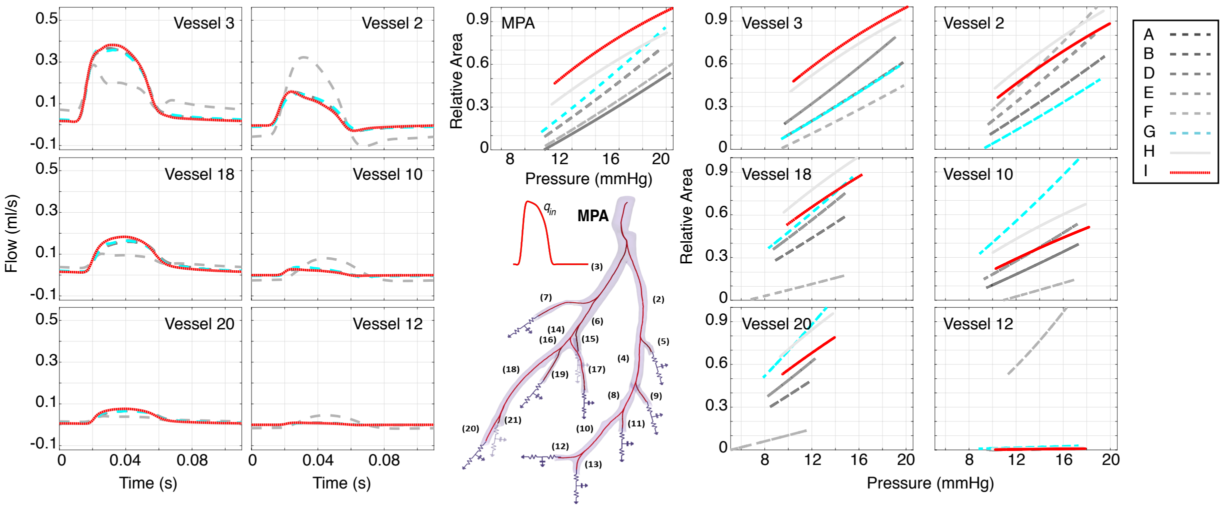


Figure 4.8: Flow (left side) and pressure-area (right side) predictions obtained using the MCMC posterior sample for the parameters from all the models considered – see Section 4.3.3 and Tables 4.2 and 4.5 for a summary of the models, which are denoted by A-I in the figure legend. The median flow predictions and pressure versus standardised cross-sectional area predictions are shown for seven of the 21 blood vessels (see Figures B.3 and B.4 for all the other vessels). The area, A_i is standardised per vessel i to lie between [0,1] using the expression: $\frac{A_i - l_i}{u_i - l_i}$, where l_i, u_i are the maximum and minimum area value for vessel i, listed in Figure B.4.

4.6.5 Parameter posteriors

The posterior correlations and the marginal posterior distributions are investigated for the linear models A and B and non-linear model I (Figure 4.3). The marginal posterior distributions have one clear mode and correlations between the parameters are negligible for the linear models. When the non-linear model is used, $s(f_1, f_2, f_3)$ in eq (2.5) is plotted rather than the individual f_1, f_2, f_3 parameters, since the term $f_2 r_0$ is close to 0 (compare prior range in Table 4.2 to posterior uncertainty interval in Table 4.5), leading to unidentifiability of f_1 and f_3.

4.6.6 Future experimental design

It is tested by a synthetic study if parameters f_1 and f_2 in the linear model with exponential radius-dependent stiffness eq (2.5) become influential as complementary data from downstream vessels are added. Synthetic data from this model (model E in Table 4.2) were generated, and, as described in Section 4.2.2, additive correlated Gaussian errors were added to them. Thus 20 data sets were created with different error instantiations, and MCMC was applied to infer the data-generating parameter values. Figure 4.9 shows the agglomerated MCMC posterior distributions from all 20 data instantiations, and the true parameter values were superimposed (if the inference procedure is correct, the peak of the agglomerated distributions should coincide with the true parameter values). This is purely for visualisation purposes, since agglomerated results over different data sets is a non-conventional Bayesian approach. For a fully Bayesian approach, the marginal posterior distribution was calculated, as well as the joint posterior distribution of the

Data	Median $p(\theta_i \mid \mathbf{y})$	Median $p(\theta \mid \mathbf{y})$
1 vessel	(9.8e-08, 0.004, 0.0001, 4.30, 5.22, 1.30)	2.8e+03
3 vessels	(9.9e-08, 0.004, 0.0003, 6.78, 9.61, 2.61)	3.4e+04
21 vessels	(9.9e-08, 0.004, 0.001, 38.4, 27.0, 6.39)	3.9e+06

Table 4.4: Inference results obtained using synthetic data, to which additive, correlated Gaussian errors were added, from 1 vessel (MPA), 3 vessels (MPA and its two daughter vessels) and all 21 vessels. The model mismatch was included in the analysis, and the data were generated using the linear wall model with exponential stiffness, $s(f_1, f_2, f_3)$, given in eq (2.5). The median marginal and joint posterior density, of the true parameter vector, $\theta = (f_1, f_2, f_3, r_1, r_2, c)$ are presented for each of the three scenarios (median calculated from 20 data sets). Joint and marginal posteriors were computed using the MCMC samples with kernel density estimation. Parameters were scaled to the same order of magnitude.

true parameter for each of the data sets, and the median over the data sets was found, as shown in Table 4.4. For the joint posterior distribution the multivariate kernel density estimation was used. Figure 4.9 shows that the peak of the agglomerated distributions aligns with the ground truth parameter values for the influential parameters, which provides validation of the inference procedure. Even with data from more than one vessel (3 or 21 vessels), f_1 and f_2 parameters remain non-influential (close to uniform marginal posterior density). The uncertainty for all the other parameters (f_3, r_1, r_2 and c) is reduced and the distributions become increasingly focused around the true parameter values as more complementary data are used. Additionally, in Table 4.4 it is quantified how the marginal and joint posterior density values of the true parameters increase with the amount of vessel data.

4.6.7 Accuracy of emulator

The accuracy of the emulation approach results (model B) has been checked by a comparison with the conventional method (model C), confirming that the emulator does not introduce any bias in the results.

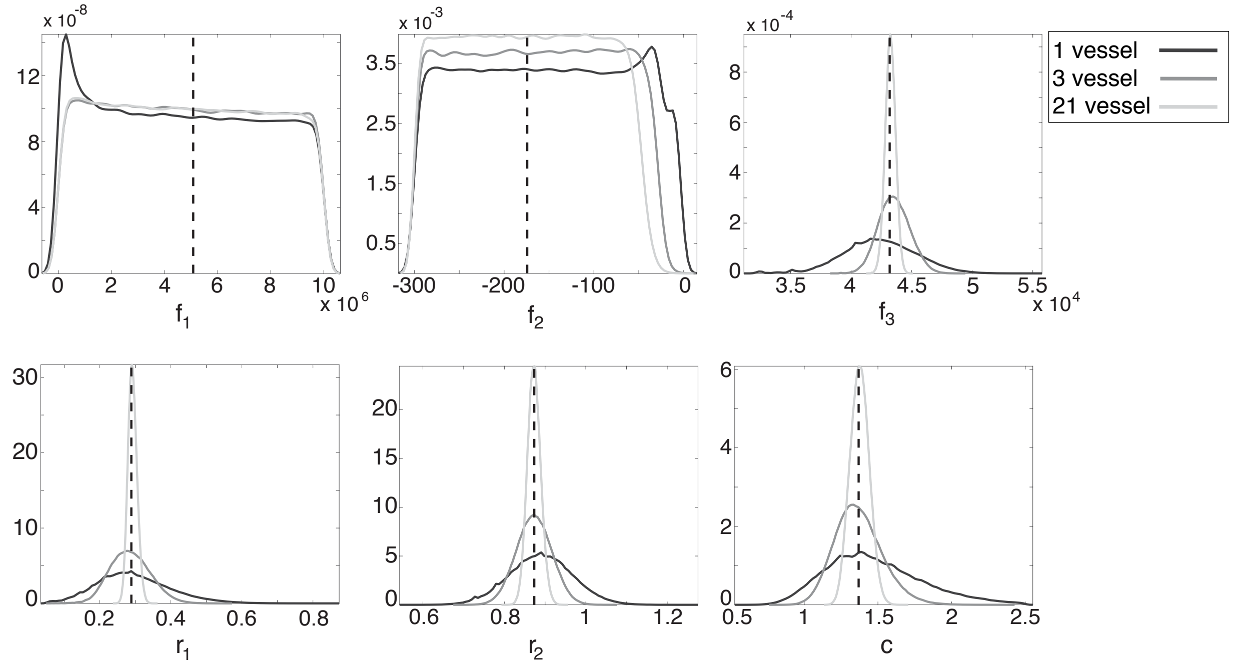


Figure 4.9: Synthetic data results obtained by agglomeration of MCMC posterior samples over 20 data instantiations. Marginal posterior densities for the parameters $f_1, f_2, f_3, r_1, r_2, c$ of the linear wall model with exponential stiffness, $s(f_1, f_2, f_3)$ given in eq (2.5) are shown. Results for 3 simulations are superimposed: one which uses synthetic data from 1 vessel (MPA) for inference – dark grey line, a second one which uses data from 3 vessels (MPA and its 2 daughter vessels) – medium grey line, and a third one which uses data from all 21 vessels – light grey line. The black dashed vertical line marks the ground truth parameter values which generated these data.

Model Abbreviation	Model	f_1 $(\times 10^5)$	f_2	f_3 $(\times 10^4)$	γ	r_1	r_2	c	Model mismatch	w $(\times 10^4)$	b	WAIC	Euclidean distance	Emulator
A	linear	0	any	5.17 (5.09, 5.25)	-	0.21 (0.19, 0.23)	0.88 (0.88, 0.89)	1.44 (1.39, 1.49)	no	-	-	408	66	+
B	linear	0	any	4.31 (3.91, 4.72)	-	0.28 (0.18, 0.42)	0.87 (0.73, 1.00)	1.35 (0.98, 1.96)	yes	5.36 (4.32, 7.09)	137 (102, 197)	-4515	551	+
C	linear	0	any	4.31 (3.91, 4.71)	-	0.29 (0.18, 0.41)	0.87 (0.73, 1.00)	1.34 (0.97, 1.94)	yes	5.41 (4.27, 7.09)	138 (101, 198)	-4515	571	-
D	linear	52.6 (2.59, 97.8)	-162 (-293, -51.1)	5.17 (5.10, 5.25)	-	0.21 (0.19, 0.23)	0.89 (0.88, 0.89)	1.45 (1.40, 1.50)	no	-	-	397	64	-
E	linear	51.1 (2.80, 97.4)	-171 (-293, -56.6)	4.32 (3.91, 4.72)	-	0.29 (0.18, 0.42)	0.87 (0.74, 1.01)	1.34 (0.96, 1.93)	yes	5.39 (4.28, 7.08)	138 (102, 200)	-4515	552	-
F	linear	0	any	*	-	2.45 (2.33, 2.50)	0.21 (0.17, 0.26)	2.08 (0.11, 2.48)	no	-	-	-107	26	-
G	linear	0	any	*	-	0.46 (0.19, 1.05)	0.86 (0.72, 1.10)	1.12 (0.45, 1.86)	yes	4.65 (3.72, 6.25)	161 (115, 262)	-4522	242	-
H	non-linear	0	any	9.17 (6.66, 12.1)	5.18 (4.38, 6.07)	0.37 (0.27, 0.47)	0.94 (0.80, 1.06)	1.60 (1.15, 2.27)	yes	4.91 (3.97, 6.44)	178 (126, 286)	-4522	384	+
I	non-linear	0.58 (0.20, 0.97)	-6.00 (-18.0, -2.13)	2.00 (1.09, 3.40)	5.09 (4.43, 6.13)	0.34 (0.24, 0.43)	0.97 (0.82, 1.09)	1.58 (1.16, 2.27)	yes	4.45 (3.62, 5.76)	196 (130, 320)	-4530	417	-

Table 4.5: Summary of MCMC results on measured data for the constitutive models considered (linear and non-linear, as indicated in Table 2.1) with model parameters $(f_1, f_2, f_3, \gamma, r_1, r_2, c)$. Whether model mismatch, defined in eq (4.6), was incorporated is indicated by yes or no, and for yes, the parameters w and b for the GP model mismatch are given. The right-most column shows whether emulation was used ('+' is yes, while '-' is no). 5000 MCMC iterations were run for the models using emulation (models A, B and H); 300,000 for the vessel-specific stiffness models not using emulation (models F and G); and 150,000 MCMC iterations for the rest of the models not using emulation (models C, D, E, I). The median posterior distribution value and the 95% credible interval from the posterior distribution are shown, as well as the WAIC score calculated from 1000 MCMC samples and the Euclidean distance obtained from the posterior median parameter values. If 21 individual stiffness parameters were inferred, marked by * in the table, the stiffness values are listed in the Appendix (Section B.6).

4.7 Discussion

In this study, several mathematical models of the pulmonary circulation are explored: a linear and a non-linear wall model with different vessel wall stiffness assumptions, and two error models capturing or ignoring the model mismatch, as defined in eq (4.6), and Bayesian analysis is used to find the model that can best predict the measured MPA blood pressure, while providing UQ associated with that pressure prediction. The validity of the parameter inference procedure is tested by a synthetic study.

4.7.1 Importance of correcting for model mismatch

Neglecting the model mismatch by obtaining point estimates based on MSE minimisation, biases parameter estimates and underestimates uncertainty in parameter and output space. This finding is based on synthetic data, for which the gold standard is known, and it tallies with results from the physiological data. The model mismatch is a consequence of making the wrong assumption about the measurement errors (i.e. iid for correlated measurement errors) and not allowing for model discrepancy between the real system and the mathematical model. This study proposes to use a method based on GPs to assess the model mismatch, which circumvents the limitations outlined above. Figure 4.2 clearly illustrates that the uncertainty in parameter space is under-dispersed when the standard method neglecting the model mismatch is used, and the true (data-generating) parameter values lie in the tail of the posterior distribution for most data sets; however, this is not the case for the proposed method of model mismatch. This is in line with results from the measured data, as evident in panel (a) of Figure 4.3, that also shows very narrow uncertainty bounds in parameter space and in output space (Figure 4.4). Moreover, the model selection criteria (WAIC in Table 4.5), clearly and consistently favouring the models which correct for the model mismatch, further strengthens this statement.

Most studies in the literature rely on minimising the Euclidean distance (i.e. MSE) in eq (4.2), which implicitly ignores the model mismatch. This approach is equivalent to maximising the likelihood in eq (4.1) under the assumption of additive Gaussian iid noise. However, in the presence of a model mismatch, the estimates that minimise the MSE are different from the estimates which maximise the likelihood in eq (4.3). The current work demonstrates that using the standard approach ignoring model mismatch, leads to biased point estimates, thus incorrect predictions, and uncertainty underestimation. Wider uncertainty bounds in output space, as seen in Figure 4.4, reflect more adequately variations in pulmonary pressure due to the natural inter-subject factors (e.g., effects of the respiratory cycle). These are well known [35] and should be contained within the uncertainty bounds of the model.

In this work, due to the limited data, the measurement and model errors are jointly captured with a single GP mismatch model. In principle, these two contributions could be disentangled by the use of a strongly informative prior on the model discrepancy function or data measurement process [20], however, this information is not available.

The current study is one of the first to focus on parameter estimation in cardiovascular modelling that has explicitly incorporated the model mismatch. A notable exception is Lei et al. [115] who explore model discrepancy in cardiac electrophysiology, and the authors also show through synthetic studies that ignoring the model-form uncertainty produces biased predictions and uncertainty underestimation, which agrees with the findings here.

4.7.2 Vessel wall stiffness

Results suggest that a linear wall model with vessel-specific stiffness outperforms, according to WAIC, all the other linear models assuming constant or exponential radius-dependent stiffness. As expected physiologically, an increased wall stiffness leads to increased systolic and pulse pressures (Figure 4.6), with more dynamic changes in these values occurring at a lower stiffness range. Table 4.5 shows that the linear models predict stiffness values within this range, suggesting accurate depiction of healthy haemodynamics, regardless of model type. Results indicate that the estimation of individual stiffnesses in a model mismatch framework (right panel of Figure 4.5) is best supported by the MPA-measured data in the context of the linear wall model, and that an exponential radius dependence of the stiffness is inconsistent with the data, as the exponential stiffness parameters are non-influential, obtaining a nearly flat posterior (Figures 4.5 and 4.9). Additionally, the model selection results support a slight radius dependence stiffness in the non-linear model (a vessel-specific stiffness for this model is not pursued as it requires vessel-specific (p_1, γ) parameters (see eq (2.4)), leading to a large number of parameters being estimated, hence extremely high computational efforts). Previous investigations [195] have shown that both wall thickness (h) and tissue properties (E) are drastically different in pulmonary arteries in pulmonary hypertension. This encourages future investigations into whether the model selection results are consistent in specimens with pulmonary hypertension.

4.7.3 Vessel wall model

A further finding, based on WAIC scores, is that the non-linear wall model is better supported by the physiological data compared to the linear wall model. Results indicate that, out of all the models investigated, the model that is most likely under the data is the non-linear model with a slight dependence on the vessel radius (the term $f_2 r_0$ in the stiffness expression $f_1 \exp(f_2 r_0) + f_3$ is close to 0, see Table 4.5). This finding agrees with other studies in the literature – e.g. the study by Valdez et al. [198] on pressure area dynamics in systemic arteries of control sheep and the study by Pilhwa et al. [114] analysing distensibility of pulmonary arteries in control mice. The study in [207] provided experimental stress-strain relations in control and hypoxic pulmonary arteries, illustrating a predominant viscoelastic effect and further suggesting that a non-linear elastic wall is more appropriate for modeling pulmonary haemodynamics.

4.7.4 Model fits

The pressure predictions shown in Figure 4.7 deviate from model to model in arteries distal to the left and right pulmonary artery (vessels 2 and 3, respectively). While predictions may look qualitatively similar, it is clear that the model used can lead to significant changes in downstream predictions. An understanding of how model type affects predictions down the pulmonary arteries is critical for future use of mathematical models in disease prognostication. For instance, pulmonary diseases like pulmonary hypertension remodel smaller arteriolar segments initially, making vessel stiffness a critical parameter in the development of disease [97, 116, 219]. The flow and pressure-area graphs show a more dramatic change between model types, which is to be expected as distal flow and dynamic area data are not available. This variability is important when considering the effects of blockages, i.e. pulmonary embolism, that can lead to obstructions in the pulmonary arteries, limiting perfusion to the alveoli for blood re-oxygenation. The pressure-area relations in Figure 4.8 show that the inferred parameters for the non-linear wall-model provide a nearly linear pressure-area curve, contrary to the findings in [207]. It is expected that the addition of dynamic area data in the likelihood function will lead to a bigger dissimilarity between the two wall models, and should be considered in future studies.

4.7.5 Parameter unidentifiability

The analysis of the linear model with exponential radius-dependent stiffness shows that using complementary data from vessels beyond MPA does not resolve the unidentifiability of some of the parameters (see Figure 4.9). Thus, additional pressure data do not carry information about the non-influential parameters. If the model has structural unidentifiabilities, subsequent predictions are unreliable, and can lead to spurious diagnoses or sub-optimal treatments [165, 166]. For this reason, it is imperative that in the exponential radius-dependent models, the entire expression $s(f_1, f_2, f_3)$ in eq (2.5) is interpreted, and not the individual parameters, f_1, f_2, f_3.

4.7.6 Future experimental design

The analysis further reveals that when complementary data are used, the parameter values are more accurately estimated (Figure 4.9) and a quantification of the uncertainty reduction in parameter space is provided (Table 4.4). These findings may be used in future experimental design, when deciding whether to record measurements in vessels beyond MPA. Furthermore, results in Figure 4.9 show that the gold standard parameter values are accurately inferred, validating the inference procedure.

4.7.7 Real-time treatment planning

A long-term goal of this project is real-time, personalised treatment planning. Therefore, once the model selection procedure finds the "best" model, predictions from that model should be computationally efficient. This work shows that this can be accomplished using efficient surrogate models in place of the computationally expensive PDE model (see Table B.2 in the Appendix).

4.8 Limitations and future directions

The WAIC model selection results based on which the non-linear model is preferred over the linear model is only valid using the MPA-measured data in the particular control mouse analysed. Moreover, it would be interesting to compare the performance of the asymptotically-based WAIC approach for model selection to an approach which does not rely on asymptotics, e.g., marginal likelihood [62], whose calculation, however, comes at a significantly higher computational cost. Additionally, as discussed in [20], the correction for model mismatch can be improved by using more informative priors than the standard GP smoothness prior that was applied in the current study. Mathematical model improvement is an ongoing process, with models becoming ever more detailed and complex. Multi-scale vessel wall models that include fluid-tissue interactions at individual cell level may be too complex for inference, but could also help refine prior knowledge. Running forward simulations with both high and medium fidelity models for space filling design in parameter space and then fitting a GP to the differences in output space can give a more realistic prior for future inference applications.

This study only analyses two parameter types, the vessel stiffness and boundary conditions. Several vessel wall models were investigated, but only one boundary condition model, estimating scaling factors that adjust Windkessel parameters in each terminal vessel. However, these methods can be carried out using other boundary conditions, such as the structured tree model [139, 152, 219]. The process used for analysing the stiffness in 21 vessels could also be applied to a more detailed analysis of vessel-specific boundary condition parameters. Moreover, there are several sources of uncertainty unaccounted for in the analysis: network geometric parameters (vessels' radii and lengths), network connectivity (location of vessel bifurcations and trifurcations), and network size (number of vessels), which use fixed values from the image segmentation process [38] (the uncertainty in the network is quantified in Chapter 5). Additionally, the MPA inflow boundary condition could be replaced by a coupling of the MPA with a right ventricle model [130].

4.9 Conclusions

The focus of this study was on parameter inference and UQ using state-of-the-art Bayesian analysis techniques in a 1D fluid-dynamics model of the pulmonary circulation, with geometry extracted from micro-CT images.

An important contribution is the thorough exploration of several mathematical models (a linear and a non-linear wall model with different vessel wall stiffness assumptions: constant, radius-dependent or vessel-specific stiffness), and error models (via the inclusion of a model mismatch). A Bayesian model selection tool (WAIC) was implemented to find the model that can most accurately predict the MPA pressure and provide the UQ in the pressure predictions.

This study clearly demonstrates that the widely used approach focusing on least-squares fit, thus ignoring the model mismatch, biases parameter estimates and model predictions, and underestimates the uncertainty in parameter and output space. These issues were circumvented by incorporating the model mismatch using GPs.

Additionally, results indicate that the MPA-measured pressure data best supports the non-linear wall model with a weak exponential radius-dependent stiffness.

Lastly, the synthetic study validates the inference procedure, and demonstrates that utilising complementary data distal to the MPA increases the parameter estimation accuracy. Additionally, the analysis quantifies the uncertainty reduction when complementary data are used, which may help better design future experiments.

Chapter 5

Uncertainty of network geometry, connectivity and size

Computational fluid dynamics (CFD) models are emerging tools for assisting in diagnostic assessment of cardiovascular disease. Recent advances in image segmentation has made subject-specific modelling of the cardiovascular system a feasible task. Uncertainty in image segmentation propagate to CFD model predictions, making quantification of segmentation-induced uncertainty crucial for subject-specific models. This study quantifies the variability of one-dimensional (1D) CFD predictions by propagating the uncertainty of network geometry and connectivity to blood pressure and flow predictions. Multiple segmentations of a single excised mouse lung using different pre-segmentation parameters are analysed. A custom algorithm extracts vessel length, vessel radii, and network connectivity for each segmented pulmonary network. Probability density functions are computed for vessel radius and length and then sampled to propagate uncertainties to haemodynamic predictions in a fixed network. In addition, the uncertainty of model predictions to changes in network size and connectivity is computed.

Note: This chapter is adapted from a study by Colebank et al. [37] on which I am the second author. Here I present methods implemented by myself (GP methods in the inverse uncertainty quantification analysis, for which I take full responsability) and collaborators (everything else), to which I actively participated through discussions in weekly Skype meetings and in a 2-week visit to NC State University. Partial results are presented, with an emphasis on results that I obtained. The reader is referred to the original paper [37] for full details and results.

5.1 Introduction

Definitive diagnosis of pulmonary hypertension (PH), defined as a mean pulmonary arterial blood pressure ≥ 25 mmHg, requires a series of medical tests including invasive right-heart catheterization and non-invasive computed topography (CT) imaging of the heart and lungs [110]. Diagnostic protocols interpret each data source independently to make an ultimate decision about disease classification and severity [63], but recent studies [151, 190] have proposed assimilation of haemodynamics and imaging data with CFD modelling, providing insight into the structure and function of the pulmonary system.

Medical imaging and image segmentation have emerged as powerful non-invasive tools for disease diagnostics [5, 49, 142], providing an abundance of data for analysing the structure and function of the cardiovascular system under physiological and pathological conditions [110]. Advances in image segmentation include semi- and fully-automated algorithms for geometric reconstruction of complex vascular regions [84, 201]. However, inherent uncertainty is present as most image segmentation software require manual specification of the image intensity thresholds (pre-segmentation parameters) between background and foreground.

Haemodynamic predictions (e.g., cross-sectional averaged flow and pressure) [98] in the pulmonary vasculature are often computed using either three-dimensional (3D) or one-dimensional (1D) [151] CFD models. 3D models predict local flow patterns with more precision [190] but are computationally expensive, making it difficult to perform multiple forward model evaluations for uncertainty quantification, i.e. UQ [90]. For instance, Sankaran et al. [168] computed 3D CFD model sensitivity to coronary stenosis diameters, using surrogate model approximations to combat high computational cost. However, they did not account for possible changes in network connectivity (i.e. location of bifurcations or trifurcations) nor for the uncertainty from the initial segmentations of the vasculature. In contrast, 1D models are more computationally efficient, reducing the need for surrogates and allowing for investigations into variability of network connectivity. Moreover, a recent study [2] of the coronary vasculature showed that 1D models attain similar haemodynamic predictions as 3D when using appropriate boundary conditions. Recent studies analysed 1D systemic arterial models [161, 200] to understand how uncertainty in network structure impacts haemodynamics. Fossan et al. [60] devised an optimisation strategy to determine the number of vessels needed to match haemodynamic predictions in the coronary arteries, and Huberts et al. [90] used polynomial chaos expansion to quantify the sensitivity of flow predictions to variations in vessel radius. In contrast to the systemic circulation, the pulmonary vasculature is more compliant, branches more rapidly, and operates at a much lower mean pressure, indicating that results from the systemic circulation may not be valid for comparison.

The total uncertainty in the haemodynamic prediction is a combination of uncertainty in the model parameters and uncertainty from the modelling framework. As noted above, several previous studies have studied uncertainty with respect to prescribed haemodynamic parameters and the 1D approximation, but to our knowledge this is the first known investigation of the impact of uncertainties in network reconstruction on CFD simulations in the pulmonary vasculature. Specifically, in this chapter it is examined how pre-segmentation parameters impact estimated vessel radius, vessel length, and network connectivity, and this uncertainty is propagated to haemodynamic predictions in the pulmonary circulation. To do so, multiple segmentations of a microcomputed tomography (micro-CT) image from a mouse pulmonary arterial tree are analysed. The uncertainty resulting from the multiple segmentations is propagated using a 1D CFD model by constructing the model domain from each segmentation. Inverse UQ is performed by constructing probability density functions (pdfs) for vessel radii and lengths from network segmentations, and then propagating their uncertainty through to the model outputs, pulmonary blood flow and pressure (forward UQ) via Monte Carlo sampling. Uncertainty in haemodynamic predictions is quantified by analysing three sets of predictions (depicted in Figure 5.1); 1) predictions using 25 segmented networks (total variation); 2) predictions from a representative network with fixed connectivity when drawing realisations of length and radius perturbations (geometric parameter variation); and 3) predictions from the same representative network when geometric parameters are fixed, but connectivity and network size are varied (network variation). UQ is an essential component of the model analysis when computational models are integrated into clinical protocols. The animal dataset used here [202, 188] serves as a preliminary step in understanding disease progression and has potential for extrapolation to human PH.

5.2 Materials and methods

5.2.1 Experimental data

The experimental data are described in Section 2.1.

5.2.2 Image analysis

The image analysis was performed by Colebank with no input from me, however it is summarised here for clarity of the analysis to follow.

Image segmentation

The micro-CT image is a gray-scale image (shown in panel (a) of Figure 5.4) which is transformed to a binary map identifying the vascular (“foreground”) and non-vascular (“background”) regions using global thresholding and image segmentation in *ITK-SNAP* [221]. Global thresholding is a pre-segmentation technique requiring a priori selection of thresholds to specify the image intensity bounds of the foreground. Threshold bounds are traditionally selected in an ad-hoc manner to ensure that the foreground is captured [56, 133, 151]. In addition, *ITK-SNAP* requires specification of a smoothing parameter to determine the boundary between the foreground and background (see Figure 5.2). Due to the experimental protocol and use of perfused contrast, the image segmented in this study did not contain high intensity voxels from other anatomical features (e.g., the veins or the heart) within the region of interest. Therefore, only the lower threshold (θ_1) and smoothing (θ_2) pre-segmentation parameters required specification.

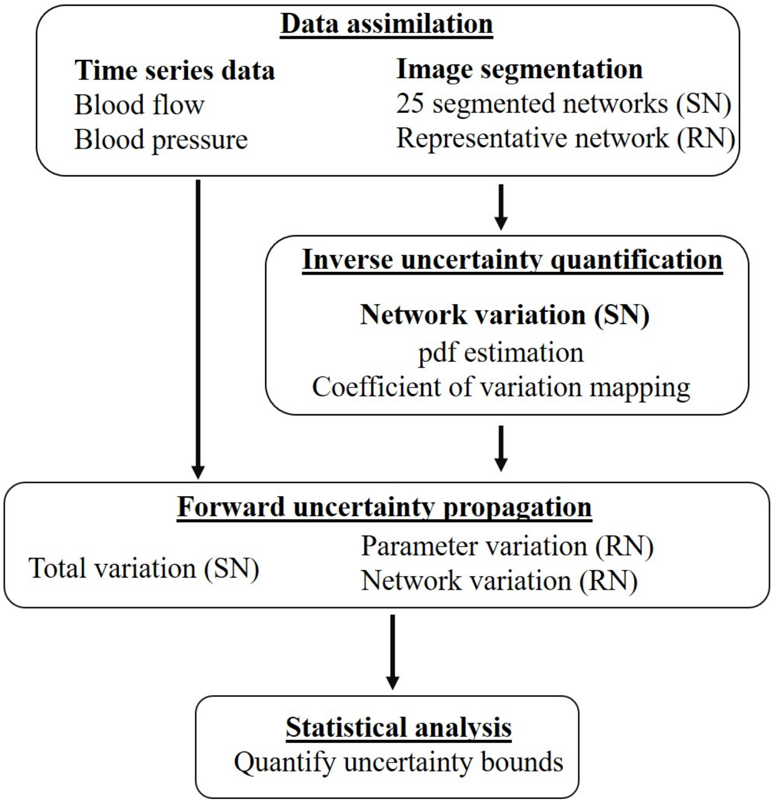


Figure 5.1: Workflow for uncertainty quantification of haemodynamics. Multiple segmentations were performed to construct the segmented networks (SNs), of which one network was selected as the representative network (RN), see Section 5.2.6 for details. Inverse uncertainty quantification (UQ) was performed on the 25 SNs by constructing probability density functions (pdfs) for vessel radius and length measurements. The 25 SNs were used in model simulations to understand the total variation, while the pdfs for the vessel radii and lengths were used to propagate uncertainty in the parameter variation study of a representative network. Lastly, the structure of the representative network was changed to understand the variation induced by network connectivity. Pressure and flow predictions are compared from the three sources of variation. Figure taken from our study in [37].

Acceptable intervals for (θ_1, θ_2) were determined to preserve the foreground for the large vessels across segmentations. 25 realisations of pre-segmentation parameter sets were drawn from a uniform distribution, i.e. $\theta_1 \in [20, 45], \theta_2 \in [3, 8]$, where these ranges were chosen based on qualitative inspection of the vasculature images. As shown in Figure 5.3, the foreground for distal vascular segments changes significantly when (θ_1, θ_2) are varied, but maintains features for the large, proximal vessels. Different combinations of (θ_1, θ_2) give rise to different vessel radii or lengths, number of vessels and vessel connectivity.

To segment the micro-CT image, a semi-automated segmentation algorithm available in *ITK-SNAP* (active contour evolution) was used, while ensuring that the largest arteries carrying the majority of the blood volume were captured.

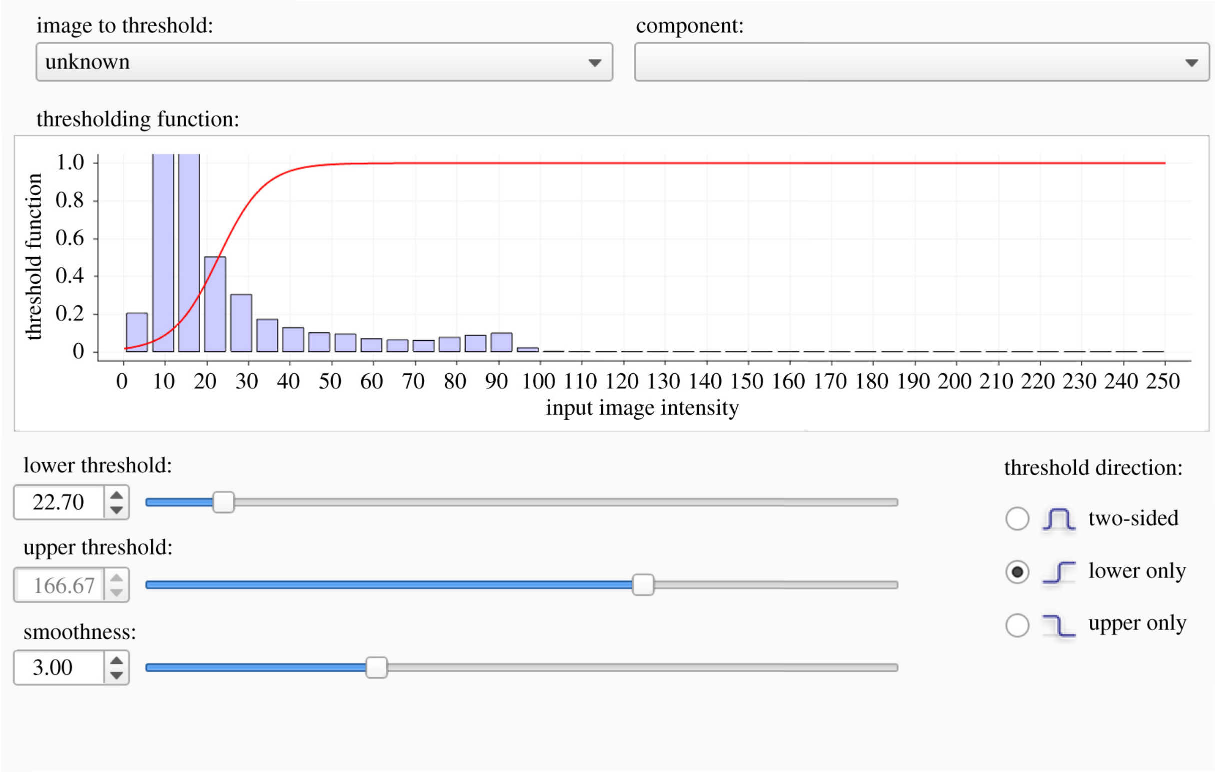


image to threshold: unknown

component:

thresholding function:

lower threshold: 22.70

upper threshold: 166.67

smoothness: 3.00

threshold direction:

○ two-sided
● lower only
○ upper only

Figure 5.2: *ITK-SNAP* interface for prescribing pre-segmentation parameters (lower threshold, smoothness). Voxel intensities (x-axis) are converted to probabilities via the threshold function (y-axis). Different pre-segmentation parameters change the form of the red curve, based on which discrimination between the foreground and background is done. Here, a lower threshold on image intensities was assumed, as shown by the constant value of one in the threshold function for all values greater than the lower threshold. Figure taken from our study in [37].

Network reconstruction

Segmented geometries were exported as surface meshes and converted to VTK polygonal files using *Paraview* [197] (Kitware, Clifton Park, NY, USA). Surface mesh VTK files were imported into *VMTK* (www.vmtk.org) [5] to extract vessel segment centrelines, lengths, and radii using native scripts. Custom Matlab algorithms, which can be found in `https://github.com/mjcolebank/Segmentation_CFD` were used to extract the network connectivity from the extracted quantities and identify all the vessels in each network. Subsequently, a connectivity matrix identifying the geometry of the tree was constructed and used in the 1D fluid-dynamics model. Figure 5.4 illustrates the workflow starting from the micro-CT image segmentation and ending with the connected network representation.

Figure 5.5 shows an example network with radii estimates at each point along the network and within a single vessel. To proceed with calculations, the vessel radius was fixed to be the mean over the centre 80% of the individual estimates, which mitigates the impact of extreme diameters in the ostium regions at either end of each segment. Connected graphs were constructed using the centreline data and a connectivity matrix was created by linking vessels, represented

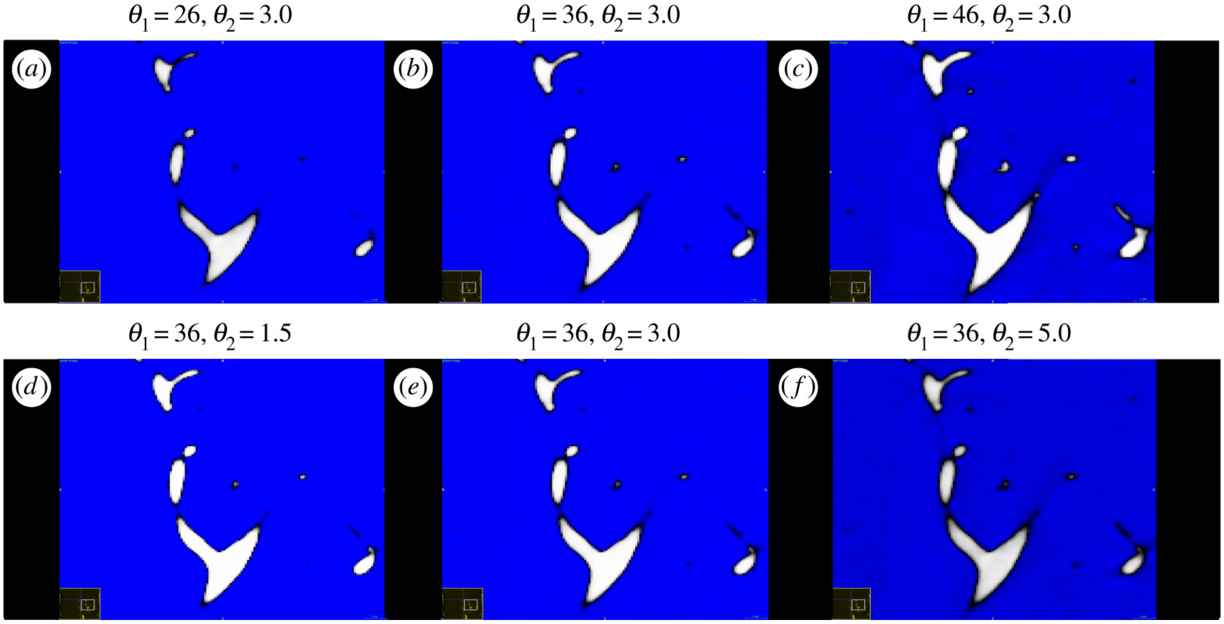

$\theta_1 = 26, \theta_2 = 3.0$
$\theta_1 = 36, \theta_2 = 3.0$
$\theta_1 = 46, \theta_2 = 3.0$
(a)
(b)
(c)
$\theta_1 = 36, \theta_2 = 1.5$
$\theta_1 = 36, \theta_2 = 3.0$
$\theta_1 = 36, \theta_2 = 5.0$
(d)
(e)
(f)

Figure 5.3: Qualitative differences in foreground (white) of distal vascular segments when changing the lower threshold for the voxel intensities (θ_1) and the smoothing parameter (θ_2). Top: changes in foreground with θ_1; bottom: changes in foreground with θ_2. Figure taken from our study in [37].

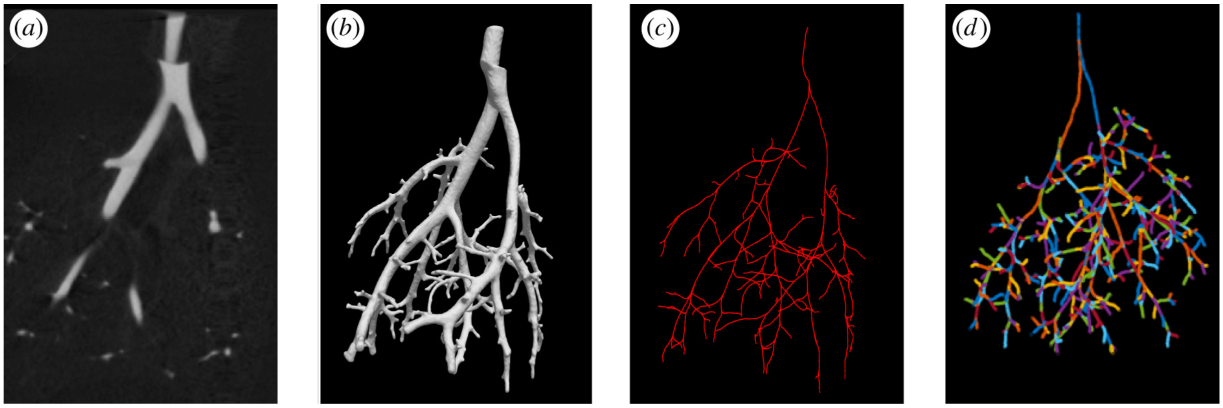


Figure 5.4: Image to network workflow. a) The foreground visible in the image file; b) The 3D rendering of the vascular foreground; c) Centrelines obtained using VMTK; d) A graph representation of the network used in the 1D model with vessels (edges) and bifurcations (nodes) identified using custom MATLAB algorithms, which can be found in `https://github.com/mjcolebank/Segmentation_CFD` (the different colours are used to distinguish where the vessels begin and end). Figure taken from our study in [37].

by their length and radius, and bifurcations. The CFD model used for haemodynamics modelling assumes a binary structure, with each generation of the tree being formed by a new set of vessels.

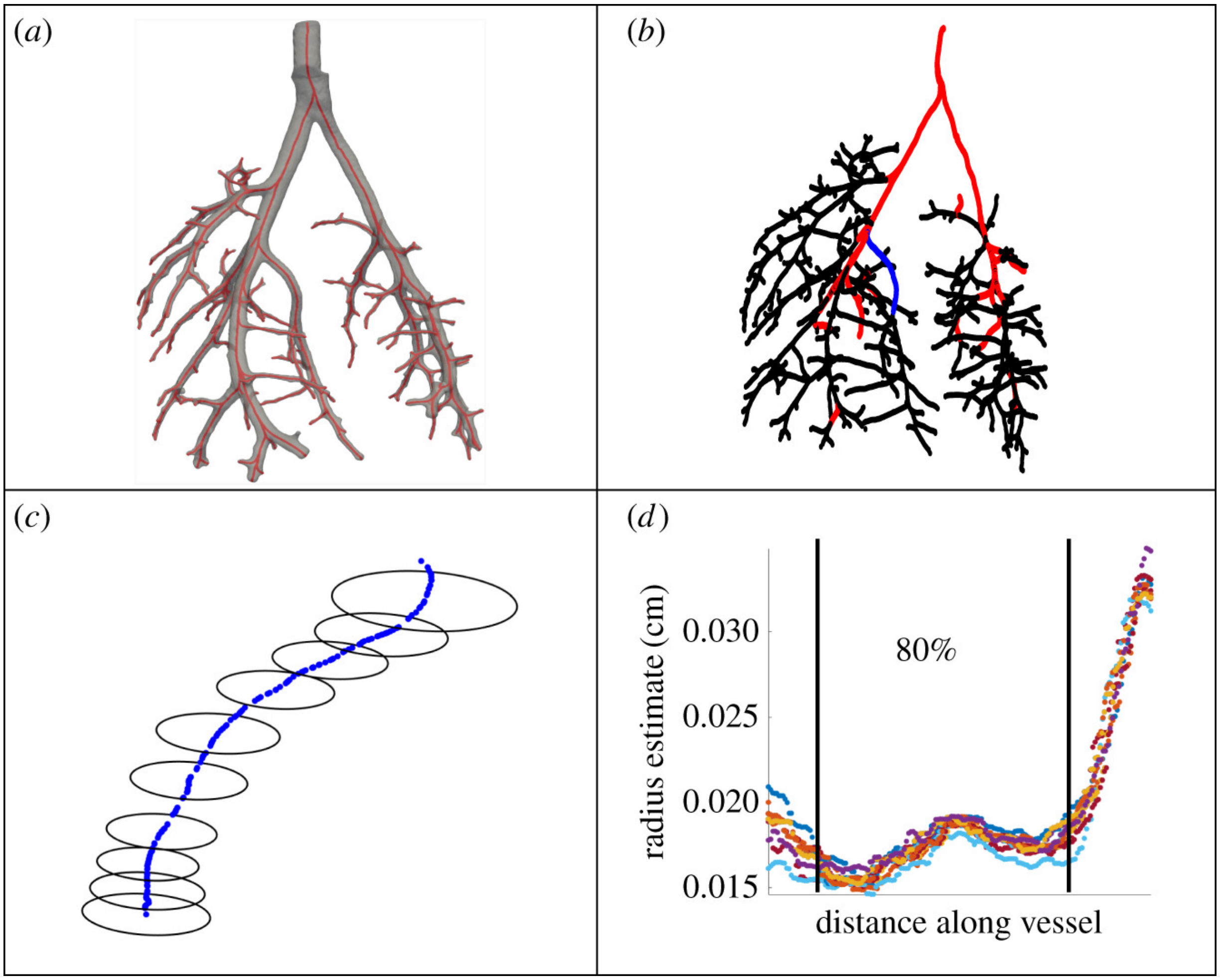


Figure 5.5: Components of an arterial tree. a) 3D segmentation of network; b) centreline representation of a tree with the 32 vessel-subset (red and blue); c) magnification of the vessel in blue depicting radius estimates; d) radius estimates along the vessel in blue, where the centre 80% of points were used to calculate the mean radius. Figure taken from our study in [37].

5.2.3 Mathematical model

The mathematical model used is indicated in Table 2.1 and described in Section 2.2.

5.2.4 Parameterisation

The haemodynamics modelling parameters include those describing the vascular structure (radius, length, and stiffness), the fluid dynamics (including viscosity, density, and the boundary layer thickness), and the inflow and outflow boundary conditions. The inflow, viscosity, density, and wall stiffness were assumed fixed (the inflow came from measurements, the kinematic viscosity was fixed to 0.0462 cm^{2}s^{-1}, the blood density to 1.057 g cm^{-3} and the stiffness to 37.5 mmHg based on literature [151, 159, 215]) and independent of the network geometry [103, 152, 151], while parameters specifying the vessel radius, length, and Windkessel outflow boundary conditions were allowed to depend on the network structure [60, 151].

5.2.5 Inverse uncertainty quantification

Inverse UQ was employed to estimate vessel length and radius pdfs over the 25 segmented networks. To compare measurements across segmentations, pdfs were computed for radius and length from a 32-vessel subset after data standardisation. Two estimation techniques, kernel density estimation (KDE) [180] and Gaussian process (GP) density estimation [160], were used to compare estimated pdfs. I implemented the GP density estimation technique, while Colebank implemented the KDE technique. In this chapter both techniques are summarised and results from both are presented for comparison. In addition, to investigate the variation in radius and length measurements across the 32 vessels, I implemented a GP regression model with input-dependent noise to remedy the issues of non-constant variance, i.e. heteroscedasticity, across vessels.

Data standardisation

A subset of 32 pulmonary vessels of various calibre (see panel (b) of Figure 5.5) was selected from the 25 segmented networks. The 32 vessels were visible in all 25 networks and contained radius and length measurements that encompass the full range of measurements in the networks. Length and radius measurements were standardised using

$$s^*_{i,j} = \frac{s_{i,j} - \bar{s}_i}{\sigma_{s_i}} \tag{5.1}$$

where $s_{i,j}$, $s = r, l$ are the measured radii or lengths from the i^{th} vessel and j^{th} segmentation, and s_i and σ_{s_i} are the mean and the standard deviations of the radii or lengths of the i^{th} vessel across the 25 networks.

Note that it is the measured vessel radius from the image segmentation process that is taken to be the reference radius r_0 in eqns (2.2) and (2.3).

Density estimation

The pdfs for radius and length were estimated using density estimation. These pdfs, constructed from the 32-vessel subset, capture the overall variation in the length and radius across all the segmented networks.

Logistic GP density estimation: A number (n) of independently-obtained measurements $s^*_1, \ldots$ s^*_n allows to construct a probability density p in a finite region Ω. An estimate for the unknown probability density p could be obtained by maximising the log likelihood function:

$$\mathsf{L}(p) = \sum_{i=1}^{n} \log p(s^*_i), \tag{5.2}$$

constrained to:

$$\int_{\Omega} p(s^*)ds^* = 1, \tag{5.3}$$

$$p(s^*) > 0 \text{ for any } s^* \in \Omega. \tag{5.4}$$

Optimising the log likelihood in eq (5.2) leads to a limiting solution of a mixture of Dirac delta functions located at the measurements, hence a prior about the unknown density is needed for realistic estimates to be obtained [160]. By defining f to be an unconstrained latent function, logistic density transform [160] may be employed, and is defined as:

$$p(s^*) = \frac{\exp\left(f(s^*)\right)}{\int_{\Omega} \exp(f(a))da}, \tag{5.5}$$

which satisfies the constraints in eq (5.3). A GP prior can be placed on the latent functions f to smooth the density estimates, and the smoothness can be controlled via the covariance functions for the GP model. Following [160], this study uses a GP model which favours density estimates for which the tails of the distribution (i.e. outside the domain of the data) eventually go to zero:

$$\begin{aligned} f(s^*) &= \mathbf{h}(s^*)^{\mathrm{T}}\beta + g(s^*), \\ g(s^*) &\sim \mathscr{GP}(\mathbf{0}, \mathbf{K}). \end{aligned} \tag{5.6}$$

where $\mathbf{h}(s^*)$ are the basis functions for the input s^*: $\mathbf{h}(s^*) = (s^* \; s^{*2})$ (i.e. $\mathbf{h}(s^*)$ is an $l \times 1$ vector, where $l = 2$ for 1D input) and β are the weights (an $l \times 1$ vector). The coefficient of s^{*2} should be negative to make the estimates in the tails of the distribution go to zero, however the prior alone cannot impose this [160]. By placing a Gaussian prior on the weights:

$$\beta \sim \mathscr{MVN}(\mathbf{b}, \mathbf{B}), \tag{5.7}$$

they can be integrated out from the joint prior distribution to obtain:

$$f(s^*) \sim \mathscr{GP}(\mathbf{h}(s^*)^{\mathrm{T}}\mathbf{b}, k(s^*, s^{*'}) + \mathbf{h}(s^*)^{\mathrm{T}}\mathbf{B}\mathbf{h}(s^{*'})). \tag{5.8}$$

The Ω space is discretised into m sub-regions (or intervals in 1D), and the coordinates of the sub-regions are stored into a matrix $\mathbf{X}$: $m \times d$ (d: parameter dimensionality, i.e. $d = 1$ in this case), where the i^{th} row stores the centre point of the i^{th} sub-region. Given $\mathbf{X}$, the GP prior model in eq (5.8) becomes:

$$p(\mathbf{f}|\mathbf{X}, \gamma) = \mathscr{MVN}(\mathbf{f}|\mathbf{H}\mathbf{b}, \mathbf{K} + \mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}}), \tag{5.9}$$

where $\mathbf{f}$ is a column-vector of size $m \times 1$ containing the latent functions for each of the m sub-regions, $\mathbf{K}$ is an $m \times m$ covariance matrix determined by the input points in $\mathbf{X}$ and by the covariance hyperparameters γ, $\mathbf{H}$ is an $m \times l$ matrix containing the basis functions evaluated at the input points. Following [160], the following settings were used: $\mathbf{b} = \mathbf{0}$ (an $l \times 1$ vector) and $\mathbf{B} = 10^2\mathbf{I}$ (an $l \times l$ matrix), i.e. $\beta \sim \mathscr{MVN}(\mathbf{0}, 10^2\mathbf{I})$, and the number of grid points m was set to 400. In addition, the covariance function for the GP model was taken to be the non-stationary neural network (a non-stationary kernel was needed as numerical instabilities caused by a high condition number of the covariance matrix which needs inverting were encountered with stationary kernels, such as squared exponential or Matérn).

After the discretisation (assuming a regular grid), the likelihood of an observation belonging to the i^{th} sub-region is:

$$\mathscr{L}_i = \frac{\exp(f_i)}{\sum_{j=1}^{m} \exp(f_j)}. \tag{5.10}$$

The number of observations in the i^{th} sub-region is denoted by y_i, thus $\mathbf{y}$ is an $m \times 1$ vector containing the observation count from all m sub-regions, i.e. $\sum_{i=1}^{m} y_i = n$. The overall likelihood based on all n points from the m regions is based on the Multinomial distribution, and the log likelihood is given by:

$$\log p(\mathbf{y}|\mathbf{f}) = \log\left(\prod_{i=1}^{m}\left(\frac{\exp(f_i)}{\sum_{j=1}^{m}\exp(f_j)}\right)^{y_i}\right) \tag{5.11}$$

$$= \sum_{i=1}^{m}\left(y_i \log(\exp(f_i)) - y_i \log\left(\sum_{j=1}^{m}\exp(f_j)\right)\right) \tag{5.12}$$

$$= \sum_{i=1}^{m}(y_i f_i) - \log\left(\sum_{j=1}^{m}\exp(f_j)\right)\sum_{i=1}^{m} y_i \tag{5.13}$$

$$= \mathbf{y}^{\mathrm{T}}\mathbf{f} - n\log\left(\sum_{j=1}^{m}\exp(f_j)\right) \tag{5.14}$$

Therefore, the overall likelihood is:

$$p(\mathbf{y}|\mathbf{f}) = \exp\left(\mathbf{y}^{\mathrm{T}}\mathbf{f} - n\log\left(\sum_{j=1}^{m}\exp(f_j)\right)\right). \tag{5.15}$$

The prior in eq (5.9) and the likelihood in eq (5.15) can be combined using Bayes' theorem to obtain the conditional posterior distribution of the latent functions, i.e.

$$p(\mathbf{f}|\mathbf{y},\mathbf{X},\gamma) = \frac{p(\mathbf{f}|\mathbf{X},\gamma)p(\mathbf{y}|\mathbf{f})}{\int p(\mathbf{f}|\mathbf{X},\gamma)p(\mathbf{y}|\mathbf{f})d\mathbf{f}}. \tag{5.16}$$

With a GP prior (eq (5.9)) and a non-Gaussian likelihood (eq (5.15)), the posterior distribution (eq (5.16)) is non-Gaussian, thus it was approximated using Laplace approximation [160] (the default approximation method in the toolbox).

Kernel density estimation: Kernel density estimation (KDE) [180] is a non-parametric technique for estimating unknown probability distributions. KDE constructs the pdf $p(.)$ for vessel length and radius using

$$p(s^*) = \frac{1}{nH} \sum_{i=1}^{n} k\left(\frac{s^* - s_i^*}{H}\right) \tag{5.17}$$

where s_i^* denotes the standardised measurement of the i^{th} vessel, n is the number of samples used for the density estimate, H is the bandwidth parameter (giving the smoothness of the density estimate, playing the role of the lengthscale in the GP covariance function), and k is the kernel function, assumed to be a Gaussian kernel, i.e. a Gaussian kernel is placed (centred) on each of the data points, and these kernels are summed to give the kernel density estimate. Two approaches for finding the optimal KDE bandwidth were considered: Silverman's rule-of-thumb [180] and maximum likelihood leave-one-out cross-validation (MLCV) [79]. The former calculates the bandwidth under the assumption that the underlying density being estimated is Gaussian, by using the median absolute deviation, and has been shown to minimise the integrated mean squared error of the density estimate. These estimates were calculated using the *ksdensity* function from `Matlab`'s Statistics and Machine Learning Toolbox. The latter (MLCV) leaves one point out at a time and computes the probability density estimate for the point left out based on all the other points (see eq (5.18)). Leaving one point out affects the calculation of the standardised quantities and requires re-calculation of the standardised measurements for each iteration.

$$p(s_{-i}^*) = \frac{1}{(n-1)H^{\text{MLCV}}} \sum_{j=1\ j\neq i}^{n} k\left(\frac{s_j^* - s_{-i}^*}{H^{\text{MLCV}}}\right) \tag{5.18}$$

where s_{-i} denotes the data point which was left out of the sample. The optimal bandwidth parameter H^{MLCV} is that that maximises the log-likelihood of the KDE:

$$H^{\text{MLCV}} = \max_{H>0} \left(\frac{1}{n}\sum_{i=1}^{n} \log\left(\sum_{j=1\ j\neq i}^{n} k\left(\frac{s_j^* - s_{-i}^*}{H}\right)\right) - \log((n-1)H)\right). \tag{5.19}$$

The bandwidth parameter determines the smoothness of the probability density.

GP regression with input-dependent noise

Given that the standard deviations of the vessels' radii and lengths, i.e. σ_{l_i} and σ_{r_i}, have different magnitudes from vessel to vessel, the coefficient of variation was used

$$c_v^{s_i} = \frac{\sigma_{s_i}}{\bar{s}_i}, \tag{5.20}$$

to compare the variability of the radius and length measurements between vessels (the higher the coefficient of variation, the higher the variability around the mean). The variance of the measurements exhibits heteroscedasticity, as smaller vessel segments are more sensitive to pre-segmentation parameters leading to non-constant variance. Hence, a GP regression model with input-dependent noise was used, following an approach by Goldberg et al. [75]. This model assumes Gaussian-distributed noisy data with both the mean and the variance as latent functions, modelled using two independent GPs (to be compared to standard GP which only assumes a GP on the mean and the noise variance is constant across the input range).

$$\begin{aligned}\mathbf{y}|\mathbf{X},\mathbf{f}^{(1)},\mathbf{f}^{(2)} &\sim \mathscr{MVN}(\mathbf{f}^{(1)}(\mathbf{X}),\sigma^2\exp(\mathbf{f}^{(2)}(\mathbf{X})),\\ \mathbf{f}^{(1)}(\mathbf{X})|\gamma^{(1)} &\sim \mathscr{GP}(\mathbf{m}^{(1)}(\mathbf{X}),\mathbf{K}^{(1)}|\gamma^{(1)}),\\ \mathbf{f}^{(2)}(\mathbf{X})|\gamma^{(2)} &\sim \mathscr{GP}(\mathbf{m}^{(2)}(\mathbf{X}),\mathbf{K}^{(2)}|\gamma^{(2)}),\\ \gamma^{(1)}\gamma^{(2)} &\sim p(\gamma^{(1)})p(\gamma^{(2)}),\end{aligned} \tag{5.21}$$

where $\gamma^{(i)}$ contains the covariance function hyperparameters for the i^{th} GP, and σ^2 is a constant observation noise variance which the model simplifies to for input-independent noise ($\sigma^2\exp(0)$). The formulation for the noise variance in eq (5.21), i.e. a GP prior is placed on the log of the noise variance, ensures that the variance stays always positive.

The noisy data ($\mathbf{y}$) are the coefficient of variation $c_v^{s_i}$, whose mean and variance across the vessels' radii and lengths ($\mathbf{X}$) are latent functions on which two independent GP prior models are placed. The Matérn 5/2 covariance function was used as a kernel for both GPs (as chosen based on the data, e.g. cross-validation, see Section C.1 in the appendix for details), and their hyperparameters were found by optimisation of the log marginal likelihood.

5.2.6 Forward uncertainty quantification

Forward UQ propagates model and parameter uncertainties to simulated quantities of interest. To analyse the variability in model predictions, three sets of simulations were set up that determine (i) the total variation of haemodynamic predictions associated with segmentation, (ii) the variation to changes in vessel radius and length (geometric parameters), and (iii) the variation to changes in network size and connectivity. The first set of simulations (i) used the 25 segmented networks, whereas the last two (ii-iii) were conducted in a representative network. This part of the work was carried out entirely by Colebank, and the methodology is summarised here.

Total variation

The haemodynamics predicted for each of the 25 segmented networks were used to quantify the total variation of flow and pressure predictions across the networks in the MPA, LPA, and RPA. The observed variation is attributed to several sources of uncertainty, including the model geometric parameters and the network size and connectivity. Once the total variation was calculated, the relative contributions from the geometric parameter and network variation were obtained.

Representative network

A representative network was used to examine the variation in vessel radius and length (geometric parameters) and changes in network size and connectivity. The MPA pressure waveform for each of the 25 segmented networks was computed, then the point-wise average MPA pressure across the 25 networks was obtained. The network with the smallest least squares cost between its MPA pressure waveform and the averaged waveform was designated as the representative network.

Geometric parameter variation

Cumulative distribution functions (cdfs, strictly increasing functions defined on the interval [0,1]) and their inverse (icdfs) can be computed from the pdfs of the geometric parameters: (standardised) length and radius measurements. Given a random variable X with pdf $f_X(x)$, its cdf $F_X(x)$ is given by:

$$F_X(x) = \int_{-\infty}^{x} f_X(u)du, \tag{5.22}$$

and can be obtained by numerical integration if f does not have an analytical form. In addition, the inverse cdf $F_X^{-1}(u)$, where $u = F_X(x)$ can be found by root-finding algorithms [144] if f does not have an analytical form (i.e. find the unique root x of the equation $u - F_X(x) = 0$). Having cdf and icdf allows inverse transform sampling of the (standardised) radius and length measurements.

Given a random variable X, its cdf $F_X(x)$ and icdf $F_X^{-1}(u)$ are strictly increasing functions, and the inverse transform sampling proceeds as follows:

- Generate a draw from the uniform distribution: $u \sim \mathscr{U}(0,1)$.
- Compute $x = F_X^{-1}(u)$.

Below an explanation as to why this works is given. Let us check if we can find a a strictly monotone transformation $T : [0,1] \to \mathbb{R}$ such that $T(U) \overset{\mathrm{d}}{=} X$, i.e. $T^{-1} : \mathbb{R} \to [0,1]$ such that $U \overset{\mathrm{d}}{=} T^{-1}(X) \sim \mathscr{U}(0,1)$.

$$F_X(x) = p(X \leq x) = p(T(U) \leq x) = p(U \leq T^{-1}(x)) = T^{-1}(x) \text{ for } u \sim \mathscr{U}(0,1), \tag{5.23}$$

So $F_X(x)$ is the inverse function of $T(x)$. Next we check if we can prove if $T(u) = F_X^{-1}(u)$ holds. We know that

$$T^{-1}(T(u)) = u. \tag{5.24}$$

Replacing x in eq (5.23) by $T(u)$ we obtain

$$F_X(T(u)) = T^{-1}(T(u)). \tag{5.25}$$

Thus, by combining eqns (5.24) and (5.25), we obtain

$$F_X(T(u)) = u. \tag{5.26}$$

By applying F_X^{-1} on both sides of eq (5.26), we get

$$T(u) = F_X^{-1}(u), \tag{5.27}$$

as desired. Thus,

$$F_X^{-1}(u) = T(u) = x. \tag{5.28}$$

Hence we can generate X from the strictly increasing function $F_X^{-1}(U)$.

Thus, we can apply the inverse transform sampling approach (Monte Carlo sampling) to draw samples for the (standardised) radius and length measurements which are subsequently used for obtaining model predictions (enabling forward UQ), as follows:

1. Draw $u \sim \mathscr{U}(0,1)$.
2. Obtain the standardised measurements s^*, thus the original measurements s as:

$$s^* = F_{s^*}^{-1}(u) \implies F_{s^*}^{-1}(u) = \frac{s-\bar{s}}{\sigma_s} = \frac{s-\bar{s}}{c_v^s \bar{s}}, \tag{5.29}$$

$$\implies s = \bar{s}(c_v^s F_{s^*}^{-1}(u) + 1), \tag{5.30}$$

   by using eqns (5.1) and (5.20).

3. Run 1D CFD model with new s, i.e. new radius and length measurements for every vessel, and new Windkessel parameters (outflow boundary conditions) as they depend on the newly drawn measurements, see Section 2.2.4 and [37] for the mathematical details on this.
4. Repeat steps 1-3 $M = 10^4$ times.

Network variation

The effect of network variation (i.e. truncation) was simulated by iteratively eliminating terminal vessel pairs from the representative network, i.e. starting at the smallest branches and moving towards the proximal vasculature, vessels with the smallest volume ($V = \pi r^2 l$) were eliminated, while ensuring that Windkessel boundary conditions were adjusted for each simulation, mathematical details are in [37].

5.3 Results

A selection of results are presented, with an emphasis on the results that I produced, for a full set of results, please see [37].

5.3.1 Inverse uncertainty quantification

Figure 5.6 shows the length and radii KDE for the 32 representative vessels computed using Silverman's rule, maximum likelihood cross-validation, and GPs. Computations using Silverman's rule exhibit overfitting, while the maximum likelihood cross-validation over-smooths the density relative to the GP. In summary, the GP density estimation provides the most robust approximation for the KDE, which is therefore chosen in the forward uncertainty propagation in Section 5.3.2.

In addition, Figure 5.7 (top panels) shows the GP regression for the coefficient of variation for the radius and length measurements, i.e. c_v^r and c_v^l, while bottom panels depict the latent variance. The value for the coefficient of variation increases as vessels get smaller (i.e. as the radius and length decrease). The mean variance for c_v^l increases as the length decreases, yet the mean variance of c_v^r has a sharp decrease for the smallest vessels. Both GP models stay above the minimum variability of $20\mu m$ (image resolution), as given in the study by Vanderpool et al. [202].

5.3.2 Forward uncertainty quantification

The MPA flow data were used as an inflow boundary condition, hence it does not change in any of the simulations. The average pressure prediction (over over 25 networks) is shown for the MPA, LPA, and RPA along with ± two standard deviations in the left column of Figure 5.8. Mean, systolic, diastolic and pulse pressure and max flow, min flow, and total volume, are given in Table 5.1. The flow distribution to the LPA is much larger than the RPA, a consequence of the larger radius of the LPA that allows for greater fluid flow.

The middle column of Figure 5.8 shows the variation in the model predictions due to the parameter variation in the representative network. That is, Figure 5.8 displays the 10,000 model predictions (obtained via forward UQ as part of the Monte Carlo sampling of the vessel radii and lengths, presented in Section 5.2.6), along with the mean and ± two standard deviations from the mean. The variation in the MPA, LPA, and RPA systolic and pulse pressures is significantly larger than the variation in the diastolic and mean pressures (see Table 5.1). The flow predictions in the LPA and RPA have larger variability in mean and max flow in comparison to the minimum flow.

The variation attributed to network size and connectivity was calculated by fixing each vessel's radius and length in the representative network before reducing the full network iteratively,

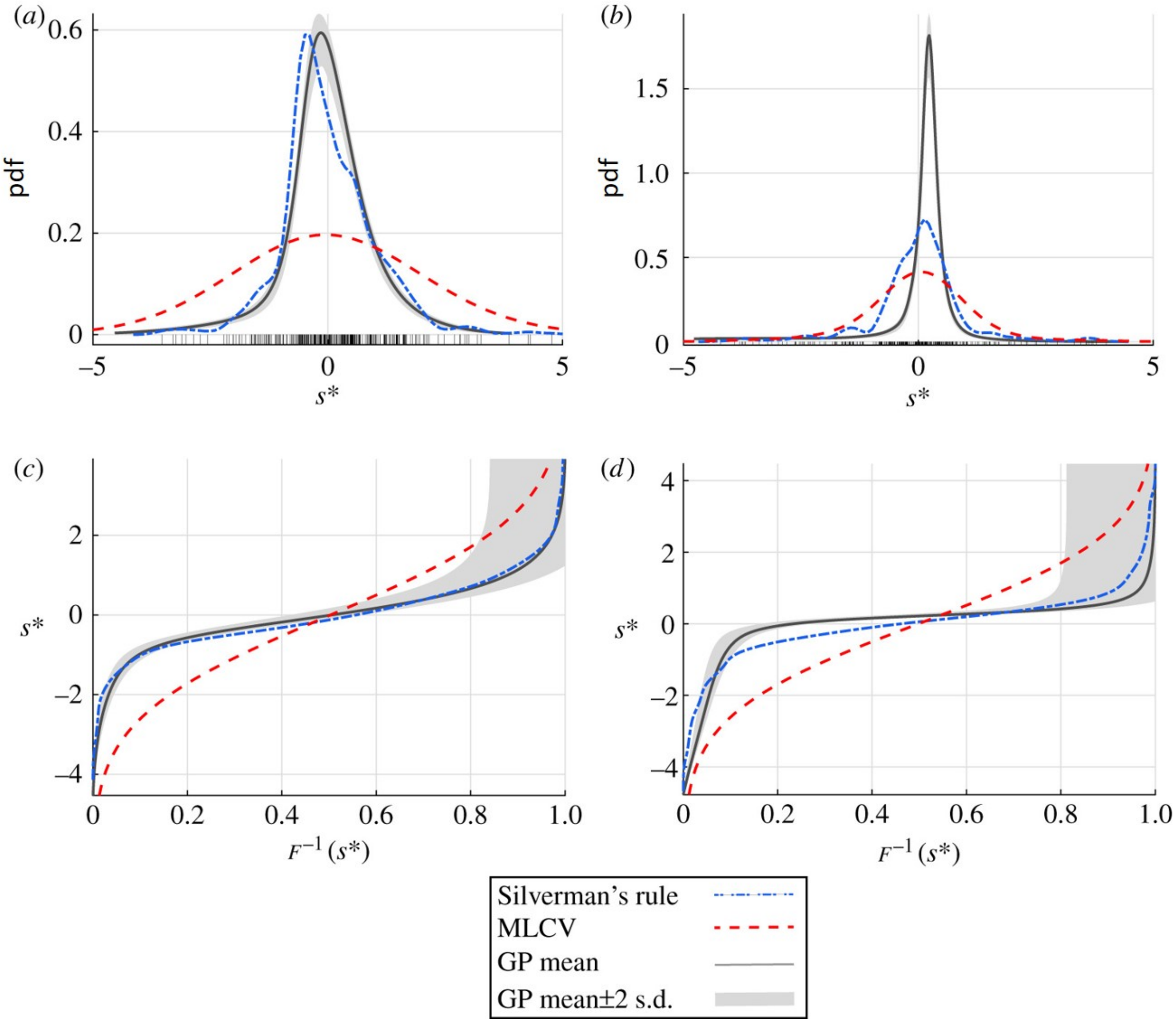
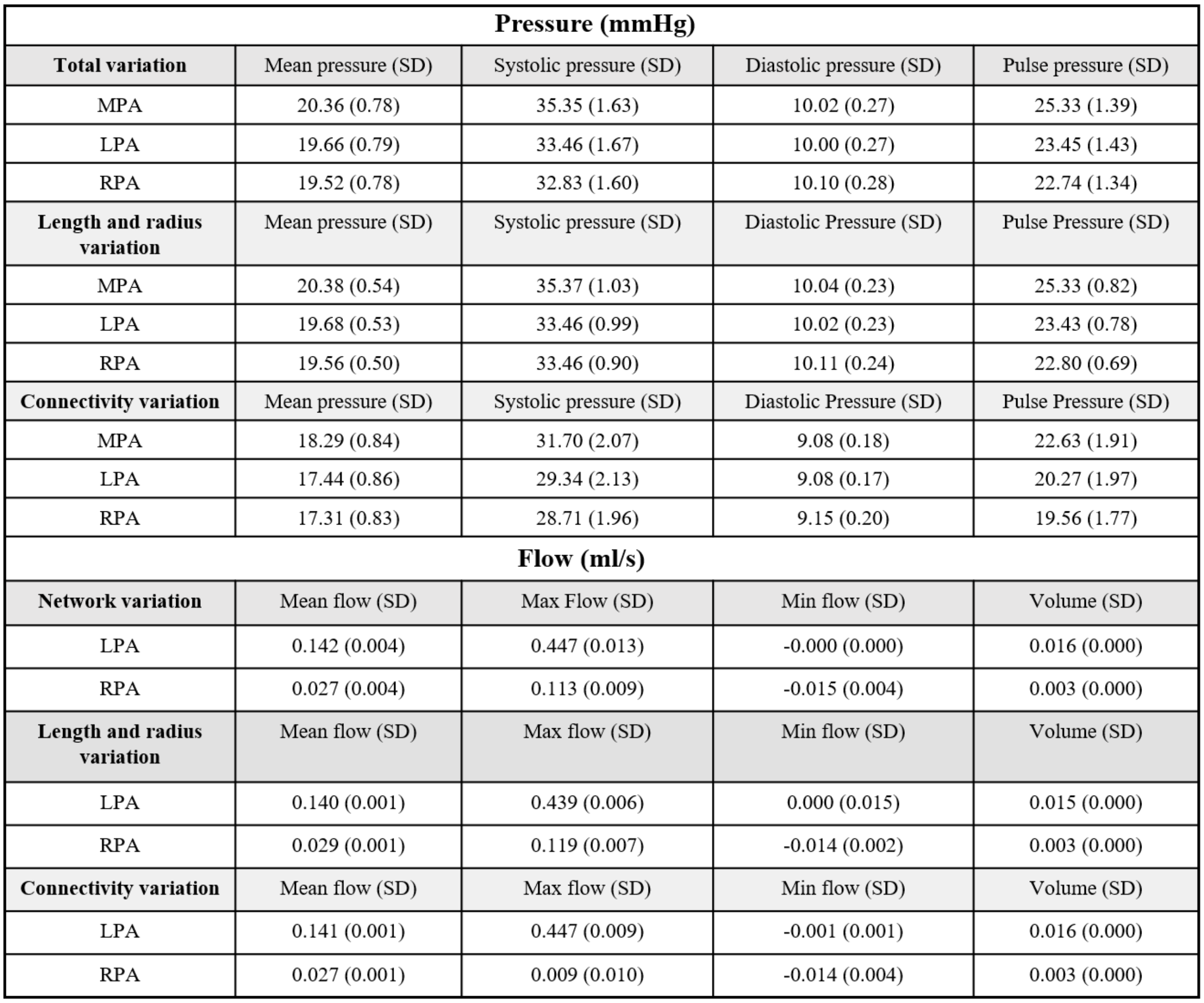


Figure 5.6: Density estimates (a) and (b) and inverse cumulative distribution functions (c) and (d) for the standardised radius and length values, respectively, measured in the 32-vessel subset. The bandwidth parameters used for the length and radius KDEs were determined using Silverman's rule (blue, dash dot) and maximum likelihood cross-validation (MLCV, red, dashed). The Gaussian process (GP) mean and 95% confidence interval are shown as a solid curve with grey bands. Standardised values are denoted by the black tick marks in panels (a) and (b). Figure taken from our study in [37].

and can be seen in the right column of Figure 5.8. Overall, reducing the number of vessels from 219 (largest network) to 3 (smallest network) introduces a pressure drop of approximately 10 mmHg in the pressure predictions of all three pulmonary arteries.

5.4 Discussion

Recent advances in image segmentation have made subject-specific modelling of PH feasible, yet the modelling process still comprises segmentation-induced uncertainty that propagates through to simulation results. This is the first known study to explicitly quantify the variability



Pressure (mmHg)				
Total variation	Mean pressure (SD)	Systolic pressure (SD)	Diastolic pressure (SD)	Pulse pressure (SD)
MPA	20.36 (0.78)	35.35 (1.63)	10.02 (0.27)	25.33 (1.39)
LPA	19.66 (0.79)	33.46 (1.67)	10.00 (0.27)	23.45 (1.43)
RPA	19.52 (0.78)	32.83 (1.60)	10.10 (0.28)	22.74 (1.34)
Length and radius variation	Mean pressure (SD)	Systolic pressure (SD)	Diastolic Pressure (SD)	Pulse Pressure (SD)
MPA	20.38 (0.54)	35.37 (1.03)	10.04 (0.23)	25.33 (0.82)
LPA	19.68 (0.53)	33.46 (0.99)	10.02 (0.23)	23.43 (0.78)
RPA	19.56 (0.50)	33.46 (0.90)	10.11 (0.24)	22.80 (0.69)
Connectivity variation	Mean pressure (SD)	Systolic pressure (SD)	Diastolic Pressure (SD)	Pulse Pressure (SD)
MPA	18.29 (0.84)	31.70 (2.07)	9.08 (0.18)	22.63 (1.91)
LPA	17.44 (0.86)	29.34 (2.13)	9.08 (0.17)	20.27 (1.97)
RPA	17.31 (0.83)	28.71 (1.96)	9.15 (0.20)	19.56 (1.77)
Flow (ml/s)				
Network variation	Mean flow (SD)	Max Flow (SD)	Min flow (SD)	Volume (SD)
LPA	0.142 (0.004)	0.447 (0.013)	-0.000 (0.000)	0.016 (0.000)
RPA	0.027 (0.004)	0.113 (0.009)	-0.015 (0.004)	0.003 (0.000)
Length and radius variation	Mean flow (SD)	Max flow (SD)	Min flow (SD)	Volume (SD)
LPA	0.140 (0.001)	0.439 (0.006)	0.000 (0.015)	0.015 (0.000)
RPA	0.029 (0.001)	0.119 (0.007)	-0.014 (0.002)	0.003 (0.000)
Connectivity variation	Mean flow (SD)	Max flow (SD)	Min flow (SD)	Volume (SD)
LPA	0.141 (0.001)	0.447 (0.009)	-0.001 (0.001)	0.016 (0.000)
RPA	0.027 (0.001)	0.009 (0.010)	-0.014 (0.004)	0.003 (0.000)

Table 5.1: Forward uncertainty quantification results. Statistics based on the pressure and flow predictions in the first pulmonary bifurcation are displayed when studying total variation, geometric parameter variation, and network variation. Predictions from the total variation include simulations in the 25 segmented networks. The geometric parameter variation is based on 10,000 Monte Carlo realisations. Lastly, the network variation is based on 219 vessels reduced iteratively to three vessels in the network (MPA, LPA, and RPA). Figure taken from our study in [37].

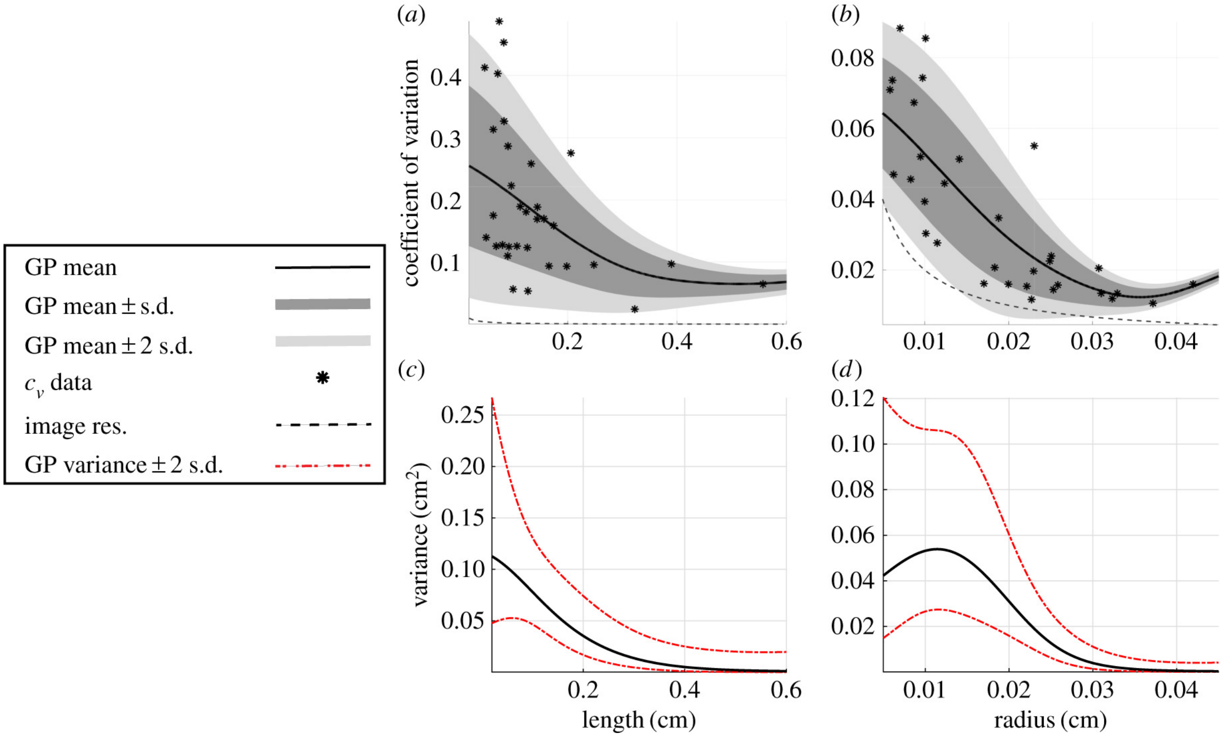


Figure 5.7: Gaussian Process (GP) regression using non-constant variance for the relationship between length and radius and their coefficient of variation (c_v). The GP means and standard deviations were computed from the c_v data obtained from the 32-vessel subset (asterisks) and plotted against the analytical bound of the image resolution (dash-dot curve), as given in [202]. The mean of the GPs and ± one and two standard deviations (s.d.) from the mean are shown in (a) and (b) in black, dark grey, and light grey, respectively. The variance of the GPs in (c) and (d) were predicted using an additional GP and provided a mean (black) and variance (dashed curve) for the variance estimate. Both mean curves in (a) and (b) are above the uncertainty bound of the imaging protocol. Figure taken from our study in [37].

of 1D CFD blood flow and pressure predictions arising from uncertainty in pre-segmentation parameterisation. Three types of segmentation-induced variations were investigated: the total variation arising from changes in pre-segmentation parameters, variation due to changes in vessel length and radius (geometric parameter variation), and variation with respect to network connectivity and size (network variation). Results suggest that variation in network structure is the greater contributor to uncertainty in haemodynamic predictions, consistent with what is known of the pulmonary vascular physiology. Moreover, the methodology developed herein can be used to generate a 1D model network for any vascular system.

5.4.1 Inverse uncertainty quantification

KDEs and GPs are commonly used techniques for density estimation [122, 160], but this study is the first to use GPs in density estimation for vascular measurements. Typically, prior assumptions are forced on the unknown parameter distributions by assuming a parametric parameter

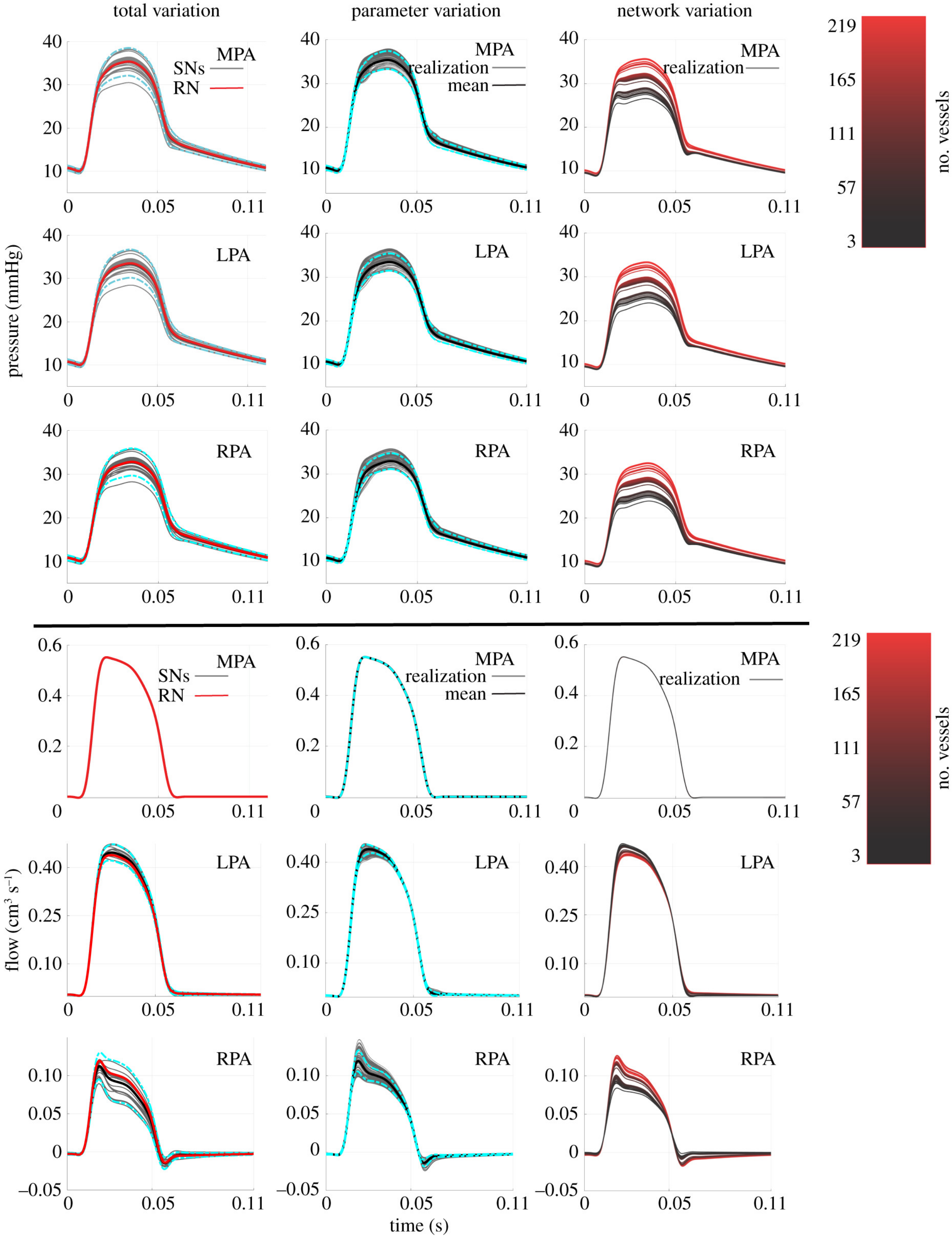


Figure 5.8: Pressure and flow predictions in the first pulmonary bifurcation when studying total variation, parameter variation, and network variation. Predictions from the total variation (1st column) include simulations in the 25 segmented networks (SNs), the representative network (RN, in red), and ± two standard deviations (s.d.) from the mean (blue, dash-dot). The parameter variation plots (2nd column) show the 10,000 Monte Carlo realisations (grey) along with the mean (black) ± two s.d. from the mean (blue, dash-dot). Lastly, the network variation predictions (3rd column) show the predictions when using 219 vessels in the network (bright red) up until the network is reduced to the MPA, LPA, and RPA (black). Figure taken from our study in [37].

distribution. By estimating the density directly from repeated measurements, a non-parametric density was constructed, describing the uncertainty of the measurements across segmentations without prior assumptions. As shown in Figure 5.6, the three density estimates are similar in the mode of the distribution (approximately zero); however the GP density estimation allows for additional UQ in both the pdf and cdf estimates [160]. The pdfs for radius and length were constructed independently, thus ignoring any correlation between the radius and length measurements. A pdf estimation method that accounts for correlation between the measurements of the two quantities should be investigated further, e.g. 2D GP density estimation [160] could be used to find the joint pdf for the radius and length measurements, and rejection sampling [27] could be subsequently employed for sampling the measurements from the joint pdf.

The GP regression analysis of the coefficient of variation against the (standardised) measurements of radius and length revealed that the coefficient of variation for the measurements increased as the measurements decreased in value. This suggests that smaller vessels are subject to larger fluctuations in measurements when varying pre-segmentation parameters, i.e. across different segmentations. Similar conclusions have been made in simulations of coronary arteries [200], as the smaller regions of the vasculature were susceptible to higher segmentation error.

5.4.2 Forward uncertainty quantification

The total network size obtained from the segmentation procedure has several effects on the model output. As shown in Table 5.1, changes in network topology due to segmentation induced a variation in systolic pressure that was nearly 6 times larger than the variation of diastolic pressure. Moreover, the total variation for the systolic and pulse pressure was larger in comparison to the mean pressure and the diastolic pressure. These pressure metrics are typically used in diagnostic tools of diseases such as PH [63], as well as for risk assessment in patients with congenital heart disease [100]. This further indicates a need for UQ when using these models for cardiovascular disease diagnostics and risk assessment.

The standard deviation of the diastolic pressure resulting from geometric parameter (radius and length) variation was greater than that resulting from network (size and connectivity) variation. This suggests that changes in vessel dimensions and nominal boundary conditions can ultimately raise the diastolic pressure of the system, which is expected in the case of chronic vascular remodelling [63]. Geometric parameter (radius and length) variation only accounted for approximately 30% of the total variation in the pulse pressure and had less of an effect on all other pressure and flow quantities when compared to the network variation. Larger networks encompassing the entire pulmonary tree correspond to more vessels, thus more uncertain estimates of radius and length, which leads to higher uncertainty of the haemodynamic parameter estimates [168].

The largest effects on pressure and flow waveform predictions in the network are attributed to changes in network connectivity and size (Figure 5.8). Network variation produces larger standard deviations in systolic and pulse pressures vs geometric parameter variation (see Table 5.1), suggesting that the configuration of vessels in the pulmonary system may play an important role in haemodynamic predictions. It is known that network remodeling is common with pulmonary vascular disease [63, 190].

Changes in network size will lead to changes in optimal values of the parameters describing stiffness, compliance, and vascular resistance during parameter inference, as these estimated parameters depend on the size of the network used in CFD simulations. This further indicates that uncertainty in the network structure must be taken into account when using 1D CFD models for clinical decision making [38].

5.4.3 Limitations and future work

Several potential limitations of this study can be addressed in future investigations. First, negligible tapering is assumed in each vessel, which could play a role in proximal artery dynamics. Second, the CFD model assumptions ignore radius dependent stiffness, which may be important in pulmonary arteries [207] and could change the model sensitivity to network size and vessel dimensions. However, the focus of this study was to quantify how changes in the model domain attributed to pre-segmentation parameters impact haemodynamic predictions for fixed material parameters. The imaging process has inherent uncertainties that affect the image intensities in the data, which subsequently affect the ability to accurately segment and reconstruct the vasculature. These uncertainties ultimately propagate to the 3D meshing and estimation of vessel centerlines, including the radius and length in the 1D model. Previous work has quantified the uncertainty in 3D coronary artery geometry using both stochastic collocation methods and machine learning, showing that uncertainty in stenosis severity can alter predictions in fractional flow reserve [168]. Future studies will integrate uncertainty at the level of segmentation and propagate this uncertainty to the radius estimation.

Conventional mean and standard deviation calculations were provided as familiar metrics for comparison. An alternative approach is to perform formal global sensitivity analysis. State-of-the-art methods are based on Sobol indices defined via conditional variances of different order [38, 90, 122, 168]; however, their computation via Monte Carlo or quasi-Monte Carlo simulations is computationally expensive. This computational complexity is aggravated by the fact that the image segmentation includes manual user input and the parameter space can therefore only be sampled at a coarse level. A potential way to alleviate this problem is to use statistical emulation, e.g. using GPs, to compute first order and total effects indices. This can, in principle, follow the method described in [122], by adapting and extending existing approaches and software tools; see https://github.com/samcoveney/maGPy. However, this exploration is beyond the scope of the present study and provides an interesting direction for future research. The frequently-used three-element Windkessel model are considered as the boundary condition for the 1D model, yet this model greatly simplifies the physiological resistance beyond the segmented vessels. In contrast, structured tree boundary conditions [138, 139, 152] can provide an additional level of complexity for approximating downstream resistance and attempt to capture network structure beyond the limits of image segmentation. In addition, the experimental protocol inhibited the same mouse from being used for both the haemodynamic and imaging data. Nevertheless, the current methodology still captures variability in model predictions due to uncertainty in the vessel dimensions and network structure. Future human-based studies could incorporate non-invasive flow and imaging data from the same patient in the model. Finally, future subject-specific models of the pulmonary vasculature would be enhanced by allowing for more physiological traits of the network, e.g. trifurcations.

5.5 Conclusions

This chapter presents the first known investigation of the impact of uncertainties in imaging-based network reconstruction on CFD simulations in the pulmonary vasculature. This work identifies the uncertainties pertaining to image pre-segmentation parameters by explicitly measuring the variation in radius and length measurements of a subset of vascular segments using state-of-the-art non-parametric techniques, as well as the variation in size and connectivity of an expansive pulmonary vascular network. Results showed that the network variation has the most influence on predictions of blood pressure and flow, while changes in vessel length and radius have less impact on haemodynamic predictions. For example, network variation produces roughly two times larger standard deviations in systolic and pulse pressures vs radius and length parameter variation.

Chapter 6

Accelerating MCMC with emulation

In this chapter MCMC with emulation of the unnormalized log posterior distribution using Gaussian Processes is adopted as a viable parameter estimation and uncertainty quantification tool in computationally expensive models described by differential equations. Such models typically incur onerous computational costs due to repeated numerical integrations as part of an adaptive parameter estimation procedure. Gradient-driven Hamiltonian/Lagrangian algorithms: Hamiltonian Monte Carlo (HMC), No U-turn sampler (NUTS), Riemann Manifold Riemann Manifold Hamiltonian Monte Carlo (RMHMC) and Lagrangian Dynamical Monte Carlo (LDMC) coupled with emulation are explored, both in a delayed acceptance framework and a standard framework (no delayed acceptance).

In the first part of this chapter a comparative evaluation study is performed to assess the performance of the methods proposed on a series of models described by differential equations: ordinary and partial differential equations, including a 1D fluid-dynamics model of the pulmonary blood circulation. In the second part of this chapter full proofs of convergence to the asymptotically exact posterior distribution for the proposed algorithms are provided.

Note: This chapter is adapted from two papers on which I am the first author: ‘Emulation-accelerated Hamiltonian Monte Carlo algorithms for parameter estimation and uncertainty quantification in differential equation models’ (in preparation), ‘MCMC with Gaussian Processes for fast parameter estimation and uncertainty quantification in a 1D fluid-dynamics model of the pulmonary circulation’ (submitted to IJNMBE).

6.1 Introduction

Parameter estimation and uncertainty quantification (UQ) in systems of non-linear ordinary and partial differential equations (ODEs/PDEs) is a topical research area given the emergence of complex mathematical models expressed via ODEs or PDEs. Such models are heavily used throughout all science and engineering fields to understand the underlying mechanisms behind a process (e.g. biological systems [212], or physiology [124]). However, mathematical modelling on its own cannot be used in practice as the differential equations depend on unknown parameters that typically cannot be measured directly, and thus need to be estimated from limited and noisy measurements. This is when statistical inference becomes an invaluable tool, allowing estimation of these parameters in a robust and coherent manner within a Bayesian or frequentist framework.

Parameter estimation is however a challenging task to accomplish as the ODEs/PDEs cannot be solved analytically, instead they must be integrated using numerical schemes. While this may not be a problem if the numerical integration is performed only a few times, it quickly becomes a major hindrance if incorporated within an adaptive parameter estimation procedure requiring thousands of ODE/PDE evalutions, incurring high computational costs. In addition, non-identifiable parameters caused either by the model formulation or the insufficient amount of data, and any strong parameter correlations further complicates the statistical analysis. Point estimate approaches based on maximising the match between the measurements and the data simulated via the ODEs/PDEs (by minimising an objective function, typically the Euclidean distance) do not allow proper exploration of all likely parameter values, thus ignoring their uncertainty. In contrast, Bayesian methods employing probabilistic models naturally overcome these issues by providing probability distributions over parameters. These methods combine the data likelihood and the prior distribution into the posterior distribution using Bayes Theorem, and MCMC algorithms can subsequently be used to sample from this distribution; the samples drawn (generally tens of thousands) are likely to have generated the measured data. For a review on uncertainty and variability in computational and mathematical models described by PDEs with an application to cardiac physiology, the reader is referred to [124]. In addition, Wilkinson [212] reviews the application of Bayesian methods to several biological systems defined using ODEs in bioinformatics (e.g. protein informatics) and computational systems biology (e.g. quantitative network models).

The main disadvantage of the MCMC Bayesian methods is that the data likelihood can only be calculated by a numerical integration of the ODEs/PDEs, carried out repeatedly for different parameter values, thus rendering the sampling process slow. In addition, finding efficient algorithms that return high effective sample sizes (ESS) in a reasonable time frame is challenging, especially if there are strong correlations between the parameters, which would retard the convergence of standard MCMC algorithms, such as Metropolis-Hastings (M-H) [196]. The implication of using standard MCMC algorithms on such problems is that a small step size is needed to obtain a reasonable acceptance rate, which in turn means that low ESS (or high auto-correlation) is obtained. This problem can be alleviated by using more advanced MCMC algorithms, such as the HMC algorithm [132]. HMC introduces an auxiliary variable and makes use of the gradient of the posterior distribution for more informed moves in parameter space.

However, while it has been shown on numerous occasions that the HMC algorithm outperforms the random-walk algorithms in terms of efficiency (e.g. for a 100-dimensional multivariate Gaussian distribution as a target density, see Ch. 5 in [17] or [174]), it has rarely been applied to non-linear ODE or PDE models. Four noticeable exceptions are [102], where HMC is used to infer the parameters of an ODE model of intracellular processes, [117], where HMC is applied to a PDE-based model of tumor growth, [22], where an HMC extension, the Riemann Manifold HMC (RMHMC) is employed for parameter inference in a PDE model of steady state heat conduction, or [174], where HMC algorithms are applied to a set of ODEs describing dynamic causal models.

The major drawback of applying HMC to ODE/PDE models is the high computational cost associated with numerically integrating the ODEs/PDEs a large number of times. HMC trajectories are simulated by following a set of deterministic Hamiltonian dynamics steps in parameter space. Throughout each trajectory, the ODE/PDEs are evaluated multiple times (for calculating the likelihood and its gradient) until a proposal is made, unlike the M-H algorithm, which requires one single ODE/PDE evaluation for a proposal to be made. To reduce the computational burden, several approaches have been proposed in the literature. In a special class of ODE models (steady state data models), Kramer et al. [102] make use of a special property of steady state data to obtain output sensitivities (i.e. derivatives of the model output with respect to the unknown parameters) required in the Hamiltonian equations through analytical calculations. Given that most ODE/PDE models are dynamic time data models for which the output sensitivities can only be obtained via numerical integration, the approach adopted in [102] is not generalisable to any ODE/PDE model. Sengupta et al. [174] compare the performance in terms of computational speed and accuracy of three methods for calculating the likelihood gradients: finite differences[1], forward sensitivities[2] and the adjoint method[3], and the latter was shown to be superior (specific details of these methods can also be found in [173]). Similarly, Bui-Thanh et al. [22] apply the adjoint method in a PDE system to compute the first, second and third order derivatives of the likelihood as part of the RMHMC algorithm. The approaches taken in [22, 174] can nevertheless still be too computationally expensive for large class problems.

The study in [31] introduces the stochastic-gradient HMC, and the main idea is to subsample the data, which introduces noise, and thus the full-data gradients in the Hamiltonian equations are replaced by stochastic gradients, which can result in reduced exploration efficiency and accuracy [9]. Yet another approach proposed to speed up HMC involves the replacement of the expensive likelihood (or posterior distribution) with computationally cheaper surrogate models [156, 223] for the likelihood. Surrogate models can be split into three categories: data-driven, projection-based reduced models, and multi-fidelity (hierarchical) models. Data-driven models (data fit interpolation and regression models) are constructed by an input-output map for the

[1] $\frac{df(x)}{dx} = \frac{f(x+h)-f(x)}{h}$, where $h > 0$ is a small constant

[2] The ODE system is augmented to include the gradient of the states with respect to the parameters (called the sensitivity derivative equation)

[3] Avoids solving the sensitivity derivative equation by the use of a linear adjoint equation which is easier to numerically integrate

original (high-fidelity) model, and methods to do this include polynomial chaos expansion [164], Gaussian Processes (GPs), also known as Kriging, [41, 42, 43, 143, 157, 169], multivariate adaptive regression splines [176], random basis functions [223], neural networks [30], support vector machines [145]; for a review on data-driven surrogate models, the reader is referred to [3, 6, 158]. Projection-based models reduce the parameter space dimensionality by projecting the governing equations into a basis of orthonormal vectors – see [6] for a review and [61, 189] for an application to structural dynamics, and to large-scale statistical inverse problems. Multi-fidelity models are surrogate models created from the original (high-fidelity, complex) model by reducing the numerical resolution, or by simplifying the physical process – see [6] for a review and [147, 148] for applications to cardiac electrophysiology.

Given that the models considered in this chapter have a fairly low number of parameters, a probabilistic inference method based on GPs is used to construct a surrogate for the complex (high-fidelity) model. Now HMC can be coupled with the surrogate model [222], and two main approaches can be taken to accomplish this: drawing samples from the approximate posterior distribution defined by the surrogate model ('emulator'), or drawing samples from the asymptotically exact posterior distribution ('simulator'). The first route, taken in multiple studies [10, 43, 51, 57, 140, 143, 170, 213], samples from the surrogate posterior distribution, resulting in substantial gains in computational efficiency (since the expensive model is no longer used), however the accuracy is sacrificed. To date, no study employing this approach has presented a way to control the error introduced by the bias (though Cotter et al. [44] theoretically show that the bias could in principle be bounded, but no practical suggestions to do so are offered).

However, there are methods that correct for the bias by using the surrogate for the proposal only (see studies in [156, 223]), or by incrementally refining the surrogate model as MCMC proceeds, while the asymptotic exactness is ensured [223]. Other studies employ similar approaches guaranteeing asymptotic convergence to the posterior distribution [39, 40, 77, 217], e.g. the study in [40] uses forward simulations from the expensive model to continually refine a local approximation of the log unnormalised posterior, however the algorithm depends on various heuristic parameters, which critically affect the computational efficiency and may be difficult to tune in practice. The delayed acceptance (DA) scheme, introduced in [33, 76] is another exact method, and it does not depend on any heuristically set terms. This method, employed in [33, 47, 85, 7, 149, 177], is a two-stage acceptance procedure, with two separate acceptance/rejection decisions. The first decision is a computationally fast pre-filter step based on the surrogate model, which upon rejection of a proposed new parameter avoids carrying out the computationally expensive second step based on the original model.

The current study follows an idea adopted by studies in [108, 156] and employs HMC in combination with statistical emulation using Gaussian Processes(GPs). An emulator is thus created for the log unnormalised posterior distribution. Throughout the trajectory, HMC runs at low computational costs in the surrogate space, and the Metropolis-Hastings acceptance/rejection step at the end of the trajectory is based on the ratio of the true posterior distributions. This requires one single numerical integration of the ODEs/PDEs throughout the HMC trajectory (to obtain one parameter proposal), which substantially reduces the computational complexity. The algorithm is exact in the sense of converging to the true posterior distribution asymptotically, assuming no discretisation errors are introduced from the numerical integration of the ODEs/PDEs[4].

In addition, the current work extends this framework to algorithms which advance HMC: No U-turn sampler (NUTS) [87], RMHMC [74], and Lagrangian Dynamical Monte Carlo (LDMC) [109], with Bayesian optimisation for hyperparameter tuning [208]. RMHMC & LDMC make use of curvature information from the posterior distribution through second-order derivatives, which is an advantage over HMC or NUTS, that only utilise first-order derivatives. These emulation Hamiltonian/Lagrangian Monte Carlo algorithms are developed within a DA framework, and it is investigated if the DA scheme brings any computational gains over the standard algorithm. The methodological contribution consists of providing proofs of converge to the correct posterior distribution for all algorithms proposed. The correctness of the samplers' implementation is checked using the Geweke consistency test [71]. Moreover, the performance of these algorithms is critically assessed on a series of complex non-linear ODE/PDE models, and the comparison focuses on finding the algorithm which gives the best trade-off in terms of accuracy and efficiency.

6.2 Methods

6.2.1 HMC coupled with emulation using GPs

The HMC algorithm can be coupled with GPs as part of the GPHMC algorithm, an idea initially proposed by Rasmussen in [156]. The need for using the GPHMC algorithm over the plain HMC algorithm is illustrated in Algorithm 1f, which compares the two algorithms in terms of the total number of expensive model (ODE/PDE) evaluations required to calculate the data log likelihood and its gradient, which is attached in the caption of the algorithm. Specific differences between HMC and GPHMC are marked by the text in blue colour. Algorithm 1f is just a conceptual outline, the reader is referred to Algorithms 1h, 1i, 1j and 1k in Section D.2 for a detailed pseudocode.

Algorithm 1f clearly illustrates that HMC run in the original log likelihood space requires many more expensive model evaluations compared to HMC run in the surrogate log likelihood space since the former requires the numerical integration of the differential equations at every leapfrog step throughout the Hamiltonian dynamics (eq (2.28)), while the latter only requires

[4] investigation of the discretisation errors is beyond the scope of this study.

Algorithm 1f Conceptual outline for Hamiltonian Monte Carlo (HMC) vs HMC coupled with emulation using Gaussian Processes (GPHMC) algorithm. The total number of model evaluations required for running each algorithm is: HMC – $\mathbf{SL(d+1)}$ vs GPHMC – $\mathbf{S}$

```
Define a $d$-dimensional vector $\theta$ with $\theta_k$ the $k^{\text{th}}$ element, $k = 1 \ldots d$; S: number of HMC samples; L: number of HMC trajectory steps; $p(\mathbf{y}|\theta)$: simulator data likelihood (eq (2.19))
for i = 1 : S do
    for j = 1 : L do
        Solve ODEs/PDEs to get $\log p(\mathbf{y}|\theta^j)$ and $\frac{\partial \log p(\mathbf{y}|\theta^j)}{\partial \theta_k^j}$
    end for
    Propose $\theta^L$
    Solve ODEs/PDEs to get $\log p(\mathbf{y}|\theta^L)$ and accept/reject in a M-H step
end for
```

```
Define a $d$-dimensional vector $\theta$ with $\theta_k$ the $k^{\text{th}}$ element, $k = 1 \ldots d$; S: number of HMC samples; L: number of HMC trajectory steps; $\tilde{p}(\mathbf{y}|\theta)$: emulator data likelihood (eq (6.2))
for i = 1 : S do
    for j = 1 : L do
        Use GPs to predict $\log \tilde{p}(\mathbf{y}|\theta^j)$ and $\frac{\partial \log \tilde{p}(\mathbf{y}|\theta^j)}{\partial \theta_k^j}$
    end for
    Propose $\theta^L$
    Solve ODEs/PDEs to get $\log p(\mathbf{y}|\theta^L)$ and accept/reject in a M-H step
end for
```

one single numerical integration at the end of the HMC trajectory per HMC sample. The implication is excessive computational costs, which can be substantially reduced by moving the HMC scheme to the surrogate posterior space defined by the statistical emulator:

$$\tilde{p}(\theta|\mathbf{y}) \propto \tilde{p}(\mathbf{y}|\theta)p(\theta), \tag{6.1}$$

where $p(\theta)$ is the prior distribution, and $\tilde{p}(\mathbf{y}|\theta)$ is the surrogate data likelihood given by

$$\tilde{p}(\mathbf{y}|\theta, \sigma^2) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{\tilde{\mathscr{S}}(\theta)}{2\sigma^2}\right), \tag{6.2}$$

where $\tilde{\mathscr{S}}(\theta)$ is the value of the residual sum-of-squares $\sum_{i=1}^n (y_i - m_i(\theta))^2$ predicted by the emulator for the particular θ.

An HMC trajectory typically has in the order of $L = 100$ to 1000 steps, which if carried out in the original space would require in the order of $100 \times (d+1)$ to $1000 \times (d+1)$ (d: number of parameters) expensive model evaluations per HMC sample, thus a reduction in the computational complexity by about two to three orders of magnitude is obtained. The term $d+1$ is the sum of one ODE/PDE evaluation to find the log likelihood, and d ODE/PDE evaluations to find the numerical derivatives using a first-order differencing scheme with respect to each of the d parameters. The GPHMC algorithm is guaranteed to converge to the correct posterior distribution, see Section 6.9 for the proof. Therefore, even if the emulator is a poor representation of the simulator, the algorithm is still mathematically guaranteed to converge to the true posterior distribution, though at a decreased convergence rate.

6.3 Methodological contribution

6.3.1 GPHMC extensions

The first methodological contribution is to provide a proof of convergence to the correct posterior distribution for Rasmussen's GPHMC algorithm [156] (see Section 6.9.3), since this proof seems to have been omitted from the paper in which it was initially proposed [156], or from subsequent studies using this algorithm [117]. Furthermore, the GPHMC algorithm is extended to allow other HMC improvement algorithms, namely NUTS, RMHMC and LDMC, and proofs of convergence for these algorithms are also provided (see Sections 6.9.4, 6.9.6 and 6.9.8). In addition, a novel scheme which embeds the GPHMC algorithm into a delayed acceptance [76] framework is proposed, to investigate whether the DA scheme can bring computational gains over the standard GPHMC algorithm. In literature [76, 177] it has been hypothesized that the delayed acceptance scheme could potentially speed up simulations. This scheme, proposed in [33], and slightly modified by Sherlock et al. [76, 177], is a two-stage acceptance procedure, with two M-H acceptance/rejection steps. The first step is a computationally fast pre-filter step, in which proposed values are accepted/rejected based on the emulator. Detailed balance with respect to the true posterior distribution is ensured through the second M-H step, which invokes the original posterior distribution for the acceptance/rejection of the those samples accepted in the first step. In other words, the scheme avoids carrying out the expensive step for samples rejected in the first step by the emulator, which justifies the potential computational gain. As in [76, 177], the first acceptance probability based on the emulator is given in eq (6.3) and the second acceptance probability based on the simulator is expressed in eq (6.4):

$$\alpha_1(\boldsymbol{\theta}^*|\boldsymbol{\theta}) \;=\; 1 \wedge \frac{\tilde{p}(\boldsymbol{\theta}^*|\mathbf{y})q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}{\tilde{p}(\boldsymbol{\theta}|\mathbf{y})q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}, \tag{6.3}$$

$$\alpha_2(\boldsymbol{\theta}^*|\boldsymbol{\theta}) \;=\; 1 \wedge \frac{p(\boldsymbol{\theta}^*|\mathbf{y})}{p(\boldsymbol{\theta}|\mathbf{y})}\frac{\tilde{p}(\boldsymbol{\theta}|\mathbf{y})}{\tilde{p}(\boldsymbol{\theta}^*|\mathbf{y})}, \tag{6.4}$$

where $\tilde{p}(.)$ is the approximate posterior distribution, constructed using the emulator, $p(.)$ is the exact posterior distribution, obtained using the simulator, and $q(.)$ is the proposal distribution.

Proofs of convergence for the GPHMC algorithm and its improvement algorithms coupled with DA are provided in Sections 6.9.2, 6.9.5 and 6.9.7. To check the mathematical and coding correctness of all the samplers proposed, the Geweke consistency test [71], reviewed in Section 2.3.6, is employed.

6.3.2 Adapted GPHMC for unknown constraints

Furthermore, the GPHMC algorithm is extended to handle a priori unknown constraints on the joint parameter space, which is addressed by constructing a GP classifier, that automatically learns the infeasible parameter regions. For some complex biophysical models, there are certain parameter values and combinations for which the underlying physical assumptions of the model are violated, or the solver used is inappropriate for the problem, producing no outputs from the simulation software. While the second matter can be tackled by e.g. trying a different solver, decreasing the convergence threshold value, or increasing the number of discretisation steps, there is no clear resolution for the first issue. This may be caused by the inappropriateness of the parameter ranges, which are generally chosen by varying one parameter at a time, while fixing all the other parameters to biologically realistic values. However, when parameters are simultaneously changed, e.g. in the fluid-dynamics model, given the fixed network geometry and inflow, the resulting parameter combinations may break physiological assumptions (e.g. a large arterial stiffness may not be compatible with high compliance downstream). We stress that a multivariate classifier should only be implemented upon thoroughly checking the suitability of the solver. For parameter values in the 'invalid' domain, different solvers may be tried. If the crash is independent of the solver used, this suggests that the crash is of a more fundamental nature (i.e. violation of the physical model assumptions).

In a standard MCMC simulation based on the actual model, an invalid parameter vector can be assigned zero likelihood. Hence, if such a parameter vector is proposed, it will be rejected in the Metropolis-Hastings acceptance/rejection step. However, dealing with such invalid regions in the context of emulation requires some extra care. A naive and straightforward approach is to set the likelihood for an invalid parameter vector to a very small value close to zero (i.e. the log likelihood to a negative value with large modulus) when training the GP emulator. However, this approach is unlikely to lead to a good emulator. A sudden shift to an extreme value will drive the lengthscale of the GP kernel to a very small value in the hyperparameter estimation step, which in turn will cause ripples in the GP interpolant and hence overfitting in the valid regime. This issue is addressed by introducing a GP classifier. Let λ denote a binary variable to indicate if the parameter vector θ falls into a valid regime ($\lambda = 1$) or invalid regime ($\lambda = 0$). Given a set of parameter vector - label pairs obtained during the initial design and exploration phase of the GPHMC algorithm (see below for details),

$$\mathscr{H} = \{(\theta_1, \lambda_1), \ldots, (\theta_n, \lambda_n)\}$$

a GP classifier (reviewed in Section 2.3.7) can be trained to predict the probability $p(\lambda = 1|\theta, \mathscr{H})$. These probabilities can be used to modify the prior:

$$\tilde{p}(\theta) = p(\theta)p(\lambda = 1|\theta, \mathscr{H})/Z, \tag{6.5}$$

where $Z = \int p(\boldsymbol{\theta})p(\lambda = 1|\boldsymbol{\theta},\mathscr{H})d\boldsymbol{\theta}$ is a normalisation constant, which cancels out in the Metropolis-Hastings acceptance/rejection step. When running in the emulated space, the sampler uses the modified prior $\tilde{p}(\boldsymbol{\theta})$ instead of the original prior $p(\boldsymbol{\theta})$ to avoid moving into invalid parameter regions (where the probability of success $p(\lambda = 1|\boldsymbol{\theta},\mathscr{H})$ is low). A similar idea of combining a GP classifier with a GP emulator has been proposed in the context of Bayesian optimisation with unknown constraints [65, 135].

Pseudocode for GPHMC coupled with a GP classifier can be found in Algorithm 11. A diagram summarising Rasmussen's GPHMC algorithm can be seen in Figure 6.1. Figures 6.2 and 6.3 contrast HMC without and with the DA method, both of which are used within the GPHMC algorithm [156].

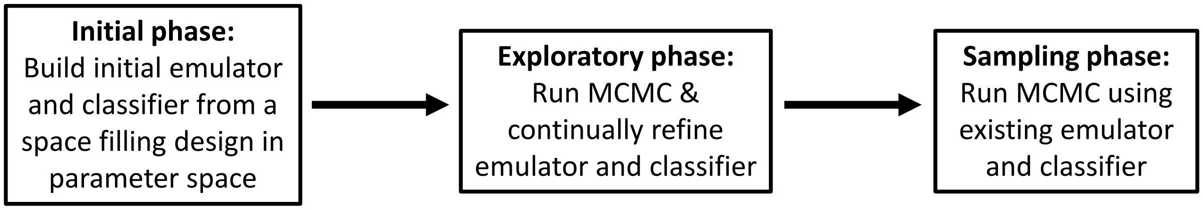

Initial phase:
Build initial emulator and classifier from a space filling design in parameter space
Exploratory phase:
Run MCMC & continually refine emulator and classifier
Sampling phase:
Run MCMC using existing emulator and classifier

Figure 6.1: Workflow of the GPHMC algorithm [156]. The emulator and classifier constructed in the initial phase are continually refined as HMC is run in the exploratory phase. HMC in the sampling phase proceeds by drawing samples from the asymptotically exact posterior distribution, with the use of the emulator and classifier, which are no longer updated.

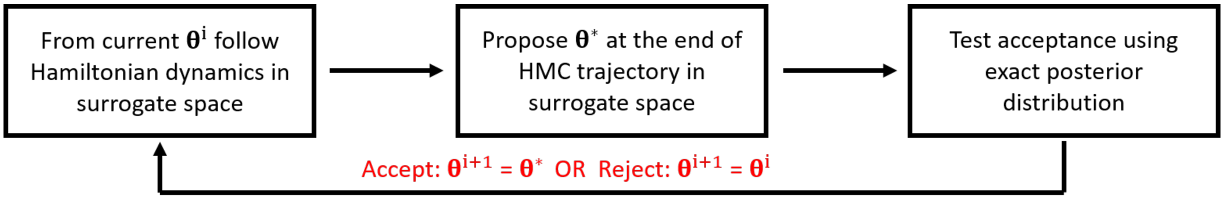

From current $\boldsymbol{\theta}^i$ follow Hamiltonian dynamics in surrogate space
Propose $\boldsymbol{\theta}^*$ at the end of HMC trajectory in surrogate space
Test acceptance using exact posterior distribution
Accept: $\boldsymbol{\theta}^{i+1} = \boldsymbol{\theta}^*$ OR Reject: $\boldsymbol{\theta}^{i+1} = \boldsymbol{\theta}^i$

Figure 6.2: Standard (no delayed acceptance) HMC algorithm used within the GPHMC algorithm [156].

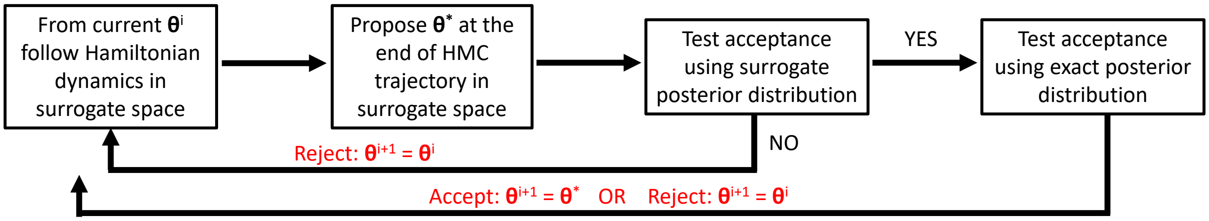

From current $\boldsymbol{\theta}^i$ follow Hamiltonian dynamics in surrogate space
Propose $\boldsymbol{\theta}^*$ at the end of HMC trajectory in surrogate space
Test acceptance using surrogate posterior distribution
YES
Test acceptance using exact posterior distribution
NO
Reject: $\boldsymbol{\theta}^{i+1} = \boldsymbol{\theta}^i$
Accept: $\boldsymbol{\theta}^{i+1} = \boldsymbol{\theta}^*$ OR Reject: $\boldsymbol{\theta}^{i+1} = \boldsymbol{\theta}^i$

Figure 6.3: Delayed acceptance HMC algorithm used within the GPHMC algorithm [156].

GPHMC coupled with a classifier can be described as follows:

- *Initial design stage.* Starting from a space filling design in parameter space to capture the compact support of the parameters, e.g. using a Latin hypercube [119] or a Sobol sequence

[16], integrate the ODEs/PDEs numerically for each parameter vector to get the true data log likelihood values and the success labels. Use these points to build a GP emulator and a GP classifier. Only those parameter vectors which yield successful ODE/PDE simulations are added to the list of training points for the GP regression model. All points regardless of whether or not they provide a successful simulation are added to the list of training points for the GP classifier, to enable the classifier to learn the infeasible regions that break the biophysical assumptions.

- *Exploratory phase.* Gather information about the target distribution by running HMC on the emulated surrogate log posterior of the PDE parameters, with the proposed point at the end of the HMC trajectory being subject to a M-H step based on the true posterior distribution (see Section 6.9 for more details on this). The emulator and classifier are sequentially refined (i.e. optimum covariance hyperparameters are found by maximisation of the log marginal likelihood of the GP training points – see eq (2.64) or (2.79)) as new points are accepted. Accepted parameter vectors are iteratively added as further training points to those used in the initial design stage. The points in the initial design stage are gradually removed from the list of training points as they tend to come from low posterior density regions and can bias the inference results. As HMC explores the parameter space, it gets closer to the equilibrium distribution (burn-in phase). This ensures the algorithm sequentially zooms into the regions of high posterior probability. Following [156], the emulated 'potential energy' of the HMC algorithm (see Section 2.3.3) is set to

$$\tilde{E}(\theta^*) = \frac{\mathbb{E}(f(\theta^*)|D) - \sqrt{\mathrm{var}(f(\theta^*)|D)}}{2\sigma^2} + \frac{n}{2}\log(2\pi\sigma^2) - \log\tilde{p}(\theta^*). \tag{6.6}$$

  Here σ^2 is the variance of the errors, $\tilde{p}(\theta^*)$ is the modified prior distribution, see eq (6.5) (which is common to both the emulator and the simulator), $f(.)$ is the emulated residual sum-of-squares function, $\mathbb{E}(f(\theta^*)|D)$ is the GP posterior predictive mean given the training points D (see eq (2.66) or (2.74)) and $\sqrt{\mathrm{var}(f(\theta^*)|D)}$ is the GP posterior predictive standard deviation (see eq (2.67) or (2.75)) for the residual sum-of-squares of the data at unseen parameter configurations θ^* conditional on the training points D. This drives the exploration into regions of high posterior probability (low value of $\mathbb{E}(.)$) or high uncertainty (large value of $\sqrt{\mathrm{var}(.)}$). If $\sqrt{\mathrm{var}(.)}$>3 (for residual sum-of-square values standardised to unit variance) along the HMC trajectory, the algorithm steps into a region of high uncertainty, where the GP needs to be further trained, thus the simulation is stopped prematurely before reaching the end of the trajectory and an expensive model evaluation is performed at this point. This follows the same exploitation-exploration trade-off principle as in optimistic acquisition functions used for Bayesian optimisation, see Section IV in [175].

- *Sampling phase.* Use the emulator and the classifier constructed in the exploratory phase to draw samples from the target distribution using HMC, or any of its variants: NUTS, RMHMC, LDMC. At this stage, the emulator and the classifier are no longer updated. The emulated 'potential energy' in the HMC algorithm is set to:

$$\tilde{E}(\theta^*) = \frac{\mathbb{E}(f(\theta^*)|D)}{2\sigma^2} + \frac{n}{2}\log(2\pi\sigma^2) - \log\tilde{p}(\theta^*). \tag{6.7}$$

Note that the numerator in the first term is the expected sum-of-squares error, which combined with the normalisation term in the middle gives the log likelihood of the data, and the final term is the log prior. Proposed points are accepted/rejected in a M-H step according to the simulator (see Section 6.9 for more details). If the rejection rate is too high (e.g. above 35%), this indicates that the emulated posterior distribution is not an accurate enough representation of the original posterior distribution[5] and an extension of the exploratory phase is needed.

Note that for problems where there is no evidence of constraints in parameter space, a classifier is not needed, and the original prior distribution $p(\theta^*)$ is used in eqns (6.6) and (6.7), and not the modified prior distribution $\tilde{p}(\theta^*)$ expressed in eq (6.5).

In the GPHMC algorithm, derivatives of the emulated log posterior are calculated by using the fact that GPs are closed under differentiation, i.e. the derivatives of a GP are also GPs (though with different covariance structures), provided the kernel is differentiable, see Section 2.3.7 for details and Section D.1 for specific derivative forms up to and including third order.

6.3.3 Setting the mass matrix for RMHMC/LDMC

For second-order algorithms (RMHMC, LDMC) obtaining the metric tensor is necessary. This is not trivial when emulating the objective function, and this section contains a discussion on the approach taken.

Eqns (2.39) and (2.40) show what the mass matrix can be set to to ensure that it is a proper metric tensor, i.e. positive definite, and distances $\Delta\theta^{\mathrm{T}}\mathbf{M}(\theta)\Delta\theta$ between probability distribution functions $p(\mathbf{y},\theta)$ and $p(\mathbf{y},\theta+\Delta\theta)$ on the Riemann manifold are coordinate-independent (see Section 3.2 in [24] for more details, in particular eq (3.28)). Unfortunately, given that in our work we emulate the objective function (residual sum-of-squares) instead of the model output, this results in loss of information, and hence inability of calculating the expectation in eqns (2.39) and (2.40). More precisely, for the model defined in eq (2.19), we may take the log to obtain

$$\log p(\mathbf{y}|\theta,\sigma^2) = \left(-\frac{n}{2}\log(2\pi\sigma^2)\right) - \frac{\sum_{i=1}^{n}(y_i - m_i(\theta))^2}{2\sigma^2}, \tag{6.8}$$

[5] The accuracy of the emulator can be checked by diagnostics [8].

and the corresponding first and second-order derivatives are:

$$\frac{\partial \log p(\mathbf{y}|\theta,\sigma^2)}{\theta_j} = \frac{1}{\sigma^2}\sum_{i=1}^{n}\left((y_i - m_i(\theta))\frac{\partial m_i(\theta)}{\partial \theta_j}\right), \tag{6.9}$$

$$\frac{\partial^2 \log p(\mathbf{y}|\theta,\sigma^2)}{\theta_k\theta_j} = \frac{1}{\sigma^2}\sum_{i=1}^{n}\left(-\left(\frac{\partial m_i(\theta)}{\partial \theta_k}\frac{\partial m_i(\theta)}{\partial \theta_j}\right) + (y_i - m_i(\theta))\frac{\partial^2 m_i(\theta)}{\partial \theta_k \partial \theta_j}\right). \tag{6.10}$$

By using the fact that

$$\mathbb{E}(\mathbf{y}|\theta) = \mathbf{m}(\theta), \tag{6.11}$$

we can negate the expression in eq (6.10) and take the expectation with respect to $\mathbf{y}|\theta$:

$$\mathbb{E}_{\mathbf{y}|\theta}\left(-\frac{\partial^2 \log p(\mathbf{y}|\theta,\sigma^2)}{\theta_k\theta_j}\right) = \frac{1}{\sigma^2}\sum_{i=1}^{n}\left(\frac{\partial m_i(\theta)}{\partial \theta_k}\frac{\partial m_i(\theta)}{\partial \theta_j}\right). \tag{6.12}$$

Given that we emulate the expression $\sum_{i=1}^{n}(y_i - m_i(\theta))^2$, we do not know the individual terms inside the sum, hence taking the expectation of the expression in eq (6.12) is impossible. Instead, we set $\mathbf{M}$ to be the observed Fisher Information matrix (the matrix of negative second-order derivatives of the log likelihood), plus the negative Hessian matrix of the log prior, i.e.

$$M_{i,j} = -\frac{\partial^2 \log p(\mathbf{y},\theta)}{\partial \theta_i \partial \theta_j} = -\frac{\partial^2 \log p(\mathbf{y}|\theta)}{\partial \theta_i \partial \theta_j} + \left(-\frac{\partial^2 \log p(\theta)}{\partial \theta_i \partial \theta_j}\right). \tag{6.13}$$

However, this is not guaranteed to be positive definite, and so it is not a proper metric tensor, unlike the quantity involving the expected Fisher information matrix in eqns (2.39) and (2.40). In addition, distances $\Delta\theta^{\mathrm{T}}\mathbf{M}(\theta)\Delta\theta$ between probability distribution functions $p(\mathbf{y},\theta)$ and $p(\mathbf{y},\theta+\Delta\theta)$ on the manifold may depend on the model parameterisation, i.e. under reparameterisation $\theta \rightarrow \phi$, $\Delta\phi^{\mathrm{T}}\mathbf{M}(\phi)\Delta\theta \neq \Delta\theta^{\mathrm{T}}\mathbf{M}(\theta)\Delta\theta$. This further invalidates the term involving the observed Fisher Information matrix in eq (6.13) as a proper metric tensor on Riemann manifolds (see Section 3.2 in [24] for more details).

We consider another form for the metric tensor:

$$M_{i,j} = \frac{\partial \log p(\mathbf{y},\theta)}{\partial \theta_i}\frac{\partial \log p(\mathbf{y},\theta)}{\partial \theta_j}, \tag{6.14}$$

which makes use of the empirical Fisher information matrix for one single data set. In expectation, the terms in eqns (6.14) and (6.13) are expected to be similar for a large number of data sets and uninformative priors. Should we have multiple data sets, the empirical Fisher information plus the negative Hessian for the log prior is the mean over all data sets:

$$M_{i,j} = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{\partial \log p(\mathbf{y}^k,\theta)}{\partial \theta_i}\frac{\partial \log p(\mathbf{y}^k,\theta)}{\partial \theta_j}\right). \tag{6.15}$$

The quantity in eq (6.14) is positive semi-definite by definition, however we found that its inversion can sometimes be numerically unstable due to a high condition number of the matrix (which can happen if the minimum eigenvalue is very close to zero, while the maximum eigenvalue is orders of magnitude larger, making their ratio, i.e. the condition number large, $> 10^{15}$). In addition, it has been shown that the convergence of the algorithm used may be negatively affected by a small K, see Section 3.2 in [24] (for $K \to \infty$, the quantity in eq (6.15)) is the middle term in eq (2.40)).

An alternative form which we consider for the metric tensor is a hybrid version between forms in eqns (6.13) and (6.14):

$$M_{i,j} = \lambda \left(-\frac{\partial^2 \log p(\mathbf{y}, \theta)}{\partial \theta_i \partial \theta_j} \right) + (1 - \lambda) \left(\frac{\partial \log p(\mathbf{y}, \theta)}{\partial \theta_i} \frac{\partial \log p(\mathbf{y}, \theta)}{\partial \theta_j} \right), \tag{6.16}$$

where $\lambda \in [0, 1]$, is chosen to be large enough as to produce a positive definite $\mathbf{M}$. The form in eq (6.16) is motivated by the fact that the two terms are expected to be similar in expectation for a large number of data sets and uninformative priors.

In this work it was found that setting the metric tensor as per eq (6.16) does not work for all parameter values tried by the sampler (in some cases, parameter values throughout the trajectory tend to take extreme, unrealistic values, subsequently leading to warning messages from the ODE solver which is called for the end trajectory value). In that case, the approach taken is akin to a Quasi-Newton method in optimisation [19], i.e. if at any point throughout the trajectory, the mass matrix is numerically unstable, the simulation within the trajectory is stopped prematurely before reaching the end. A new simulation is started from the beginning of the trajectory and the HMC algorithm is used instead of RMHMC or LDMC for that particular iteration. The resulting posterior samples will have been drawn using a hybrid version of HMC and RMHMC or LDMC, and the proposed algorithm is called Quasi-HMC-RMHMC or Quasi-HMC-LDMC. This algorithm naturally satisfies detailed balance since each sample is drawn using a valid sampler (either HMC or RMHMC/LDMC).

Adopting a Hessian-based approach in MCMC has been done in a few previous studies. For example, Qi et. al. [146] used a modified version of the Hessian ($\mathbf{H} = \mathbf{H} + \lambda \mathbf{I}$ to ensure that the negative Hessian is positive definite) as the covariance matrix of the Gaussian proposal distribution in the M-H algorithm. In addition, Zhang et. al. [220] set the mass matrix in the HMC algorithm equal to the negative Hessian, which they compute based on a fixed window of past HMC samples. The authors take the past chain samples in a way that ensures that the negative Hessian is always positive definite, however this is not guaranteed to work for problems in which the log posterior density is not convex, since the negative Hessian is no longer positive definite. A similar idea is also adopted in another study on stochastic Quasi-Newton Langevin Monte Carlo algorithm [182]. Additionally, Dahlin et. al. [48] incorporated the Hessian into a particle M-H algorithm as a way to set the covariance of the proposal, and the authors transform the negative Hessian into a positive definite matrix by adding a diagonal matrix: $\mathbf{H} = \mathbf{H} + \lambda \mathbf{I}$, where $\lambda = \max(0, -2\lambda_{\min})$, where $\lambda_{\min}$ is the minimum negative eigenvalue. However, the workability of such an approach is limited by the strict assumption that the log posterior is convex, which is difficult to check in high-dimensional (>2) spaces.

6.3.4 Brief discussion on efficiency for all algorithms proposed

The noDA-GPNUTS algorithm requires evaluation of the differential equations several times (equal to the number of tree doublings, i.e. tree height) along the trajectory before making a proposal; see the proof in Section 6.9.8, in particular eq (6.34) on page 187, which shows that the simulator potential function $E(\theta)$ needs evaluating for every tree doubling j. In contrast, DA-GPNUTS only evaluates the ODEs/PDEs once at the end of the trajectory, for the final proposed point (as do all the other algorithms investigated: noDA-GPHMC, DA-GPHMC, noDA-GPRMHMC, DA-GPRMHMC, noDA-GPLDMC, DA-GPLDMC). This implies that the noDA-GPNUTS requires a much larger number of forward evaluations (roughly one order of magnitude larger) than the DA-GPNUTS. For this reason, this algorithm was not implemented. All the other algorithms: noDA-GPHMC (which is the standard GPHMC), DA-GPHMC, DA-GPNUTS, noDA-GPRMHMC, DA-GPRMHMC, noDA-GPLDMC, DA-GPLDMC were applied on a number of ODE/PDE problems, presented in Section 6.4.

6.4 ODE/PDE test examples

This section describes the test examples on which the methods outlined above were applied.

6.4.1 Sinusoidal example

The first example is a linear ODE model, that is a sinusoidal toy problem, defined via the expression

$$\frac{d^2 f}{dt^2} + B^2 f = 0, \tag{6.17}$$

or equivalently,

$$\frac{df}{dt} = z, \quad \frac{dz}{dt} + B^2 f = 0, \tag{6.18}$$

and which has an analytical solution:

$$f(t) = A \sin(B(t + C)), \tag{6.19}$$

where A: amplitude, $\frac{2\pi}{B}$: period, C: phase shift are the unknown parameters, estimated from the data, and $t \in [0, 2\pi]$, i.e. only one period is considered to ensure unimodality of the likelihood.

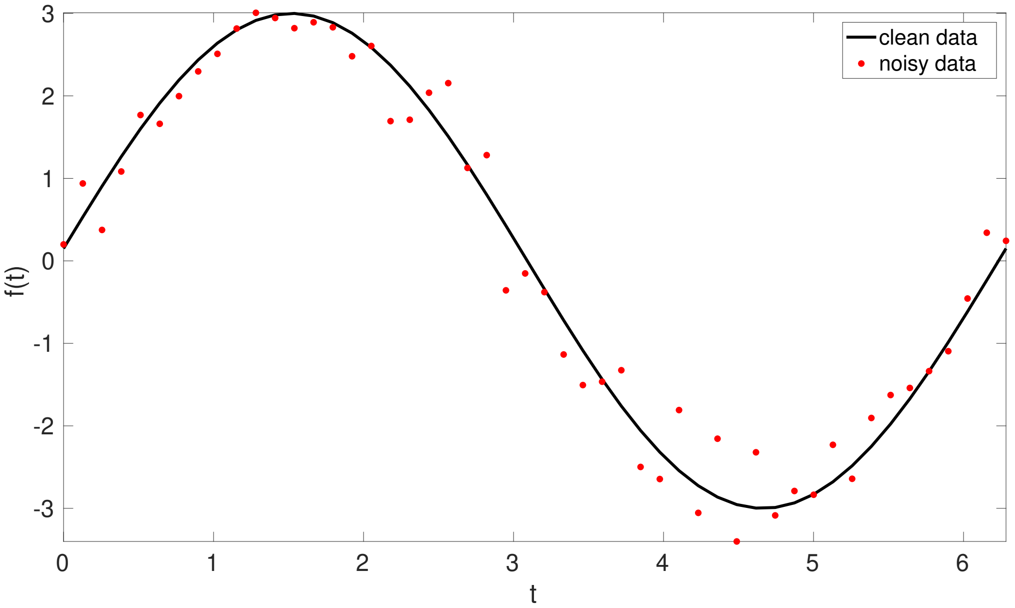


Figure 6.4: An example of noise-free data generated from the sinusoidal model using eq (6.17) (continuous black line). To this iid additive Gaussian noise with variance 0.12 was added (red dots), and the noisy data were used in the inference procedure.

This linear ODE example with closed form solution allows to check the mathematical and coding correctness of the samplers' implementation (using Geweke consistency test, reviewed in Section 2.3.6) at affordable costs.

Data were simulated at 50 equally spaced time points in the range $[0, 2\pi]$. The true parameter values were set as: $A = 3, B = 1$ and $C = 0.05$. Iid additive Gaussian noise was added to the clean data generated using eq (6.17), and the variance of the noise σ^2 was set to 0.12. An illustration of the data is given in Figure 6.4.

The parameters A, B, C were inferred, and the noise variance was kept fixed at its true value. This simplification was relaxed in the next examples. A Gaussian prior distribution was placed on the log of the parameters to ensure positivity ($\log A \sim \mathscr{N}(\log 4, 0.02)$, $\log B \sim \mathscr{N}(\log 1, 0.01)$, $\log C \sim \mathscr{N}(\log 0.05, 0.05)$). These priors were chosen to impose unimodality of the posterior distribution.

6.4.2 FitzHugh-Nagumo

The FitzHugh-Nagumo model, developed by FitzHugh [58] and Nagumo et al. [131] to model the behaviour of spike potentials in the giant axon of neurons, is defined as a non-linear ODE model:

$$\frac{dV}{dt} = \gamma\left(V - \frac{V^3}{3} + R\right), \quad \frac{dR}{dt} = \frac{1}{\gamma}(V - \alpha + \beta R). \tag{6.20}$$

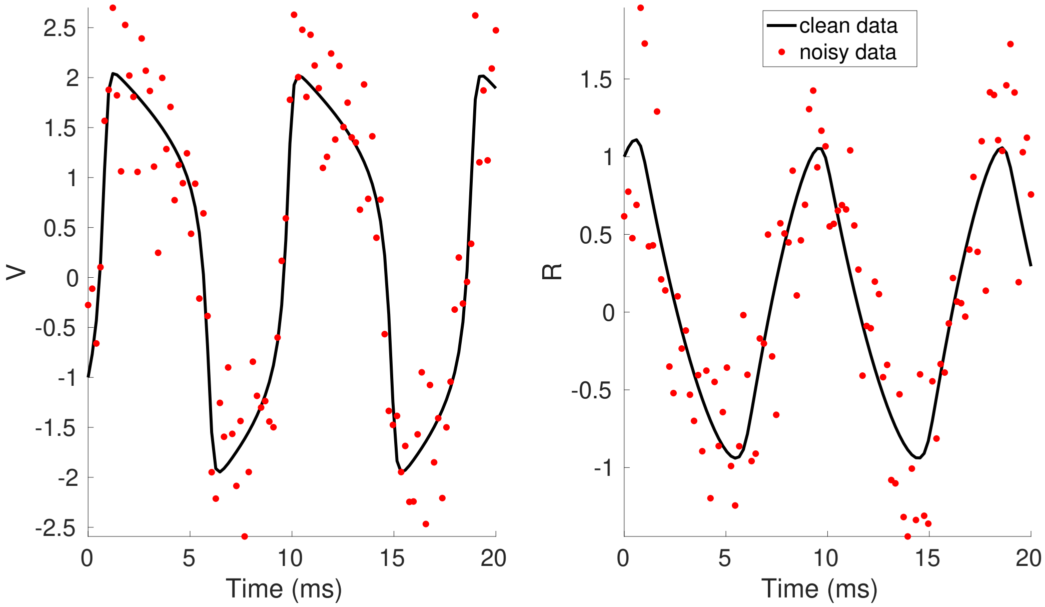


Figure 6.5: An example of noise-free data generated from the FitzHugh-Nagumo model using eq (6.20) (continuous black line). To these iid additive Gaussian noise with variance 0.25 - left signal and 0.16 - right signal (red dots) was added. Data from the two signals were used for the inference procedure.

The system describes the reciprocal dependency between the voltage V across an axon membrane (characterising the self-excitation of the axon membrane) and the recovery R, acting as outwards currents (providing a feedback response). The model has been used as a mathematical representation for cardiac dynamics [80] and neuro-degenerative diseases [21]. Equation (6.20) defines a highly non-linear likelihood surface [155] as the three parameters α, β, γ are varied.

Following [25], data from the model were generated with the following parameter values: $\alpha = 0.2, \beta = 0.2, \gamma = 3$, and initial values $(V_0, R_0) = (-1, 1)$. 100 time points equally spaced in the interval $[0, 20]$ ms were used. Iid additive Gaussian noise was added to the data, with the following variances: $\sigma_V^2 = 0.25, \sigma_R^2 = 0.16$. A depiction of the data can be seen in Figure 6.5.

The estimation procedure inferred five parameters: $\alpha, \beta, \gamma, \sigma_V^2, \sigma_R^2$ from data from the two "species" V and R. Following the study in Chapter 8 of [112], a Gamma(2,1) prior was set for parameters α, β, γ, while for the noise variances: σ_V^2, σ_R^2 a weakly informative Inverse-Gamma(0.001, 0.001) prior was used.

6.4.3 Biochemical signalling pathway

The biochemical signalling pathway model characterised by the non-linear ODE system in equations (6.21) uses the Michaelis-Menten kinetic law to describe the activation of a protein R into its active form R_{pp} in the presence of an enzyme S, followed by the degradation of the enzyme

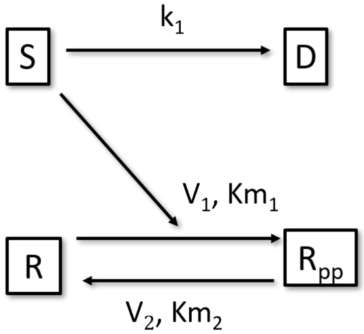


Figure 6.6: Graphical representation of the protein signalling pathway in eq (6.21). The model uses the Michaelis-Menten kinetic law to describe the activation of a protein R into its active form R_{pp} in the presence of an enzyme S, followed by the degradation of the enzyme into its inactive form D. Figure adapted from Chapter 8 in [112].

into its inactive form D (Figure 6.6), see Chapter 8 in [112],

$$\frac{dS}{dt} = -k_1 S, \quad \frac{dD}{dt} = k_1 S, \quad \frac{dR}{dt} = -\frac{V_1 RS}{K_{m_1}+R} + \frac{V_2 R_{pp}}{K_{m_2}+R_{pp}}, \quad \frac{dR_{pp}}{dt} = \frac{V_1 RS}{K_{m_1}+R} - \frac{V_2 R_{pp}}{K_{m_2}+R_{pp}}. \tag{6.21}$$

Cell signalling has been used in cancer modelling [118] and modelling of neuro-degenerative diseases [99]. The system of eqns in (6.21) depends on five kinetic parameters: k_1, V_1, K_{m_1}, V_2, K_{m_2} which control how fast the biochemical processes involving the five "species" (S, D, R, R_{pp}) take place.

Following the study in Chapter 8 of [112], data were generated from the model with the following parameter values: $k_1 = 0.05, V_1 = 0.2, K_{m_1} = 0.1, V_2 = 0.1, K_{m_2} = 0.1$, and initial values $(S_0, D_0, R_0, R_{pp_0}) = (1, 0, 1, 0)$. 20 data points were used within the interval $[0, 100]$ s, measured at time points {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 18, 20, 25, 30, 40, 60, 80, 100}. Iid additive Gaussian noise was added to the data, with the variance of 0.0004 for all species. A depiction of the data can be seen in Figure 6.7.

Five parameters: $k_1, V_1, K_{m_1}, V_2, K_{m_2}$ were estimated from data from the four "species": S, D, R, R_{pp}. Following the study in Chapter 8 of [112], a Gamma(1,1) prior was set for parameters k_1, V_1, K_{m_1}, V_2, K_{m_2}, while for the noise variances: $\sigma^2_S, \sigma^2_D, \sigma^2_R, \sigma^2_{R_{pp}}$, a weakly informative Inverse-Gamma(0.001, 0.001) prior was used.

6.4.4 Real-world application: fluid-dynamics model of the pulmonary blood circulation

The mathematical model used is indicated in Table 2.1 and is described in Section 2.2. The experimental data used are described in Section 2.1. Measured blood pressure (a time series consisting of 512 temporal pressure points) in the main pulmonary artery (MPA) was used to infer the various biophysical parameters: s, r_1, r_2, c in a fixed vessel network obtained from

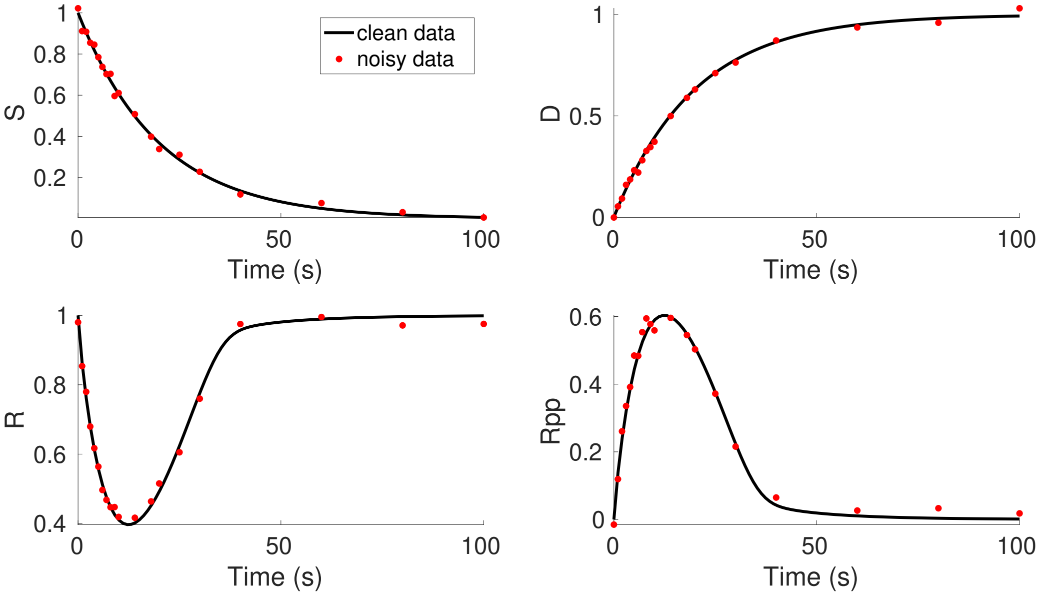


Figure 6.7: An example of noise-free data generated from the biochemical signalling pathway model using eq (6.21) (continuous black line). To these we added iid additive Gaussian noise with variance 0.0004 for all four signals (red dots). Data from all four signals were used for the inference procedure.

one image segmentation. The parameters estimated are a constant vessel stiffness and three Windkessel adjustment parameters: s, r_1, r_2, c, which lie within biologically plausible ranges, as established by the authors of [151]: $s \in [7 \times 10^4, 5 \times 10^5], r_1 \in [-0.5, 1.92], r_2 \in [-0.5, 1.0], c \in [-2.5, 1.5]$). A rescaled $\text{Beta}(1,1)$ prior for the parameters was used to enforce that they lie within the prescribed ranges, while for the noise variance σ^2 a weakly informative Inverse-Gamma(0.001, 0.001) prior was used (iid errors were assumed, see Chapter 7 for a discussion).

6.5 Simulations

6.5.1 Software

The statistical methods were implemented in *Matlab* (Mathworks, Natick, MA) and simulations were run on a RedHat Enterprise Linux 6 machine with Intel(R) Xeon(R) CPU E5-2680 v2 2.80GHz and 32GB RAM. The GP models were constructed using the *GPstuff* toolbox [203] and the MCMC convergence and efficiency diagnostics (multivariate potential scale reduction factor, MPSRF [18] and ESS [95]) made use of functions from the MCMC toolbox [107]. NUTS, RMHMC and LDMC algorithms were run using the Matlab implementations developed by the authors of the papers where these algorithms were proposed [74, 87, 109]. For the numerical integration of the ODEs, the *ode15s* Matlab solver was used for the FitzHugh-Nagumo model, and the *ode23* solver for the biochemical signalling pathway model (the change of solvers is motivated by stiffness of the system, i.e. ode15s is used for stiff systems). The PDEs of the 1D fluid-dynamics model were numerically integrated using a two step Lax-Wendroff scheme [113] implemented in *C*++ by Olufsen et al. [139, 151].

6.5.2 Method implementation details

GP kernel

For the GP emulator of the residual sum-of-squares, a squared exponential kernel (eq (2.54)) was used, which is infinitely differentiable[6], allowing to analytically compute[7] first-order derivatives of the GP posterior predictive mean and variance (eqns (D.1), (D.2), (D.6) and (D.7)) needed in HMC, and second and third-order derivatives of the GP posterior predictive mean, needed in RMHMC and LDMC (note that differentiation is a linear operator, so the derivative of a GP is again a GP for differentiable kernels; see Section 9.4 in [157]).

GP mean

For the sinusoidal, FitzHugh-Nagumo and pulmonary models a zero mean GP prior was used, while for the biochemical pathway model a second-order polynomial for the mean basis functions was used, e.g. in 5D:

$$\mathbf{h}(\mathbf{x}) = [1 \quad x_1 \quad \dots \quad x_5 \quad x_1^2 \quad \dots \quad x_5^2 \quad x_1x_2 \quad x_1x_3 \quad \dots \quad x_4x_5]^{\mathrm{T}}, \tag{6.22}$$

to discourage the emulated RSS values become zero outside the region where there are no data, and ensure high accuracy of the emulator in the high posterior probability region, where most data lie. In regions of no data, the posterior relies completely on the prior, hence for a zero mean GP prior, the emulated RSS values will tend towards zero. In addition, following the work by Riihimaki et al. [160], a weakly informative prior distribution for the regression coefficients β was imposed by setting $\mathbf{b} = \mathbf{0}$ and $\mathbf{B} = 10^2\mathbf{I}$, i.e. $\beta \sim \mathscr{MVN}(\mathbf{0}, 10^2\mathbf{I})$. A justification of using a non-zero mean GP prior for the biochemical model is given in Section 6.6.3.

[6] For example, the Matèrn 3/2 (eq (2.57)) or 5/2 kernels (eq (2.58)) cannot be used for the RMHMC and LDMC algorithms, as they are only once and twice respectively differentiable.

[7] The analytical derivative was checked against the numerical derivative; a difference in Euclidean space below a small threshold, e.g. 0.1%, was taken as an indication that the two were in agreement.

Multiple GPs

For problems with multiple "species" (biochemical and FitzHugh-Nagumo examples), the RSS for every "species" was emulated independently, thus eq (6.2) becomes

$$\tilde{p}(\mathbf{y}|\theta,\sigma^2) = \prod_{j=1}^{J}\left(\left(\frac{1}{\sqrt{2\pi\sigma_j^2}}\right)^{n_j}\exp\left(-\frac{\tilde{\mathscr{S}}_j(\theta)}{2\sigma_j^2}\right)\right), \tag{6.23}$$

where $\tilde{\mathscr{S}_j}(\theta)$ is the emulated RSS for the j^{th} "species", σ_j^2 is the noise variance, n_j is the number of data points, and J is the total number of "species".

GP compact support

The emulator needs compact support. While for the real-world problem the bounds for the parameters were taken to be biologically meaningful, for the other three applications, we chose the bounds based on an initial exploration, e.g. initially a wide interval was chosen, and 1D and 2D plots of the true residual sum-of-square values against the parameters helped refine this range. For example, parameter values producing signals extremely different from the data, and thus high residual sum-of-square values (relative to the minimum value, e.g. 100 times larger) should be excluded. Choosing ranges in this manner may work in 1D or 2D space, further refinement may be needed in joint parameter space for dimensions higher than three – see Section 6.5.2 for further details. Thus, the compact support was chosen as follows: for the sinusoidal example: $A \in [2,7], B \in [0.5,1.7], C \in [0.01,0.1]$; for the FitzHugh-Nagumo model: $\alpha \in [1,4], \beta \in [10^{-3},1], \gamma \in [10^{-3},1]$; for the biochemical signalling pathway: $k_1 \in [0.03,0.07], V_1 \in [0.05,0.5], K_{m_1} \in [0.01,1], V_2 \in [0.05,0.2]$, $K_{m_2} \in [0.01,0.3]$; for the pulmonary fluid-dynamics model: $s \in [7\times10^4, 5\times10^5], r_1 \in [-0.5,1.92], r_2 \in [-0.5,1.0], c \in [-2.5,1.5]$).

Data sets

For all three synthetic examples (sinusoidal, FitzHugh-Nagumo and biochemical pathway) 10 synthetic data sets were generated with different noise instantiations. For the pulmonary example one single measured data set was available.

GPHMC phases

Initial design stage: At the initial stage of the GPHMC algorithm [156] in which the initial emulator is created, a number of training points must be selected. This number can be determined by quantifying the efficiency of the MCMC sampler in the beginning of the exploratory phase (the acceptance rate of the sampler running on the initial emulator should not be too low, e.g. below 20%). 1500 points were collected for the sinusoidal and FitzHugh-Nagumo (out of which 500 points with lowest RSS value were used as training points for the emulator), 600 points for the pulmonary fluid-dynamics model (all points were used as training points for the emulator), and 3000 points (out of which 500 points with lowest RSS value were used as training points for the emulator) for the biochemical example. This can act as an initial refinement of the initial joint range if very large RSS values relative to the minimum recorded are recorded (an idea similar to a history matching approach [213]).

Exploratory phase: In the exploratory phase of the GPHMC algorithm [156] a minimum number of training points should be stored (as to boost computational efficiency), while preserving a high enough emulator accuracy (as quantified by GP diagnostics [8]), and a high acceptance rate in the sampling phase (>65% [132]). Initially $100 \times d$ (d: parameter dimensionality) training points can be used, and if the acceptance rate in the sampling phase is sub-optimal (<65%), the exploratory phase can be extended to allow a larger number of training points to be collected. Generally this number depends on the parameter dimensionality and the complexity of the posterior distribution. For example, the rule of thumb presented in the study by Jones et al. [94] $10 \times d$ was found inadequate; while this rule of thumb may work for Bayesian optimisation, which is the context in which it was originally proposed, it provided a sub-optimal emulator in an MCMC context. The algorithm in the exploratory phase was run for 400 iterations for the sinusoidal example and the accepted samples were added as further training points to the list from the initial stage design (400 training points were saved at the end of the exploratory to be used in the sampling phase), 500 iterations for the FitzHugh-Nagumo (450 points at the end of the exploratory), 1000 iterations for the pulmonary fluid-dynamics model (400 points at the end of the exploratory), and for the biochemical example the sampler was run to obtain 1500 training points to be used in the sampling phase.

The samples collected in the exploratory phase were used to refine the GP emulator (i.e. update the covariance hyperparameters by reoptimising the log marginal likelihood of the GP training points – see eq (2.64) or (2.79)) and to get closer to the equilibrum distribution.

The noise variance was kept fixed in the first part of the exploratory phase to enable learning the parameters while avoiding changes in curvature which would be induced by varying the noise variance. One can initially use an informed guess, as given by RSS divided by the number of points, where the RSS value was obtained from fitting a non-linear regression model to the data. The acceptance rate in the exploratory phase of the algorithm can then act as an objective metric to assess the appropriateness of the noise variance σ^2 value. If the acceptance rate is too low, e.g. <10%, this indicates a higher σ^2 value in eqns (2.19) and (6.2) is needed to reduce the discrepancy between the emulated and original likelihood (or posterior). If the acceptance rate is too large, e.g. >90%, this suggests that the σ^2 value should be reduced to avoid flattening out the likelihood (or posterior) landscape (see eq (2.19)), and encouraging exploration of too wide of a region without focusing on the area of interest (i.e. a low σ^2 value will make the likelihood in eq (2.19) more peaked, while a higher σ^2 will flatten out the likelihood). The σ^2 value can sequentially be altered by, say, 10%, and the exploratory phase re-run with a monitoring of the acceptance rate. For the sinusoidal and FitzHugh-Nagumo examples, we kept σ^2 to its true value, for the biochemical example we set $\sigma^2 = 0.008$, and for the pulmonary model we set $\sigma^2 = 1.1$. The noise variance was sampled in later stages, except the sinusoidal for which the noise variance was fixed to its true value along the entire simulation as a simplifying assumption. In the exploratory phase, the step size and number of leapfrog steps can be uniformly sampled within some fairly narrow pre-specified ranges [132], chosen to ensure an acceptable acceptance rate (e.g. above 20%), especially at the beginning of this phase when the emulator may not be a very good representation of the simulator. Fairly low step sizes and number of leapfrog steps were used, as follows: sinusoidal example – $\varepsilon = 0.005, L = 20$, FitzHugh-Nagumo – $\varepsilon \in [0.001, 0.005], L \in [20, 30]$, biochemical – $\varepsilon \in [0.0005, 0.002], L \in [50, 100]$, pulmonary: $\varepsilon = 0.003, L = 10$.

Sampling phase: In the sampling phase of GPHMC [156] 100 samples were collected and used as burn-in samples (chosen to ensure that MPSRF ≤ 1.1 [18]), and 2000 samples were subsequently drawn and used for inference. Besides sampling the model parameters, the noise variance was also sampled. Every algorithm (DA-GPHMC, noDA-GPHMC, DA-GPNUTS, DA-GPRMHMC, noDA-GPRMHMC, DA-GPLDMC, noDA-GPLDMC) was run 10 times from different random number generator seeds and different starting values for the parameters, selected from the points collected in the exploratory phase, to make it less likely to start in a low probability region. For each of the 10 simulations, ESS across all parameters was recorded, as well as ESS divided by the number of model (ODE/PDE) evaluations, and ESS divided by the total CPU time for the entire simulation. Each of the 10 simulations was timed three times, and the best CPU time out of the three was used.

Bayesian optimisation for HMC performance tuning

The Hamiltonian/Lagrangian Monte Carlo algorithms (HMC, RMHMC, LDMC) making use of Bayesian optimisation [208] for hyperparameter tuning require an initial GP for the normalised squared jumping distance in eq (2.41), as a function of the step size and the number of leapfrog steps, which need compact support. The ranges for the step size and the number of leapfrog steps were carefully selected to ensure a high effective sample size (ESS) [95], while no anticorrelation was induced (which would make ESS greater than the total number of MCMC samples) [73]. For the sinusoidal example: for HMC, $L \in \{1, \dots 40\}, \varepsilon \in [10^{-4}, 5 \times 10^{-3}]$, and for RMHMC and LDMC, $L \in \{1, \dots 25\}, \varepsilon \in [10^{-2}, 10^{-1}]$; for the FitzHugh-Nagumo example: for HMC, $L \in \{1, \dots 500\}, \varepsilon \in [10^{-4}, 10^{-2}]$, and for RMHMC and LDMC, $L \in \{1, \dots 50\}, \varepsilon \in [10^{-3}, 10^{-1}]$; for the biochemical example: for HMC, $L \in \{1, \dots 500\}, \varepsilon \in [10^{-4}, 10^{-2}]$; for the pulmonary example: for HMC, $L \in \{1, \dots 50\}, \varepsilon \in [10^{-4}, 5 \times 10^{-3}]$, and for RMHMC and LDMC, $L \in \{1, \dots 25\}, \varepsilon \in [10^{-2}, 10^{-1}]$. The initial GP for the normalised squared jumping distance was constructed based on 20 (ε, L) parameter vectors, and for each parameter vector 10 MCMC samples for the ODE/PDE parameters were obtained and used to estimate the expectation in eq (2.41). The initial parameter values fed into the sampler came from points collected in the exploratory phase as to start from a region of high posterior probability. This is important since the optimal step size and number of steps is dependent on the region in parameter space that the sampler is exploring, and thus HMC hyperparameters which are optimal in the region of high posterior probability are needed. Following [208] a squared exponential kernel for the GP of the normalised squared jumping distance was used, whose hyperparameters were optimised in a maximum likelihood framework. 50 iterations of the Bayesian optimisation algorithm were then run to select ε and L values that maximise the acquisition function. The optimum value for L was used as an *upper bound* for the number of leapfrog steps, and optimum ε was used as the step size in the sampling phase of the algorithm.

Parameter transformations

To improve numerical stability and reduce round-off errors [141], the original parameters θ_i were scaled in the order of one for building the GP emulator: $\theta_i \in [l_i, u_i] \rightarrow \dfrac{\theta_i}{s_i} \in [-1, 1]$, where s_i is a scaling factor which ensures $\dfrac{\theta_i}{s_i} \in [-1, 1]$. In addition, the Hamiltonian/Lagrangian Monte Carlo algorithms require unbounded parameters: $\theta_i \in [l_i, u_i] \rightarrow \log_e \dfrac{\theta_i - l_i}{u_i - \theta_i} \in \mathbb{R}$ for hard bounds (for our real-world application, parameter values outside these bounds are not plausible biologically), or $\log_e \theta_i$ for soft bounds (for the other three applications, for which the only constraint was to be positive). The transformed parameters were then mapped back via the inverse transformation into the original domain $\theta_i \in [l_i, u_i]$ before being inserted into the ODE/PDE simulator.

6.6 Numerical Results

Simulations for all test examples and methods registered a MPSRF ≤ 1.1, thus there was no evidence of lack of convergence.

6.6.1 Sinusoidal

Geweke consistency test: mathematical and coding correctness of the sampler

The mathematical and coding correctness of the samplers' implementation was performed using the Geweke consistency test [71] reviewed in Section 2.3.6. This was done for the linear ODE sinusoidal example to ensure no high computational costs attributed to a large number of ODE numerical integrations were incurred. Figure 6.8 shows QQ plots, i.e. quantiles of the prior distribution against quantiles of the ensemble of posterior distributions (see eq (2.52)) for all parameters (A,B,C) for three of the algorithms: DA-GPHMC (top panel), noDA-GPHMC (middle panel), DA-GPNUTS (bottom panel). The points lie on the equality line, taken as evidence that the implementation of the three samplers is correct. This conclusion is extrapolated to the other samplers: (no)DA-GPRMHMC and (no)DA-GPLDMC since by looking at the proofs in Section 6.9, one can notice that they are a straightforward extension of (no)DA-GPHMC.

DA vs noDA

It was investigated whether the delayed acceptance scheme offers any gains in computational efficiency. To quantify efficiency, effective sample size, reviewed in Section 2.3.5 was used. The minimum, median and maximum ESS across all d parameters can be defined as:

$$\text{MinESS} = \min_i \left(\frac{N}{1+2\sum_{l=1}^{\infty} \rho_l^i} \right), \tag{6.24}$$

$$\text{MedianESS} = \text{median}_i \left(\frac{N}{1+2\sum_{l=1}^{\infty} \rho_l^i} \right), \tag{6.25}$$

$$\text{MaxESS} = \max_i \left(\frac{N}{1+2\sum_{l=1}^{\infty} \rho_l^i} \right). \tag{6.26}$$

Median ESS (eq (6.25)) across all parameters was calculated for all DA-HMC algorithms against their noDA alternative (with the exception of NUTS, as justified in Section 6.3.4). Figure 6.9 shows the median ESS normalised by the total number of MCMC samples N, as well as the median ESS normalised by the CPU time, and by the number of forward (ODE) evaluations for 10 data sets. The distributions over the 10 data sets for DA compared to noDA appear to overlap and there is no evidence that the DA scheme brings any advantages, a claim formally investigated by a hypothesis test:

$$H_0 : \mu_{\text{ESS}}^{\text{noDA}} = \mu_{\text{ESS}}^{\text{DA}} \quad \text{versus} \quad H_1 : \mu_{\text{ESS}}^{\text{noDA}} \neq \mu_{\text{ESS}}^{\text{DA}}. \tag{6.27}$$

In eq (6.27), $\mu_{\text{ESS}}^{\text{noDA}}$ is the mean ESS for the noDA method in the sample of 10 data sets, while $\mu_{\text{ESS}}^{\text{DA}}$ is the mean ESS for the DA method in the sample of 10 data sets. The test revealed a p-value > 0.05, hence the null hypothesis cannot be rejected, supporting the visual finding. Given that the next goal is to compare the different algorithms, an arbitrary choice to go with the DA algorithms was made.

Accuracy

Parameter space: For each of the algorithms (DA-GPHMC, DA-GPNUTS, DA-GPRMHMC and DA-GPLDMC) and ODE parameters (A,B,C), Figure 6.10 shows the distribution over 10 data sets of the biases in parameter space, obtained by subtracting the true parameter from the

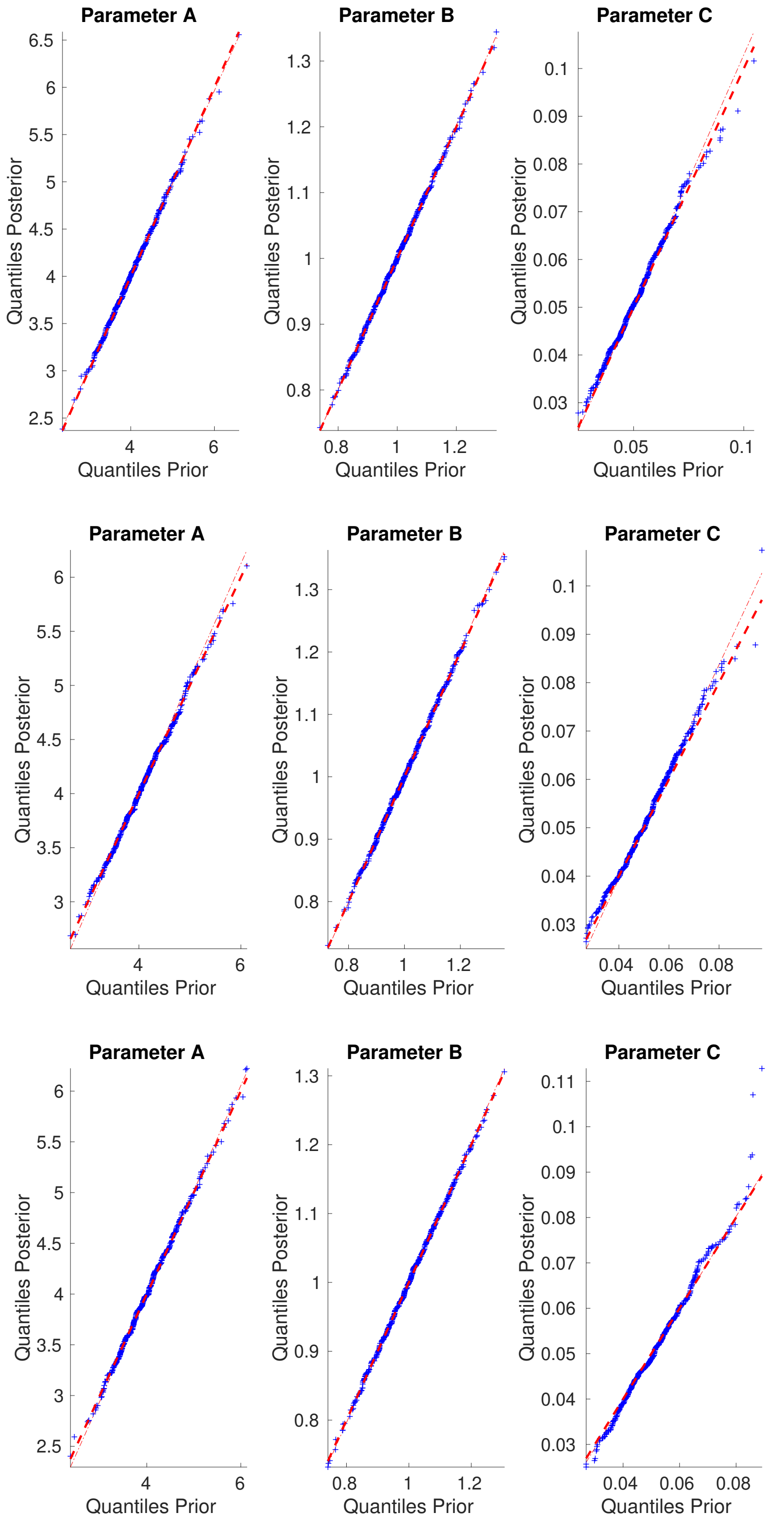


Figure 6.8: Geweke consistency test [71] to check the mathematical and coding correctness of three samplers: DA-GPHMC (top panel), noDA-GPHMC (middle panel), DA-GPNUTS (bottom panel) for the three parameters (A, B, C) of the sinusoidal example. Quantiles of the prior distribution are shown against the quantiles of the ensemble of posterior distributions, see Section 2.3.6 for more details. Points lying on the equality line (red dotted line) indicate correctness in the implementation of the sampler.

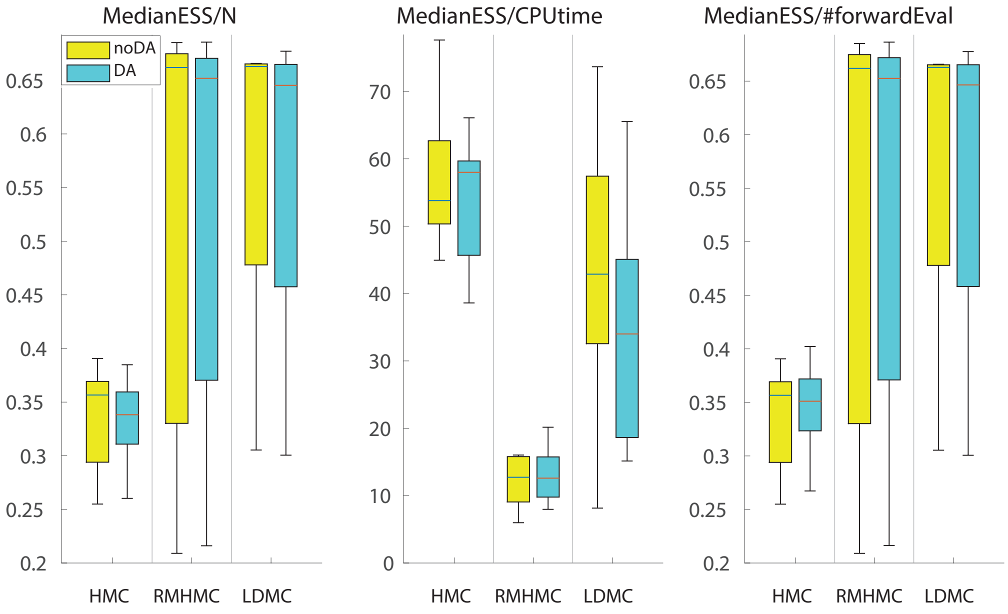


Figure 6.9: Efficiency in terms of ESS of DA-GPHMC, DA-GPRMHMC, DA-GPLDMC versus their noDA version for the sinusoidal example. The distribution of median ESS (eq (6.25)) over 10 data sets is shown.

posterior median, which for a d-dimensional parameter vector θ is defined as:

$$\mathbf{b}_k(\theta) = \mu_k(\theta) - \theta_{\text{true}}, \quad k = 1 \ldots 10, \tag{6.28}$$

where $\mathbf{b}_k(\theta)$ is the parameter bias vector for the k^{th} data set, $\mu_k(\theta)$ is the parameter posterior median vector for the k^{th} data set, and θ_{true} is the true (data-generating) parameter vector.

In addition, Table 6.1 illustrates the mean of the posterior medians (i.e. the mean of the 10 posterior medians from the 10 data sets) for the ODE parameters drawn with all emulation algorithms, and the corresponding standard deviation, i.e.

$$\begin{aligned} \bar{\theta} &= \frac{1}{K}\sum_{k=1}^{K} \mu_k(\theta), \quad K = 10, \\ \sigma(\theta) &= \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}(\mu_k(\theta) - \bar{\theta})^2}. \end{aligned} \tag{6.29}$$

If the parameters are well estimated, $\bar{\theta}$ should be close to θ_{true}.

Moreover, for each of the algorithms and ODE parameters, Figure 6.11 shows the distribution over 10 data sets of the biases in parameter space, obtained by subtracting the true parameter from the agglomerated posterior samples (i.e. posterior samples combined from all 10 data sets),

Algorithm	A	B	C
DA-GPHMC	3.0337 (0.0659)	1.0013 (0.0052)	0.0507 (0.0036)
DA-GPNUTS	3.0348 (0.0645)	1.0012 (0.0052)	0.0508 (0.0032)
DA-GPRMHMC	3.0358 (0.0656)	1.0011 (0.0053)	0.0505 (0.0035)
DA-GPLDMC	3.0348 (0.0650)	1.0014 (0.0052)	0.0506 (0.0035)
True value	3	1	0.05

Table 6.1: Parameter estimates and standard deviations for the sinusoidal example for each of the methods compared. The mean and standard deviation of the posterior medians for 10 data sets, calculated using eq (6.29) is shown. The true parameter values are also displayed.

which for a d-dimensional parameter vector θ is defined as:

$$\mathbf{b}(\theta) = \theta_{\cup_{k=1}^{K}} - \theta_{\text{true}}, \quad K = 10, \tag{6.30}$$

where $\theta_{\cup_{k=1}^{K}}$ is the union of K sets of parameter posterior samples corresponding to the K data sets.

In Figure 6.12, for parameter A and each of the algorithms, the distribution of the biases in parameter space is displayed for each of the 10 data sets individually, obtained by subtracting the true parameter from the posterior samples for each data set, which for a d-dimensional parameter vector θ is defined as:

$$\mathbf{b}(\theta) = \theta_k - \theta_{\text{true}}, \quad k = 1 \ldots K, \quad K = 10, \tag{6.31}$$

where θ_k represents the set of parameter posterior samples for the k^{th} data set.

By analysing Figures 6.10, 6.11 and Table 6.1, it can be noted that all four algorithms register a very similar performance in terms of accuracy, and the parameters seem to have been learnt well, especially parameters B and C (the distributions lie around the zero bias line). For parameter A a slight bias is observed. Investigation of Figure 6.12 reveals two negative offsets (i.e. distributions below the zero bias line), six positive offsets and two zero offsets, hence no systematic overestimation is present, therefore the bias is a consequence of a small number of data sets (10). As the number of data sets increases, this parameter would be better estimated. As an experiment, a fair coin can be tossed eight times, and the event of obtaining six or more heads can be assigned to the event of obtaining six or more positive offsets. Calculations indicate that the probability of obtaining six or more heads out of eight tosses is 0.145, hence assuming that the algorithm is correctly implemented, the probability of obtaining six positive offsets out of eight tosses is 0.145, thus it is not unlikely.

Functional space: The performance of the algorithms in functional space is quantified using R^2, which indicates how good the fit is between the data and the signal generated with the estimated parameter values, e.g. with the posterior median. R^2 is defined in terms of the residual

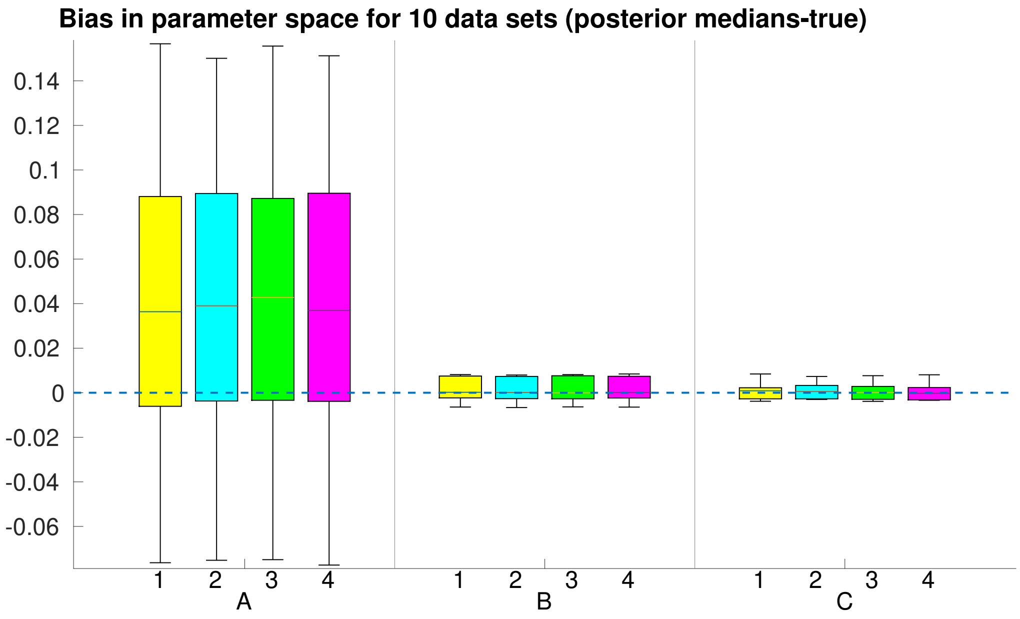


Figure 6.10: Bias in parameter space for each parameter (A, B, C) of the sinusoidal example. The bias is given by the difference between the posterior median value and the true parameter value (see eq (6.28)). The posterior median value is the median of all posterior samples drawn using each of the algorithms compared (1: DA-GPHMC, 2: DA-GPNUTS, 3: DA-GPRMHMC, 4: DA-GPLDMC). The distribution of the biases over 10 data sets is shown. The horizontal dashed line indicates zero bias.

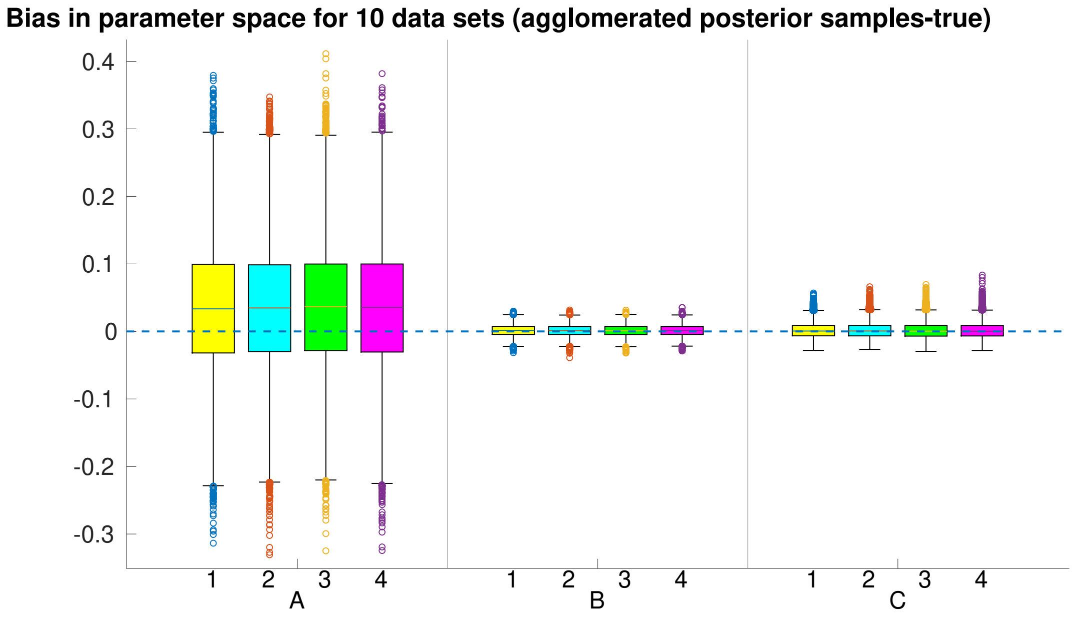


Figure 6.11: Bias in parameter space for each parameter (A, B, C) of the sinusoidal example. The bias is given by the difference between the posterior samples and the true parameter value (see eq (6.30)). The posterior samples were drawn using each of the algorithms compared (1: DA-GPHMC, 2: DA-GPNUTS, 3: DA-GPRMHMC, 4: DA-GPLDMC). The distribution of the agglomerated biases is shown for 10 data sets. The horizontal dashed line indicates zero bias.

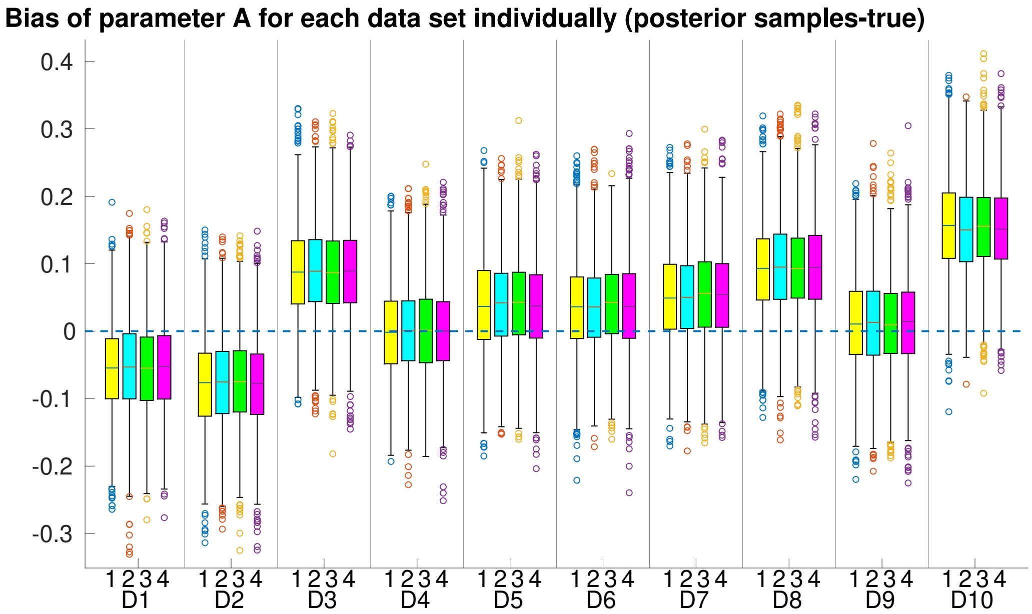


Figure 6.12: Bias in parameter space for parameter A of the sinusoidal example. The bias is given by the difference between the posterior samples and the true parameter value (see eq (6.31)). The posterior samples were drawn using each of the algorithms compared (1: DA-GPHMC, 2: DA-GPNUTS, 3: DA-GPRMHMC, 4: DA-GPLDMC). The distribution of the biases is shown for each of the 10 data sets individually. The horizontal dashed line indicates zero bias. D_i stands for the i^{th} data set.

sum-of-squares RSS and the total sum-of-squares SS_{total}:

$$R^2 = 1 - \frac{\text{RSS}}{\text{SS}_{\text{total}}}, \quad \text{RSS} = \sum_{i=1}^{n} (y_i - m_i(\theta))^2 = \sum_{i=1}^{n} \varepsilon_i^2, \quad \text{SS}_{\text{total}} = \sum_{i=1}^{n} \left(y_i - \frac{1}{n} \sum_{i=1}^{n} y_i \right)^2, \tag{6.32}$$

thus R^2 lies within [0,1], and the higher R^2 is, the better the fit is.

Figure 6.13 shows the distribution of the R^2 values for each of the 10 data sets, for all four algorithms. R^2 is very high (~ 0.97) for all methods, suggesting that all algorithms perform equally well in terms of predicting a signal which is very similar to the data.

Parameter UQ: UQ for the parameter estimates for all methods is also provided. Figure 6.14 displays the marginal posterior distributions for the parameters A, B, C obtained with 1D kernel density estimation from the posterior samples drawn with the four methods (a random data set was used). To obtain the marginal posterior distributions, the kernel smoothing function estimate was used for univariate data with the optimal bandwidth for normal densities [14]. The marginal distributions are comparable, backing up previous findings that all four methods perform very similarly.

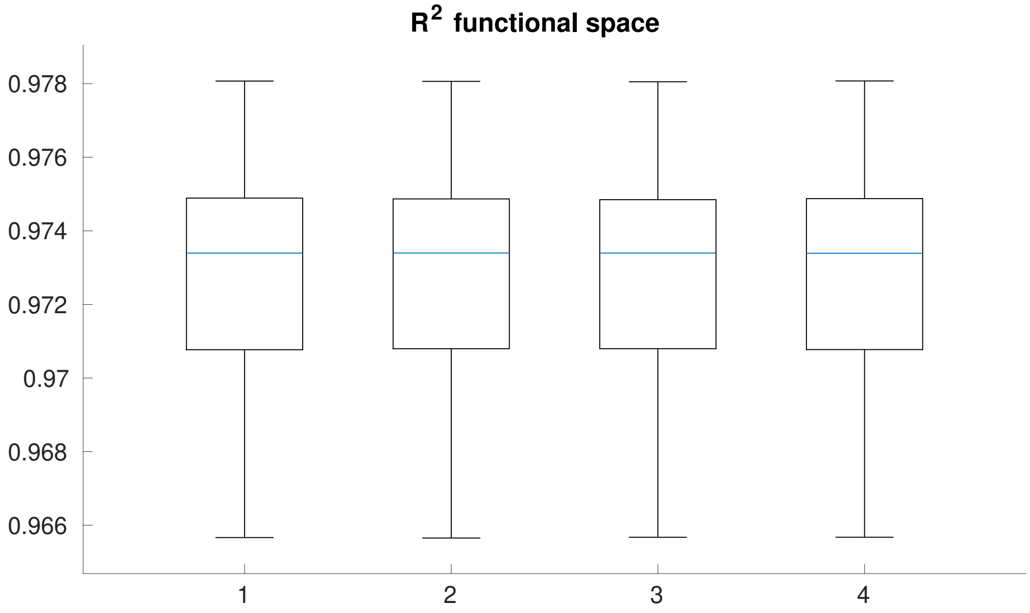


Figure 6.13: Accuracy quantification in functional space via R^2 for the sinusoidal example. R^2 was computed using eq (6.32) with the posterior median value obtained from posterior samples drawn using each of the algorithms compared (1: DA-GPHMC, 2: DA-GPNUTS, 3: DA-GPRMHMC, 4: DA-GPLDMC). The distribution of R^2 over 10 data sets is shown.

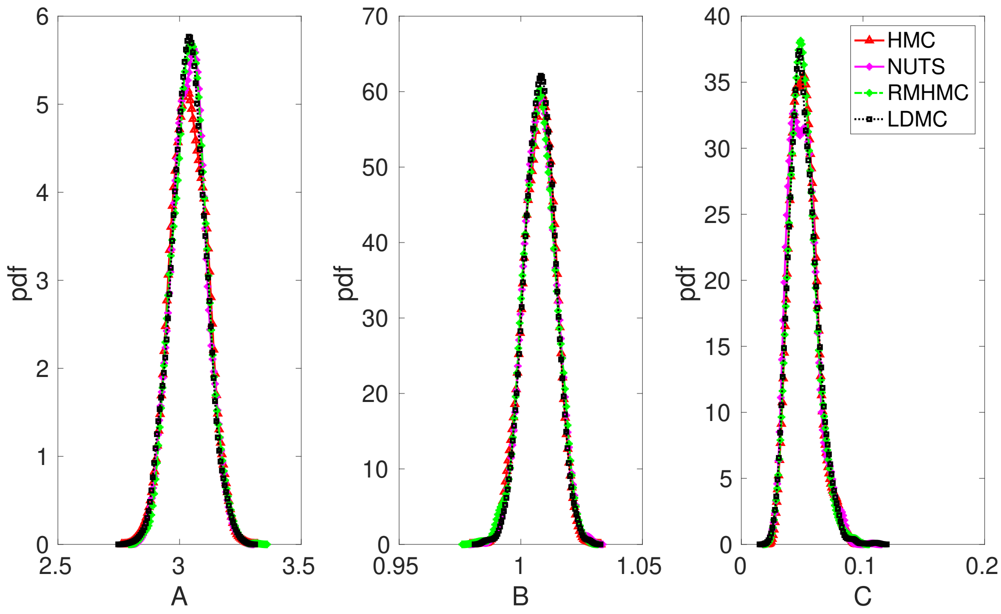


Figure 6.14: Parameter uncertainty quantification: marginal posterior distributions obtained via 1D kernel density estimation from the posterior samples of each parameter A, B, C of the sinusoidal example. The posterior samples for the ODE parameters were drawn using each of the algorithms compared (DA-GPHMC, DA-GPNUTS, DA-GPRMHMC, DA-GPLDMC). Marginal distributions for one random data set are shown.

Efficiency

The different methods were next compared in terms of efficiency, as quantified using min, median and max ESS (see eqns (6.24)-(6.26)) normalised by the total number of MCMC samples *N*, by the CPU time, and by the number of forward (model) evaluations[8]. The latter measure is the only one that can generalise to ODE/PDE models for which a forward model evaluation is computationally expensive. Results based on all three efficiency measures are presented in Figure 6.15, showing the distribution of these quantities over 10 data sets. When inspecting ESS/*N* (left panel), great variability can be observed between the distributions for the minESS/*N*, medianESS/*N* and maxESS/*N* for the first-order methods (HMC and NUTS), in contrast with the higher-order methods (RMHMC and LDMC). In terms of minESS/*N*, RMHMC and LDMC are clearly superior to HMC and NUTS, while in terms of medianESS/*N* all methods are more comparable, and similarly in terms of maxESS/*N*, with the HMC algorithm having moderate advantage. The same pattern is observed when inspecting ESS normalised by the number of model evaluations (right panel), which is expected, considering the high acceptance rate (>95% across all methods), i.e. the number of model evaluations is very close to *N*. When it comes to ESS normalised by the CPU time (middle panel), it becomes apparent that the RMHMC algorithm loses its advantage due to incurring high computational costs, and performs worst in terms of median and max. LDMC appears best in terms of minESS/CPUtime, with the other three methods being very similar. In terms of medianESS/CPUtime, HMC and NUTS are quite comparable with LDMC, with a slight advantage, which becomes more pronounced when looking at maxESS/CPUtime, with HMC being clearly superior.

6.6.2 Fitz-Hugh Nagumo

Quasi algorithms for non-positive definite negative Hessian matrix

For the FitzHugh-Nagumo model difficulties with running the higher-order methods (RMHMC and LDMC) were encountered due to the negative Hessian matrix of the log posterior not being positive definite, as illustrated in Figure 6.16 displaying regions in 2D parameter space for which the negative Hessian is not positive definite. As outlined in Section 6.3.3, for this example, the RMHMC or LDMC methods were replaced by the Quasi-HMC-RMHMC or Quasi-HMC-LDMC, and for the RMHMC/LDMC component of the method, eq (6.16) was used to set the metric tensor. The Quasi-HMC-LDMC algorithm registered a percentage of 25% (average over the 10 data sets) of HMC-drawn samples, and 75% of LDMC-drawn samples out of the total number of MCMC samples. Also, the Quasi-HMC-RMHMC algorithm registered a percentage of 34% (average over the 10 data sets) of HMC-drawn samples, and 66% of RMHMC-drawn samples out of the total number of MCMC samples. For the FitzHugh-Nagumo, results in

[8] Given that the number of parameters for the sinusoidal example is three, the min, median and max ESS correspond to the ESS for each of the three parameters.

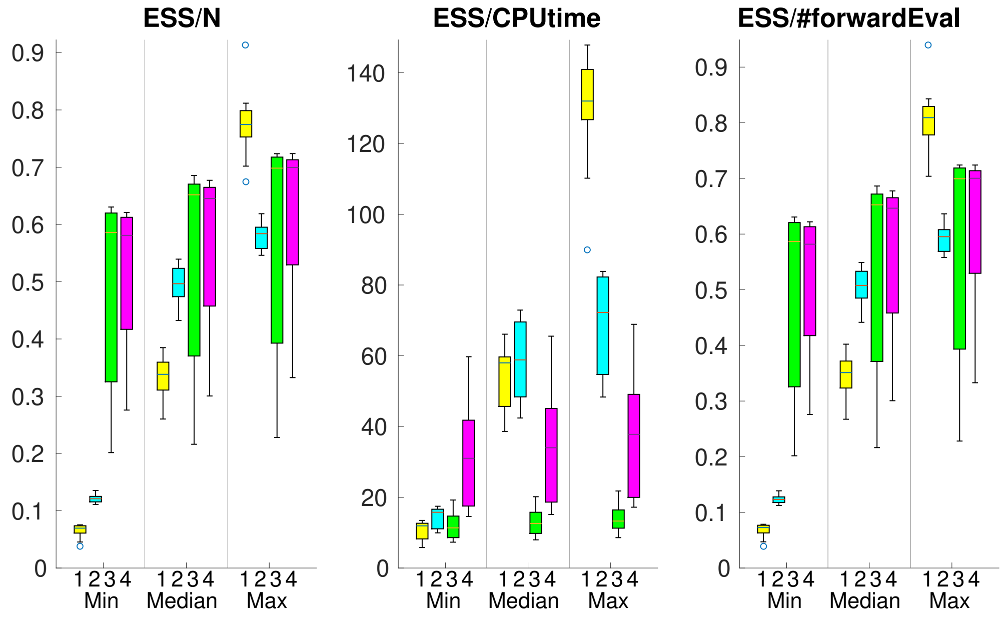


Figure 6.15: Efficiency in terms of ESS of the algorithms compared (1: DA-GPHMC, 2: DA-GPNUTS, 3: DA-GPRMHMC, 4: DA-GPLDMC) for the sinusoidal example. Min, median, max ESS were calculated as per eqns (6.24)-(6.26). The distribution over 10 data sets is shown.

subsequent sections are shown for these Quasi algorithms.

DA vs noDA

Figure 6.17 reveals mostly an overlap between the distributions over 10 data sets of normalised ESS for the DA and noDA algorithms, which is backed up by a p-value > 0.05 from the hypothesis test (see eq (6.27)) for all efficiency measures, except MedianESS/CPUtime for HMC. Thus, there is no difference in efficiency between the DA and noDA algorithms, except with respect to one particular measure (MedianESS/CPUtime) for one algorithm (HMC), for which DA appears slightly better (Figure 6.17). Similarly to the sinusoidal example, the DA algorithms are taken forward for method comparison.

Accuracy

Parameter space: Table 6.2 illustrates the mean and standard deviation of the posterior medians over 10 data sets (see eq (6.29)) for the ODE parameters drawn with all emulation methods, and of the noise variances, sampled with Gibbs. Figure 6.18 shows the distribution over data sets of the biases in parameter space, obtained by subtracting the true parameter from the agglomerated posterior samples (eq (6.30)) for all methods. Figure 6.18 and Table 6.2 reveal that all four algorithms perform very similarly in terms of accuracy, with the distribution of biases for the ODE parameters and the two noise variances containing the zero bias line. A slight bias (underestimation) is observed for the α parameter from Figure 6.18, which is a consequence of

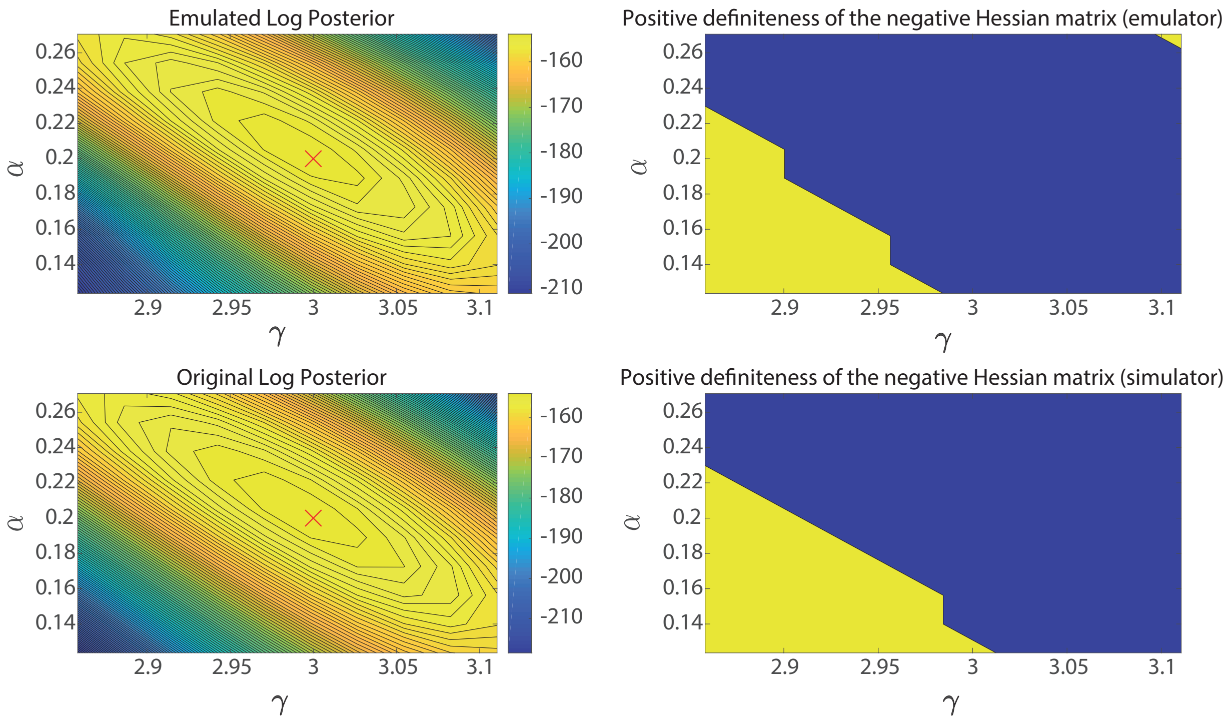


Figure 6.16: Displaying the positive definiteness of the negative Hessian matrix for the emulated (top) and original (bottom) log posterior distribution of two of the parameters (the third parameter is kept fixed at its true value) for the FitzHugh-Nagumo model. Blue is positive definite, yellow is non-positive definite. The red cross in the log unnormalised posterior distribution marks the true parameter value.

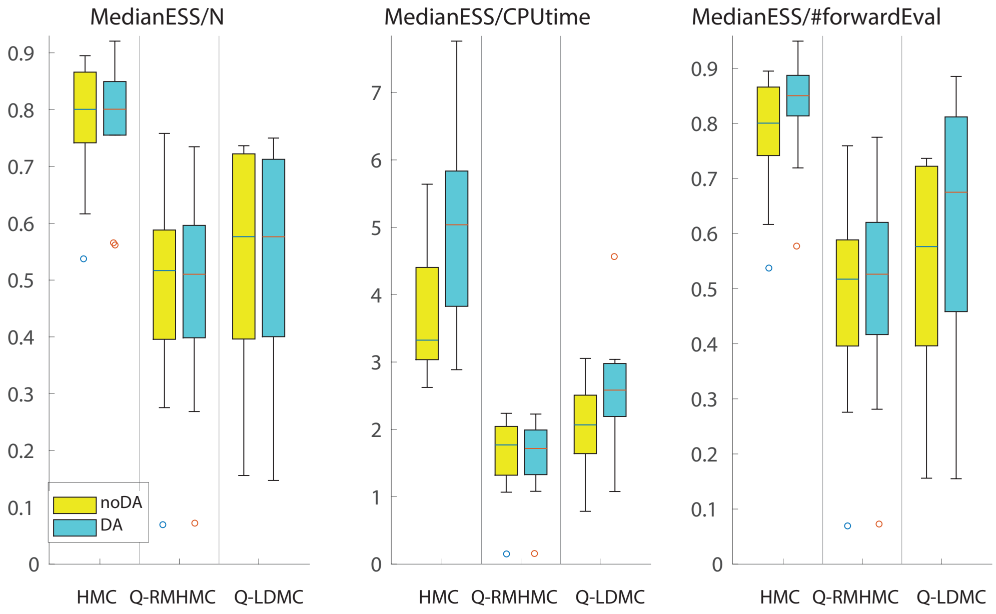


Figure 6.17: Efficiency in terms of ESS of DA-GPHMC, DA-GPQuasi-HMC-RMHMC, DA-GPQuasi-HMC-LDMC versus their noDA version for the FitzHugh-Nagumo example. Median ESS is calculated as per eq (6.25). The distribution of median ESS over 10 data sets is shown.

Algorithm	α	β	γ	σ_V^2 & σ_R^2
DA-GPHMC	0.1840 (0.0282)	0.2507 (0.0991)	2.9821 (0.0465)	0.2587 (0.0355) & 0.1677 (0.0183)
DA-GPNUTS	0.1837 (0.0285)	0.2494 (0.1010)	2.9829 (0.0468)	0.2588 (0.0356) & 0.1679 (0.0173)
DA-GPQuasi-RMHMC-HMC	0.1844 (0.0288)	0.2610 (0.1006)	2.9834 (0.0466)	0.2592 (0.0351) & 0.1680 (0.0178)
DA-GPQuasi-LDMC-HMC	0.1842 (0.0279)	0.2629 (0.1030)	2.9825 (0.0471)	0.2596 (0.0356) & 0.1680 (0.0181)
True value	0.2	0.2	3	0.25 & 0.16

Table 6.2: Parameter estimates and standard deviations for the FitzHugh-Nagumo example for each of the methods compared (note that the noise variances were sampled using Gibbs sampling). The mean and standard deviation of the posterior medians for 10 data sets, calculated using eq (6.29) is shown. The true parameter values are also displayed.

a small number of data sets (10), and not a systematic underestimation for all data sets. The coin tossing experiment reveals that the probability of obtaining seven or more tails (equivalent to obtaining seven or more negative offsets) out of 10 tosses is 0.172 (out of the 10 offsets, seven were negative and three were positive).

Functional space: Figure 6.19 shows the distribution of the R^2 (eq (6.32)) values over data sets for all methods and each "species". The different methods perform very similarly, and one of the signals appears to be learnt better ($R^2 \sim 0.9$) than the other ($R^2 \sim 0.7$).

Parameter UQ: Regarding UQ, all the methods are on a par, as evident in Figure 6.20 showing the marginal posterior distributions for the ODE parameters and noise variances for one random data set.

Efficiency

The methods are assessed in terms of efficiency, as quantified using min, median and max ESS (see eqns (6.24)-(6.26) normalised by the total number of MCMC samples N, by the CPU time, and by the number of forward (model) evaluations, and the latter measure is the only one that can generalise to ODE/PDE models for which a forward model evaluation is computationally expensive. Results based on all three measures are presented in Figure 6.21, which shows the distribution of these quantities over 10 data sets. When inspecting ESS/N (left panel), much larger variability between the distributions for the minESS/N, medianESS/N and maxESS/N can be seen for the first-order methods (HMC and NUTS), unlike the higher-order methods

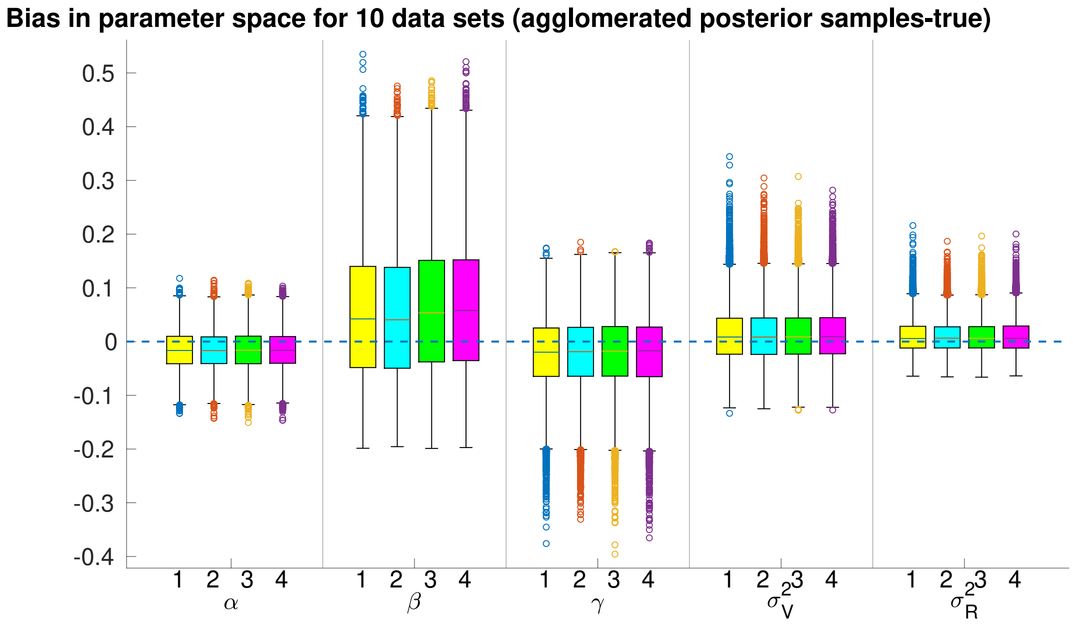


Figure 6.18: Bias in parameter space for each parameter ($\alpha, \beta, \gamma, \sigma_V^2, \sigma_R^2$) of the FitzHugh-Nagumo example. The bias is given by the difference between the posterior samples and the true parameter value (see eq (6.30)). The posterior samples were drawn using each of the algorithms compared (1: DA-GPHMC, 2: DA-GPNUTS, 3: DA-GPQuasi-HMC-RMHMC, 4: DA-GPQuasi-HMC-LDMC), and Gibbs sampling was used for sampling the noise variances. The distribution of the agglomerated biases for 10 data sets is shown. The horizontal dashed line indicates zero bias.

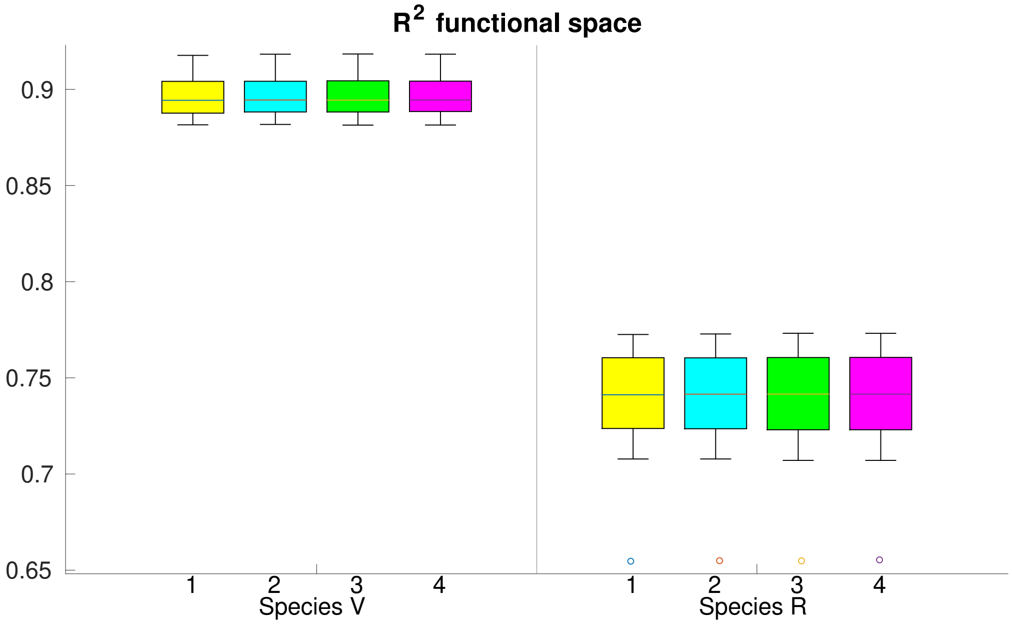


Figure 6.19: Accuracy quantification in functional space via R^2 for the FitzHugh-Nagumo example. R^2 was computed using eq (6.32) with the posterior median value obtained from posterior samples drawn using each of the algorithms compared (1: DA-GPHMC, 2: DA-GPNUTS, 3: DA-GPQuasi-HMC-RMHMC, 4: DA-GPQuasi-HMC-LDMC). The distribution of R^2 over 10 data sets is shown.

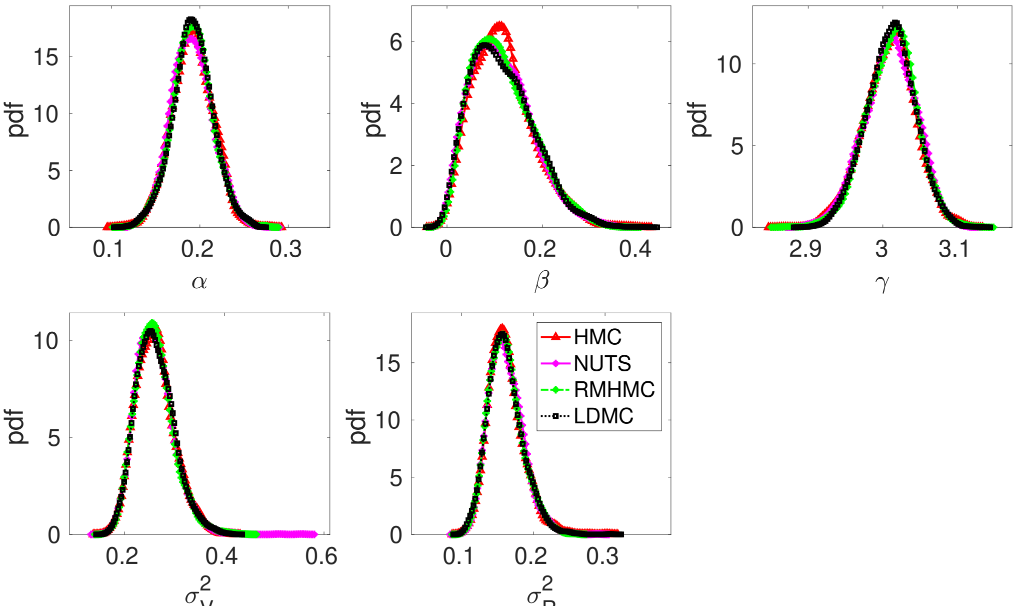


Figure 6.20: Parameter uncertainty quantification: marginal posterior distributions obtained via 1D kernel density estimation from the posterior samples of each parameter $\alpha, \beta, \gamma, \sigma_V^2, \sigma_R^2$ of the FitzHugh-Nagumo example. The posterior samples for the ODE parameters were drawn using each of the algorithms compared (DA-GPHMC, DA-GPNUTS, DA-GPQuasi-HMC-RMHMC, DA-GPQuasi-HMC-LDMC), and Gibbs sampling was used for sampling the noise variances. The marginal distributions are shown for one random data set.

(Quasi-HMC-RMHMC and Quasi-HMC-LDMC). Another observation is that NUTS tends to perform systematically worst. In terms of minESS/N, HMC, Quasi-HMC-RMHMC and Quasi-HMC-LDMC are comparable, while in terms of medianESS/N and maxESS/N, HMC seems to have an advantage. The same observations can be made when inspecting ESS normalised by the number of model evaluations (right panel), which is expected, considering the high acceptance rate (>80% across all methods), i.e. the number of model evaluations is close to N. Regarding ESS normalised by the CPU time (middle panel), the Quasi-HMC-RMHMC algorithm is penalised for the high computational costs, and tends to perform equally poorly as NUTS when compared to HMC and Quasi-HMC-LDMC. The latter two are comparable with respect to minESS/CPUtime, however HMC is preferred when looking at medianESS/CPUtime and maxESS/CPUtime.

6.6.3 Biochemical pathway

A zero mean GP versus a quadratic mean GP

For the biochemical example, a zero mean GP model for the RSS was first implemented. This model encouraged adding 'extreme' RSS values (high relative to the low RSS region) to the list of training points as a consequence of stepping into a region of high uncertainty of the

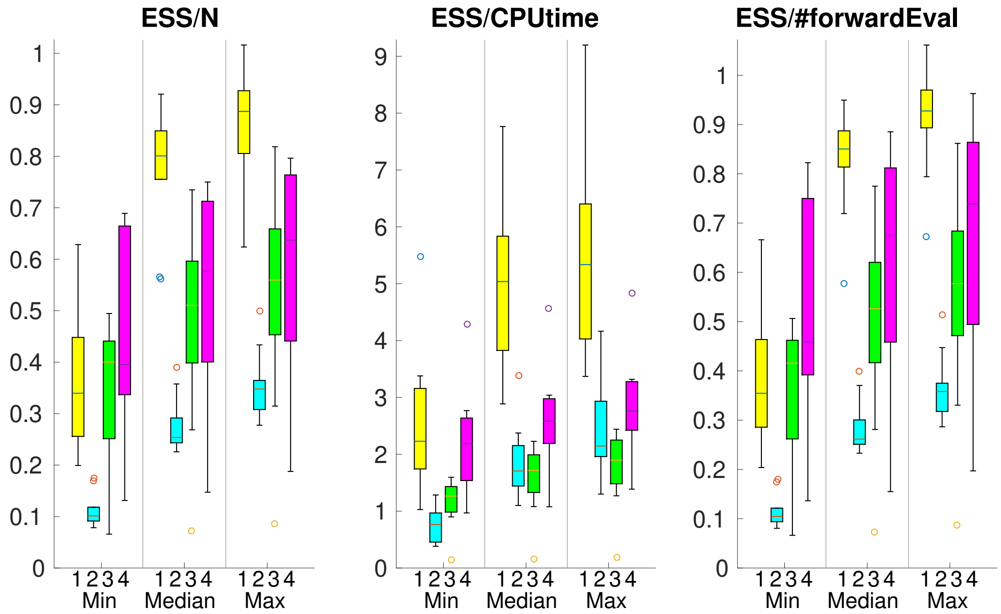


Figure 6.21: Efficiency in terms of ESS of the algorithms compared (1: DA-GPHMC, 2: DA-GPNUTS, 3: DA-GPQuasi-HMC-RMHMC, 4: DA-GPQuasi-HMC-LDMC) for the FitzHugh-Nagumo example. Min, median, max ESS are calculated as per eqns (6.24)-(6.26). The distribution over 10 data sets is shown.

emulator. The effect this has is illustrated in Figure 6.22, comparing the original and emulated log posterior in 2D (two of the parameters were varied while the others were kept fixed to their true values). Figure 6.22 reveals spurious modes and structures in the emulated log posterior, which are not present in the original log posterior. The emulator is not a faithful representation of the simulator, which leads to poor performance in the sampling phase.

Next a quadratic mean function for the GP prior was employed, which potentially places high prior density values on the region where most data are, i.e. around the mode of the posterior distribution, and lower density values away from the mode, thus moves of the emulator away from the high posterior probability are discouraged. Consequently, the acceptance rate in the sampling phase increases. For example, Figure 6.23 shows posterior samples collected in the sampling phase for all five parameters for one data set. The top two figures show posterior samples generated from the zero mean GP log posterior and two different initialisations, while the bottom two figures show samples drawn from the quadratic mean GP log posterior. It is obvious that the chain mixing based on the latter GP model is better than the former. In addition, Table 6.3 displays quantitative metrics, demonstrating that the quadratic mean GP leads to better performance (in terms of acceptance rate and ESS) compared to the zero mean GP model. Therefore, in the next sections, results produced with a quadratic mean function GP prior model are presented.

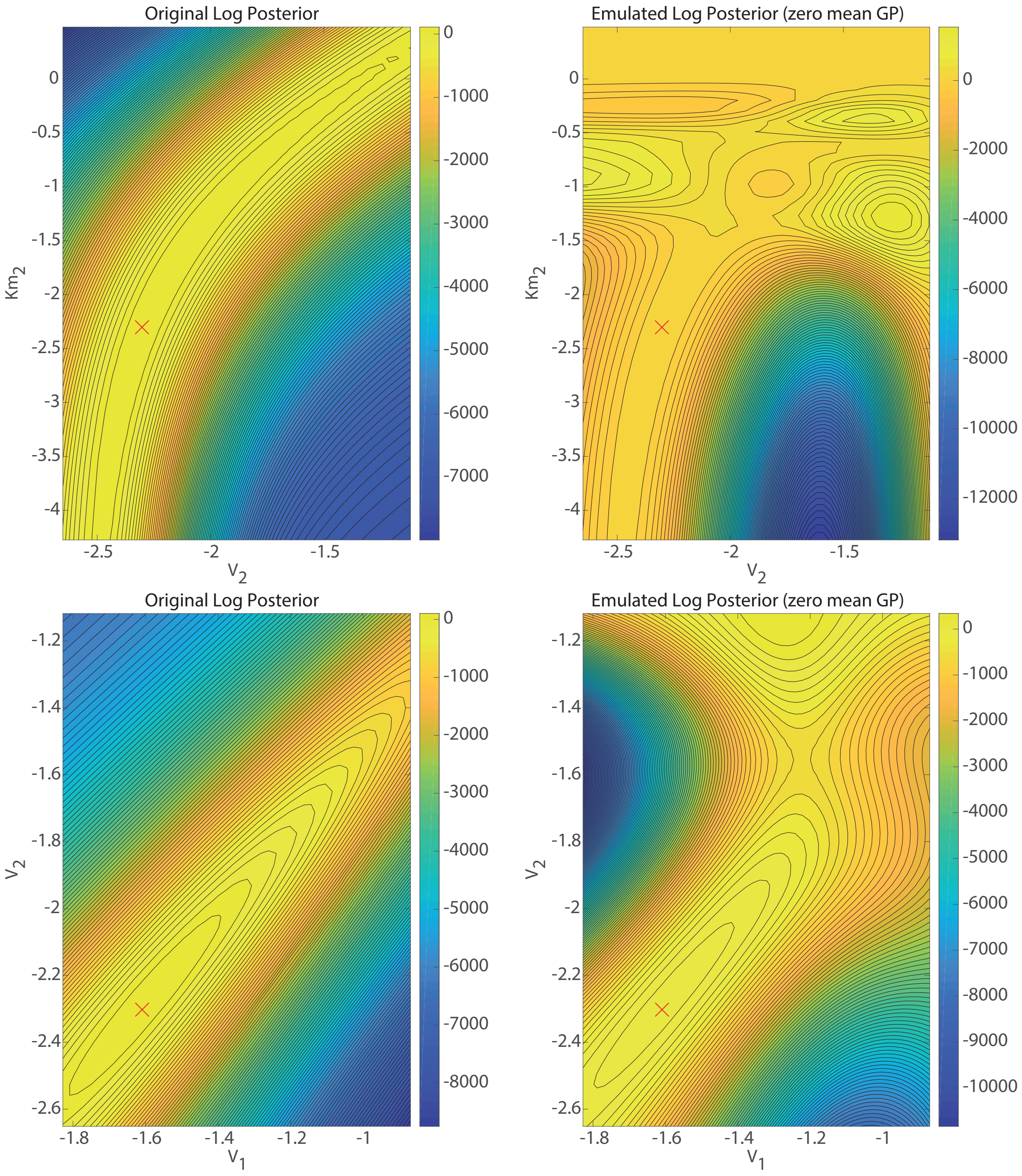


Figure 6.22: Biochemical pathway model: Showing the inappropriateness of using the zero mean GP prior by a comparison of the emulated log unnormalised posterior (constructed in the exploratory phase of the GPHMC algorithm) to the original log posterior for two pairs of parameters (when two of the parameters are varied, the other parameters are kept fixed at their true values). The red cross in the plot of the log unnormalised posterior distribution marks the true parameter value. The parameter values on the log scale are shown on the (x,y) axis.

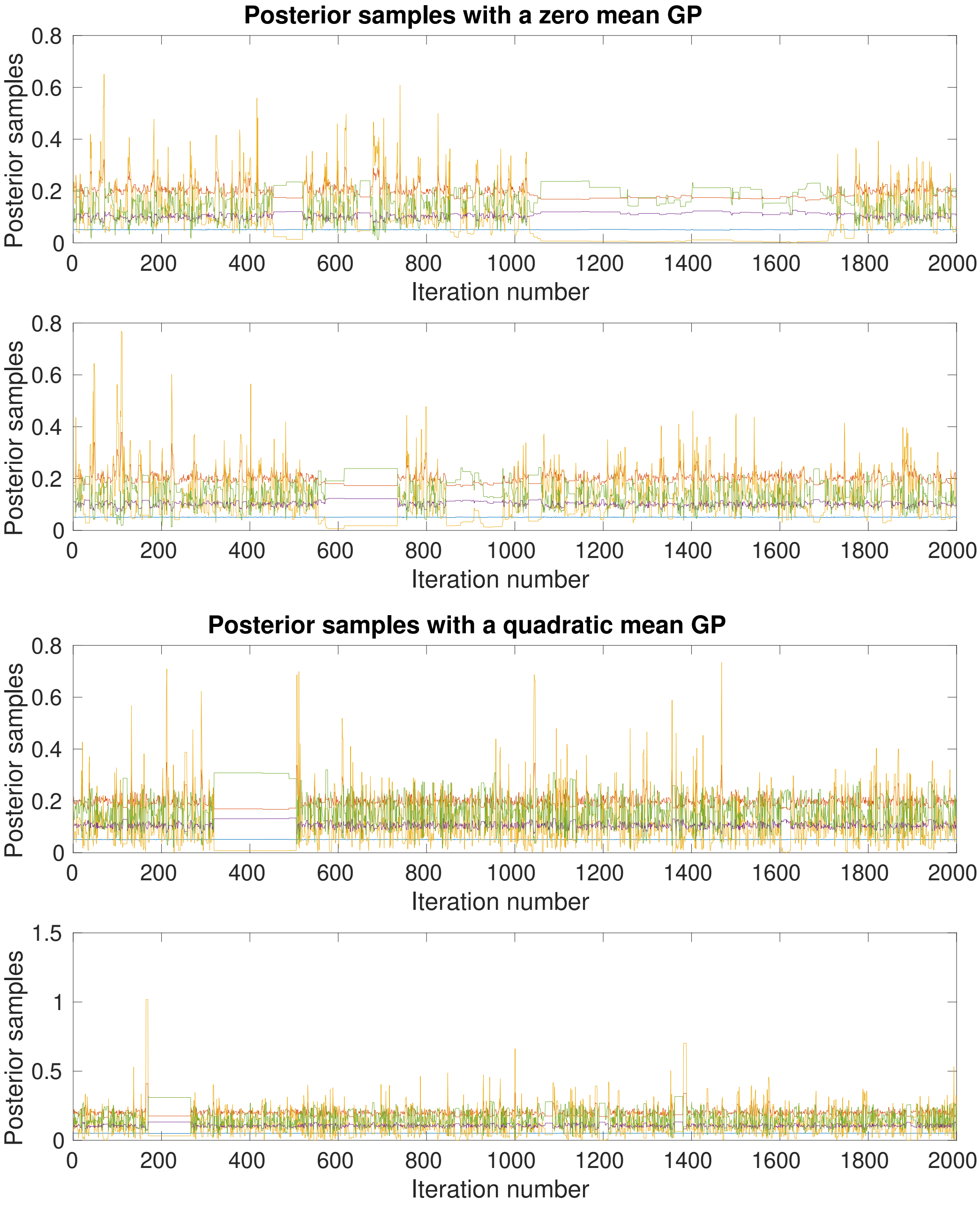


Figure 6.23: Displaying occasional chain stagnation ('stickiness') in the sampling phase for the biochemical pathway model when a zero mean GP (top two panels) or quadratic mean GP (bottom two panels) is used for the emulation of the residual-sum-of-squares. The chains consist of posterior samples of the five parameters for two different initialisations for one data set.

GP model	Acceptance rate	ESS/N
Zero-mean GP	57%	(0.24, 0.04, 0.02, 0.03, 0.06)
Quadratic mean GP	77%	(0.55, 0.48, 0.24, 0.32, 0.42)

Table 6.3: Hamiltonian Monte Carlo (HMC) results for two GP models (zero mean vs quadratic mean GP) of the log unnormalised posterior for the biochemical signalling pathway model in eq (6.21). The acceptance rate and effective sample size (ESS) normalised by the number of HMC iterations N is the median over 10 algorithm initialisations for one data set.

Quasi algorithms for non-positive definite negative Hessian matrix

When running the higher-order methods (RMHMC and LDMC), difficulties were encountered due to the negative Hessian matrix of the log posterior not being positive definite, as illustrated in Figure 6.24 showing regions in 2D parameter space for which the negative Hessian matrix is not positive definite. Large regions where the negative Hessian matrix is not positive definite can be observed. Similarly to the FitzHugh-Nagumo example, the Quasi algorithms described in Section 6.3.3 were implemented, however these were unable to perform satisfactorily. There were two types of problems: for some data sets, the Quasi algorithms returned samples drawn mostly with the HMC algorithm, i.e. the percentage of HMC-drawn samples was roughly 85-90%, while 10-15% of the samples were drawn with RMHMC or LDMC, the reason being a high condition number ($> 10^{15}$) of the mass matrix set using eq (6.16). For other data sets, the Bayesian optimisation employed for RMHMC or LDMC tuned the hyperparameters to very low values (higher values would encourage the sampler to step into regions where the matrix has high condition number), leading to the sampler making tiny, ineffective moves, resulting in low ESS. These issues illustrate failure of second-order HMC methods (RMHMC, LDMC) when combined with emulation of the log unnormalised posterior distribution due to the sub-optimality of the mass matrix. Hence, these algorithms were excluded from the comparison, and subsequent results for the biochemical pathway example are presented for HMC and NUTS only.

DA vs noDA

The distributions over 10 data sets of the normalised minimum, median and maximum ESS (calculated using eqns (6.24)-(6.26)) for the two schemes mostly overlap, as illustrated in Figure 6.25. The hypothesis testing (see eq (6.27)) revealed a p-value > 0.05 for all measures, except MinESS/CPutime, MedianESS/CPUtime, MaxESS/CPUtime, and MaxESS/number of forward evaluations, for which DA seems to have a slight advantage (Figure 6.25). For consistency with previous examples, the DA algorithms were taken forward.

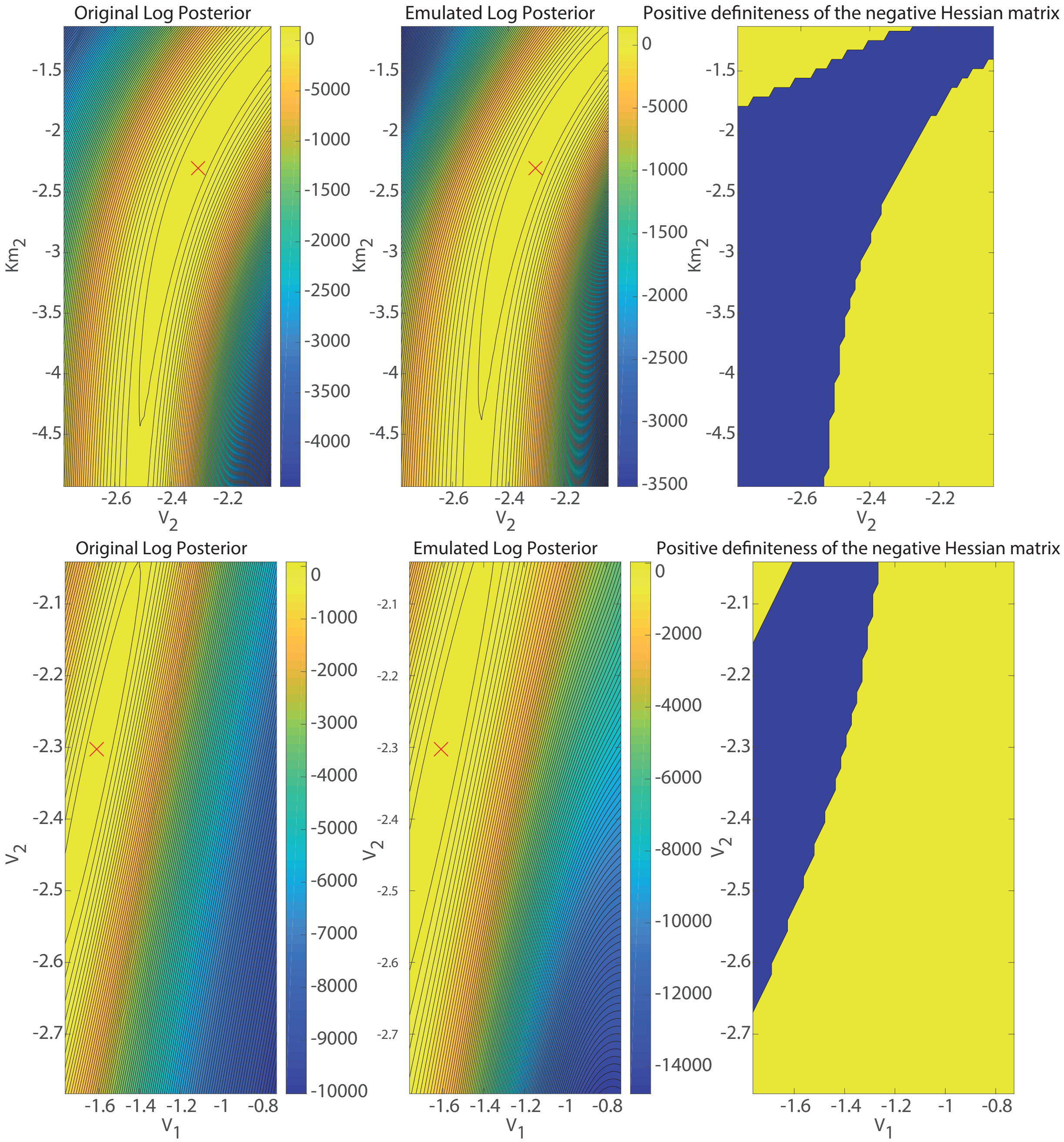


Figure 6.24: Biochemical pathway model: Displaying the positive definiteness of the negative Hessian matrix (right panel) for the emulated log unnormalised posterior distribution in the middle panel, to be compared to the original log unnormalised posterior distribution in the left panel for two of the parameters (the other parameters are kept fixed at their true values). Blue is positive definite, yellow is non-positive definite. The red cross in the plot of the log unnormalised posterior distribution marks the true parameter value. The parameter values on the log scale are shown on the (x,y) axis.

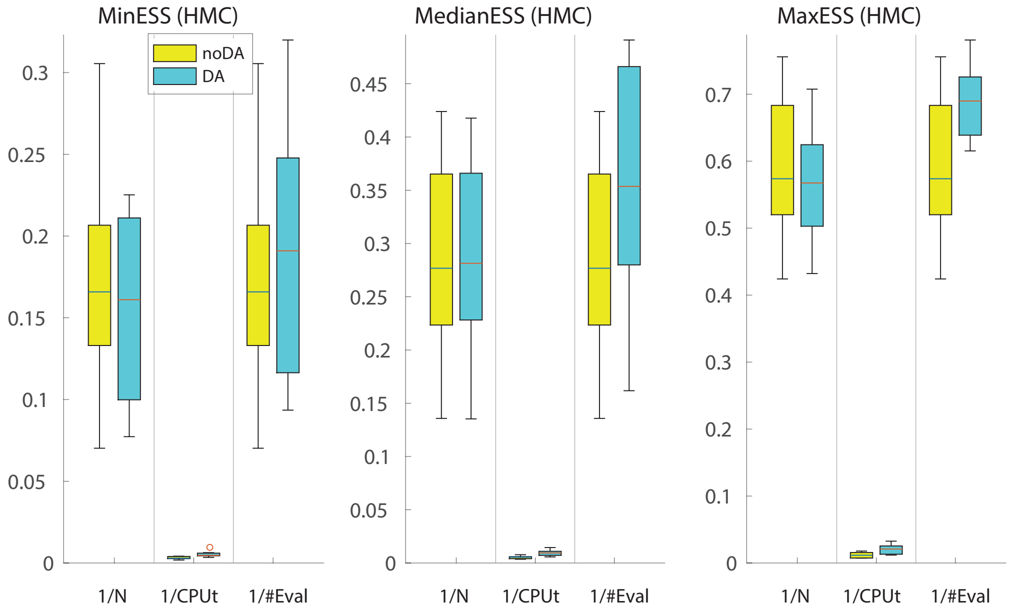


Figure 6.25: Efficiency in terms of ESS of DA-GPHMC versus noDA-GPHMC for the biochemical pathway example. Minimum ESS, median ESS and maximum ESS are calculated as per eqns (6.24)-(6.26). The distribution of min, median, max ESS over 10 data sets is shown. Section 6.6.3 contains justification as to why the algorithms Riemann Manifold Hamiltonian Monte Carlo and Lagrangian Dynamical Monte Carlo are not included in the comparison.

Accuracy

Parameter space: Table 6.4 illustrates the mean and standard deviation of the posterior medians based on 10 data sets (see eq (6.29)) for the ODE parameters drawn with all emulation methods, and of the noise variances, sampled with Gibbs. Figure 6.26 displays the distribution

Algorithm	k_1	V_1	K_{m_1}	V_2	K_{m_2}	σ_S^2 & σ_D^2 & σ_R^2 & $\sigma_{R_{pp}}^2$
DA-GPHMC	0.0506 (0.0008)	0.2021 (0.0142)	0.1166 (0.0516)	0.0996 (0.0049)	0.1142 (0.0233)	0.0005 (0.0001) & 0.0005 (0.0001) & 0.0006 (0.0001) & 0.0005 (0.0001)
DA-GPNUTS	0.0506 (0.0008)	0.2021 (0.0146)	0.1167 (0.0518)	0.0997 (0.0052)	0.1150 (0.0268)	0.0005 (0.0001) & 0.0005 (0.0001) & 0.0006 (0.0001) & 0.0005 (0.0001)
True value	0.05	0.2	0.1	0.1	0.1	0.0004 & 0.0004 & 0.0004 & 0.0004

Table 6.4: Parameter estimates and standard deviations for the biochemical pathway example for each of the methods compared (note that the noise variances were sampled using Gibbs sampling). The mean and standard deviation of the posterior medians for 10 data sets, calculated using eq (6.29) are shown. The true parameter values are also displayed.

over data sets of the biases in parameter space, obtained by subtracting the true parameter from the agglomerated posterior samples (eq (6.30)) for all methods. Figure 6.26 and Table 6.4 reveal that the methods perform very similarly in terms of accuracy, with the distribution of biases for the ODE parameters and the "species" noise variances including the zero bias line. A slight bias (overestimation) is observed for the k_1 parameter. The coin tossing experiment reveals that the probability of obtaining six or more heads (equivalent to obtaining six or more positive offsets) out of eight tosses is 0.145 (out of the 10 offsets, two were negative, six were positive and two had zero bias).

Functional space: Figure 6.27 shows the distribution of the R^2 (eq (6.32)) values over data sets for all methods and each "species". The different methods perform similarly well, giving a very large R^2 (~ 0.99).

Parameter UQ: When analysing the marginal posterior distributions for the ODE parameters and noise variances for one random data set (see Figure 6.28), it can be observed that the distributions generated with the different methods overlap greatly.

Efficiency

Method comparison: The performance of the methods with respect to efficiency is investigated, as quantified using min, median and max ESS (see eqns (6.24)-(6.26)) normalised by the total number of MCMC samples N, by the CPU time, and by the number of forward (model) evaluations, and the latter measure is the only one generalisable to ODE/PDE models for which a forward model evaluation is computationally burdensome. Results based on all three measures are presented in Figure 6.29, which shows the distribution of these quantities over 10 data sets. In terms of normalised MinESS and MedianESS the HMC algorithm has a superior performance to NUTS, while the performance between the two algorithms is comparable with respect to normalised MaxESS.

Chain stagnation: Figure 6.23 displays the posterior samples for the five parameters obtained by running the sampler from two different initialisations for one data set. By focusing on the simulation using a quadratic mean GP (bottom two figures), fairly long periods of rejections can be observed, after which the sampler recovers, with good mixing exhibited.

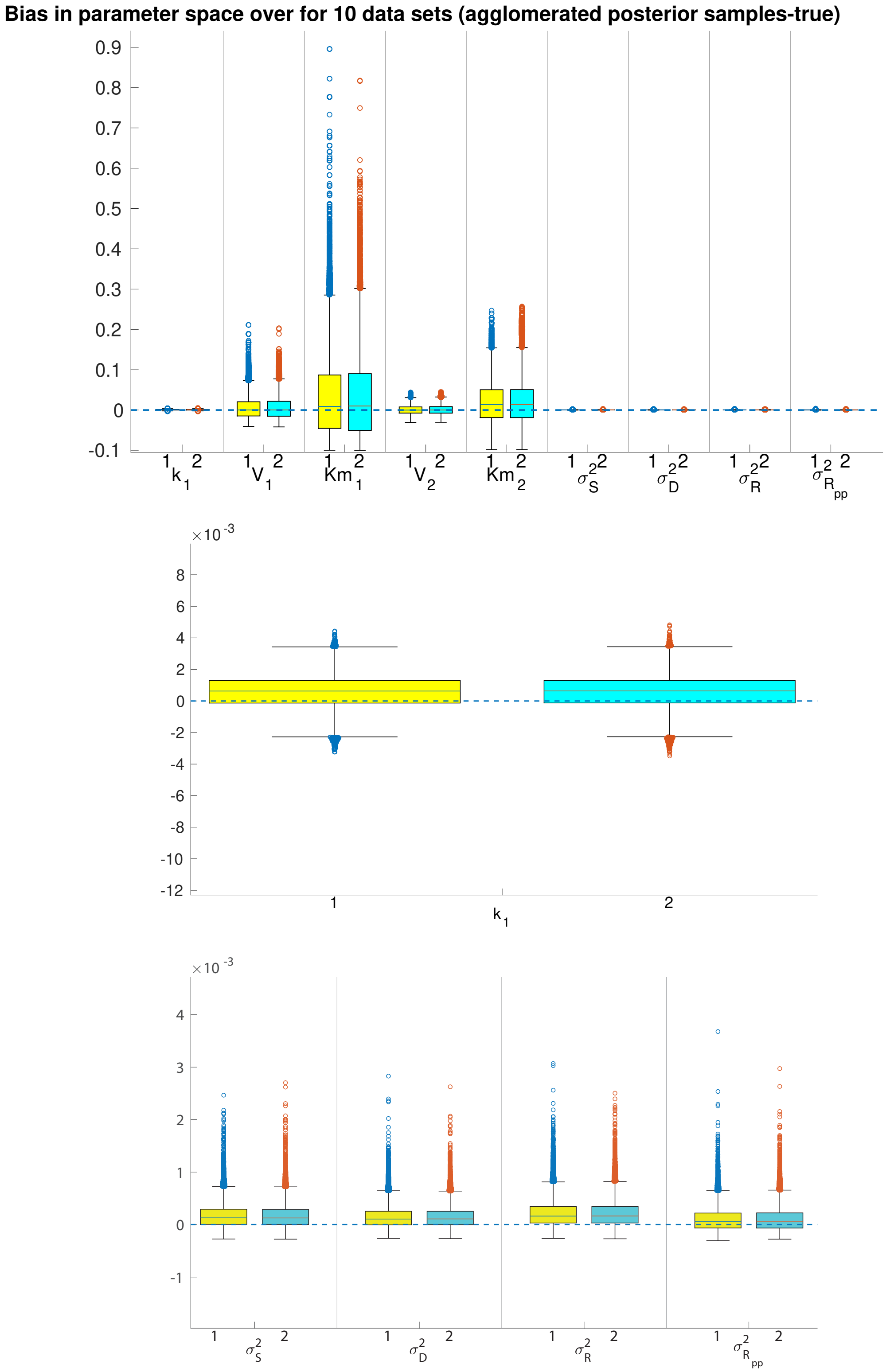


Figure 6.26: Bias in parameter space for each parameter (k_1 – zoomed in in the middle panel, $V_1, K_{m_1}, V_2, K_{m_2}, \sigma^2_S, \sigma^2_D, \sigma^2_R, \sigma^2_{R_{pp}}$ – noise variances zoomed in in the bottom panel) of the biochemical pathway example. The bias is given by the difference between the posterior samples and the true parameter value (see eq (6.30)). The posterior samples were drawn using each of the algorithms compared (1: DA-GPHMC, 2: DA-GPNUTS), and Gibbs sampling was used for sampling the noise variances. The distribution of the agglomerated biases for 10 data sets is shown. The horizontal dashed line indicates zero bias.

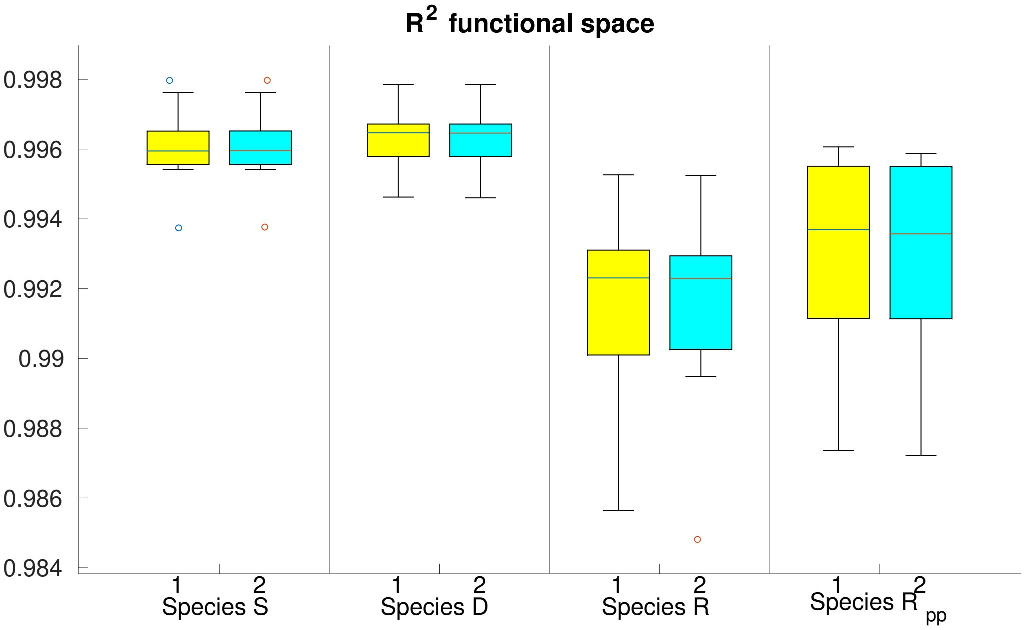


Figure 6.27: Accuracy quantification in functional space via R^2 for the biochemical pathway example. R^2 was computed using eq (6.32) with the posterior median value obtained from posterior samples drawn using each of the algorithms compared (1: DA-GPHMC, 2: DA-GPNUTS). The distribution of R^2 over 10 data sets is shown.

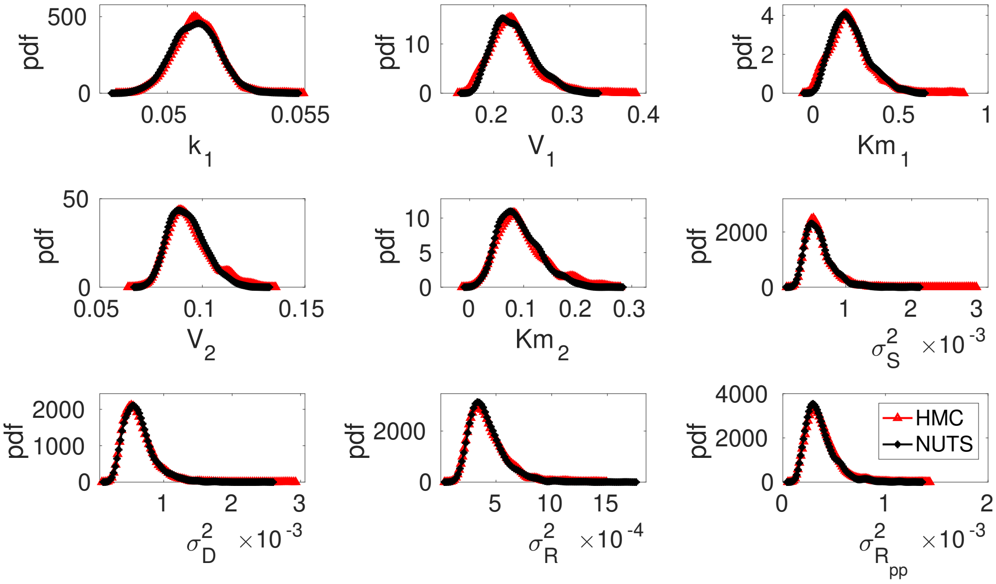


Figure 6.28: Parameter uncertainty quantification: marginal posterior distributions obtained via 1D kernel density estimation from the posterior samples of each parameter $k_1, V_1, K_{m_1}, V_2, K_{m_2}, \sigma_S^2, \sigma_D^2, \sigma_R^2, \sigma_{R_{pp}}^2$ of the biochemical pathway example. The posterior samples for the ODE parameters were drawn using each of the algorithms compared (1: DA-GPHMC, 2: DA-GPNUTS), and Gibbs sampling was used for sampling the noise variances. The marginal distributions for one random data set are shown.

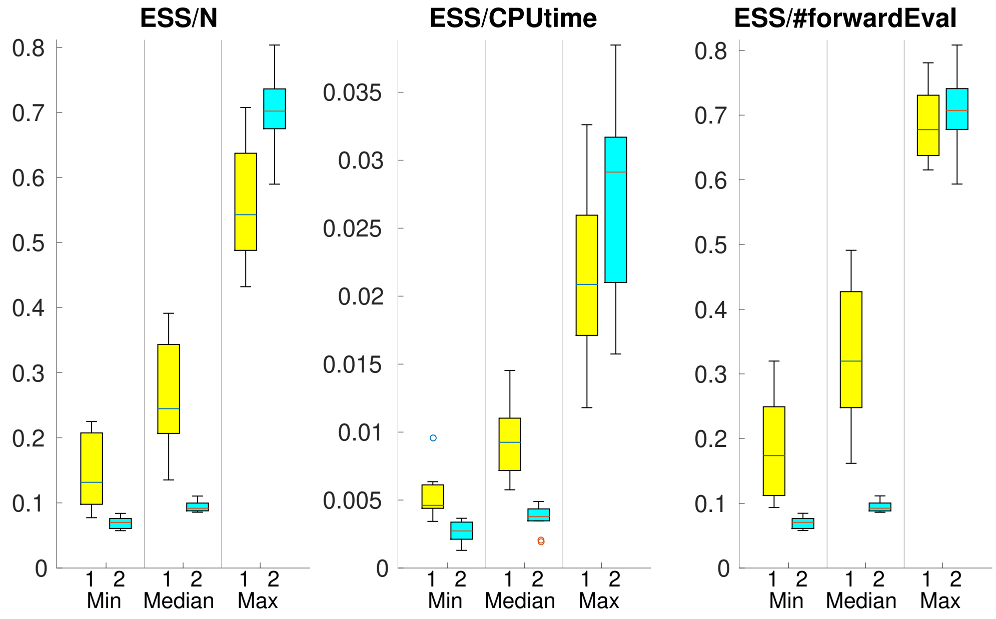


Figure 6.29: Efficiency in terms of ESS of the algorithms compared (1: DA-GPHMC, 2: DA-GPNUTS) for the biochemical pathway example. Min, median, max ESS were calculated as per eqns (6.24)-(6.26). The distribution over 10 data sets is shown.

6.6.4 Real-world application: 1D fluid-dynamics model of the pulmonary blood circulation

DA vs noDA

Figure 6.30 displays the distributions of the normalised ESS *over all four parameters* (see eq (2.51)) for both schemes: DA and noDA, and all algorithms, for the particular data set available. The distributions overlie, except for RMHMC with respect to ESS/CPUtime (DA seems to have an advantage, see Figure 6.30). This tallies with results from the hypothesis test, testing for equality of normalised ESS means across the distributions over parameters of the DA and noDA methods (i.e. p-value > 0.05 for all metrics and algorithms except ESS/CPUtime for RMHMC). These findings suggest no difference in efficiency between the DA and noDA schemes for this particular data set, except for RMHMC with respect to ESS/CPUtime. Similarly to the previous three examples, the DA algorithms were taken forward for method comparison.

Accuracy

Table 6.5 shows the posterior median and 95% credible interval of the PDE parameters obtained from the posterior samples generated with all the methods (DA-GPHMC, DA-GPNUTS, DA-GPRMHMC, DA-GPLDMC), and of the noise variance, sampled with Gibbs sampling; the standard deviation of these estimates is also attached. These estimates are for one single measured (real) data set, for which the ground truth parameter values are unknown. In addition,

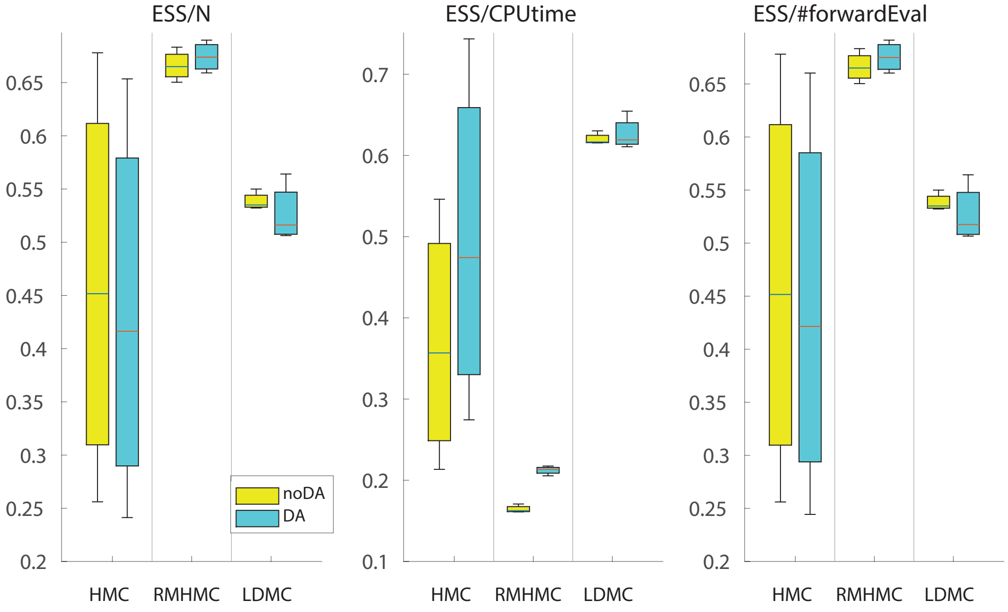


Figure 6.30: Efficiency in terms of ESS of DA-GPHMC, DA-GPRMHMC, DA-GPLDMC versus their noDA version for the fluid-dynamics pulmonary example. ESS was calculated as per eq (2.51). The distribution of ESS over the four parameters for one single data set is shown.

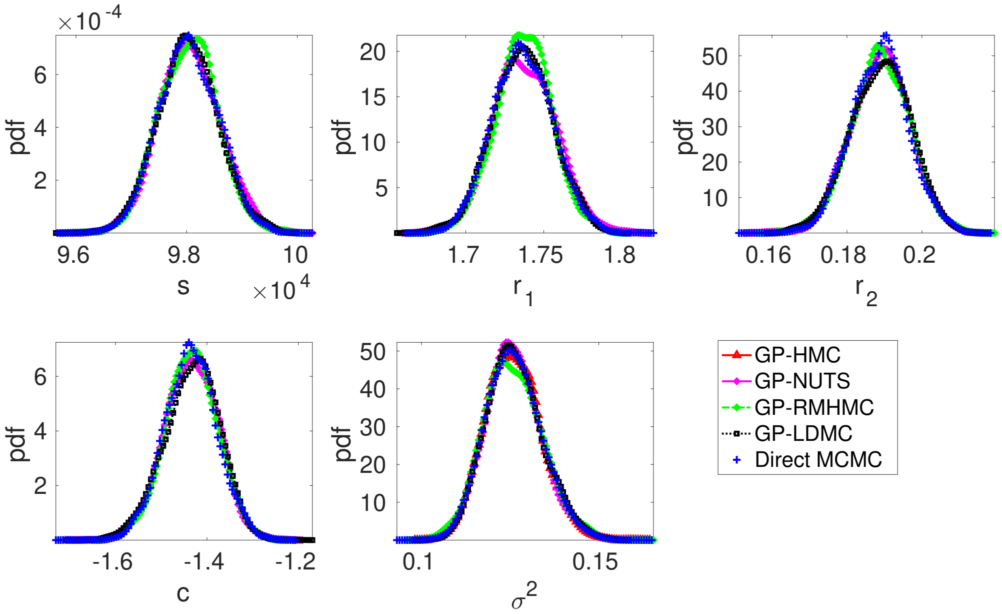


Figure 6.31: Parameter uncertainty quantification: marginal posterior distributions obtained via 1D kernel density estimation from the posterior samples of each parameter s, r_1, r_2, c, σ^2 of the fluid-dynamics pulmonary example. The posterior samples for the PDE parameters were drawn using each of the algorithms compared (1: DA-GPHMC, 2: DA-GPNUTS, 3: DA-GPRMHMC, 4: DA-GPLDMC), and Gibbs sampling was used for sampling the noise variances. The Adaptive Metropolis [82] algorithm was used to draw samples from the exact posterior distribution (Direct MCMC). This enables to test if the emulation approach gives any bias. The marginal distributions for one available data set are shown.

Figure 6.31 displays the marginal posterior distributions of the parameters. In the absence of a gold standard, to test whether the emulation approach gives any bias, results obtained with a long-run MCMC sampler (the Adaptive Metropolis (AM) algorithm [82]) drawing samples directly from the asymptotically exact posterior distribution, are also shown and are taken as a proxy for the gold standard. HMC running directly on the original posterior distribution would incur excessive computational costs (see Section 6.2.1 for a discussion on this), hence in our work we opt for a random-walk algorithm (AM).

Table 6.5 and Figure 6.31 suggest that all methods provide very similar results. The overlapping distributions for the different algorithms indicate that the methods provide samples from approximately the same distribution. In the absence of a proper gold standard – a consequence of the fact that real data are used, for which the true parameter values are unknown – this agreement between the predicted posterior probability distributions across the emulation algorithms and the long-run MCMC sampler (direct MCMC) can be taken as a proxy for accuracy. This statement is backed up by a very high R^2 value (see Table 6.5) registered by all methods, indicating a very good fit to the measured data.

Algorithm	s	r_1	r_2	c	σ^2	R^2
DA-GPHMC	98043 (97004, 99159)	1.7371 (1.6997, 1.7765)	0.1892 (0.1740, 0.2035)	-1.4339 (-1.5495, -1.3264)	0.1256 (0.1119, 0.1420)	0.99
DA-GPNUTS	98021 (96998, 99136)	1.7364 (1.6996, 1.7748)	0.1891 (0.1743, 0.2026)	-1.4322 (-1.5409, -1.3259)	0.1256 (0.1116, 0.1425)	0.99
DA-GPRMHMC	98038 (97040, 99052)	1.7374 (1.7023, 1.7716)	0.1889 (0.1728, 0.2035)	-1.4340 (-1.5505, -1.3260)	0.1255 (0.1109, 0.1432)	0.99
DA-GPLDMC	98024 (96967, 99132)	1.7366 (1.7006, 1.7731)	0.1893 (0.1734, 0.2037)	-1.4294 (-1.5591, -1.3280)	0.1258 (0.1122, 0.1429)	0.99

Table 6.5: Accuracy in parameter and functional space for the statistical inference performed on the fluid-dynamics pulmonary application. The parameter posterior medians and 95% credible interval are shown for all the emulation methods employed (note that the noise variance was sampled using Gibbs sampling). R^2, computed as in eq (6.32), is also displayed.

Efficiency

The methods are evaluated with respect to efficiency, quantified using ESS (see eq (2.51)) normalised by the total number of MCMC samples N, by the CPU time, and by the number of forward (model) evaluations. Results based on all three measures are given in Figure 6.32, which displays the distribution of these quantities *over all four parameters* for the single data set analysed. The RMHMC algorithm is superior to all algorithms when analysing ESS/N (left

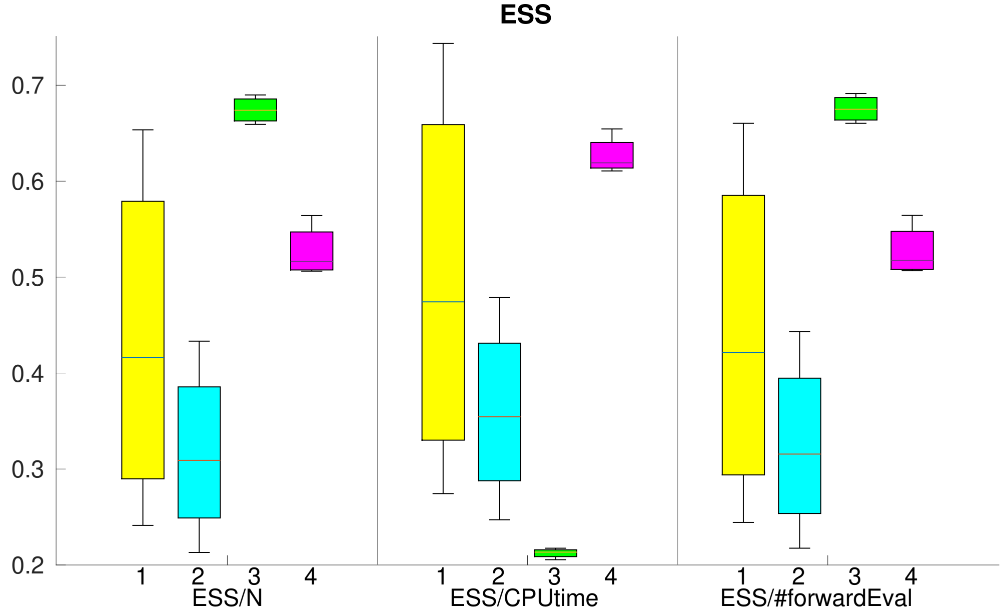


Figure 6.32: Efficiency in terms of ESS of the algorithms compared (1: DA-GPHMC, 2: DA-GPNUTS, 3: DA-GPRMHMC, 4: DA-GPLDMC) for the fluid-dynamics pulmonary example. ESS was calculated as per eq (2.51). The distribution of ESS over the four parameters for one single data set is shown.

panel) or ESS/#forwardEval (right panel), however it loses its advantage, becoming the worst when the CPU time is accounted for (middle panel). Additionally, NUTS systematically performs more poorly than the other parameters. LDMC is clearly superior to HMC when looking at the minimum or median ESS/*N*, ESS/CPUtime or ESS/#forwardEval, and HMC is better when looking at the maximum ESS/*N*, ESS/CPUtime or ESS/#forwardEval. The distribution of these three quantities is much more variable for HMC and NUTS than for RMHMC and LDMC.

6.6.5 Advantage of surrogate modelling

Table 6.6 displays the number of model solves (ODEs/PDEs) required to obtain one single sample drawn using the conventional HMC versus the emulation HMC (GPHMC) for all four test examples considered in this study. It is clear that the use of the surrogate model in place of the computationally expensive ODE/PDE simulator leads to a substantial reduction in the number of model evaluations.

6.7 Discussion

This study proposes to accelerate Hamiltonian/Lagrangian Monte Carlo algorithms by coupling them with Gaussian Processes for emulation of the log unnormalised posterior distribution. Proofs of convergence to the asymptotically exact posterior distribution were provided for these

Test example	Algorithm	Number of model solves per HMC sample
sinusoidal	conventional HMC	80 (44)
	emulation HMC	1
FitzHugh-Nagumo	conventional HMC	284 (162)
	emulation HMC	1
biochemical pathway	conventional HMC	1224 (703)
	emulation HMC	1
fluid-dynamics	conventional HMC	113 (62)
	emulation HMC	1

Table 6.6: Number of model evaluations (ODEs/PDEs) required to obtain one single HMC sample drawn using conventional HMC versus emulation HMC (GPHMC) algorithm (mean and standard deviation) for all test examples considered in this study. The number of model solves for the conventional HMC is $L(d+1)$, where L is the number of leapfrog steps and d is the parameter dimensionality. The term $d+1$ is the sum of one model evaluation to find the log likelihood, and d model evaluations to find its numerical derivatives by a first-order differencing scheme with respect to each of the d parameters. For the sinusoidal model $d=3$ and optimum $L=39$, for the FitzHugh-Nagumo model $d=3$ and optimum $L=141$, for the biochemical pathway model $d=5$ and optimum $L=407$, and for the fluid-dynamics model, $d=4$ and optimum $L=44$. The optimum L value was obtained with Bayesian optimisation. HMC was run with the number of leapfrog steps drawn from a uniform distribution with lower bound being 1 and upper bound being optimum L [208]. For the sinusoidal, FitzHugh-Nagumo and biochemical pathway model, optimum L and number of model evaluations are reported for a random data set.

algorithms, and the mathematical and coding correctness of the samplers' implementation was validated by Geweke consistency tests (Figure 6.8). Moreover, a comparative evaluation study was carried out to assess the performance of the methods on a series of models described by ODEs or PDEs (sinusoidal, FitzHugh-Nagumo, biochemical pathway and fluid-dynamics pulmonary model). The aim was to identify the most computationally efficient and accurate parameter inference and UQ tool to be applied to non-linear ODE or PDE models which typically incur onerous computational costs caused by repeated numerical integrations. In addition, it was investigated whether the delayed acceptance scheme used in conjunction with these algorithms can further offer computational gains over the standard algorithms.

6.7.1 A discussion on the algorithms compared

The following algorithms were compared: noDA-GPHMC (i.e. standard GPHMC), DA-GPHMC, DA-GPNUTS, noDA-GPRMHMC, DA-GPRMHMC, noDA-GPLDMC, DA-GPLDMC. While the standard GPHMC was originally proposed in [156], all the other algorithms are methodologically new and my own work. For all algorithms but NUTS, Bayesian optimisation was used to tune the hyperparameters: step size and number of leapfrog steps. NUTS has an in-built mechanism to tune the number of steps recursively, while the step size is tuned with stochastic optimisation [87].

GPNUTS

For the NUTS algorithm, only the DA-GPNUTS was implemented, and not noDA-GPNUTS. This is because the noDA-GPNUTS algorithm requires evaluation of the differential equations several times along the trajectory before making a proposal (see Section 6.9.8, in particular eq (6.34) on page 184). In contrast, DA-GPNUTS only evaluates the ODEs/PDEs once at the end of the trajectory (as do all the other algorithms investigated: noDA-GPHMC, DA-GPHMC, noDA-GPRMHMC, DA-GPRMHMC, noDA-GPLDMC, DA-GPLDMC). This implies that noDA-GPNUTS would incur a larger computational burden than DA-GPNUTS, since the number of forward evaluations is roughly one order of magnitude higher for noDA-GPNUTS compared to DA-GPNUTS.

Quasi algorithms for non-positive definite negative Hessian matrix

While the HMC and NUTS algorithms make use of the first-order derivative of the log posterior, the RMHMC and LDMC algorithms use second and third-order derivative of the log posterior, thus they utilise the Fisher information matrix and its derivative. As discussed in Section 6.3.3, due to emulating the log posterior instead of the signal, the expected Fisher information matrix could not be used, instead the observed Fisher information matrix (the negative Hessian matrix of the log posterior) was used, which is however not guaranteed to be positive definite and is not a valid metric tensor, as required by RMHMC and LDMC. While using the negative Hessian worked for the sinusoidal and pulmonary model, it failed for the FitzHugh-Nagumo and biochemical pathway models, which had regions in parameter space for which the negative Hessian was non-positive definite (see Figures 6.16 and 6.24). In that case a different form for the metric tensor was adopted, see eq (6.16), which ensures that the mass matrix can never be non-positive definite for a large enough λ. However, it can have a high condition number ($> 10^{15}$).

To deal with this, an approach akin to a Quasi-Newton method in optimisation [19] was taken, i.e. if at any point throughout the trajectory, the matrix is numerically unstable, the simulation within the trajectory is stopped prematurely before reaching the end. A new simulation is started from the beginning of the trajectory and the HMC algorithm is used instead of RMHMC/LDMC for that particular iteration. The resulting posterior samples will have been drawn using a hybrid version of HMC and RMHMC/LDMC, an algorithm thus called Quasi-HMC-RMHMC or Quasi-HMC-LDMC. This approach achieved high efficiency for the FitzHugh-Nagumo, which registered roughly two thirds of RMHMC-drawn samples and three quarters of LDMC-drawn samples, however it achieved very little efficiency gain for the biochemical pathway, for which a high condition number of the mass matrix was repeatedly registered. This led either to a small percentage of RMHMC or LDMC-drawn samples (<15%), or to low-valued optimised RMHMC/LDMC hyperparameters obtained based on the standard RMHMC or LDMC algorithms (higher values would encourage the sampler to step into regions where the matrix has high condition number), which resulted in small moves, thus an ineffective sampler.

In general, approaches utilising the Hessian are prone to facing positive definiteness issues for non-convex problems, and ad-hoc modifications to impose positive definiteness of the matrix, discussed in Section 6.3.3, may change the topology of the log posterior. Consider for example a saddle point for which the Hessian has both positive and negative eigenvalues. By adding twice the minimum negative eigenvalue as a diagonal term, the saddle point is ignored, and the non-convex problem is wrongly transformed into a convex one, ultimately converging to the wrong parameter values.

6.7.2 Emulation of the model output

The reason for the inability of computing the expected Fisher information matrix in eq (2.40) (which should be used in RMHMC/LDMC) is the fact that the log unnormalised posterior distribution was emulated, and not the model output (signal). This leads to loss of information as the (time) pointwise log posterior values cannot be recovered, only the sum of the pointwise values is known (see Section 6.3.3 for specific equations and details). A solution to this is to emulate the signal. Different strategies for the emulation of multivariate outputs have been proposed in the literature [42], e.g. ensemble of single-output emulators (MS), or multivariate-output Gaussian Process (MO) [126, 224]. While the first strategy (MS) fits independent emulators to each model output point (i.e. if the signal is measured at k time points, k independent emulators will be fitted), it ignores any correlation between the points. However it allows every emulator have its own kernel hyperparameters, and if k computer cores are available, they can all be used in parallel for k emulator predictions at the cost of one. The second strategy (MO) fits a k-dimensional Gaussian Process to the multivariate output, thus correlation is naturally incorporated, and the computational burden is equivalent to that of fitting a single output emulator. The downside is that the kernel hyperparameters are shared between dimensions, however there is no reason to assume that changes in input will affect the different outputs in the same way [42].

6.7.3 Advantage of delayed acceptance

This study reveals that for most ODE/PDE models considered there is no evidence that the DA scheme brings any computational gains when coupled with Hamiltonian/Lagrangian Monte Carlo algorithms. The efficiency of DA-GPHMC, noDA-GPHMC (i.e. standard GPHMC as proposed in [156]), DA-GPRMHMC, noDA-GPRMHMC, DA-GPLDMC, noDA-GPLDMC, as measured in terms of ESS normalised by the total number of MCMC samples, ESS normalised by CPU time and by number of model (forward) evaluations, is comparable between the DA and noDA algorithms. This is displayed in Figures 6.9, 6.17, 6.25 and 6.30, showing overlapping distributions, with a few exceptions, e.g. for MedianESS/CPUtime for HMC in the Fitz-Hugh Nagumo model, or ESS/CPUtime for the biochemical pathway model, or ESS/CPUtime for GPRMHMC in the pulmonary model, for which the DA scheme appeared slightly better. This statement was validated by a formal hypothesis test testing for equal sample means of normalised ESS, and a p-value>0.05 was registered for most algorithms and metrics used, except a few, as given above.

While an MCMC with DA approach has been taken in previous studies in the literature [7, 33, 47, 76, 85, 149, 177], to the author's best knowledge, this study is the first to use it in conjunction with Hamiltonian/Lagrangian Monte Carlo algorithms. Previous work has compared standard MCMC algorithms to DA-MCMC algorithms. For example, Golithly et al. [76] showed that the DA scheme can lead to improvements in computational efficiency in a particle MCMC algorithm applied to stochastic kinetic models. Additionally, Banterle et al. [7] and Quinoz et al. [149] showed that DA brings computational advantage when applied to M-H algorithms on large data sets, for which data sub-sampling is employed. However, these MCMC algorithms are based on a random-walk, which is known to have a lower acceptance rate (and efficiency) than the gradient-driven Hamiltonian Monte Carlo algorithms (see Ch. 5 in [17] or [174]). Therefore, for the former algorithms there is more potential for the DA scheme to be advantageous than for the latter algorithms (i.e. if a rejection is more likely, a higher number of computationally expensive model evaluations are avoided by the first stage employing the surrogate model[9]).

[9] provided the surrogate is an accurate representation of the simulator

This may help explain why previous studies employing DA with random-walk MCMC have benefited (from a computational point of view) from using DA, which is hardly the case for this study using DA with Hamiltonian/Lagrangian Monte Carlo algorithms.

6.7.4 Accuracy

The accuracy in parameter and functional space proved to be very similar between the different methods for all ODE/PDE models considered, see Figures 6.11, 6.13, 6.18, 6.19, 6.26, 6.27 and Table 6.5. In addition, for the toy examples, it was shown that the algorithms were able to learn the true parameter values that generated the data (Tables 6.1, 6.2, 6.4). The marginal posterior distributions constructed from the MCMC posterior samples showed overlapping distributions, indicating that the uncertainty quantification was on a par for all methods (Figures 6.14, 6.20, 6.28, 6.31).

6.7.5 Efficiency

ESS normalised by the total number of MCMC samples

In terms of ESS normalised by the total number of MCMC samples N (left panel in Figures 6.15, 6.21, 6.29, 6.32), it was found that the performance of DA-GPNUTS was generally inferior to that of the other algorithms (DA-GPHMC, DA-GPRMHMC, DA-GPLDMC). A possible explanation is that for DA-GPNUTS the hyperparameter (step size and number of steps) tuning is performed in the emulated log posterior entirely, based on samples accepted at the emulator stage, due to the construction of the algorithm (see the proof in Section 6.9). In contrast, for the other three algorithms the tuning is performed by taking into account the simulator, i.e. based on samples accepted at the simulator stage, thus the simulator plays a role in the choice of optimum hyperparameters, which positively affects efficiency. In terms of minESS/N generally the RMHMC and LDMC algorithms perform better than HMC, while in terms of medianESS/N or maxESS/N no clear pattern is observed (sometimes RMHMC and LDMC are better, other times HMC is preferred). A much larger discrepancy can be seen between minESS/N and maxESS/N for HMC and NUTS than RMHMC and LDMC for which ESS/N varies much less across parameters. This is a consequence of the latter two algorithms using a mass matrix set via the curvature of the log posterior (metric tensor), while HMC and NUTS use an identity matrix as the mass matrix, and the optimum step size is restricted by the lowest marginal variance. That implies that for HMC/NUTS all the work falls on the back of the step size (and number of steps) to optimise efficiency, while for RMHMC/LDMC this is alleviated by the use of the metric tensor. For this reason, a first-order algorithm like HMC or NUTS can be more inefficient (e.g. in terms of minESS/N) for problems with large discrepancies between the lowest and largest marginal variance. Generally, RMHMC and LDMC perform similarly, an exception is the pulmonary model which registers better performance for RMHMC than LDMC.

ESS normalised by the CPU time

In terms of ESS normalised by the CPU time (middle panel in Figures 6.15, 6.21, 6.29, 6.32), the following observations can be made. RMHMC incurs high computational costs due to the use of the implicit integrator in the simulation of the Hamiltonian dynamics, becoming the algorithm with the worst performance when the CPU time is taken into account. In terms of MinESS/CPUtime and MedianESS/CPUtime, NUTS performs second worst (or worst for the biochemical pathway), except for the sinusoidal example; no systematic pattern is observed for MaxESS/CPUtime. Inspection of the minESS/CPUtime reveals that LDMC tends to be the best algorithm, while medianESS/CPUtime shows no clear pattern (sometimes HMC is better, other times LDMC is preferred). This makes sense considering that HMC has the advantage of avoiding calculations of higher-order derivatives, but requires a larger number of steps to be made, while LDMC calculates higher-order derivatives, but only needs a few steps. In terms of maxESS/CPUtime, HMC is consistently best. However, maxESS is an inflated measure since ideally an algorithm that gives the highest number of uncorrelated samples across all parameters is preferred, hence minESS (or medianESS) is a better measure.

ESS normalised by the number of forward evaluations

In terms of ESS normalised by the number of model (forward) evaluations (right panel in Figures 6.15, 6.21, 6.29, 6.32), a similar pattern as for ESS/N is observed, which is expected given the high acceptance rate (>80%), meaning that the number of model evaluations in close to the total number of MCMC samples. This finding also helps explain why no advantage of the DA scheme was found: generally most proposals are accepted at the emulator (first) stage, and are subsequently subject to the accept/reject at the simulator (second) stage.

The above interpretations for the FitzHugh-Nagumo model apply to the Quasi-RMHMC-HMC/Quasi-LDMC-HMC instead of the standard RMHMC/LDMC.

An algorithm performance metric that is generalisable to any model, regardless of its computational complexity, is ESS/number of forward solves. For models for which a single solve is very computationally expensive, e.g. in the order of hours, the CPU time would be dominated by the computational load from the forward solves, thus ESS/CPU time would not be an appropriate metric to compare strictly the performance of the algorithms since the computational load from the algorithms would be dominated by that of the forward solve. In contrast, ESS normalised by the CPU time is a more suitable efficiency measure for ODE/PDE models for which a forward evaluation incurs low or moderate computational cost, for which emulation might not bring any computational advantages.

The comparative observations made about HMC, NUTS, LDMC and RMHMC are based on empirical findings, to the author's knowledge, no theoretical study comparing the efficiency between the algorithms has been performed.

6.7.6 A zero mean GP versus a quadratic mean GP

For the sinusoidal, FitzHugh-Nagumo and pulmonary models, a zero mean GP prior for the RSS was implemented, producing an emulator faithful to the simulator, which was reflected in efficient and accurate inference results.

For the biochemical example, initially a zero mean GP model was employed. However, this model led to a sub-optimal emulator, see Figure 6.22. This could be a consequence of the fact that 'extreme' RSS values (relative to the low RSS region) were added to the list of training points as the sampler stepped into a region of high uncertainty. In this region, in the absence of data, the posterior relied on the prior for inference, hence the predicted emulated RSS value would be close to 0, despite the true RSS value being high[10]. The result is a sub-optimal performance (acceptance rate and efficiency) of the sampler in the sampling phase.

This issue was alleviated by the use of a quadratic mean GP prior, which potentially places high prior density values on the region where most data points are, i.e. around the mode of the posterior distribution, and lower density values away from the mode, thus the emulator is discouraged to move far away from the high posterior probability region. This modification considerably improved the acceptance rate and the efficiency (see Figure 6.23 and Table 6.3).

6.7.7 Limitations and future improvements for the biochemical pathway example

The biochemical example was a hard problem to emulate due to the high correlations manifested through long ridges in the log posterior landscape (see Figure 6.22), thus large regions in parameter space needed covering by the emulator. This required a large number of training points, which in turn affected efficiency (high CPU times)[11] due to different operations (to compute the GP predictive mean and up to its third-order derivative) involving the high-dimensional covariance matrix (inversion of the covariance matrix was no longer performed in the sampling phase since the emulator was no longer refined). Such operations performed repeatedly as part of the MCMC scheme restricted the number of training points used, which resulted in a somewhat sub-optimal emulator. The consequence was occasional stagnations of the chain, i.e. 'stickiness', see Figure 6.23, caused by a mismatch between the emulator and the simulator in the tails of the target distribution: proposals were accepted at the emulator stage, but rejected at the simulator stage. The use of a larger number of training points in conjunction with sparse GPs [184], which

[10] The proposed points in this area would be rejected based on the simulator in the M-H accept/reject step.

[11] Care has to be taken to ensure that the computational times in approaches employing the emulator remain lower than when the simulator solely is used, else the entire purpose of emulation is defied.

optimally select a lower number of points retaining the maximum information at reduced costs could overcome the issues presented and constitute future work. This strategy could potentially be coupled with continuous refinement of the emulator when the sampler steps into a region of high uncertainty, similar to the study in [40], to avoid deciding when to stop the exploratory phase during which the emulator is refined. An aside observation is that sampler ’stickiness’ is a notorious issue in pseudo-marginal MCMC problems [53, 129], in which the estimator of the target distribution is inconsistent with the true target distribution in the tails. In addition, emulating the model output instead of the objective function, as discussed in Section 6.7.2 will allow to apply the RMHMC and LDMC algorithms on this model, which as previously highlighted, was not possible due to numerical instabilities of the mass matrix.

6.7.8 Future work for the fluid-dynamics model

The method comparison could also be performed on a more physiologically and complex fluid-dynamics model that includes vessel-specific stiffness, as described in Chapter 4. This modification will significantly increase the model complexity, presumably requiring a much larger number of training points needed to train the GP emulator to ensure a dense enough coverage of the parameter space. O’Hagan [137] notes that GP emulation is likely to be implemented effectively with up to 50 inputs on modern computing platforms. The number of training points required to optimally cover a 50D input space depends on the complexity of the problem (e.g. smoothness of the log posterior), and is restricted by the $\mathrm{O}(n^3)$ computational complexity of GP emulators (due to the covariance matrix inversion). Conventional GPs may be replaced by sparse GPs [184] for the emulator. In addition, the model mismatch discussed in Chapter 4 should be incorporated into future analysis.

6.7.9 Connection to cardiovascular modelling

Cardiovascular models have been proved to have real potential in enabling better understanding of cardiovascular (patho)physiology, as well as assist clinicians in the diagnosis, prognostication and treatment of cardiovascular diseases [91]. Moreover, cardiovascular modelling can aid in surgical interventions, in the design and evaluation of medical devices, and even in the inference of unknown and immeasurable parameters from measured data [91]. Before being used as a decision support system in the clinic, these models must first be adapted to patient-specific conditions and an assessment of their credibility (uncertainty quantification, UQ) based on a comparison between model predictions and clinical data, must be performed. For example, studies in [89, 91] discuss the requirements that clinically applicable cardiovascular models must meet and the advances required to do so. In addition, the study in [64] demonstrates that biophysical parameters of mathematical cardiovascular models have genuine predictive value for disease prognostication.

However, fast parameter estimation and computational efficiency are paramount for clinical decision making. The benchmark study in this chapter comparing several state-of-the-art sampling methods, particularly adapted to complex and expensive computational models, sheds light on their relative computational efficiencies, and is novel in the context of biophysical modelling. Among cardiovascular mathematical modelling studies, a single study in cardiac electrophysiology was found, which uses a similar idea to that proposed here: Dhamala et al. [50] employ the Metropolis-Hastings algorithm coupled with GPs and DA with a surrogate model for the log posterior distribution, and they take inspiration from a study by Lê et al. [117], who apply the HMC algorithm coupled with GPs to a tumor growth model. However, these studies apply one single algorithm to the problem of interest, without any comparative evaluation of its performance.

Furthermore, the inference procedure proposed here deals with unknown constraints in parameter space, caused by the violation of the physical model assumptions. In such problems, naively employing a GP-MCMC method will yield inaccurate results, hence the use of a multivariate classifier that automatically learns the infeasible parameter domain is proposed, and to the author's best knowledge, this is novel in the cardiovascular and mathematical biology research community.

In addition, considering the number of equivalent number of forward evaluations replaced by the GP surrogate (Table 6.6), it is clear that the method proposed is a key enabler for using ODE/PDE mathematical models as a model-based disease diagnostic in the clinic.

Therefore, this work is a stepping stone towards a decision support system for personalised medicine that can help clinical practitioners to make informed decisions in real time.

6.8 Conclusions

This study proposes to accelerate Hamiltonian/Lagrangian Monte Carlo algorithms by coupling them with Gaussian Processes for emulation of the log unnormalised posterior distribution. Proofs of convergence to the asymptotically exact posterior distribution for these algorithms were provided, and the mathematical and coding correctness of the samplers' implementation was performed by Geweke consistency tests. Investigations into whether the delayed acceptance scheme can offer computational gains over the standard algorithms were done. A comparative evaluation study was carried out to assess the performance of the methods on a series of models described by ODEs and PDEs, including toy problems and a real-world application of the fluid-dynamics in a pulmonary blood circulation model. The aim was to identify the most computationally efficient and accurate parameter inference and UQ tool to be applied to non-linear ODE or PDE models, which typically incur onerous computational costs caused by repeated numerical integrations. Results showed no advantage of the delayed acceptance scheme over the standard algorithms with respect to several efficiency measures based on the effective sample size for most methods and ODE/PDE models considered. Additionally, the methods estimated the true parameter values well, with all methods performing similarly across the ODE/PDE models considered. The Lagrangian Dynamical Monte Carlo and Riemann Manifold Hamiltonian Monte Carlo tended to register the highest efficiency (in terms of effective sample size normalised by the number of forward model evaluations), followed by the Hamiltonian Monte Carlo, and the No U-turn sampler tended to be the least efficient.

6.9 Proofs of convergence

In this part of the chapter, proofs of convergence to the asymptotically exact posterior distribution are provided for the emulation Hamiltonian/Lagrangian Monte Carlo algorithms.

6.9.1 HMC with emulation and DA (DA-GPHMC)

Define

$$p(\theta|\mathbf{y}) \;=\; \frac{\exp(-E(\theta))}{Z_p}; \;\; \tilde{p}(\theta|\mathbf{y}) \;=\; \frac{\exp(-\tilde{E}(\theta))}{\tilde{Z}_p},$$

where $E(\theta) = -(\log p(\theta) + \log p(\mathbf{y}|\theta))$ is the true potential function and $\tilde{E}(\theta) = -(\log p(\theta) + \log \tilde{p}(\mathbf{y}|\theta))$ is the surrogate potential function, and $Z_p = \int_\theta p(\theta)p(\mathbf{y}|\theta)d\theta = p(\mathbf{y})$ is a normalising constant ensuring that the integral of the probability distribution $p(\theta|\mathbf{y})$ is 1. Also, $\tilde{Z}_p = \int_\theta p(\theta)\tilde{p}(\mathbf{y}|\theta)d\theta$ is a normalising constant.

The algorithms starts with a parameter vector θ and proceeds as follows:

1. Sample a vector of auxiliary 'momentum' variables $\mathbf{r}$ from $\mathscr{MVN}(\mathbf{0},\mathbf{M})$ in a Gibbs step, i.e.

   $$p(\mathbf{r}) \;=\; \frac{\exp(-K(\mathbf{r}))}{Z_k},$$

   where $K(\mathbf{r}) \;=\; \frac{1}{2}\mathbf{r}^T\mathbf{M}^{-1}\mathbf{r}$ is the kinetic energy ($\mathbf{M}$ - mass matrix) and $Z_k = \exp\left(-\frac{1}{2}\log\left((2\pi)^d|\mathbf{M}|\right)\right)$ is a normalising constant ensuring that the integral of the probability distribution $p(\mathbf{r})$ is 1. Note that $\tilde{p}(\mathbf{r}|\theta) = p(\mathbf{r}|\theta) = p(\mathbf{r})$ since it is expressed independently of $E(\theta)$.

2. Propose $(\theta^*, \mathbf{r}^*)$ by following a set of Hamiltonian dynamics steps, which are deterministic (so the transition probability is $q(\theta^*, \mathbf{r}^*|\theta, \mathbf{r}) = 1$) in emulated space, and accepting the new configuration with the following acceptance probability (emulator-based):

   $$\alpha_1(\theta^*, \mathbf{r}^*|\theta, \mathbf{r}) \;=\; 1 \wedge \frac{\exp(-\tilde{H}[\theta^*, \mathbf{r}^*])}{\exp(-\tilde{H}[\theta, \mathbf{r}])},$$

   where $\tilde{H}(\theta, \mathbf{r}) = \tilde{E}(\theta) + K(\mathbf{r})$ is the Hamiltonian function.

3. If the configuration has been accepted, accept the parameters with the second stage acceptance probability (simulator-based)

$$\alpha_2(\theta^*, \mathbf{r}^*|\theta, \mathbf{r}) \; = \; 1 \wedge \frac{\exp(-E(\theta^*))}{\exp(-E(\theta))} \frac{\exp(-\tilde{E}(\theta))}{\exp(-\tilde{E}(\theta^*))}.$$

This algorithm preserves detailed balance with respect to $p(\theta|\mathbf{y})$.

PROOF

One needs to show that

$$\alpha_2(\theta^*, \mathbf{r}^*|\theta, \mathbf{r})\alpha_1(\theta^*, \mathbf{r}^*|\theta, \mathbf{r})q(\theta^*, \mathbf{r}^*|\theta, \mathbf{r})p(\mathbf{r}|\theta)p(\theta|\mathbf{y})$$

is invariant with respect to the parameter exchange $(\theta, \mathbf{r}) \leftrightarrow (\theta^*, \mathbf{r}^*)$. The individual terms are as follows:

$$\begin{aligned}
p(\theta|\mathbf{y}) \; &= \; \frac{\exp(-E(\theta))}{Z_p}, \\
p(\mathbf{r}|\theta) \; &= \; \frac{\exp(-K(\mathbf{r}))}{Z_k}, \\
q(\theta^*, \mathbf{r}^*|\theta, \mathbf{r}) \; &= \; 1, \\
\alpha_1(\theta^*, \mathbf{r}^*|\theta, \mathbf{r}) \; &= \; 1 \wedge \frac{\exp(-\tilde{H}[\theta^*, \mathbf{r}^*])}{\exp(-\tilde{H}[\theta, \mathbf{r}])}, \\
\alpha_2(\theta^*, \mathbf{r}^*|\theta, \mathbf{r}) \; &= \; 1 \wedge \frac{\exp(-E(\theta^*))}{\exp(-E(\theta))} \frac{\exp(-\tilde{E}(\theta))}{\exp(-\tilde{E}(\theta^*))},
\end{aligned}$$

Multiplying these individual terms together gives:

$$\begin{aligned}
&\left(\frac{1}{Z_p Z_k}\right) \frac{\exp(-E(\theta))}{\exp(-\tilde{E}(\theta))} \exp(-\tilde{E}(\theta) - K(\mathbf{r}))\alpha_1(\theta^*, \mathbf{r}^*|\theta, \mathbf{r})\alpha_2(\theta^*, \mathbf{r}^*|\theta, \mathbf{r}) \\
= \; &\left(\frac{1}{Z_p Z_k}\right) \exp(-\tilde{H}(\theta, \mathbf{r}))\alpha_1(\theta^*, \mathbf{r}^*|\theta, \mathbf{r}) \frac{\exp(-E(\theta))}{\exp(-\tilde{E}(\theta))} \alpha_2(\theta^*, \mathbf{r}^*|\theta, \mathbf{r}) \\
= \; &\left(\frac{1}{Z_p Z_k}\right) \left[\exp(-\tilde{H}[\theta, \mathbf{r}]) \wedge \exp(-\tilde{H}[\theta^*, \mathbf{r}^*])\right] \left[\frac{\exp(-E(\theta))}{\exp(-\tilde{E}(\theta))} \wedge \frac{\exp(-E(\theta^*))}{\exp(-\tilde{E}(\theta^*))}\right],
\end{aligned}$$

which is invariant with respect to the parameter swap $(\theta, \mathbf{r}) \leftrightarrow (\theta^*, \mathbf{r}^*)$, as required.

Hence, the algorithm defines a transition kernel that leaves the joint posterior distribution,

$$p(\theta, \mathbf{r}|\mathbf{y}) = p(\theta|\mathbf{y})p(\mathbf{r}|\theta)$$

invariant, and thus the target distribution $p(\theta|\mathbf{y})$ invariant, as the auxiliary variables can be integrated out:

$$\int p(\theta, \mathbf{r}|\mathbf{y}) d\mathbf{r} = p(\theta|\mathbf{y}).$$

6.9.2 RMHMC with emulation and DA (DA-GPRMHMC)

For the RMHMC algorithm, the Hamiltonian function in surrogate space is:

$$\tilde{H}(\theta, \mathbf{r}) = \tilde{E}(\theta) + K(\mathbf{r}|\theta).$$

Hence,

$$p(\mathbf{r}|\theta) = \frac{\exp\left(-K(\mathbf{r}|\theta)\right)}{Z_k(\theta)},$$

where $K(\mathbf{r}|\theta) = \frac{1}{2}\mathbf{r}^T\mathbf{M}(\theta)^{-1}\mathbf{r}$, and $Z_k(\theta) = \exp\left(-\frac{1}{2}\log\left((2\pi)^d|\mathbf{M}(\theta)|\right)\right)$ is a normalising factor ensuring that the integral of the probability distribution $p(\mathbf{r}|\theta)$ is 1. Here, $\mathbf{M}(\theta)$ is a function of every θ along the leapfrog trajectory. All the other terms and the proof are similar to those for DA-GPHMC in Section 6.9.1.

6.9.3 HMC with emulation and no DA (noDA-GPHMC)

Define

$$p(\theta|\mathbf{y}) \;=\; \frac{\exp(-E(\theta))}{Z_p};\;\; \tilde{p}(\theta|\mathbf{y}) \;=\; \frac{\exp(-\tilde{E}(\theta))}{\tilde{Z}_p},$$

where $E(\theta) = -(\log p(\theta) + \log p(\mathbf{y}|\theta))$ is the true potential function and $\tilde{E}(\theta) = -(\log p(\theta) + \log \tilde{p}(\mathbf{y}|\theta))$ is the surrogate potential function, and $Z_p = \int_\theta p(\theta)p(\mathbf{y}|\theta)d\theta = p(\mathbf{y})$ is a normalising constant ensuring that the integral of the probability distribution $p(\theta|\mathbf{y})$ is 1. Also, $\tilde{Z}_p = \int_\theta p(\theta)\tilde{p}(\mathbf{y}|\theta)d\theta$ is a normalising constant.

The algorithms starts with a parameter vector θ and proceeds as follows:

1. Sample a vector of auxiliary 'momentum' variables $\mathbf{r}$ from $\mathscr{MVN}(\mathbf{0}, \mathbf{M})$ in a Gibbs step

   $$p(\mathbf{r}) \;=\; \frac{\exp(-K(\mathbf{r}))}{Z_k},$$

   where $K(\mathbf{r}) \;=\; \frac{1}{2}\mathbf{r}^T\mathbf{M}^{-1}\mathbf{r}$ is the kinetic energy ($\mathbf{M}$ - mass matrix), and
   $Z_k = \exp\left(-\frac{1}{2}\log\left((2\pi)^d|\mathbf{M}|\right)\right)$ is a normalising constant ensuring that the integral of the probability distribution $p(\mathbf{r})$ is 1. Note that $\tilde{p}(\mathbf{r}|\theta) = p(\mathbf{r}|\theta) = p(\mathbf{r})$ since it is expressed independently of $E(\theta)$.

2. Propose $(\theta^*, \mathbf{r}^*)$ by following a set of deterministic Hamiltonian dynamics steps in the

surrogate space, and the proposal probability ratio is:

$$\frac{q(\theta^*,\mathbf{r}^*|\theta,\mathbf{r})}{q(\theta,\mathbf{r}|\theta^*,\mathbf{r}^*)} = \frac{1}{1} = 1,$$

3. Accept the parameters $(\theta^*,\mathbf{r}^*)$ in a M-H step with the following probability:

$$\begin{aligned}\alpha(\theta^*,\mathbf{r}^*|\theta,\mathbf{r}) &= 1 \wedge \frac{p(\theta^*|\mathbf{y})p(\mathbf{r}^*|\theta^*)q(\theta,\mathbf{r}|\theta^*,\mathbf{r}^*)}{p(\theta|\mathbf{y})p(\mathbf{r}|\theta)q(\theta^*,\mathbf{r}^*|\theta,\mathbf{r})}\\ &= 1 \wedge \frac{\exp(-E(\theta^*))}{\exp(-E(\theta))}\frac{\exp(-K(\mathbf{r}^*))}{\exp(-K(\mathbf{r}))}.\end{aligned}$$

This algorithm preserves detailed balance with respect to $p(\theta|\mathbf{y})$.

PROOF

$$\begin{aligned}& p(\theta|\mathbf{y})p(\mathbf{r}|\theta)q(\theta^*,\mathbf{r}^*|\theta,\mathbf{r})\alpha(\theta^*,\mathbf{r}^*|\theta,\mathbf{r})\\ = \;& \frac{\exp(-E(\theta))}{Z_p}\frac{\exp(-K(\mathbf{r}))}{Z_k}\left[1 \wedge \frac{\exp(-E(\theta^*))\exp(-K(\mathbf{r}^*))}{\exp(-E(\theta))\exp(-K(\mathbf{r}))}\right]\\ = \;& \left(\frac{1}{Z_pZ_k}\right)[\exp(-H(\theta,\mathbf{r})) \wedge \exp(-H(\theta^*,\mathbf{r}^*))]\end{aligned}$$

which is invariant with respect to the parameter swap $(\theta,\mathbf{r}) \leftrightarrow (\theta^*,\mathbf{r}^*)$, as required.

Hence, the algorithm defines a transition kernel that leaves the joint posterior distribution,

$$p(\theta,\mathbf{r}|\mathbf{y}) = p(\theta|\mathbf{y})p(\mathbf{r}|\theta)$$

invariant, and thus the target distribution $p(\theta|\mathbf{y})$ invariant, as the auxiliary variables can be integrated out:

$$\int p(\theta,\mathbf{r}|\mathbf{y})d\mathbf{r} = p(\theta|\mathbf{y}).$$

6.9.4 RMHMC with emulation and no DA (noDA-GPRMHMC)

For the RMHMC algorithm, the Hamiltonian function in surrogate space is:

$$\tilde{H}(\theta,\mathbf{r}) = \tilde{E}(\theta) + K(\mathbf{r}|\theta).$$

Hence,

$$p(\mathbf{r}|\theta) = \frac{\exp\left(-K(\mathbf{r}|\theta)\right)}{Z_k(\theta)},$$

where $K(\mathbf{r}|\theta) = \frac{1}{2}\mathbf{r}^T\mathbf{M}(\theta)^{-1}\mathbf{r}$, and $Z_k(\theta) = \exp\left(-\frac{1}{2}\log\left((2\pi)^d|\mathbf{M}(\theta)|\right)\right)$ is a normalising factor ensuring that the integral of the probability distribution $p(\mathbf{r}|\theta)$ is 1. Here, $\mathbf{M}(\theta)$ is a function of every θ along the leapfrog trajectory. All the other terms and the proof are similar to those of noDA-GPHMC in Section 6.9.3.

6.9.5 LDMC with emulation and DA (DA-GPLDMC)

The Lagrangian Dynamical Monte Carlo (LDMC) algorithm uses 'velocity' instead of 'momentum' as the auxiliary variable, thus the sampling follows the Lagrangian dynamics rather than the Hamiltonian dynamics.

Define

$$p(\theta|\mathbf{y}) \;=\; \frac{\exp(-E(\theta))}{Z_p}; \;\; \tilde{p}(\theta|\mathbf{y}) \;=\; \frac{\exp(-\tilde{E}(\theta))}{\tilde{Z}_p},$$

where $E(\theta) = -(\log p(\theta) + \log p(\mathbf{y}|\theta))$ is the true potential function and $\tilde{E}(\theta) = -(\log p(\theta) + \log \tilde{p}(\mathbf{y}|\theta))$ is the surrogate potential function, and $Z_p = \int_\theta p(\theta)p(\mathbf{y}|\theta)d\theta = p(\mathbf{y})$ is a normalising constant ensuring that the integral of the probability distribution $p(\theta|\mathbf{y})$ is 1. Also, $\tilde{Z}_p = \int_\theta p(\theta)\tilde{p}(\mathbf{y}|\theta)d\theta$ is a normalising constant.

The algorithms starts with a parameter vector θ and proceeds as follows:

1. Sample a vector of auxiliary 'velocity' variables $\mathbf{v}$ from $\mathscr{MVN}(\mathbf{0}, \mathbf{M}(\theta)^{-1})$ in a Gibbs step, i.e.

   $$p(\mathbf{v}|\theta) \;=\; \frac{\exp(-K(\mathbf{v}|\theta))}{Z_k(\theta)},$$

   where $K(\mathbf{v}|\theta) \;=\; \frac{1}{2}\mathbf{v}^T\mathbf{M}(\theta)\mathbf{v}$ and $Z_k(\theta) = \exp\left(-\frac{1}{2}\log\left((2\pi)^d|\mathbf{M}(\theta)^{-1}|\right)\right)$ is a normalising factor ensuring that the integral of the probability distribution $p(\mathbf{v}|\theta)$ is 1. Note that $\tilde{p}(\mathbf{v}|\theta) = p(\mathbf{v}|\theta)$ since it is expressed independently of $E(\theta)$.

2. Propose $(\theta^*, \mathbf{v}^*)$ by following a set of Lagrangian dynamics steps, which are deterministic, but do not preserve volume in phase space, and accept the new configuration with the following acceptance probability (emulator-based):

   $$\alpha_1(\theta^*, \mathbf{v}^*|\theta, \mathbf{v}) \;=\; 1 \wedge \frac{\dfrac{\exp(-\tilde{H}[\theta^*, \mathbf{v}^*])}{Z_k(\theta^*)}}{\dfrac{\exp(-\tilde{H}[\theta, \mathbf{v}])}{Z_k(\theta)}} \left|\frac{\partial(\theta^*, \mathbf{v}^*)}{\partial(\theta, \mathbf{v})}\right|,$$

   where $\tilde{H}(\theta, \mathbf{v}) = \tilde{E}(\theta) + K(\mathbf{v}|\theta)$.

3. If the configuration has been accepted, accept the parameters with the second stage acceptance probability (simulator-based):

   $$\alpha_2(\theta^*, \mathbf{v}^*|\theta, \mathbf{v}) \;=\; 1 \wedge \frac{\exp(-E(\theta^*))}{\exp(-E(\theta))} \frac{\exp(-\tilde{E}(\theta))}{\exp(-\tilde{E}(\theta^*))}.$$

This algorithm preserves detailed balance with respect to $p(\theta|\mathbf{y})$.

PROOF

We need to show that

$$\alpha_2(\theta^*,\mathbf{v}^*|\theta,\mathbf{v})\alpha_1(\theta^*,\mathbf{v}^*|\theta,\mathbf{v})q(\theta^*,\mathbf{v}^*|\theta,\mathbf{v})p(\mathbf{v}|\theta)p(\theta|\mathbf{y})$$

is invariant with respect to the parameter exchange $(\theta,\mathbf{v}) \leftrightarrow (\theta^*,\mathbf{v}^*)$. The individual terms are as follows:

$$\begin{aligned}
p(\theta|\mathbf{y}) &= \frac{\exp(-E(\theta))}{Z_p}, \\
p(\mathbf{v}|\theta) &= \frac{\exp(-K(\mathbf{v}|\theta))}{Z_k(\theta)}, \\
\frac{q(\theta^*,\mathbf{v}^*|\theta,\mathbf{v})}{q(\theta,\mathbf{v}|\theta^*,\mathbf{v}^*)} &= \left|\frac{\partial(\theta,\mathbf{v})}{\partial(\theta^*,\mathbf{v}^*)}\right|, \text{ by transformation of pdfs of random variables} \\
\alpha_1(\theta^*,\mathbf{v}^*|\theta,\mathbf{v}) &= 1 \wedge \frac{\dfrac{\exp(-\tilde{H}[\theta^*,\mathbf{v}^*])}{Z_k(\theta^*)}}{\dfrac{\exp(-\tilde{H}[\theta,\mathbf{v}])}{Z_k(\theta)}} \left|\frac{\partial(\theta^*,\mathbf{v}^*)}{\partial(\theta,\mathbf{v})}\right|, \\
\alpha_2(\theta^*,\mathbf{v}^*|\theta,\mathbf{v}) &= 1 \wedge \frac{\exp(-E(\theta^*))}{\exp(-E(\theta))}\frac{\exp(-\tilde{E}(\theta))}{\exp(-\tilde{E}(\theta^*))},
\end{aligned}$$

Multiplying these individual terms together gives:

$$\begin{aligned}
& \left(\frac{1}{Z_p Z_k(\theta)}\right) q(\theta^*,\mathbf{v}^*|\theta,\mathbf{v})\alpha_1(\theta^*,\mathbf{v}^*|\theta,\mathbf{v})\alpha_2(\theta^*,\mathbf{v}^*|\theta,\mathbf{v})\frac{\exp(-E(\theta))}{\exp(-\tilde{E}(\theta))}\exp(-\tilde{E}(\theta)-K(\mathbf{v}|\theta)) \\
= & \left(\frac{1}{Z_p Z_k(\theta)}\right)\frac{\exp(-E(\theta))}{\exp(-\tilde{E}(\theta))}\alpha_2(\theta^*,\mathbf{v}^*|\theta,\mathbf{v})\exp(-\tilde{H}(\theta,\mathbf{v}))q(\theta^*,\mathbf{v}^*|\theta,\mathbf{v})\alpha_1(\theta^*,\mathbf{v}^*|\theta,\mathbf{v}) \\
= & \left(\frac{1}{Z_p Z_k(\theta)}\right)\left[\frac{\exp(-E(\theta))}{\exp(-\tilde{E}(\theta))} \wedge \frac{\exp(-E(\theta^*))}{\exp(-\tilde{E}(\theta^*))}\right]\exp(-\tilde{H}(\theta,\mathbf{v}))q(\theta^*,\mathbf{v}^*|\theta,\mathbf{v}) \\
& \left[1 \wedge \frac{\dfrac{\exp(-\tilde{H}[\theta^*,\mathbf{v}^*])}{Z_k(\theta^*)}}{\dfrac{\exp(-\tilde{H}[\theta,\mathbf{v}])}{Z_k(\theta)}}\left|\frac{\partial(\theta^*,\mathbf{v}^*)}{\partial(\theta,\mathbf{v})}\right|\right] \\
= & \left(\frac{1}{Z_p}\right)\left[\frac{\exp(-E(\theta))}{\exp(-\tilde{E}(\theta))} \wedge \frac{\exp(-E(\theta^*))}{\exp(-\tilde{E}(\theta^*))}\right] \\
& \left[\frac{\exp(-\tilde{H}(\theta,\mathbf{v}))}{Z_k(\theta)}q(\theta^*,\mathbf{v}^*|\theta,\mathbf{v}) \wedge \frac{\exp(-\tilde{H}[\theta^*,\mathbf{v}^*])}{Z_k(\theta^*)}q(\theta,\mathbf{v}|\theta^*,\mathbf{v}^*)\right],
\end{aligned}$$

which is invariant with respect to the parameter swap $(\theta,\mathbf{v}) \leftrightarrow (\theta^*,\mathbf{v}^*)$, as required.

Hence, the algorithm defines a transition kernel that leaves the joint posterior distribution,

$$p(\theta,\mathbf{v}|\mathbf{y}) = p(\theta|\mathbf{y})p(\mathbf{v}|\theta)$$

invariant, and thus the target distribution $p(\theta|\mathbf{y})$ invariant, as the auxiliary variables can be integrated out:

$$\int p(\theta, \mathbf{v}|\mathbf{y})d\mathbf{v} = p(\theta|\mathbf{y}).$$

6.9.6 LDMC with emulation and no DA (noDA-GPLDMC)

Define

$$p(\theta|\mathbf{y}) \;=\; \frac{\exp(-E(\theta))}{Z_p};\;\; \tilde{p}(\theta|\mathbf{y}) \;=\; \frac{\exp(-\tilde{E}(\theta))}{\tilde{Z}_p},$$

where $E(\theta) = -(\log p(\theta) + \log p(\mathbf{y}|\theta))$ is the true potential function and $\tilde{E}(\theta) = -(\log p(\theta) + \log \tilde{p}(\mathbf{y}|\theta))$ is the surrogate potential function, and $Z_p = \int_\theta p(\theta)p(\mathbf{y}|\theta)d\theta = p(\mathbf{y})$ is a normalising constant ensuring that the integral of the probability distribution $p(\theta|\mathbf{y})$ is 1. Also, $\tilde{Z}_p = \int_\theta p(\theta)\tilde{p}(\mathbf{y}|\theta)d\theta$ is a normalising constant.

The algorithms starts with a parameter vector θ and proceeds as follows:

1. Sample a vector of auxiliary 'velocity' variables $\mathbf{v}$ from $\mathscr{MVN}(\mathbf{0}, \mathbf{M}(\theta)^{-1})$ in a Gibbs step, i.e.

   $$p(\mathbf{v}|\theta) \;=\; \frac{\exp(-K(\mathbf{v}|\theta))}{Z_k(\theta)},$$

   where $K(\mathbf{v}|\theta) \;=\; \frac{1}{2}\mathbf{v}^T\mathbf{M}(\theta)\mathbf{v}$ and $Z_k(\theta) = \exp\left(-\frac{1}{2}\log\left((2\pi)^d|\mathbf{M}(\theta)^{-1}|\right)\right)$ is a normalising factor ensuring that the integral of the probability distribution $p(\mathbf{v}|\theta)$ is 1. Note that $\tilde{p}(\mathbf{v}|\theta) = p(\mathbf{v}|\theta)$ since it is expressed independently of $E(\theta)$.

2. Propose $(\theta^*, \mathbf{v}^*)$ by following a set of Lagrangian dynamics steps, which are deterministic, but do not preserve volume in phase space, and the proposal probability ratio is:

   $$\frac{q(\theta^*, \mathbf{v}^*|\theta, \mathbf{v})}{q(\theta, \mathbf{v}|\theta^*, \mathbf{v}^*)} \;=\; \left|\frac{\partial(\theta, \mathbf{v})}{\partial(\theta^*, \mathbf{v}^*)}\right|, \text{ by transformation of pdfs of random variables}$$

3. Accept the parameters $(\theta^*, \mathbf{v}^*)$ in a M-H step with the following probability:

   $$\begin{aligned}\alpha(\theta^*, \mathbf{v}^*|\theta, \mathbf{v}) &= 1 \wedge \frac{p(\theta^*|\mathbf{y})p(\mathbf{v}^*|\theta^*)q(\theta, \mathbf{v}|\theta^*, \mathbf{v}^*)}{p(\theta|\mathbf{y})p(\mathbf{v}|\theta)q(\theta^*, \mathbf{v}^*|\theta, \mathbf{v})} \\ &= 1 \wedge \frac{\exp(-E(\theta^*))}{\exp(-E(\theta))} \frac{\frac{\exp(-K(\mathbf{v}^*|\theta^*))}{Z_k(\theta^*)}}{\frac{\exp(-K(\mathbf{v}|\theta))}{Z_k(\theta)}} \left|\frac{\partial(\theta^*, \mathbf{v}^*)}{\partial(\theta, \mathbf{v})}\right|.\end{aligned}$$

PROOF

$$
\begin{aligned}
& p(\boldsymbol{\theta}|\mathbf{y})p(\mathbf{v}|\boldsymbol{\theta})q(\boldsymbol{\theta}^*,\mathbf{v}^*|\boldsymbol{\theta},\mathbf{v})\alpha(\boldsymbol{\theta}^*,\mathbf{v}^*|\boldsymbol{\theta},\mathbf{v}) \\
= \quad & \frac{\exp(-E(\boldsymbol{\theta}))}{Z_p}\frac{\exp(-K(\mathbf{v}|\boldsymbol{\theta}))}{Z_k(\boldsymbol{\theta})}q(\boldsymbol{\theta}^*,\mathbf{v}^*|\boldsymbol{\theta},\mathbf{v})\left[1 \wedge \frac{\exp(-E(\boldsymbol{\theta}^*))}{\exp(-E(\boldsymbol{\theta}))}\frac{\dfrac{\exp(-K(\mathbf{v}^*|\boldsymbol{\theta}^*))}{Z_k(\boldsymbol{\theta}^*)}}{\dfrac{\exp(-K(\mathbf{v}|\boldsymbol{\theta}))}{Z_k(\boldsymbol{\theta})}}\left|\frac{\partial(\boldsymbol{\theta}^*,\mathbf{v}^*)}{\partial(\boldsymbol{\theta},\mathbf{v})}\right|\right] \\
= \quad & \left(\frac{1}{Z_p}\right)\left[\frac{\exp(-H(\boldsymbol{\theta}^*,\mathbf{v}^*))}{Z_k(\boldsymbol{\theta}^*)}q(\boldsymbol{\theta},\mathbf{v}|\boldsymbol{\theta}^*,\mathbf{v}^*) \wedge \frac{\exp(-H(\boldsymbol{\theta},\mathbf{v}))}{Z_k(\boldsymbol{\theta})}q(\boldsymbol{\theta}^*,\mathbf{v}^*|\boldsymbol{\theta},\mathbf{v})\right],
\end{aligned}
$$

which is invariant with respect to the parameter swap $(\boldsymbol{\theta},\mathbf{v}) \leftrightarrow (\boldsymbol{\theta}^*,\mathbf{v}^*)$, as required. Hence, the algorithm defines a transition kernel that leaves the joint posterior distribution,

$$p(\boldsymbol{\theta},\mathbf{v}|\mathbf{y}) = p(\boldsymbol{\theta}|\mathbf{y})p(\mathbf{v}|\boldsymbol{\theta})$$

invariant, and thus the target distribution $p(\boldsymbol{\theta}|\mathbf{y})$ invariant, as the auxiliary variables can be integrated out:

$$\int p(\boldsymbol{\theta},\mathbf{v}|\mathbf{y})d\mathbf{v} = p(\boldsymbol{\theta}|\mathbf{y}).$$

6.9.7 NUTS with emulation and DA (DA-GPNUTS)

NUTS with emulation carries out Hamiltonian dynamics (with the leapfrog method) on the combined momentum and parameter space $(\boldsymbol{\theta},\mathbf{r})$ in emulated log posterior space. It collects configurations along the leapfrog trajectory to create two sets B and $C \subset B$. A balanced binary tree is constructed by repeatedly doubling the number of points visited along the trajectory. B contains all the (position, momentum) $(\boldsymbol{\theta},\mathbf{r})$ configurations collected throughout the trajectory, while C contains a subset of these parameters to which transitioning does not violate detailed balance, a condition checked using a real slice variable u. The algorithm effectively defines a transition kernel $T(\boldsymbol{\theta}^*,\mathbf{r}^*,C_{\text{new}}|\boldsymbol{\theta},\mathbf{r},C_{\text{old}})$ that leaves the joint posterior distribution,

$$p(\boldsymbol{\theta},\mathbf{r},u,B,C,C_{\text{old}}|\mathbf{y}) = p(\boldsymbol{\theta}|\mathbf{y})\tilde{p}(\mathbf{r},u,B,C,C_{\text{old}}|\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{y})\tilde{p}(\mathbf{r}|\boldsymbol{\theta})\tilde{p}(u,B,C|\boldsymbol{\theta},\mathbf{r})\tilde{p}(C_{\text{old}}|C)$$

invariant, and hence the target distribution $p(\boldsymbol{\theta}|\mathbf{y})$ invariant, as the auxiliary variables can be integrated out:

$$\int p(\boldsymbol{\theta},\mathbf{r},u,B,C,C_{\text{old}}|\mathbf{y})d\mathbf{r}dudBdCdC_{\text{old}} = p(\boldsymbol{\theta}|\mathbf{y}).$$

Note that $\tilde{p}(\mathbf{r}|\boldsymbol{\theta}) = p(\mathbf{r}|\boldsymbol{\theta}) = p(\mathbf{r})$ since it is expressed independently of $E(\boldsymbol{\theta})$.

If (efficient) NUTS is run on the surrogate posterior distribution from the emulator, detailed balance in emulated space holds (see proof in Section A.5), and the following is true:

$$\begin{aligned}T(\theta^*,\mathbf{r}^*,C_{\text{new}}|\theta,\mathbf{r},C_{\text{old}})\tilde{p}(\theta|\mathbf{y})\tilde{p}(\mathbf{r}|\theta)\tilde{p}(u|\theta,\mathbf{r})\tilde{p}(B,C|\theta,\mathbf{r},u,\varepsilon)\tilde{p}(C_{\text{old}}|C) = \\ T(\theta,\mathbf{r},C_{\text{old}}|\theta^*,\mathbf{r}^*,C_{\text{new}})\tilde{p}(\theta^*|\mathbf{y})\tilde{p}(\mathbf{r}^*|\theta^*)\tilde{p}(u|\theta^*,\mathbf{r}^*)\tilde{p}(B,C|\theta^*,\mathbf{r}^*,u,\varepsilon)\tilde{p}(C_{\text{new}}|C),\end{aligned} \tag{6.33}$$

Hence, in emulated posterior distribution space, detailed balance is satisfied. Next, detailed balance is proven to hold for the original (exact) posterior distribution.

Denote by $\theta = \theta^0, \quad \mathbf{r} = \mathbf{r}^0, \quad \theta^* = \theta^J, \quad \mathbf{r}^* = \mathbf{r}^J,$

$C_{\text{new}} = C^J_{\text{new}}; \quad \cap_j C^j_{\text{new}} = \emptyset; \quad C^j_{\text{old}} \cup C^j_{\text{new}} = C^j; \quad C = C^J; \quad C^j_{\text{old}} \cup C^j_{\text{new}} = C^{j+1}_{\text{old}}; \quad C_{\text{old}} = C^J_{\text{old}},$

where J: tree height.

If θ^* is accepted by NUTS at the emulator level with first-stage acceptance probability:

$$\alpha_1(\theta^*,\mathbf{r}^*,C_{\text{new}}|\theta,\mathbf{r},C_{\text{old}}) = \prod_{j=1}^{J} \alpha_{1j}(\theta^j,\mathbf{r}^j,C^j_{\text{new}}|\theta^{j-1},\mathbf{r}^{j-1},C^j_{\text{old}}) = \prod_{j=1}^{J}\left[1 \wedge \frac{|C^j_{\text{new}}|}{|C^j_{\text{old}}|}\right],$$

the second-stage delayed acceptance step is included, which accepts the final configuration $(\theta^*,\mathbf{r}^*) = (\theta^J,\mathbf{r}^J)$ with the following probability:

$$\begin{aligned}\alpha_2(\theta^*|\theta) &= 1 \wedge \frac{p(\theta^*|\mathbf{y})}{p(\theta|\mathbf{y})}\frac{\tilde{p}(\theta|\mathbf{y})}{\tilde{p}(\theta^*|\mathbf{y})} \\ &= 1 \wedge \frac{\exp(-E(\theta^*))}{\exp(-E(\theta))}\frac{\exp(-\tilde{E}(\theta))}{\exp(-\tilde{E}(\theta^*))}.\end{aligned}$$

The aim is to show that the combined transition kernel

$$\alpha_2(\theta^*|\theta)T(\theta^*,\mathbf{r}^*,C_{\text{new}}|\theta,\mathbf{r},C_{\text{old}})$$

leaves the distribution $p(\theta,\mathbf{r},u,B,C|\mathbf{y})$ invariant.

PROOF

$$\begin{aligned}& \alpha_2(\theta^*|\theta)T(\theta^*,\mathbf{r}^*,C_{\text{new}}|\theta,\mathbf{r},C_{\text{old}})p(\theta,\mathbf{r},u,B,C,C_{\text{old}}|\mathbf{y},\varepsilon) \\ = \; & \alpha_2(\theta^*|\theta)\frac{\exp(-E(\theta))}{\exp(-\tilde{E}(\theta))}T(\theta^*,\mathbf{r}^*,C_{\text{new}}|\theta,\mathbf{r},C_{\text{old}})\tilde{p}(\theta|\mathbf{y})\tilde{p}(\mathbf{r}|\theta)\tilde{p}(u|\theta,\mathbf{r}) \\ & \tilde{p}(B,C|\theta,\mathbf{r},u,\varepsilon)\tilde{p}(C_{\text{old}}|C) \\ = \; & \left(\frac{\exp(-E(\theta))}{\exp(-\tilde{E}(\theta))} \wedge \frac{\exp(-E(\theta^*))}{\exp(-\tilde{E}(\theta^*))}\right)T(\theta^*,\mathbf{r}^*,C_{\text{new}}|\theta,\mathbf{r},C_{\text{old}})\tilde{p}(\theta|\mathbf{y})\tilde{p}(\mathbf{r}|\theta) \\ & \tilde{p}(u|\theta,\mathbf{r})\tilde{p}(B,C|\theta,\mathbf{r},u,\varepsilon)\tilde{p}(C_{\text{old}}|C)\end{aligned}$$

The first factor is invariant with respect to a swap of the arguments $\theta \leftrightarrow \theta^*$. Equation (6.33) implies that the last six factors are invariant with respect to a swap of the arguments $(\theta,\mathbf{r},C_{\text{old}}) \leftrightarrow$

$(\theta^*, \mathbf{r}^*, C_{\text{new}})$ via a sequence of intermediate state swaps $(\theta^{j-1}, \mathbf{r}^{j-1}, C^j_{\text{old}}) \leftrightarrow (\theta^j, \mathbf{r}^j, C^j_{\text{new}})$. Together this implies that $\alpha_2(\theta^*|\theta)T(\theta^*, \mathbf{r}^*, C_{\text{new}}|\theta, \mathbf{r}, C_{\text{old}})p(\theta, \mathbf{r}, u, B, C, C_{\text{old}}|\mathbf{y}, \varepsilon)$ is invariant with respect to a swap of the arguments $(\theta, \mathbf{r}, C_{\text{old}}) \leftrightarrow (\theta^*, \mathbf{r}^*, C_{\text{new}})$. Hence detailed balance holds:

$$\begin{aligned} & \alpha_2(\theta^*|\theta)T(\theta^*, \mathbf{r}^*, C_{\text{new}}|\theta, \mathbf{r}, C_{\text{old}})p(\theta, \mathbf{r}, u, B, C, C_{\text{old}}|\mathbf{y}, \varepsilon) \\ = \; & \alpha_2(\theta|\theta^*)T(\theta, \mathbf{r}, C_{\text{old}}|\theta^*, \mathbf{r}^*, C_{\text{new}})p(\theta^*, \mathbf{r}^*, u, B, C, C_{\text{new}}|\mathbf{y}, \varepsilon) \end{aligned}$$

and the combined transition kernel leaves the joint distribution $p(\theta, \mathbf{r}, u, B, C|\mathbf{y}, \varepsilon)$ invariant, as required.

6.9.8 NUTS with emulation and no DA (noDA-GPNUTS)

NUTS with emulation carries out Hamiltonian dynamics (with the leapfrog method) on the combined momentum and parameter space $(\theta, \mathbf{r})$ in emulated log posterior space. It collects configurations along the leapfrog trajectory to create two sets B and $C \subset B$. A balanced binary tree is constructed by repeatedly doubling the number of points visited along the leapfrog trajectory. B contains all the (position, momentum) $(\theta, \mathbf{r})$ configurations collected throughout the trajectory, while C contains a subset of these parameters to which transitioning does not violate detailed balance, a condition checked using a real slice variable u. The algorithm effectively defines a transition kernel $T(\theta^*, \mathbf{r}^*|\theta, \mathbf{r}, C)$ that leaves the joint posterior distribution,

$$p(\theta, \mathbf{r}, u, B, C|\mathbf{y}) = p(\theta|\mathbf{y})\tilde{p}(\mathbf{r}, u, B, C|\theta) = p(\theta|\mathbf{y})\tilde{p}(\mathbf{r}|\theta)\tilde{p}(u, B, C|\theta, \mathbf{r})$$

invariant, and hence the target distribution $p(\theta|\mathbf{y})$ invariant, as the auxiliary variables can be integrated out:

$$\int p(\theta, \mathbf{r}, u, B, C|\mathbf{y}) d\mathbf{r} du dB dC = p(\theta|\mathbf{y}).$$

Note that $\tilde{p}(\mathbf{r}|\theta) = p(\mathbf{r}|\theta) = p(\mathbf{r})$ since it is expressed independently of $E(\theta)$.

Define

$$p(\theta|\mathbf{y}) \; = \; \frac{\exp(-E(\theta))}{Z_p}; \; \tilde{p}(\theta|\mathbf{y}) \; = \; \frac{\exp(-\tilde{E}(\theta))}{\tilde{Z}_p},$$

where $E(\theta) = -(\log p(\theta) + \log p(\mathbf{y}|\theta))$ is the true potential function and $\tilde{E}(\theta) = -(\log p(\theta) + \log \tilde{p}(\mathbf{y}|\theta))$ is the surrogate potential function, and $Z_p = \int_\theta p(\theta)p(\mathbf{y}|\theta)d\theta = p(\mathbf{y})$ is a normalising constant ensuring that the integral of the probability distribution $p(\theta|\mathbf{y})$ is 1. Also, $\tilde{Z}_p = \int_\theta p(\theta)\tilde{p}(\mathbf{y}|\theta)d\theta$ is a normalising constant.

The naive NUTS algorithm is first set up (which will help with the set-up of the efficient NUTS), starting with a parameter vector θ:

1. Sample a vector of auxiliary ‘momentum’ variables $\mathbf{r}$ from $\mathscr{MVN}(\mathbf{0}, \mathbf{M})$ in a Gibbs step

   $$p(\mathbf{r}) \;=\; \frac{\exp(-K(\mathbf{r}))}{Z_k},$$

   where $K(\mathbf{r}) \;=\; \frac{1}{2}\mathbf{r}^T\mathbf{M}^{-1}\mathbf{r}$ is the kinetic energy ($\mathbf{M}$ - mass matrix) and $Z_k = \exp\left(-\frac{1}{2}\log\left((2\pi)^d|\mathbf{M}|\right)\right)$ is a normalising constant ensuring that the integral of the probability distribution $p(\mathbf{r})$ is 1.

2. Sample the slice variable $u|\theta,\mathbf{r} \sim \text{Uniform}(0, \exp(-\tilde{E}(\theta) - K(\mathbf{r})))$ in a Gibbs step.

3. Sample B, C from their conditional distribution $\tilde{p}(B,C|\theta,\mathbf{r},u,\varepsilon)$ in a Gibbs step, such that

   $$\tilde{p}(B,C|\theta,\mathbf{r},u,\varepsilon) \propto \mathbb{1}((\theta,\mathbf{r}) \in C),$$
   $$\tilde{p}(B,C|\theta,\mathbf{r},u,\varepsilon) = \tilde{p}(B,C|\theta^*,\mathbf{r}^*,u,\varepsilon).$$

4. Move from $(\theta,\mathbf{r})$ to $(\theta^*,\mathbf{r}^*)$ with uniform probability over C:

   $$q(\theta^*,\mathbf{r}^*|\theta,\mathbf{r},C) = \frac{\mathbb{1}((\theta^*,\mathbf{r}^*) \in C)}{|C|}.$$

   Next, by noting that

   $$\begin{aligned} p(\theta,\mathbf{r},B,C,u|\mathbf{y},\varepsilon) &= p(\theta,\mathbf{r}|\mathbf{y})\tilde{p}(u|\theta,\mathbf{r})\tilde{p}(B,C|\theta,\mathbf{r},u,\varepsilon) \\ &= p(\theta|\mathbf{y})p(\mathbf{r}|\theta)\tilde{p}(u|\theta,\mathbf{r})\tilde{p}(B,C|\theta,\mathbf{r},u,\varepsilon) \\ &= \frac{\exp(-E(\theta))}{Z_p}\frac{\exp(-K(\mathbf{r}))}{Z_k}\frac{\mathbb{1}(u \le \exp(-\tilde{E}(\theta) - K(\mathbf{r})))}{\exp(-\tilde{E}(\theta) - K(\mathbf{r}))}\tilde{p}(B,C|\theta,\mathbf{r},u,\varepsilon) \\ &= \left(\frac{1}{Z_pZ_k}\right)\frac{\exp(-E(\theta))}{\exp(-\tilde{E}(\theta))}\tilde{p}(B,C|\theta,\mathbf{r},u,\varepsilon), \text{ if } \mathbb{1}(.) = 1, \end{aligned}$$

   accept the parameters $(\theta^*,\mathbf{r}^*)$ in a M-H step with the following probability:

   $$\alpha(\theta^*,\mathbf{r}^*|\theta,\mathbf{r}) = 1 \wedge \frac{\exp(-E(\theta^*))}{\exp(-\tilde{E}(\theta^*))}\frac{\exp(-\tilde{E}(\theta))}{\exp(-E(\theta))}$$

This algorithm preserves detailed balance with respect to $p(\theta|\mathbf{y})$.

PROOF

$$
\begin{aligned}
& p(\theta,\mathbf{r}|\mathbf{y})\tilde{p}(u|\theta,\mathbf{r})\tilde{p}(B,C|\theta,\mathbf{r},u,\varepsilon)q(\theta^*,\mathbf{r}^*,|\theta,\mathbf{r},C)\alpha(\theta^*,\mathbf{r}^*|\theta,\mathbf{r},C) \\
= & \left(\frac{1}{Z_pZ_k}\right)\frac{\exp(-E(\theta))}{\exp(-\tilde{E}(\theta)}\tilde{p}(B,C|\theta,\mathbf{r},u,\varepsilon)\frac{\mathbb{1}((\theta^*,\mathbf{r}^*)\in C)}{|C|}\left[1\wedge\frac{\exp(-E(\theta^*))}{\exp(-\tilde{E}(\theta^*))}\frac{\exp(-\tilde{E}(\theta))}{\exp(-E(\theta))}\right] \\
= & \left(\frac{1}{Z_pZ_k}\right)\tilde{p}(B,C|\theta,\mathbf{r},u,\varepsilon)\frac{\mathbb{1}((\theta^*,\mathbf{r}^*)\in C)}{|C|}\left[\frac{\exp(-E(\theta))}{\exp(-\tilde{E}(\theta))}\wedge\frac{\exp(-E(\theta^*))}{\exp(-\tilde{E}(\theta^*))}\right] \\
= & \left(\frac{1}{Z_pZ_k}\right)\tilde{p}(B,C|\theta^*,\mathbf{r}^*,u,\varepsilon)\frac{\mathbb{1}((\theta,\mathbf{r})\in C)}{|C|}\left[\frac{\exp(-E(\theta^*))}{\exp(-\tilde{E}(\theta^*))}\wedge\frac{\exp(-E(\theta))}{\exp(-\tilde{E}(\theta))}.\right]
\end{aligned}
$$

Hence, the algorithm is invariant with respect to the parameter swap $(\theta,\mathbf{r}) \leftrightarrow (\theta^*,\mathbf{r}^*)$, as required.

Next the efficient NUTS algorithm is set up, starting with a parameter vector θ:

1. Sample a vector of auxiliary ‘momentum’ variables $\mathbf{r}$ from $\mathscr{MVN}(\mathbf{0},\mathbf{M})$ in a Gibbs step

   $$p(\mathbf{r}) \;=\; \frac{\exp(-K(\mathbf{r}))}{Z_k},$$

   where $K(\mathbf{r}) \;=\; \frac{1}{2}\mathbf{r}^T\mathbf{M}^{-1}\mathbf{r}$ is the kinetic energy ($\mathbf{M}$ - mass matrix) and
   $Z_k = \exp\left(-\frac{1}{2}\log\left((2\pi)^d|\mathbf{M}|\right)\right)$ is a normalising constant ensuring that the integral of the probability distribution $p(\mathbf{r})$ is 1.

2. Sample the slice variable $u|\theta,\mathbf{r} \sim \text{Uniform}(0,\exp(-\tilde{E}(\theta)-K(\mathbf{r})))$ in a Gibbs step.

3. Sample B,C from their conditional distribution $\tilde{p}(B,C|\theta,\mathbf{r},u,\varepsilon)$ in a Gibbs step, such that

   $$
   \begin{aligned}
   \tilde{p}(B,C|\theta,\mathbf{r},u,\varepsilon) &\propto \mathbb{1}((\theta,\mathbf{r})\in C), \\
   \tilde{p}(B,C|\theta,\mathbf{r},u,\varepsilon) &= \tilde{p}(B,C|\theta^*,\mathbf{r}^*,u,\varepsilon).
   \end{aligned}
   $$

4. Deterministically divide the set C into C_{new} and C_{old}, such that $C_{\text{old}}\cup C_{\text{new}} = C, C_{\text{old}}\cap C_{\text{new}} = \emptyset$, and sample C_{old} and C_{new} with probability:

   $$p(C_{\text{new}}|C) = p(C_{\text{old}}|C) = 1$$

5. Propose to move from $(\theta^{j-1},\mathbf{r}^{j-1}) \in C^j_{\text{old}}$ to $(\theta^j,\mathbf{r}^j) \in C^j_{\text{new}}$ by following a set of deterministic Hamiltonian dynamics steps in the surrogate space. Switch from C^j_{old} to C^j_{new} with probability $\mathbb{1}(C^{'j}_{\text{old}} = C^j_{\text{new}})\mathbb{1}(C^{'j}_{\text{new}} = C^j_{\text{old}})$ and choose an element $(\theta^j,\mathbf{r}^j) \in C^j_{\text{new}}$ with uniform probability $\dfrac{\mathbb{1}((\theta^j,\mathbf{r}^j)\in C^j_{\text{new}})}{|C^j_{\text{new}}|}$, where $|C^j_{\text{new}}|$ defines the cardinality of the

set C^j_{new}:

$$\begin{aligned} q(\theta^j,\mathbf{r}^j,C^j_{\text{new}}|\theta^{j-1},\mathbf{r}^{j-1},C^j_{\text{old}}) &= \mathbb{1}(C^{j'}_{\text{old}} = C^j_{\text{new}})\mathbb{1}(C^{j'}_{\text{new}} = C^j_{\text{old}})\frac{\mathbb{1}((\theta^j,\mathbf{r}^j) \in C^j_{\text{new}})}{|C^j_{\text{new}}|} \\ &= \frac{\mathbb{1}((\theta^j,\mathbf{r}^j) \in C^j_{\text{new}})}{|C^j_{\text{new}}|}, \end{aligned}$$

assuming $\mathbb{1}(C^{j'}_{\text{old}} = C^j_{\text{new}})\mathbb{1}(C^{j'}_{\text{new}} = C^j_{\text{old}}) = 1$

6. Accept the parameters $(\theta^j,\mathbf{r}^j,C^j_{\text{new}})$ in a M-H step with the following probability:

$$\begin{aligned} \alpha(\theta^j,\mathbf{r}^j,C^j_{\text{new}}|\theta^{j-1},\mathbf{r}^{j-1},C^j_{\text{old}}) = 1 \wedge \frac{\exp(-E(\theta^j))}{\exp(-\tilde{E}(\theta^j))}\frac{\exp(-\tilde{E}(\theta^{j-1}))}{\exp(-E(\theta^{j-1}))}\frac{|C^j_{\text{new}}|}{|C^j_{\text{old}}|} \\ \frac{\mathbb{1}((\theta^{j-1},\mathbf{r}^{j-1}) \in C^j_{\text{old}})}{\mathbb{1}((\theta^j,\mathbf{r}^j) \in C^j_{\text{new}})}, \end{aligned}$$

Propose to move from $(\theta,\mathbf{r}) \in C_{\text{old}}$ to $(\theta^*,\mathbf{r}^*) \in C_{\text{new}}$ by following a set of deterministic Hamiltonian dynamics steps in the surrogate space via a sequence of intermediate states. Thus, the transition kernel T is repeatedly applied after every tree doubling, i.e. steps (5) and (6) are repeated a number of times equal to the tree height:

$$\begin{aligned} T(\theta^*,\mathbf{r}^*,C_{\text{new}}|\theta,\mathbf{r},C_{\text{old}}) &= \prod_{j=1}^{J} q(\theta^j,\mathbf{r}^j,C^j_{\text{new}}|\theta^{j-1},\mathbf{r}^{j-1},C^j_{\text{old}})\alpha(\theta^j,\mathbf{r}^j,C^j_{\text{new}}|\theta^{j-1},\mathbf{r}^{j-1},C^j_{\text{old}}) \\ &= \prod_{j=1}^{J}\left\{\frac{\exp(-\tilde{E}(\theta^{j-1}))}{\exp(-E(\theta^{j-1}))}\left[\frac{\exp(-E(\theta^{j-1}))}{\exp(-\tilde{E}(\theta^{j-1}))}\frac{\mathbb{1}((\theta^j,\mathbf{r}^j) \in C^j_{\text{new}})}{|C^j_{\text{new}}|}\wedge \right.\right. \\ &\left.\left. \frac{\exp(-E(\theta^j))}{\exp(-\tilde{E}(\theta^j))}\frac{\mathbb{1}((\theta^{j-1},\mathbf{r}^{j-1}) \in C^j_{\text{old}})}{|C^j_{\text{old}}|}\right]\right\}, \end{aligned} \tag{6.34}$$

where:
$\theta = \theta^0, \mathbf{r} = \mathbf{r}^0$: the initial values, J: the number of tree doublings (tree height) before reaching the final proposed values $\theta^* = \theta^J, \mathbf{r}^* = \mathbf{r}^J$.

C_{new} contains all the points visited at the last doubling, i.e. $C_{\text{new}} = C^J_{\text{new}}$ and

$$\cap_j C^j_{\text{new}} = \emptyset; \quad C^j_{\text{old}} \cup C^j_{\text{new}} = C^j; \quad C = C^J; \quad C^j_{\text{old}} \cup C^j_{\text{new}} = C^{j+1}_{\text{old}}; \quad C_{\text{old}} = C^J_{\text{old}}.$$

One can similarly express

$$\begin{aligned}
T(\theta,\mathbf{r},C_{\text{old}}|\theta^*,\mathbf{r}^*,C_{\text{new}}) &= \prod_{j=1}^{J} q(\theta^{j-1},\mathbf{r}^{j-1},C^{j}_{\text{old}}|\theta^{j},\mathbf{r}^{j},C^{j}_{\text{new}})\alpha(\theta^{j-1},\mathbf{r}^{j-1},C^{j}_{\text{old}}|\theta^{j},\mathbf{r}^{j},C^{j}_{\text{new}}) \\
&= \prod_{j=1}^{J} \left\{ \frac{\exp(-\tilde{E}(\theta^j))}{\exp(-E(\theta^j))} \left[\frac{\exp(-E(\theta^j))}{\exp(-\tilde{E}(\theta^j))} \frac{\mathbb{1}((\theta^{j-1},\mathbf{r}^{j-1}) \in C^{j}_{\text{old}})}{|C^{j}_{\text{old}}|} \wedge \right.\right. \\
&\left.\left. \frac{\exp(-E(\theta^{j-1}))}{\exp(-\tilde{E}(\theta^{j-1}))} \frac{\mathbb{1}((\theta^{j},\mathbf{r}^{j}) \in C^{j}_{\text{new}})}{|C^{j}_{\text{new}}|} \right] \right\}
\end{aligned}$$

This algorithm preserves detailed balance with respect to $p(\theta|\mathbf{y})$.

PROOF

$$\begin{aligned}
&p(\theta,\mathbf{r}|\mathbf{y})\tilde{p}(u|\theta,\mathbf{r})\tilde{p}(B,C|\theta,\mathbf{r},u,\varepsilon)p(C_{\text{old}}|C)q(\theta^*,\mathbf{r}^*,C_{\text{new}}|\theta,\mathbf{r},C_{\text{old}}) \\
&\alpha(\theta^*,\mathbf{r}^*,C_{\text{new}}|\theta,\mathbf{r},C_{\text{old}}) \\
&= \left(\frac{1}{Z_p Z_k}\right) \frac{\exp(-E(\theta^0))}{\exp(-\tilde{E}(\theta^0))} \tilde{p}(B,C|\theta^0,\mathbf{r}^0,u,\varepsilon) 1 \\
&\prod_{j=1}^{J} \left\{ \frac{\exp(-\tilde{E}(\theta^{j-1}))}{\exp(-E(\theta^{j-1}))} \left[\frac{\exp(-E(\theta^{j-1}))}{\exp(-\tilde{E}(\theta^{j-1}))} \frac{\mathbb{1}((\theta^{j},\mathbf{r}^{j}) \in C^{j}_{\text{new}})}{|C^{j}_{\text{new}}|} \wedge \frac{\exp(-E(\theta^j))}{\exp(-\tilde{E}(\theta^j))} \frac{\mathbb{1}((\theta^{j-1},\mathbf{r}^{j-1}) \in C^{j}_{\text{old}})}{|C^{j}_{\text{old}}|} \right] \right\} \\
&= \left(\frac{1}{Z_p Z_k}\right) \tilde{p}(B,C|\theta^0,\mathbf{r}^0,u,\varepsilon) \left[\frac{\exp(-E(\theta^0))}{\exp(-\tilde{E}(\theta^0))} \frac{\mathbb{1}((\theta^{1},\mathbf{r}^{1}) \in C^{1}_{\text{new}})}{|C^{1}_{\text{new}}|} \wedge \frac{\exp(-E(\theta^1))}{\exp(-\tilde{E}(\theta^1))} \frac{\mathbb{1}((\theta^{0},\mathbf{r}^{0}) \in C^{1}_{\text{old}})}{|C^{1}_{\text{old}}|} \right] \\
&\prod_{j=2}^{J} \left\{ \frac{\exp(-\tilde{E}(\theta^{j-1}))}{\exp(-E(\theta^{j-1}))} \left[\frac{\exp(-E(\theta^{j-1}))}{\exp(-\tilde{E}(\theta^{j-1}))} \frac{\mathbb{1}((\theta^{j},\mathbf{r}^{j}) \in C^{j}_{\text{new}})}{|C^{j}_{\text{new}}|} \wedge \frac{\exp(-E(\theta^j))}{\exp(-\tilde{E}(\theta^j))} \frac{\mathbb{1}((\theta^{j-1},\mathbf{r}^{j-1}) \in C^{j}_{\text{old}})}{|C^{j}_{\text{old}}|} \right] \right\}
\end{aligned}$$

Moreover,

$$\begin{aligned}
&p(\theta^*,\mathbf{r}^*|\mathbf{y})\tilde{p}(u|\theta^*,\mathbf{r}^*)\tilde{p}(B,C|\theta^*,\mathbf{r}^*,u,\varepsilon)p(C_{\text{old}}|C)q(\theta,\mathbf{r},C_{\text{old}}|\theta^*,\mathbf{r}^*,C_{\text{new}})\alpha(\theta,\mathbf{r},C_{\text{old}}|\theta^*,\mathbf{r}^*,C_{\text{new}}) \\
&= \left(\frac{1}{Z_pZ_k}\right)\frac{\exp(-E(\theta^J))}{\exp(-\tilde{E}(\theta^J))}\tilde{p}(B,C|\theta^J,\mathbf{r}^J,u,\varepsilon)1 \\
&\prod_{j=1}^{J}\left\{\frac{\exp(-\tilde{E}(\theta^j))}{\exp(-E(\theta^j))}\left[\frac{\exp(-E(\theta^j))}{\exp(-\tilde{E}(\theta^j))}\frac{\mathbb{1}((\theta^{j-1},\mathbf{r}^{j-1})\in C^j_{\text{old}})}{|C^j_{\text{old}}|}\wedge\frac{\exp(-E(\theta^{j-1}))}{\exp(-\tilde{E}(\theta^{j-1}))}\frac{\mathbb{1}((\theta^j,\mathbf{r}^j)\in C^j_{\text{new}})}{|C^j_{\text{new}}|}\right]\right\} \\
&= \left(\frac{1}{Z_pZ_k}\right) \\
&\tilde{p}(B,C|\theta^J,\mathbf{r}^J,u,\varepsilon)\left[\frac{\exp(-E(\theta^J))}{\exp(-\tilde{E}(\theta^J))}\frac{\mathbb{1}((\theta^{J-1},\mathbf{r}^{J-1})\in C^J_{\text{old}})}{|C^J_{\text{old}}|}\wedge\frac{\exp(-E(\theta^{J-1}))}{\exp(-\tilde{E}(\theta^{J-1}))}\frac{\mathbb{1}((\theta^J,\mathbf{r}^J)\in C^J_{\text{new}})}{|C^J_{\text{new}}|}\right] \\
&\prod_{j=1}^{J-1}\left\{\frac{\exp(-\tilde{E}(\theta^j))}{\exp(-E(\theta^j))}\left[\frac{\exp(-E(\theta^j))}{\exp(-\tilde{E}(\theta^j))}\frac{\mathbb{1}((\theta^{j-1},\mathbf{r}^{j-1})\in C^j_{\text{old}})}{|C^j_{\text{old}}|}\wedge\frac{\exp(-E(\theta^{j-1}))}{\exp(-\tilde{E}(\theta^{j-1}))}\frac{\mathbb{1}((\theta^j,\mathbf{r}^j)\in C^j_{\text{new}})}{|C^j_{\text{new}}|}\right]\right\} \\
&= \left(\frac{1}{Z_pZ_k}\right)\tilde{p}(B,C|\theta^J,\mathbf{r}^J,u,\varepsilon)\frac{\exp(-\tilde{E}(\theta^1))}{\exp(-E(\theta^1))}\left[\frac{\exp(-E(\theta^1))}{\exp(-\tilde{E}(\theta^1))}\frac{\mathbb{1}((\theta^0,\mathbf{r}^0)\in C^1_{\text{old}})}{|C^1_{\text{old}}|}\wedge\right. \\
&\left.\frac{\exp(-E(\theta^0))}{\exp(-\tilde{E}(\theta^0))}\frac{\mathbb{1}((\theta^1,\mathbf{r}^1)\in C^1_{\text{new}})}{|C^1_{\text{new}}|}\right]\frac{\exp(-E(\theta^J))}{\exp(-\tilde{E}(\theta^J))}\prod_{j=2}^{J}\left\{\frac{\exp(-\tilde{E}(\theta^j))}{\exp(-E(\theta^j))}\right. \\
&\left.\left[\frac{\exp(-E(\theta^j))}{\exp(-\tilde{E}(\theta^j))}\frac{\mathbb{1}((\theta^{j-1},\mathbf{r}^{j-1})\in C^j_{\text{old}})}{|C^j_{\text{old}}|}\wedge\frac{\exp(-E(\theta^{j-1}))}{\exp(-\tilde{E}(\theta^{j-1}))}\frac{\mathbb{1}((\theta^j,\mathbf{r}^j)\in C^j_{\text{new}})}{|C^j_{\text{new}}|}\right]\right\} \\
&= \left(\frac{1}{Z_pZ_k}\right)\tilde{p}(B,C|\theta^J,\mathbf{r}^J,u,\varepsilon)\left[\frac{\exp(-E(\theta^1))}{\exp(-\tilde{E}(\theta^1))}\frac{\mathbb{1}((\theta^0,\mathbf{r}^0)\in C^1_{\text{old}})}{|C^1_{\text{old}}|}\wedge\frac{\exp(-E(\theta^0))}{\exp(-\tilde{E}(\theta^0))}\frac{\mathbb{1}((\theta^1,\mathbf{r}^1)\in C^1_{\text{new}})}{|C^1_{\text{new}}|}\right] \\
&\prod_{j=2}^{J}\left\{\frac{\exp(-\tilde{E}(\theta^{j-1}))}{\exp(-E(\theta^{j-1}))}\left[\frac{\exp(-E(\theta^j))}{\exp(-\tilde{E}(\theta^j))}\frac{\mathbb{1}((\theta^{j-1},\mathbf{r}^{j-1})\in C^j_{\text{old}})}{|C^j_{\text{old}}|}\wedge\frac{\exp(-E(\theta^{j-1}))}{\exp(-\tilde{E}(\theta^{j-1}))}\frac{\mathbb{1}((\theta^j,\mathbf{r}^j)\in C^j_{\text{new}})}{|C^j_{\text{new}}|}\right]\right\}
\end{aligned}$$

Since $\tilde{p}(B,C|\theta^0,\mathbf{r}^0,u,\varepsilon)=\tilde{p}(B,C|\theta^J,\mathbf{r}^J,u,\varepsilon)$, it is clear that

$$\begin{aligned}
&p(\theta,\mathbf{r}|\mathbf{y})\tilde{p}(u|\theta,\mathbf{r})\tilde{p}(B,C|\theta,\mathbf{r},u,\varepsilon)p(C_{\text{old}}|C)q(\theta^*,\mathbf{r}^*,C_{\text{new}}|\theta,\mathbf{r},C_{\text{old}}) \\
&\qquad\alpha(\theta^*,\mathbf{r}^*,C_{\text{new}}|\theta,\mathbf{r},C_{\text{old}}) \\
&= p(\theta^*,\mathbf{r}^*|\mathbf{y})\tilde{p}(u|\theta^*,\mathbf{r}^*)\tilde{p}(B,C|\theta^*,\mathbf{r}^*,u,\varepsilon)p(C_{\text{old}}|C)q(\theta,\mathbf{r},C_{\text{old}}|\theta^*,\mathbf{r}^*,C_{\text{new}}) \\
&\qquad\alpha(\theta,\mathbf{r},C_{\text{old}}|\theta^*,\mathbf{r}^*,C_{\text{new}})
\end{aligned}$$

Hence, the algorithm is invariant with respect to the parameter swap $(\theta,\mathbf{r},C_{\text{old}})\leftrightarrow(\theta^*,\mathbf{r}^*,C_{\text{new}})$ via a sequence of intermediate state swaps $(\theta^{j-1},\mathbf{r}^{j-1},C^j_{\text{old}})\leftrightarrow(\theta^j,\mathbf{r}^j,C^j_{\text{new}})$. So the joint distribution $p(\theta,\mathbf{r},u,B,C|\mathbf{y})$ is left invariant.

Chapter 7

Discussion

7.1 Conclusions

This thesis focuses on developing, adapting and implementing statistical methods to quantify uncertainty in mathematical models of the pulmonary blood circulation. The ultimate goal is using these models for decision-making in the clinic, to allow the non-invasive diagnosis of long-term PH, which is currently done with right-heart catheterisation. However, this is an invasive technique that comes with a series of risks and possible side effects, including excessive bleeding because of puncture of the vein during catheter insertion, and partial collapse of the lung. The ultimate quest, therefore, is to use MRI with mathematical modelling and statistical inference to develop a non-invasive alternative.

Substantial efforts have been put into developing mathematical models of the pulmonary circulation [151, 152], however much work still needs to be done to incorporate statistical inference into the analysis. The problem with using mathematical models alone is that the uncertainty, e.g. in model parameters, model form, data measurement process is ignored, making these models lack credibility. A rigorous testing by accounting for all sources of uncertainty is needed. An overview of the sources of uncertainty present in the mathematical models of the pulmonary blood circulation is given in Chapter 1. The present study focuses on estimating unknown parameters of a 1D fluid-dynamics model, which cannot be measured in-vivo, from measured blood flow and pressure data in healthy and diseased (hypertensive) mice. These parameters relate to the blood haemodynamics, e.g. vessel wall stiffness and boundary conditions (3-element Windkessels) for the PDEs by which the models are defined. Chapter 3 of this thesis discusses a Bayesian approach to inferring the vessel wall stiffness (assumed constant throughout the vessel network) and Windkessel *adjustment* parameters (also assumed constant) and quantifying their uncertainty in a fixed vessel network geometry (e.g. fixed vessel radii, length, number of vessels and vessel connectivity) extracted from micro-CT images. The analysis, performed with a random-walk MCMC algorithm (DRAM) makes the assumptions of iid measurement errors, and finds that PH is associated with stiffer and less compliant proximal and distal vasculature. The wall stiffness of the large vessels was larger for the diseased mouse than the healthy (control) mouse, and the compliance from the Windkessels, representative of the wall compliance of the small vessels and capillaries was smaller in PH compared to the control case. In addition, it was found that the mathematical model provides blood pressure predictions that faithfully resemble the haemodynamics for the diseased mouse, however a slight model mismatch was observed for the healthy mouse data.

The model mismatch stems from the model discrepancy between the real system and the mathematical model and the from the wrong noise model, i.e. wrongly assuming iid errors when the errors are in fact correlated. Possible causes for the measurement error correlation are: the nature of the data (i.e. the blood flow or pressure measurements at the current time point depend on measurements at previous time points), and smoothing and averaging of the data. Possible causes for the model discrepancy are: numerical errors (e.g., numerical integration of the PDEs), model assumptions (e.g., purely elastic vessel walls, or the 1D model simplification), and inconsistency between network geometry and haemodynamic data (e.g., the network geometry and the blood flow data do not come from the same mouse).

The model mismatch is incorporated in the analysis presented in Chapter 4 by assuming that the errors follow a multivariate normal distribution with a full covariance matrix, learnt using Gaussian Processes. In this chapter, haemodynamic parameters (vessel stiffness and Windkessel adjustment parameters) in a fixed network were jointly sampled with the error model hyperparameters using Bayesian methods based on random-walk Adaptive Metropolis algorithm. This was carried out for the control mouse data. This chapter emphasizes the importance of allowing for model mismatch when one is present. It is demonstrated that minimising the mean square error between the measured data and the model-predicted data, which is the conventional method widely used in the literature, leads to biased parameter estimates and incorrect predictions, and underestimates uncertainty in parameter and output space. This finding is based on synthetic data, for which the gold standard is known, and it tallies with results from the physiological data. In contrast, the proposed method based on GPs modelling the model mismatch, circumvents these issues and provides wider uncertainty bounds in parameter and output space, which accommodates for natural, physiological variations in pulmonary pressure (e.g. effects of the respiratory cycle).

A further contribution of the analysis in Chapter 4 is the thorough exploration of several mathematical models describing the vessel wall elasticity (i.e. a linear and a non-linear elastic behaviour of the vessel wall) with different vessel wall assumptions (constant, exponential radius-dependent, or vessel-specific stiffness). Model selection was performed with WAIC, and it was found that the model that can most accurately predict the available pressure data for the healthy mouse was the non-linear wall model with a weak exponential radius-dependent stiffness.

Chapters 3 and 4 deal with inference and UQ of haemodynamic parameters, and uncertainty propagation to haemodynamic predictions, in a fixed vessel network obtained based on one image segmentation. In contrast, Chapter 5 presents an analysis of the network uncertainty, i.e. network geometry: vessel radius and length, network connectivity: location of vessel bifurcations, and network size: number of vessels, resulting from multiple image segmentations with different pre-segmentation parameters. The variation in the network also leads to uncertainty in haemodynamic predictions. Using multiple image segmentations of one single mouse lung allowed quantifying the total network variation, and thus the uncertainty in the haemodynamic predictions. Subsequently, one single image segmentation was chosen and the contributions to the total network variation were separated to find the variation due to 1) network geometry and 2) network connectivity and size. GPs were used for density estimation of the vessels' radii and lengths from repeated measurements of these quantities for fixed network connectivity and size, and sampling from these densities allowed forward UQ (i.e. UQ in model predictions) via Monte Carlo sampling (with inverse transform sampling). In addition, varying the network size and connectivity for a fixed network geometry (i.e. deterministically and sequentially reducing the number of vessels), allowed to compute the model predictions, thus quantify the variability in the haemodynamic predictions. Results indicated that variation in network size and connectivity is a larger contributor to haemodynamic uncertainty than the variation in network geometry, i.e. vessel radius and length.

Chapters 3 and 4 employ random-walk MCMC algorithms in a conventional fashion, by evaluating the PDEs at every MCMC iteration for tens of thousands of iterations, rendering the parameter inference procedure unsuitable for use in the clinic due to high computational costs. Thus, Chapter 6 places a strong emphasis on computational efficiency by coupling efficient, gradient-based MCMC algorithms (Hamiltonian and Lagrangian Monte Carlo algorithms) with emulation of the log unnormalised posterior distribution using GPs, with the aim to identify the method which gives the best trade-off between accuracy and efficiency. Several algorithms, particularly adapted for computationally expensive models, were derived and implemented on the 1D fluid-dynamics model, as well as on toy models defined by differential equations. The empirical method comparison revealed that all the proposed algorithms accurately estimated the ground-truth parameter values, which were known for the toy problems. Additionally, Lagrangian Dynamical Monte Carlo and Riemann Manifold Hamiltonian Monte Carlo tended to register the highest efficiency (in terms of effective sample size normalised by the number of forward model evaluations), followed by the Hamiltonian Monte Carlo, and the No U-turn sampler tended to be least efficient.

7.2 Future work

The present study did not jointly quantify the uncertainty in haemodynamic and network parameters, this constitutes future work. Currently, the image segmentation process requires substantial user input, rendering an automated network parameter inference impossible. Efforts are being made into automating the image segmentation process [29]. Future work will also quantify the uncertainty inherent in the imaging process affecting the image intensities and leading to uncertainty in the segmentation and mesh reconstruction of the network vasculature. Furthermore, replacement of the Windkessel boundary conditions, which greatly simplify the downstream resistance, by structured tree boundary conditions [138, 139, 152], can provide an additional level of complexity for approximating downstream resistance and help incorporate the network structure beyond the truncation induced by the image segmentation.

The parameter dimension reduction from Chapter 3 was done ad-hoc and future work includes sensitivity analysis, to identify the parameters that the output is most sensitive to, following approaches in [36]. Additionally, the method comparison in Chapter 6 used the simple linear wall fluid-dynamics model with a constant vessel stiffness and iid measurement errors. Future work includes applying the most efficient algorithm identified to the non-linear wall model with an exponential radius-dependent stiffness (the model best supported by the data, as found in Chapter 4), and incorporating the model mismatch.

While in this thesis mouse data were used, the analysis performed can be translated to human data, using for example the model developed by Qureshi et al. [152], which is a two-sided model incorporating both the pulmonary arterial and venous side, unlike the model used in this study, which only models the arterial side.

Lastly, another future research direction represents the closed-loop effects arising when coupling the mathematical and statistical modelling with the clinical decision process. More specifically, medical interventions caused by model predictions must be accommodated for in the modelling framework. In a clinical application, the prediction of high pulmonary blood pressure above a critical threshold will trigger the administration of vasodilators to reduce the blood pressure, which will increase the blood vessel diameter. A statistical inference performed on the newly measured blood flow data with the old vessel diameter will lead to biased parameter inference, as demonstrated in our preliminary study [92].

Appendix A

Appendix for Chapter 2

A.1 Detailed balance proof for DR

$$
\begin{aligned}
&p(\theta^{*(2)}|\mathbf{y})q_1(\theta^*|\theta^{*(2)})[1-\alpha_1(\theta^*|\theta^{*(2)})]q_2(\theta^{(k-1)}|\theta^*,\theta^{*(2)})\alpha_2(\theta^{(k-1)}|\theta^*,\theta^{*(2)}) \\
&\qquad = p(\theta^{*(2)}|\mathbf{y})q_1(\theta^*|\theta^{*(2)})[1-\alpha_1(\theta^*|\theta^{*(2)})]q_2(\theta^{(k-1)}|\theta^*,\theta^{*(2)}) \\
&\qquad \left[1 \wedge \frac{p(\theta^{k-1}|\mathbf{y})q_1(\theta^*|\theta^{k-1})q_2(\theta^{*(2)}|\theta^*,\theta^{k-1})[1-\alpha_1(\theta^*|\theta^{k-1})]}{p(\theta^{*(2)}|\mathbf{y})q_1(\theta^*|\theta^{*(2)})q_2(\theta^{k-1}|\theta^*,\theta^{*(2)})[1-\alpha_1(\theta^*|\theta^{*(2)})]}\right] \\
&\qquad = \Big[p(\theta^{*(2)}|\mathbf{y})q_1(\theta^*|\theta^{*(2)})q_2(\theta^{k-1}|\theta^*,\theta^{*(2)})[1-\alpha_1(\theta^*|\theta^{*(2)})] \wedge \\
&\qquad\qquad p(\theta^{k-1}|\mathbf{y})q_1(\theta^*|\theta^{k-1})q_2(\theta^{*(2)}|\theta^*,\theta^{k-1})[1-\alpha_1(\theta^*|\theta^{k-1})]\Big],
\end{aligned}
$$

which is invariant with respect to the parameter swap $\theta^{k-1} \leftrightarrow \theta^{*(2)}$, hence detailed balance holds:

$$
\begin{aligned}
&p(\theta^{*(2)}|\mathbf{y})q_1(\theta^*|\theta^{*(2)})[1-\alpha_1(\theta^*|\theta^{*(2)})]q_2(\theta^{k-1}|\theta^*,\theta^{*(2)})\alpha_2(\theta^{k-1}|\theta^*,\theta^{*(2)}) = \\
&\quad p(\theta^{k-1}|\mathbf{y})q_1(\theta^*|\theta^{k-1})[1-\alpha_1(\theta^*|\theta^{k-1})]q_2(\theta^{*(2)}|\theta^*,\theta^{k-1})\alpha_2(\theta^{*(2)}|\theta^*,\theta^{k-1}).
\end{aligned}
$$

A.2 Detailed balance proof for HMC

Consider a probabilistic model $p(\mathbf{y}|\theta)$, where $\mathbf{y}$ are the data and θ the model parameters with prior probability $p(\theta)$. The corresponding posterior probability is given by $p(\theta|\mathbf{y}) \propto p(\mathbf{y}|\theta)p(\theta)$.

Define

$$
p(\theta|\mathbf{y}) \;=\; \frac{\exp(-E(\theta))}{Z_p},
$$

where $E(\theta) = -(\log p(\mathbf{y}|\theta) + \log p(\theta))$ is the potential function, and $Z_p = \int_\theta p(\theta)p(\mathbf{y}|\theta)d\theta = p(\mathbf{y})$ is a normalising constant ensuring that the integral of the probability distribution $p(\theta|\mathbf{y})$ is 1.

Starting with a parameter vector $\boldsymbol{\theta}$, the algorithm follows:

1. Sample a vector of auxiliary 'momentum' variables $\mathbf{r}$ from $\mathscr{MVN}(\mathbf{0},\mathbf{M})$ in a Gibbs step

$$p(\mathbf{r}) \;=\; \frac{\exp(-K(\mathbf{r}))}{Z_k},$$

where $K(\mathbf{r}) \;=\; \frac{1}{2}\mathbf{r}^T\mathbf{M}^{-1}\mathbf{r}$ is the kinetic energy ($\mathbf{M}$ - mass matrix) and $Z_k = \exp\left(-\frac{1}{2}\log\left((2\pi)^d|\mathbf{M}|\right)\right)$ is a normalising constant ensuring that the integral of the probability distribution $p(\mathbf{r})$ is 1. Note that $p(\mathbf{r}|\boldsymbol{\theta}) = p(\mathbf{r})$ since it is expressed independently of $\boldsymbol{\theta}$.

2. Propose $(\boldsymbol{\theta}^*,\mathbf{r}^*)$ by following a set of deterministic Hamiltonian dynamics steps in the joint $(\boldsymbol{\theta},\mathbf{r})$ space, and the proposal probability ratio is:

$$\frac{q(\boldsymbol{\theta}^*,\mathbf{r}^*|\boldsymbol{\theta},\mathbf{r})}{q(\boldsymbol{\theta},\mathbf{r}|\boldsymbol{\theta}^*,\mathbf{r}^*)} \;=\; \frac{1}{1} \;=\; 1.$$

3. Accept the parameters $(\boldsymbol{\theta}^*,\mathbf{r}^*)$ in a M-H step with the following probability:

$$\begin{aligned}\alpha(\boldsymbol{\theta}^*,\mathbf{r}^*|\boldsymbol{\theta},\mathbf{r}) \;&=\; \left[1\wedge\frac{p(\boldsymbol{\theta}^*|\mathbf{y})p(\mathbf{r}^*|\boldsymbol{\theta}^*)q(\boldsymbol{\theta},\mathbf{r}|\boldsymbol{\theta}^*,\mathbf{r}^*)}{p(\boldsymbol{\theta}|\mathbf{y})p(\mathbf{r}|\boldsymbol{\theta})q(\boldsymbol{\theta}^*,\mathbf{r}^*|\boldsymbol{\theta},\mathbf{r})}\right]\\ &=\; 1\wedge\frac{\exp(-E(\boldsymbol{\theta}^*))}{\exp(-E(\boldsymbol{\theta}))}\frac{\exp(-K(\mathbf{r}^*))}{\exp(-K(\mathbf{r}))}.\end{aligned}$$

This algorithm preserves detailed balance with respect to $p(\boldsymbol{\theta}|\mathbf{y})$.

PROOF

$$\begin{aligned}&p(\boldsymbol{\theta}|\mathbf{y})p(\mathbf{r}|\boldsymbol{\theta})q(\boldsymbol{\theta}^*,\mathbf{r}^*|\boldsymbol{\theta},\mathbf{r})\alpha(\boldsymbol{\theta}^*,\mathbf{r}^*|\boldsymbol{\theta},\mathbf{r})\\ =\;& \frac{\exp(-E(\boldsymbol{\theta}))}{Z_p}\frac{\exp(-K(\mathbf{r}))}{Z_k}\left[1\wedge\frac{\exp(-E(\boldsymbol{\theta}^*))\exp(-K(\mathbf{r}^*))}{\exp(-E(\boldsymbol{\theta}))\exp(-K(\mathbf{r}))}\right]\\ =\;& \left(\frac{1}{Z_pZ_k}\right)[\exp(-H(\boldsymbol{\theta},\mathbf{r}))\wedge\exp(-H(\boldsymbol{\theta}^*,\mathbf{r}^*))],\end{aligned}$$

which is invariant with respect to the parameter swap $(\boldsymbol{\theta},\mathbf{r})\leftrightarrow(\boldsymbol{\theta}^*,\mathbf{r}^*)$.

The algorithm defines a transition kernel that leaves the joint posterior distribution,

$$p(\boldsymbol{\theta},\mathbf{r}|\mathbf{y}) = p(\boldsymbol{\theta}|\mathbf{y})p(\mathbf{r}|\boldsymbol{\theta})$$

invariant, and hence the target distribution $p(\boldsymbol{\theta}|\mathbf{y})$ invariant, as the auxiliary variables can be integrated out:

$$\int p(\boldsymbol{\theta},\mathbf{r}|\mathbf{y})d\mathbf{r} = p(\boldsymbol{\theta}|\mathbf{y}).$$

A.3 Detailed balance proof for RMHMC

Note that in the Riemann Manifold Hamiltonian Monte Carlo (RMHMC) algorithm, the Hamiltonian function is:

$$H(\theta,\mathbf{r}) = E(\theta) + K(\mathbf{r}|\theta).$$

Hence,

$$p(\mathbf{r}|\theta) = \frac{\exp\left(-K(\mathbf{r}|\theta)\right)}{Z_k(\theta)},$$

where $K(\mathbf{r}|\theta) = \frac{1}{2}\mathbf{r}^T\mathbf{M}(\theta)^{-1}\mathbf{r}$, and $Z_k(\theta) = \exp\left(-\frac{1}{2}\log\left((2\pi)^d|\mathbf{M}(\theta)|\right)\right)$ is a normalising factor ensuring that the integral of the probability distribution $p(\mathbf{r}|\theta)$ is 1. Here, $\mathbf{M}(\theta)$ is a function of every θ along the leapfrog trajectory. All the other terms and the proof are similar to HMC in Section A.2.

A.4 Detailed balance proof for LDMC

The Lagrangian Dynamical Monte Carlo (LDMC) algorithm uses ’velocity’ instead of ’momentum’ as the auxiliary variable, thus the sampling follows the Lagrangian dynamics rather than the Hamiltonian dynamics.

Define

$$p(\theta|\mathbf{y}) \;=\; \frac{\exp(-E(\theta))}{Z_p},$$

where $E(\theta) = -(\log p(\theta) + \log p(\mathbf{y}|\theta))$ is the potential function and $Z_p = \int_\theta p(\theta)p(\mathbf{y}|\theta)d\theta = p(\mathbf{y})$ is a normalising constant ensuring that the integral of the probability distribution $p(\theta|\mathbf{y})$ is 1.

Starting with a parameter vector θ, the algorithm follows:

1. Sample a vector of auxiliary ‘velocity’ variables $\mathbf{v}$ from $\mathscr{MVN}(\mathbf{0},\mathbf{M}(\theta)^{-1})$ in a Gibbs step, i.e.

   $$p(\mathbf{v}|\theta) \;=\; \frac{\exp(-K(\mathbf{v}|\theta))}{Z_k(\theta)},$$

   where $K(\mathbf{v}|\theta) \;=\; \frac{1}{2}\mathbf{v}^T\mathbf{M}(\theta)\mathbf{v}$ and $Z_k(\theta) = \exp\left(-\frac{1}{2}\log\left((2\pi)^d|\mathbf{M}(\theta)^{-1}|\right)\right)$ is a normalising factor ensuring that the integral of the probability distribution $p(\mathbf{v}|\theta)$ is 1.

2. Propose $(\theta^*,\mathbf{v}^*)$ by following a set of Lagrangian dynamics steps, which are deterministic, but do not preserve volume in phase space, and the proposal probability ratio follows from the transformation of probability densities of random variables:

   $$\frac{q(\theta^*,\mathbf{v}^*|\theta,\mathbf{v})}{q(\theta,\mathbf{v}|\theta^*,\mathbf{v}^*)} \;=\; \left|\frac{\partial(\theta,\mathbf{v})}{\partial(\theta^*,\mathbf{v}^*)}\right|.$$

3. Accept the parameters $(\theta^*, \mathbf{v}^*)$ in a M-H step with the following probability:

$$\alpha(\theta^*, \mathbf{v}^* | \theta, \mathbf{v}) \; = \; 1 \wedge \frac{\dfrac{\exp(-H[\theta^*, \mathbf{v}^*])}{Z_k(\theta^*)}}{\dfrac{\exp(-H[\theta, \mathbf{v}])}{Z_k(\theta)}} \left| \frac{\partial(\theta^*, \mathbf{v}^*)}{\partial(\theta, \mathbf{v})} \right|,$$

where $H(\theta, \mathbf{v}) = E(\theta) + K(\mathbf{v}|\theta)$.

This algorithm preserves detailed balance with respect to $p(\theta|\mathbf{y})$.

PROOF

$$\begin{aligned}
& p(\theta|\mathbf{y}) p(\mathbf{v}|\theta) q(\theta^*, \mathbf{v}^* | \theta, \mathbf{v}) \left[1 \wedge \frac{p(\theta^*|\mathbf{y}) p(\mathbf{v}^*|\theta^*) q(\theta, \mathbf{v}|\theta^*, \mathbf{v}^*)}{p(\theta|\mathbf{y}) p(\mathbf{v}|\theta) q(\theta^*, \mathbf{v}^*|\theta, \mathbf{v})} \right] \\
= \; & \frac{\exp(-E(\theta))}{Z_p} \frac{\exp(-K(\mathbf{v}|\theta))}{Z_k(\theta)} q(\theta^*, \mathbf{v}^*|\theta, \mathbf{v}) \left[1 \wedge \frac{\dfrac{\exp(-E(\theta^*)) \exp(-K(\mathbf{v}^*|\theta^*))}{Z_k(\theta^*)}}{\dfrac{\exp(-E(\theta)) \exp(-K(\mathbf{v}|\theta))}{Z_k(\theta)}} \left| \frac{\partial(\theta^*, \mathbf{v}^*)}{\partial(\theta, \mathbf{v})} \right| \right] \\
= \; & \left(\frac{1}{Z_p} \right) \left[\frac{\exp(-H(\theta, \mathbf{v})) q(\theta^*, \mathbf{v}^*|\theta, \mathbf{v})}{Z_k(\theta)} \wedge \frac{\exp(-H(\theta^*, \mathbf{v}^*)) q(\theta, \mathbf{v}|\theta^*, \mathbf{v}^*)}{Z_k(\theta^*)} \right],
\end{aligned}$$

which is invariant with respect to the parameter swap $(\theta, \mathbf{v}) \leftrightarrow (\theta^*, \mathbf{v}^*)$. Thus, the transition kernel leaves the joint posterior distribution $p(\theta, \mathbf{v}|\mathbf{y})$ invariant, and hence the target distribution, $p(\theta|\mathbf{y})$ invariant, as the auxiliary variable $\mathbf{v}$ can be integrated out:

$$\int p(\theta, \mathbf{v}|\mathbf{y}) d\mathbf{v} = p(\theta|\mathbf{y}).$$

A.5 Detailed balance proof for NUTS

NUTS, which stands for "No U-Turn Sampler", carries out Hamiltonian dynamics (with the leapfrog method) on the combined momentum and parameter space $(\theta, \mathbf{r})$. It collects configurations along the leapfrog trajectory to create two sets B and $C \subset B$. A balanced binary tree is constructed by repeatedly doubling the number of points visited along the leapfrog trajectory. B contains all the (position, momentum) $(\theta, \mathbf{r})$ configurations collected throughout the trajectory, while C contains a subset of these parameters to which transitioning does not violate detailed balance, a condition checked using a real slice variable u. The algorithm effectively defines a transition kernel $T(\theta^*, \mathbf{r}^* | \theta, \mathbf{r}, C)$ that leaves the joint posterior distribution,

$$p(\theta, \mathbf{r}, u, B, C|\mathbf{y}) = p(\theta|\mathbf{y}) p(\mathbf{r}|\theta) p(u|\theta, \mathbf{r}) p(B, C|\theta, \mathbf{r}, u)$$

invariant, and thus the target distribution $p(\theta|\mathbf{y})$ invariant, as the auxiliary variables can be integrated out:

$$\int p(\theta, \mathbf{r}, u, B, C|\mathbf{y}) d\mathbf{r} du dB dC = p(\theta|\mathbf{y}).$$

NUTS exists in two forms: the naive version and the efficient version. The naive NUTS algorithm is first explained.

The algorithm starts with a parameter vector θ and proceeds as follows:

1. Sample a vector of auxiliary 'momentum' variables $\mathbf{r}$ from $\mathscr{MVN}(\mathbf{0}, \mathbf{M})$ in a Gibbs step

   $$p(\mathbf{r}) \; = \; \frac{\exp(-K(\mathbf{r}))}{Z_k},$$

   where $K(\mathbf{r}) \; = \; \frac{1}{2}\mathbf{r}^T\mathbf{M}^{-1}\mathbf{r}$ is the kinetic energy ($\mathbf{M}$ - mass matrix), and $Z_k = \exp\left(-\frac{1}{2}\log\left((2\pi)^d|\mathbf{M}|\right)\right)$ is a normalising constant ensuring that the integral of the probability distribution $p(\mathbf{r})$ is 1. Note that $p(\mathbf{r}|\theta) = p(\mathbf{r})$ since it is expressed independently of θ.

2. Sample the slice variable $u|\theta, \mathbf{r} \sim \text{Uniform}(0, \exp(-E(\theta) - K(\mathbf{r})))$ in a Gibbs step.

3. Sample B, C from their conditional distribution $p(B, C|\theta, \mathbf{r}, u, \varepsilon)$ in a Gibbs step.

4. Move from $(\theta, \mathbf{r})$ to $(\theta^*, \mathbf{r}^*)$ with uniform transition probability over C:

   $$T(\theta^*, \mathbf{r}^*|\theta, \mathbf{r}, C) = \frac{\mathbb{1}((\theta^*, \mathbf{r}^*) \in C)}{|C|}.$$

This algorithm preserves detailed balance with respect to $p(\theta, \mathbf{r}, u, B, C|\mathbf{y})$.

PROOF

The aim is to show that

$$\begin{gathered} T(\theta^*, \mathbf{r}^*|\theta, \mathbf{r}, C) p(\theta, \mathbf{r}|\mathbf{y}) p(u|\theta, \mathbf{r}) p(B, C|\theta, \mathbf{r}, u, \varepsilon) = \\ T(\theta, \mathbf{r}|\theta^*, \mathbf{r}^*, C) p(\theta^*, \mathbf{r}^*|\mathbf{y}) p(u|\theta^*, \mathbf{r}^*) p(B, C|\theta^*, \mathbf{r}^*, u, \varepsilon), \end{gathered} \tag{A.1}$$

where ε is the step size in the leapfrog algorithm.

By noting that

$$p(\theta, \mathbf{r}|\mathbf{y}) = p(\theta|\mathbf{y}) p(\mathbf{r}|\theta), \tag{A.2}$$

equation (A.1), which needs proving, becomes equivalent to:

$$\begin{gathered} T(\theta^*, \mathbf{r}^*|\theta, \mathbf{r}, C) p(\theta|\mathbf{y}) p(\mathbf{r}|\theta) p(u|\theta, \mathbf{r}) p(B, C|\theta, \mathbf{r}, u, \varepsilon) = \\ T(\theta, \mathbf{r}|\theta^*, \mathbf{r}^*, C) p(\theta^*|\mathbf{y}) p(\mathbf{r}^*|\theta^*) p(u|\theta^*, \mathbf{r}^*) p(B, C|\theta^*, \mathbf{r}^*, u, \varepsilon), \end{gathered} \tag{A.3}$$

The individual terms are:

$$\begin{aligned}
p(\theta|\mathbf{y}) &= \frac{\exp(-E(\theta))}{Z_p},\ E(\theta) = -(\log p(\theta) + \log p(\mathbf{y}|\theta)), \\
&\quad Z_p = \int_\theta p(\theta)p(\mathbf{y}|\theta)d\theta = p(\mathbf{y}), \\
p(\mathbf{r}|\theta) &= \frac{\exp(-K(\mathbf{r}))}{Z_k},\ Z_k = \exp(-\frac{1}{2}\log\Big((2\pi)^d|\mathbf{M}|\Big)), \\
p(u|\theta,\mathbf{r}) &= \frac{\mathbb{1}(u \leq \exp(-E(\theta) - K(\mathbf{r})))}{\exp(-E(\theta) - K(\mathbf{r}))}, \\
p(B,C|\theta,\mathbf{r},u,\varepsilon) &= p(B,C|\theta^*,\mathbf{r}^*,u,\varepsilon),\ \text{where } \varepsilon \text{ is the leapfrog step size}, \\
p(B,C|\theta,\mathbf{r},u,\varepsilon) &= \frac{\mathbb{1}((\theta,\mathbf{r}) \in C)}{|B|},\ \text{where } \mathbb{1}(.)\text{: indicator function, } |B|\text{: cardinality of the set B}, \\
T(\theta^*,\mathbf{r}^*|\theta,\mathbf{r},C) &= \frac{\mathbb{1}((\theta^*,\mathbf{r}^*) \in C)}{|C|}.
\end{aligned}$$

But,

$$\begin{aligned}
p(\theta,\mathbf{r}|u,B,C,\varepsilon) &= \frac{p(\theta,\mathbf{r},u,B,C|\varepsilon)}{p(u,B,C|\varepsilon)} \\
&= \frac{p(B,C|\theta,\mathbf{r},u,\varepsilon)p(\theta,\mathbf{r},u)}{p(u,B,C|\varepsilon)} \\
&\propto p(B,C|\theta,\mathbf{r},u,\varepsilon)p(\theta,\mathbf{r}|u) \\
&\propto \frac{\mathbb{1}((\theta,\mathbf{r}) \in C)}{|B|}\mathbb{1}(u \leq \exp(-E(\theta) - K(\mathbf{r}))) \qquad \text{(A.4)} \\
&\propto \frac{\mathbb{1}((\theta,\mathbf{r}) \in C)}{|C|}\frac{|C|}{|B|},\ \text{where } |B| \geq |C| \\
&\propto \frac{\mathbb{1}((\theta,\mathbf{r}) \in C)}{|C|}, \\
&\text{since } \mathbb{1}(u \leq \exp(-E(\theta) - K(\mathbf{r}))) = 1 \text{ if } (\theta,\mathbf{r}) \in C.
\end{aligned}$$

Thus, the following holds:

$$p(\theta,\mathbf{r}|u,B,C,\varepsilon) \propto p(\theta,\mathbf{r}|C) = \frac{\mathbb{1}((\theta,\mathbf{r}) \in C)}{|C|}. \qquad \text{(A.5)}$$

By cancellation of terms and using eq (A.4), eq (A.3) can be simplified to:

$$T(\theta^*,\mathbf{r}^*|\theta,\mathbf{r},C)p(\theta,\mathbf{r}|C) = \quad T(\theta,\mathbf{r}|\theta^*,\mathbf{r}^*,C)p(\theta^*,\mathbf{r}^*|C), \qquad \text{(A.6)}$$

which needs proving. To do so, the following can be noted:

$$\begin{aligned}
& T(\theta^*,\mathbf{r}^*|\theta,\mathbf{r},C)p(\theta,\mathbf{r}|C) \\
= \; & \frac{\mathbb{1}((\theta^*,\mathbf{r}^*)\in C)}{|C|}\frac{\mathbb{1}((\theta,\mathbf{r})\in C)}{|C|} \\
= \; & \frac{\mathbb{1}((\theta,\mathbf{r})\in C)}{|C|}\frac{\mathbb{1}((\theta^*,\mathbf{r}^*)\in C)}{|C|} \\
= \; & T(\theta,\mathbf{r}|\theta^*,\mathbf{r}^*,C)p(\theta^*,\mathbf{r}^*|C).
\end{aligned}$$

Hence, the algorithm is invariant with respect to a swap $(\theta,\mathbf{r}) \leftrightarrow (\theta^*,\mathbf{r}^*)$.

A more efficient NUTS is needed because:

1. The naive NUTS requires a large amount of memory to store all $(\theta,\mathbf{r}) \in C$.
2. The simple uniform sampling is inefficient; there exist alternative transition kernels that ensure detailed balance is satisfied with respect to a uniform distribution and that produce larger jumps on average than the simple uniform sampling method.

The efficient NUTS thus uses a more sophisticated, memory-efficient transition kernel that leaves the distribution over C invariant.

The algorithm starts with a parameter vector θ and proceeds as follows:

1. Sample a vector of auxiliary ‘momentum’ variables $\mathbf{r}$ from $\mathscr{MVN}(\mathbf{0},\mathbf{M})$ in a Gibbs step

   $$p(\mathbf{r}) \;=\; \frac{\exp(-K(\mathbf{r}))}{Z_k},$$

   where $K(\mathbf{r}) \;=\; \frac{1}{2}\mathbf{r}^T\mathbf{M}^{-1}\mathbf{r}$ is the kinetic energy ($\mathbf{M}$ - mass matrix) and
   $Z_k = \exp\left(-\frac{1}{2}\log\left((2\pi)^d|\mathbf{M}|\right)\right)$ is a normalising constant ensuring that the integral of the probability distribution $p(\mathbf{r})$ is 1. Note that $p(\mathbf{r}|\theta) = p(\mathbf{r})$ since it is expressed independently of θ.
2. Sample the slice variable $u|\theta,\mathbf{r} \sim \text{Uniform}(0,\exp(-E(\theta)-K(\mathbf{r})))$ in a Gibbs step.
3. Sample B,C from their conditional distribution $p(B,C|\theta,\mathbf{r},u,\varepsilon)$ in a Gibbs step.
4. Deterministically divide the set C into C_{new} and C_{old}, such that $C_{\text{old}} \cup C_{\text{new}} = C$, $C_{\text{old}} \cap C_{\text{new}} = \emptyset$, and sample C_{old} and C_{new} with probability:

   $$p(C_{\text{new}}|C) = p(C_{\text{old}}|C) = 1.$$

5. Propose to move from $(\theta^{j-1},\mathbf{r}^{j-1}) \in C^j_{\text{old}}$ to $(\theta^j,\mathbf{r}^j) \in C^j_{\text{new}}$ by following a set of deterministic Hamiltonian dynamics steps. Switch from C^j_{old} to C^j_{new} with probability $\mathbb{1}(C^{'j}_{\text{old}} =$

$C^{j}_{\text{new}})\mathbb{1}(C^{'j}_{\text{new}} = C^{j}_{\text{old}})$ and choose an element $(\theta^*,\mathbf{r}^*) \in C_{\text{new}}$ with uniform probability $\dfrac{\mathbb{1}((\theta^*,\mathbf{r}^*) \in C_{\text{new}})}{|C_{\text{new}}|}$, where $|C_{\text{new}}|$ defines the cardinality of the set C_{new}:

$$
\begin{aligned}
q(\theta^j,\mathbf{r}^j,C^j_{\text{new}}|\theta^{j-1},\mathbf{r}^{j-1},C^j_{\text{old}}) &= \mathbb{1}(C^{j'}_{\text{old}} = C^j_{\text{new}})\mathbb{1}(C^{j'}_{\text{new}} = C^j_{\text{old}})\frac{\mathbb{1}((\theta^j,\mathbf{r}^j) \in C^j_{\text{new}})}{|C^j_{\text{new}}|} \\
&= \frac{\mathbb{1}((\theta^j,\mathbf{r}^j) \in C^j_{\text{new}})}{|C^j_{\text{new}}|},
\end{aligned}
$$

assuming $\mathbb{1}(C^{j'}_{\text{old}} = C^j_{\text{new}})\mathbb{1}(C^{j'}_{\text{new}} = C^j_{\text{old}}) = 1$.

6. Accept the parameters $(\theta^j,\mathbf{r}^j,C^j_{\text{new}})$ in a M-H step with the following probability:

$$
\alpha(\theta^j,\mathbf{r}^j,C^j_{\text{new}}|\theta^{j-1},\mathbf{r}^{j-1},C^j_{\text{old}}) = 1 \wedge \frac{|C^j_{\text{new}}|}{|C^j_{\text{old}}|}\frac{\mathbb{1}((\theta^{j-1},\mathbf{r}^{j-1}) \in C^j_{\text{old}})}{\mathbb{1}((\theta^j,\mathbf{r}^j) \in C^j_{\text{new}})}.
$$

The sampler proposes to move from $(\theta,\mathbf{r}) \in C_{\text{old}}$ to $(\theta^*,\mathbf{r}^*) \in C_{\text{new}}$ by following a set of deterministic Hamiltonian dynamics steps via a sequence of intermediate states. Thus, the transition kernel T is repeatedly applied after every tree doubling, i.e. steps (5) and (6) are repeated a number of times equal to the tree height:

$$
\begin{aligned}
T(\theta^*,\mathbf{r}^*,C_{\text{new}}|\theta,\mathbf{r},C_{\text{old}}) &= \prod_{j=1}^{J} q(\theta^j,\mathbf{r}^j,C^j_{\text{new}}|\theta^{j-1},\mathbf{r}^{j-1},C^j_{\text{old}})\alpha(\theta^j,\mathbf{r}^j,C^j_{\text{new}}|\theta^{j-1},\mathbf{r}^{j-1},C^j_{\text{old}}), \\
&= \prod_{j=1}^{J}\left[\frac{\mathbb{1}((\theta^j,\mathbf{r}^j) \in C^j_{\text{new}})}{|C^j_{\text{new}}|} \wedge \frac{\mathbb{1}((\theta^{j-1},\mathbf{r}^{j-1}) \in C^j_{\text{old}})}{|C^j_{\text{old}}|}\right],
\end{aligned}
$$

where $\theta = \theta^0,\ \mathbf{r} = \mathbf{r}^0$: the initial values, J: the number of tree doublings (tree height) before reaching the final proposed values $\theta^* = \theta^J,\ \mathbf{r}^* = \mathbf{r}^J$. C_{new} contains all the points visited at the last doubling, i.e. $C_{\text{new}} = C^J_{\text{new}}$ and

$$
\cap_j C^j_{\text{new}} = \emptyset; \quad C^j_{\text{old}} \cup C^j_{\text{new}} = C^j; \quad C = C^J; \quad C^j_{\text{old}} \cup C^j_{\text{new}} = C^{j+1}_{\text{old}}; \quad C_{\text{old}} = C^J_{\text{old}}.
$$

This algorithm preserves detailed balance with respect to $p(\theta,\mathbf{r},u,B,C|\mathbf{y})$.

PROOF

Eq (A.3) in the naive NUTS algorithm is adapted to give

$$
\begin{gathered}
T(\theta^*,\mathbf{r}^*,C_{\text{new}}|\theta,\mathbf{r},C_{\text{old}})p(\theta|\mathbf{y})p(\mathbf{r}|\theta)p(u|\theta,\mathbf{r})p(B,C|\theta,\mathbf{r},u,\varepsilon)p(C_{\text{old}}|C) = \\
T(\theta,\mathbf{r},C_{\text{old}}|\theta^*,\mathbf{r}^*,C_{\text{new}})p(\theta^*|\mathbf{y})p(\mathbf{r}^*|\theta^*)p(u|\theta^*,\mathbf{r}^*)p(B,C|\theta^*,\mathbf{r}^*,u,\varepsilon)p(C_{\text{new}}|C).
\end{gathered}
\tag{A.7}
$$

The aim is to show that eq (A.7) holds.

Eq (A.7) can be simplified using eq (A.4) and (A.5) from the naive NUTS section to give:

$$\begin{aligned}
& T(\theta^*, \mathbf{r}^*, C_{\text{new}} | \theta, \mathbf{r}, C_{\text{old}}) p(\theta, \mathbf{r} | C) p(C_{\text{old}} | C) \\
= & \prod_{j=1}^{J} \left[\frac{\mathbb{1}((\theta^j, \mathbf{r}^j) \in C^j_{\text{new}})}{|C^j_{\text{new}}|} \wedge \frac{\mathbb{1}((\theta^{j-1}, \mathbf{r}^{j-1}) \in C^j_{\text{old}})}{|C^j_{\text{old}}|} \right] \frac{\mathbb{1}((\theta, \mathbf{r}) \in C)}{|C|} \\
= & \prod_{j=1}^{J} \left[\frac{\mathbb{1}((\theta^{j-1}, \mathbf{r}^{j-1}) \in C^j_{\text{old}})}{|C^j_{\text{old}}|} \wedge \frac{\mathbb{1}((\theta^j, \mathbf{r}^j) \in C^j_{\text{new}})}{|C^j_{\text{new}}|} \right] \frac{\mathbb{1}((\theta^*, \mathbf{r}^*) \in C)}{|C|} \\
= & T(\theta, \mathbf{r}, C_{\text{old}} | \theta^*, \mathbf{r}^*, C_{\text{new}}) p(\theta^*, \mathbf{r}^* | C) p(C_{\text{new}} | C).
\end{aligned}$$

Hence, the algorithm is invariant with respect to the parameter swap $(\theta, \mathbf{r}, C_{\text{old}}) \leftrightarrow (\theta^*, \mathbf{r}^*, C_{\text{new}})$ via a sequence of intermediate state swaps $(\theta^{j-1}, \mathbf{r}^{j-1}, C^j_{\text{old}}) \leftrightarrow (\theta^j, \mathbf{r}^j, C^j_{\text{new}})$. So the joint distribution $p(\theta, \mathbf{r}, u, B, C | \mathbf{y})$ is left invariant.

A.6 Proof of equation (2.40)

Note that

$$\text{Cov}(X,Y) = \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y).$$

And so,

$$\begin{aligned}
\text{Cov}\left(\frac{\partial \log p(\mathbf{y}|\theta)}{\partial \theta_i}, \frac{\partial \log p(\mathbf{y}|\theta)}{\partial \theta_j} \right) = \mathbb{E}_{\mathbf{y}|\theta} \left(\frac{\partial \log p(\mathbf{y}|\theta)}{\partial \theta_i} \frac{\partial \log p(\mathbf{y}|\theta)}{\partial \theta_j} \right) - \\
\mathbb{E}_{\mathbf{y}|\theta} \left(\frac{\partial \log p(\mathbf{y}|\theta)}{\partial \theta_i} \right) \mathbb{E}_{\mathbf{y}|\theta} \left(\frac{\partial \log p(\mathbf{y}|\theta)}{\partial \theta_j} \right).
\end{aligned}$$

But

$$\mathbb{E}_{\mathbf{y}|\theta} \left(\frac{\partial \log p(\mathbf{y}|\theta)}{\partial \theta_i} \right) = \int p(\mathbf{y}|\theta) \frac{1}{p(\mathbf{y}|\theta)} \frac{\partial p(\mathbf{y}|\theta)}{\partial \theta_i} d\mathbf{y} = \frac{\partial}{\partial \theta_i} \int p(\mathbf{y}|\theta) d\mathbf{y} d\theta = \frac{\partial}{\partial \theta_i} 1 = 0.$$

Hence,

$$\text{Cov}\left(\frac{\partial \log p(\mathbf{y}|\theta)}{\partial \theta_i}, \frac{\partial \log p(\mathbf{y}|\theta)}{\partial \theta_j} \right) = \mathbb{E}_{\mathbf{y}|\theta} \left(\frac{\partial \log p(\mathbf{y}|\theta)}{\partial \theta_i} \frac{\partial \log p(\mathbf{y}|\theta)}{\partial \theta_j} \right),$$

which proves the first equality of eq (2.40). To prove that the second equality holds, note that:

$$\begin{aligned}
\frac{\partial^2 \log p(\mathbf{y}|\theta)}{\partial \theta_i \partial \theta_j} &= \frac{\partial}{\partial \theta_i} \left(\frac{\partial \log p(\mathbf{y}|\theta)}{\partial \theta_j} \right) = \frac{\partial}{\partial \theta_i} \left(\frac{1}{p(\mathbf{y}|\theta)} \frac{\partial p(\mathbf{y}|\theta)}{\partial \theta_j} \right) \\
&= -\frac{1}{p(\mathbf{y}|\theta)^2} \frac{\partial p(\mathbf{y}|\theta)}{\partial \theta_i} \frac{\partial p(\mathbf{y}|\theta)}{\partial \theta_j} + \frac{1}{p(\mathbf{y}|\theta)} \frac{\partial^2 p(\mathbf{y}|\theta)}{\partial \theta_i \partial \theta_j} \\
&= -\frac{\partial \log p(\mathbf{y}|\theta)}{\partial \theta_i} \frac{\partial \log p(\mathbf{y}|\theta)}{\partial \theta_j} + \frac{1}{p(\mathbf{y}|\theta)} \frac{\partial^2 p(\mathbf{y}|\theta)}{\partial \theta_i \partial \theta_j}
\end{aligned}$$

Multiply both sides by -1 and take expectation with respect to $\mathbf{y}|\theta$ to obtain:

$$\mathbb{E}_{\mathbf{y}|\theta}\left(-\frac{\partial^2 \log p(\mathbf{y}|\theta)}{\partial\theta_i\partial\theta_j}\right)=\mathbb{E}_{\mathbf{y}|\theta}\left(\frac{\partial \log p(\mathbf{y}|\theta)}{\partial\theta_i}\frac{\partial \log p(\mathbf{y}|\theta)}{\partial\theta_j}-\frac{1}{p(\mathbf{y}|\theta)}\frac{\partial^2 p(\mathbf{y}|\theta)}{\partial\theta_i\partial\theta_j}\right).$$

But,

$$\mathbb{E}_{\mathbf{y}|\theta}\left(\frac{1}{p(\mathbf{y}|\theta)}\frac{\partial^2 p(\mathbf{y}|\theta)}{\partial\theta_i\partial\theta_j}\right)=\int\frac{1}{p(\mathbf{y}|\theta)}\frac{\partial^2 p(\mathbf{y}|\theta)}{\partial\theta_i\partial\theta_j}p(\mathbf{y}|\theta)d\mathbf{y}=\frac{\partial^2}{\partial\theta_i\partial\theta_j}\int p(\mathbf{y}|\theta)d\mathbf{y}=\frac{\partial^2 1}{\partial\theta_i\partial\theta_j}=0.$$

Hence,

$$\mathbb{E}_{\mathbf{y}|\theta}\left(-\frac{\partial^2 \log p(\mathbf{y}|\theta)}{\partial\theta_i\partial\theta_j}\right) \quad = \quad \mathbb{E}_{\mathbf{y}|\theta}\left(\frac{\partial \log p(\mathbf{y}|\theta)}{\partial\theta_i}\frac{\partial \log p(\mathbf{y}|\theta)}{\partial\theta_j}\right).$$

The proof assumes that operations ’integration’ and ’taking derivatives’ (up to 2^{nd} order) commute.

Appendix B

Appendix for Chapter 4

B.1 N-steps ahead DA-GP-MCMC algorithm

This section describes a proposed Bayesian approach for accelerating parameter estimation and UQ in expensive models, such as the fluid-dynamics model under consideration, by using a novel combination of state-of-the-art statistical inference techniques. This work takes inspiration from a paper by Rasmussen [156], and makes various modifications inspired from the statistical literature (e.g. Delayed Acceptance MCMC [33, 177]) which have the potential to further reduce the computational costs. The algorithm derived is called the N-steps ahead DA-GP-MCMC algorithm, which is proved to converge asymptotically to the correct posterior distribution, see our study [140] for a proof.

The N-steps ahead DA-GP-MCMC algorithm proceeds as follows:

- *Initial design stage.* Starting from a space filling design in parameter space, e.g. Sobol sequence [16], integrate the PDEs numerically for each parameter vector to get the true log likelihood. Use these points to build a GP emulator (surrogate model) for the log likelihood. The GP model is built on a compact parameter space, where the lower and upper limits are decided in advance.

- *Exploratory phase.* Gather information about the target distribution by running MCMC on the surrogate log posterior of the PDE parameters; the proposed point is subject to a MH accept/reject step, for which the simulator is called, i.e. the PDEs are integrated numerically – see the pseudocode in Algorithm 1g for one iteration of the algorithm. The emulator is sequentially refined (optimum covariance hyperparameters are found by maximisation of the log marginal likelihood of the hyperparameters) as new points are accepted. Following [156], the emulated approximate log posterior distribution of the MCMC algorithm is set to

$$\log \tilde{p}(\theta, \phi|\mathbf{y}) \propto \left(\mathbb{E}(f(\theta,\phi)|D) + \sqrt{\text{var}(f(\theta,\phi)|D)}\right) + \log p(\theta,\phi).$$

Here $p(\theta,\phi)$ is the prior distribution, $f(.)$ is the emulated log likelihood function, also $\mathbb{E}(f(\theta,\phi)|D)$ is the GP posterior predictive mean given the training points D (see eq. (2.66)), and $\sqrt{\text{var}(f(\theta,\phi)|D)}$ is the GP posterior predictive standard deviation (see eq. (2.67)) for the log likelihood of the physiological data at unseen parameter configurations θ,ϕ conditional on the training points D. This drives the exploration into regions with high posterior probability (large value of $\mathbb{E}(.)$) or high uncertainty (large value of $\sqrt{\text{var}(.)}$). If $\sqrt{\text{var}(.)}$>3 along the trajectory, the simulation is stopped prematurely before reaching the end of the trajectory, as the algorithm steps into a region of high uncertainty, where the GP needs to be further trained. The log likelihood is computed at this point by numerically solving the PDEs of the biophysical model. The exploratory phase is run until high accuracy of the emulator is reached (as quantified by GP diagnostics [8]).

- *Sampling phase.* Use the emulator created in the exploratory phase to draw samples from the target distribution using MCMC – see the pseudocode in Algorithm 1g, where the PDE model parameters and the GP neural network hyperparameters (error model parameters) are jointly sampled. At this stage, the emulator and the classifier are no longer updated. We set the emulated approximate log posterior distribution of the MCMC algorithm to

$$\log \tilde{p}(\theta,\phi|\mathbf{y}) \propto \mathbb{E}(f(\theta,\phi)|D) + \log p(\theta,\phi).$$

Note that the numerator in the first term is the expected log likelihood of the data, and the final term is the log prior. The end point of the trajectory is subject to a 2-stage DA Metropolis-Hastings accept/reject step, based on the simulator, see Algorithm 1g. The rejection rate is monitored, and this indicates how well the GP emulator has captured the log posterior density. A large number of rejections calls for an extension of the exploratory phase.

Note that the proposal distribution $q(.|.)$ in Algorithm 1g is that of an Adaptive Metropolis (AM) algorithm [82], i.e. a multivariate normal distribution centred at the current point, with covariance matrix adapted based on the past posterior samples. See Section 2.3.2 for details on the AM algorithm.

B.2 Bayesian hierarchical model

B.2.1 Standard Gibbs sampler

In a hierarchical Bayesian model, the probability of a selected parameter conditional on all the other parameters is equal to the probability of the parameter conditional on its Markov blanket:

$$p(m_s|\text{everything else}) \;=\; p(m_s|\text{Markov blanket of } m_s).$$

Algorithm 1g One iteration of the N-steps ahead Delayed Acceptance GP-MCMC algorithm

```
Define a probabilistic model $p(\mathbf{y}|\theta,\phi)$, where $\mathbf{y}$ are the data, $\theta$ are the PDE model parameters with prior probability $p(\theta)$, and $\phi$ are the error parameters with prior probability $p(\phi)$. Let $\tilde{p}(\mathbf{y}|\theta)$ denote a computationally cheap surrogate model. The corresponding posterior probabilities are given by $p(\theta,\phi|\mathbf{y}) \propto p(\mathbf{y}|\theta,\phi)p(\theta)p(\phi)$ and $\tilde{p}(\theta,\phi|\mathbf{y}) \propto \tilde{p}(\mathbf{y}|\theta,\phi)p(\theta)p(\phi)$.
Define N: number of proposed points before the simulator is called.
for i=2:N do
    Given the current parameter vector $(\theta_{i-1},\phi_{i-1})$, draw new parameters $(\theta_i,\phi_i)$ from the proposal distribution $q(\theta_i,\phi_i|\theta_{i-1},\phi_{i-1})$, and accept the move with the acceptance probability:
    $\alpha_1(\theta_i,\phi_i|\theta_{i-1},\phi_{i-1}) = \min\left(1, \frac{\tilde{p}(\theta_i,\phi_i|\mathbf{y})q(\theta_{i-1},\phi_{i-1}|\theta_i,\phi_i)}{\tilde{p}(\theta_{i-1},\phi_{i-1}|\mathbf{y})q(\theta_i,\phi_i|\theta_{i-1},\phi_{i-1})}\right). \quad (B.1)$
end for
The final proposed parameter vector $(\theta_N,\phi_N)$ is subject to a second stage acceptance probability:
$\alpha_2(\theta_N,\phi_N|\theta_1,\phi_1) = \min\left(1, \frac{p(\theta_N,\phi_N|\mathbf{y})}{p(\theta_1,\phi_1|\mathbf{y})}\frac{\tilde{p}(\theta_1,\phi_1|\mathbf{y})}{\tilde{p}(\theta_N,\phi_N|\mathbf{y})}\right). \quad (B.2)$
```

The Markov blanket is the set of parents, children and co-parents. For m_s, the parents are m^*, σ^{2*}, the children are $\mathbf{s} = (s_1, \ldots, s_d)$, and there is one co-parent: σ_s^2. This gives:

$$p(m_s|\text{everything else}) = p(m_s|m^*,\sigma^{2*},s_1,\ldots,s_d,\sigma_s^2).$$

The conditional probability is proportional to the joint probability:

$$p(m_s|m^*,\sigma^{2*},s_1,\ldots,s_d,\sigma_s^2) \propto p(m_s,m^*,\sigma^{2*},s_1,\ldots,s_d,\sigma_s^2).$$

The joint probability can be factorised according to the factorisation rules for directed graphical models, applied to Figure 4.1:

$$p(m_s,m^*,\sigma^{2*},s_1,\ldots,s_d,\sigma_s^2) = p(m_s|m^*,\sigma^{2*})\prod_{i=1}^{d} p(s_i|m_s,\sigma_s^2).$$

According to Figure 4.1, all distributions on the right-hand side are Gaussian, hence:

$$p(m_s|\text{everything else}) \propto \mathcal{N}(m_s|m^*,\sigma^{2*})\prod_{i=1}^{d}\mathcal{N}(s_i|m_s,\sigma_s^2).$$

Now, one can write out the expressions for the Gaussian, and keep all the terms that explicitly depend on m_s. The other terms are constant with respect to m_s and get absorbed in the normalization constant. This leads to:

$$\begin{aligned}
p(m_s|\text{everything else}) &\propto \mathscr{N}(m_s|m^*, \sigma^{2*}) \prod_{i=1}^{d} \mathscr{N}(s_i|m_s, \sigma_s^2) \\
&= \exp\left(\frac{-1}{2\sigma^{2*}}(m_s - m^*)^2 + \frac{-1}{2\sigma_s^2} \sum_{i=1}^{d} (s_i - m_s)^2 \right) \\
&\propto \exp\left(\frac{-1}{2} m_s^2 \left[\frac{d}{\sigma_s^2} + \frac{1}{\sigma^{2*}} \right] + m_s \left[\frac{m^*}{\sigma^{2*}} + \frac{1}{\sigma_s^2} \sum_{i=1}^{d} s_i \right] \right).
\end{aligned}$$

Completing the square and normalising this distribution yields:

$$p(m_s|\text{everything else}) = \mathscr{N}\left(m_s \middle| \frac{\frac{m^*}{\sigma^{2*}} + \frac{1}{\sigma_s^2} \sum_{i=1}^{d} s_i}{\frac{1}{\sigma^{2*}} + \frac{d}{\sigma_s^2}}, \left[\frac{1}{\sigma^{2*}} + \frac{d}{\sigma_s^2} \right]^{-1} \right). \tag{B.3}$$

In words: the mean of the conditional posterior distrubution is a weighted sum of $m^*, s_1, \ldots, s_d$, weighted by their respective precision. The precision of the conditional posterior distribution is a sum of the individual precisions of $m^*, s_1, \ldots, s_d$. This makes intuitively sense: each of the variables in $\{m^*, s_1, \ldots, s_d\}$ contributes a piece of information whose value is its precision. The overall precision gets higher as you pass information from more variables to m^* (i.e. its inverse, the variance, gets reduced).

For σ_s^2, one can proceed in the same way:

$$p(\sigma_s^2|\text{everything else}) = p(\sigma_s^2|\text{Markov blanket of } \sigma_s^2).$$

Again, the Markov blanket is the set of parents, children and co-parents. For σ_s^2, the parents are α^*, β^*, the children are $\mathbf{s} = (s_1, \ldots, s_d)$, and there is one co-parent: m_s. So the following is obtained:

$$p(\sigma_s^2|\text{everything else}) = p(\sigma_s^2|\alpha^*, \beta^*, s_1, \ldots, s_d, m_s).$$

The conditional probability is proportional to the joint probability:

$$p(\sigma_s^2|\text{everything else}) \propto p(\sigma_s^2, \alpha^*, \beta^*, s_1, \ldots, s_d, m_s).$$

The joint probability can be factorised according to the factorisation rules for directed graphical models, applied to Figure 4.1. Again, only those terms that explicitly depend on σ_s^2 are needed, as the other terms get absorbed in the normalization constant. Since the prior is conjugate, a distribution in closed form (an Inverse Gamma $\mathscr{IG}$ distribution) is obtained, as follows:

$$
\begin{aligned}
p(\sigma_s^2|\text{everything else}) &\propto \mathscr{IG}(\alpha^*,\beta^*)\prod_{i=1}^{d}\mathscr{N}(s_i|m_s,\sigma_s^2)\\
&\propto \left[(\sigma_s^2)^{-\alpha^*-1}\exp\left(-\frac{\beta^*}{\sigma_s^2}\right)\right]\left[\left(\frac{1}{\sigma_s^2}\right)^{\frac{d}{2}}\exp\Big(\frac{-1}{2\sigma_s^2}\sum_{i=1}^{d}(s_i-m_s)^2\Big)\right]\\
&\propto \left[(\sigma_s^2)^{-\left(\alpha^*+\frac{d}{2}\right)-1}\exp\left[-\frac{1}{\sigma_s^2}\Big(\beta^*+0.5\sum_{i=1}^{d}(s_i-m_s)^2\Big)\right]\right].
\end{aligned}
$$

Normalising this distribution yields:

$$
p(\sigma_s^2|\text{everything else}) \quad = \quad \mathscr{IG}\left(\alpha^*+\frac{d}{2},\beta^*+0.5\sum_{i=1}^{d}(s_i-m_s)^2\right). \tag{B.4}
$$

This Gibbs sampler can be used for sampling from the posterior distribution, iteratively sampling from

- $p(m_s|m^*,\sigma^{2*},s_1,\ldots,s_d,\sigma_s^2)$ in eq (B.3),
- $p(\sigma_s^2|\alpha^*,\beta^*,s_1,\ldots,s_d,m_s)$ in eq (B.4),
- $p(\mathbf{s},r_1,r_2,c|m_s,\sigma_s^2,\mathbf{y},\alpha,\beta)$,

where the sampling of $\mathbf{s}=(s_1,\ldots,s_d)$ and r_1,r_2 and c cannot be done analytically and follows a Metropolis-Hastings within Gibbs scheme.

B.2.2 A first attempt at collapsing

One can marginalise over m_s.

$$
\begin{aligned}
p(m_s, \sigma_s^2|\text{everything else}) &\propto \mathscr{I}\mathscr{G}(\sigma_s^2|\alpha^*, \beta^*)\mathscr{N}(m_s|m^*, \sigma^{2*})\prod_{i=1}^{d}\mathscr{N}(s_i|m_s, \sigma_s^2) \\
&\propto \left(\frac{1}{\sigma_s^2}\right)^{\alpha^*+1}\exp\left(\frac{-\beta^*}{\sigma_s^2}\right)\left(\frac{1}{\sigma_s^2}\right)^{d/2} \\
&\quad \exp\left(\frac{-1}{2\sigma^{2*}}(m_s - m^*)^2 + \frac{-1}{2\sigma_s^2}\sum_{i=1}^{d}(s_i - m_s)^2\right) \\
&= \left(\frac{1}{\sigma_s^2}\right)^{\alpha^*+1}\left(\frac{1}{\sigma_s^2}\right)^{d/2}\exp\left(\frac{-\beta^*}{\sigma_s^2}\right)\exp\left(\frac{-1}{2\sigma_s^2}\sum_{i=1}^{d}(s_i - \bar{s})^2\right) \\
&\quad \exp\left(\frac{-1}{2\sigma^{2*}}(m_s - m^*)^2 + \frac{-d}{2\sigma_s^2}(\bar{s} - m_s)^2\right) \\
&\propto \left(\frac{1}{\sigma_s^2}\right)^{\alpha^*+1}\left(\frac{1}{\sigma_s^2}\right)^{(d-1)/2}\exp\left(\frac{-\beta^*}{\sigma_s^2}\right)\exp\left(\frac{-1}{2\sigma_s^2}\sum_{i=1}^{d}(s_i - \bar{s})^2\right) \\
&\quad \mathscr{N}(m_s|m^*, \sigma^{2*})\mathscr{N}\left(\bar{s}|m_s, \frac{\sigma_s^2}{d}\right),
\end{aligned}
$$

where $\bar{s} = \frac{1}{d}\sum_{i=1}^{d} s_i$. m_s can now be integrated using the standard Gaussian integral

$$
\int \mathscr{N}\left(\bar{s}|m_s, \frac{\sigma_s^2}{d}\right)\mathscr{N}(m_s|m^*, \sigma^{2*})dm_s \propto \mathscr{N}\left(\bar{s}|m^*, \sigma^{2*} + \frac{\sigma_s^2}{d}\right).
$$

This gives:

$$
\begin{aligned}
p(\sigma_s^2|\text{everything else except m}_\text{s}) &\propto \left(\frac{1}{\sigma_s^2}\right)^{\alpha^*+1+(d-1)/2}\exp\left(\frac{-\beta^*}{\sigma_s^2}\right)\exp\left(\frac{-1}{2\sigma_s^2}\sum_{i=1}^{d}(s_i - \bar{s})^2\right) \\
&\quad \mathscr{N}\left(\bar{s}|m^*, \sigma^{2*} + \frac{\sigma_s^2}{d}\right) \\
&\propto \left(\frac{1}{\sigma_s^2}\right)^{\alpha^*+1+(d-1)/2}\exp\left(\frac{-\beta^*}{\sigma_s^2}\right)\exp\left(\frac{-1}{2\sigma_s^2}\sum_{i=1}^{d}(s_i - \bar{s})^2\right) \\
&\quad \left(\frac{1}{\sigma^{2*} + \frac{\sigma_s^2}{d}}\right)^{1/2}\exp\left(\frac{-(\bar{s} - m^*)^2)}{2(\sigma^{2*} + \frac{\sigma_s^2}{d})}\right).
\end{aligned}
$$

Hence, m_s can be integrated out analytically to get the marginal distribution in closed form. However, due to the additive term σ^{2*} in the denominator of the last two factors, the distribution is *not* in the family of inverse-gamma distributions, and therefore σ_s^2 cannot be sampled from directly. In order to proceed, one would have to set up a slice sampling or Metropolis-Hastings scheme. This is will almost certainly lose the efficiency gained from collapsing.

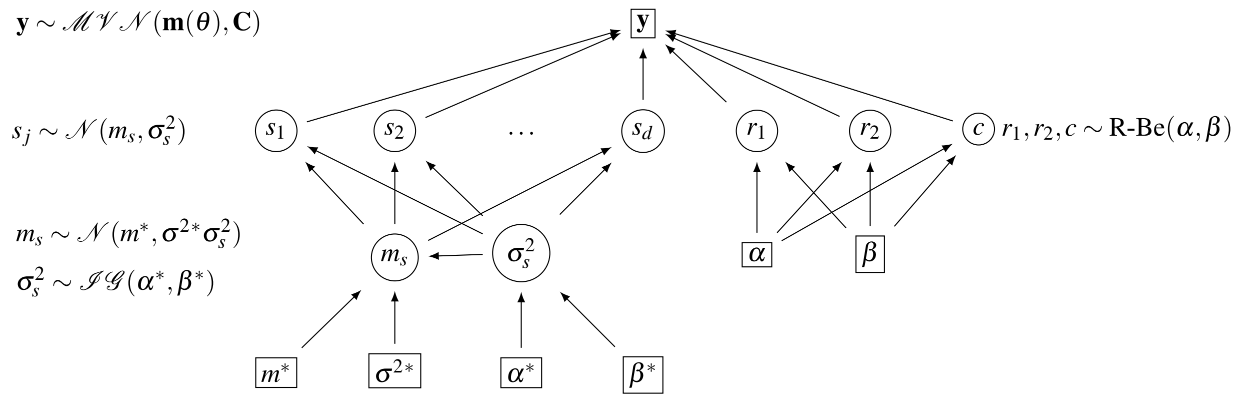

$\mathbf{y} \sim \mathscr{MVN}(\mathbf{m}(\theta), \mathbf{C})$
$\mathbf{y}$
$s_j \sim \mathscr{N}(m_s, \sigma_s^2)$
s_1
s_2
$\dots$
s_d
r_1
r_2
c
$r_1, r_2, c \sim$ R-Be(α, β)
$m_s \sim \mathscr{N}(m^*, \sigma^{2*}\sigma_s^2)$
$\sigma_s^2 \sim \mathscr{IG}(\alpha^*, \beta^*)$
m_s
σ_s^2
α
β
m^*
σ^{2*}
α^*
β^*

Figure B.1: Modified Bayesian Hierarchical model to potentially enable computationally efficient inference, i.e. the prior is modified according to eq (B.5)), which corresponds to an additional edge from σ_s^2 to m_s (to be compared to Figure 4.1). The data, denoted by $\mathbf{y}$, are assumed to follow a multivariate normal distribution $\mathscr{MVN}$ with mean $\mathbf{m}(\theta)$ and covariance matrix $\mathbf{C}$. If iid errors are assumed, $\mathbf{C}$ is a diagonal matrix, $\mathbf{C} = \sigma^2\mathbf{I}$ (where σ^2: error variance and $\mathbf{I}$: identity matrix), and if correlated errors are assumed, $\mathbf{C}$ is a full matrix. The biophysical parameters, $\theta = (s_1, \dots s_d, r_1, r_2, c)$, and the hyperparameters, m_s, σ_s^2, are apriori drawn from the distributions indicated in the graphical model (where $\mathscr{N}$: Normal distribution, $\mathscr{IG}$: Inverse-Gamma, R-Be: rescaled Beta distribution). The circle represents variable quantities, which are inferred using MCMC, and the rectangle stands for fixed quantities.

B.2.3 How to get the collapsed Gibbs sampler to work

Following studies in [4, 162] that use a trick to enable collapsing, the prior can be modified as follows:

$$\mathscr{N}(m_s|m^*, \sigma^{2*}) \;\to\; \mathscr{N}(m_s|m^*, \sigma^{2*}\sigma_s^2). \tag{B.5}$$

Graphically, this corresponds to introducing an additional edge from σ_s^2 to m_s in the hierarchical model, see Figure B.1.

$$\begin{aligned} p(m_s, \sigma_s^2|\text{everything else}) \;\; \propto \;\; & \left(\frac{1}{\sigma_s^2}\right)^{\alpha^*+1} \left(\frac{1}{\sigma_s^2}\right)^{(d-1)/2} \exp\left(\frac{-\beta^*}{\sigma_s^2}\right) \exp\left(\frac{-1}{2\sigma_s^2}\sum_{i=1}^{d}(s_i - \bar{s})^2\right) \\ & \mathscr{N}(m_s|m^*, \sigma^{2*}\sigma_s^2)\mathscr{N}\left(\bar{s}|m_s, \frac{\sigma_s^2}{d}\right). \end{aligned}$$

Integrating out m_s now gives:

$$\int \mathscr{N}\left(\bar{s}|m_s, \frac{\sigma_s^2}{d}\right)\mathscr{N}(m_s|m^*, \sigma^{2*}\sigma_s^2)dm_s \;\propto\; \mathscr{N}\left(\bar{s}|m^*, \sigma_s^2\left[\sigma^{2*} + \frac{1}{d}\right]\right),$$

leading to

$$
\begin{aligned}
& p(\sigma_s^2 | \text{everything else except for } m_s) \\
\propto \; & \left(\frac{1}{\sigma_s^2}\right)^{\alpha^*+1} \left(\frac{1}{\sigma_s^2}\right)^{(d-1)/2} \exp\left(\frac{-\beta^*}{\sigma_s^2}\right) \exp\left(\frac{-1}{2\sigma_s^2}\sum_{i=1}^{d}(s_i-\bar{s})^2\right) \\
& \mathscr{N}\left(\bar{s}|m^*, \sigma_s^2\left[\sigma^{2*}+\frac{1}{d}\right]\right) \\
\propto \; & \left(\frac{1}{\sigma_s^2}\right)^{\alpha^*+1+(d-1)/2} \exp\left(\frac{-\beta^*}{\sigma_s^2}\right) \exp\left(\frac{-1}{2\sigma_s^2}\sum_{i=1}^{d}(s_i-\bar{s})^2\right) \\
& \left(\frac{1}{\sigma_s^2}\right)^{1/2} \exp\left(\frac{-1}{2\sigma_s^2}\left[\sigma^{2*}+\frac{1}{d}\right]^{-1}(\bar{s}-m^*)^2\right) \\
= \; & \left(\frac{1}{\sigma_s^2}\right)^{\alpha^*+1+d/2} \exp\left\{\frac{-1}{\sigma_s^2}\left(\beta^*+\frac{1}{2}\sum_{i=1}^{d}(s_i-\bar{s})^2+\frac{1}{2}\left[\sigma^{2*}+\frac{1}{d}\right]^{-1}(\bar{s}-m^*)^2\right)\right\}.
\end{aligned}
$$

Hence,

$$
\begin{aligned}
p(\sigma_s^2 | \text{everything else except for } m_s) \propto \mathscr{IG}\Bigg(\sigma_s^2 \Bigg| \alpha^*+\frac{d}{2}, \Bigg[\beta^* + \frac{1}{2}\sum_{i=1}^{d}(s_i-\bar{s})^2 \\
+\frac{1}{2}\left[\sigma^{2*}+\frac{1}{d}\right]^{-1}(\bar{s}-m^*)^2\Bigg]\Bigg).
\end{aligned}
\tag{B.6}
$$

This could give a faster and more efficient sampling scheme than the original naive Gibbs sampler, following

- $p(m_s|m^*, \sigma^{2*}, s_1, \ldots, s_d, \sigma_s^2)$ in eq (B.3),
- $p(\sigma_s^2|\alpha^*, \beta^*, s_1, \ldots, s_d, m^*, \sigma^{2*})$ in eq (B.6),
- $p(\mathbf{s}, r_1, r_2, c|m_s, \sigma_s^2, \mathbf{y}, \alpha, \beta)$.

B.2.4 Final improvement: eliminating the need for Gibbs sampling

However, the mixing and convergence of the sampler can be even further improved with another round of collapsing. To simplify the notation, let $\Psi[.]$ denote all the 'prior' random variables (i.e. downstream of the data $\mathbf{y}$) except those included in the bracket. So $p(\sigma_s^2|\text{everything else except for } m_s) = p(\sigma_s^2|\Psi[m_s, \sigma_s^2])$. Here, the additional condition 'downstream of the data $\mathbf{y}$' is redundant, because $\mathbf{y}$ is not included in the Markov blanket of σ_s^2. However, this additional condition does make a difference when looking at the distribution of the vector of stiffness parameters, $\mathbf{s} = (s_1, \ldots, s_d)$. Now, take the previously derived expression for $p(\sigma_s^2|\Psi[m_s, \sigma_s^2])$ in eq (B.6), keep the terms that depend on σ_s^2 and $\mathbf{s} = (s_1, \ldots, s_d)$, ignore all the other terms (because they will be absorbed in the normalization constant), and the following is obtained:

$$p(\sigma_s^2,\mathbf{s}|\Psi[m_s,\sigma_s^2]) \propto \left(\frac{1}{\sigma_s^2}\right)^{\alpha^*+1+d/2} \exp\left\{\frac{-1}{\sigma_s^2}\left(\beta^*+\frac{1}{2}\sum_{i=1}^{d}(s_i-\bar{s})^2+\frac{1}{2}\left[\sigma^{2*}+\frac{1}{d}\right]^{-1}(\bar{s}-m^*)^2\right)\right\}.$$

Now marginalise over σ_s^2,

$$p(\mathbf{s}|\Psi[m_s\,\sigma_s^2]) = \int_0^\infty p(\sigma_s^2,\mathbf{s}|\Psi[m_s,\sigma_s^2])d\sigma_s^2,$$

and make use of the gamma integral

$$\int_0^\infty \left(\frac{1}{x}\right)^{a+1} \exp\left(\frac{-b}{x}\right) dx \;=\; \frac{\Gamma(a)}{b^a}.$$

This leads to

$$p(\mathbf{s}|\Psi[m_s,\sigma_s^2]) \propto \frac{1}{\left(\beta^*+\frac{1}{2}\sum_{i=1}^{d}(s_i-\bar{s})^2+\frac{1}{2}\left[\sigma^{2*}+\frac{1}{d}\right]^{-1}(\bar{s}-m^*)^2\right)^{(\alpha^*+d/2)}}. \tag{B.7}$$

The upshot is that σ_s^2 and m_s have effectively been eliminated altogether. So the corresponding sampling steps of both the naive and the collapsed Gibbs sampler are no longer needed. All that one has to do is run a standard MCMC scheme for $\mathbf{s}$, with the likelihood given by $p(\mathbf{y}|\mathbf{s},r_1,r_2,c)$, the prior of $\{r_1,r_2,c\}$ taken from the hierarchical Bayesian model, and the prior of $\mathbf{s}$ given by eq (B.7) (where the unknown normalisation constant cancels out in the Metropolis-Hastings ratio).

B.3 Bayesian Hierarchical model for the stiffness – prior distribution

In the Bayesian Hierarchical model the hyper-hyperparameters $(m_s,\sigma_s^2,\alpha^*,\beta^*)$ take fixed values, set as follows. The stiffness mean m_s was chosen to be within the physiological range via the fixed hyperparameters m^* and σ^{2*}. The stiffness variance σ_s^2, as chosen via the fixed hyperparameters α^*,β^*, controls the spread around m_s, thus a 90% probability of the stiffness being inside the physiological range is controlled by σ_s^2.

For the iid errors analysis, $m^*=51690$ (which is the posterior mean value obtained from the MCMC simulation for the constant stiffness model under the iid error assumption, see Table 4.5), $\sigma^{2*}=1e+08$, $\alpha^*=3$, and $\beta^*=1e+09$.

For the correlated errors analysis, we set $m^*=43075$ (which is the posterior mean value obtained from the MCMC simulation for the constant stiffness model under the correlation errors assumption, see Table 4.5), $\sigma^{2*} = 2.5e+08$, $\alpha^* = 3$, and $\beta^* = 0.6e+09$.

B.4 Error correlation parameters – prior ranges

A log uniform prior distribution for the GP neural network hyperparameters (used in the error correlation analysis) was used, with the range chosen based on maximising the profile log likelihood. The log uniform distribution was chosen to ensure a sufficiently large prior coverage. A GP was fitted to the residuals obtained by a difference between the measured data and the data generated from the mathematical model with the MAP estimate from the iid errors analysis. The profile log likelihood was obtained for each hyperparameter, i.e. by varying one hyperparameter at once (within a wide range), and the marginal log likelihood in eq (**??**) was maximised with respect to the other kernel hyperparameter. The maximised marginal log likelihood in eq (**??**) (called profile log likelihood) was plotted against the varying hyperparameter. This plot informed of the range that the varying hyperparameter should take. The range should cover a concave-looking profile log likelihood curve – the maximum of this curve is the maximum profile log likehood value [104], which should be included in the range; hyperparameter values which gave a small profile log likelihood value relative to the maximum value, e.g. below 10% of the maximum value, were excluded from the range.

B.5 Log likelihood GP kernel

For the GP regression model of the log likelihood, a squared exponential kernel was used, as chosen based on the data via 1-fold cross-validation with a score based on the absolute value of the deviation of the data $\mathbf{y}$ from the GP predictions $\hat{\mathbf{y}}$, i.e. $|\mathbf{y} - \hat{\mathbf{y}}|$. One log likelihood data point at a time was left out from the training set of the emulator, the log marginal likelihood was optimised with respect to the hyperparameters, the GP model thus obtained was used to predict the held-out log likelihood point, and the prediction was compared to the actual held-out point. The kernel which recorded the smallest deviation was the best.

B.6 Additional results

While in Section 4.6 results for a few blood vessels are shown, here the complete results for all 21 blood vessels are attached.

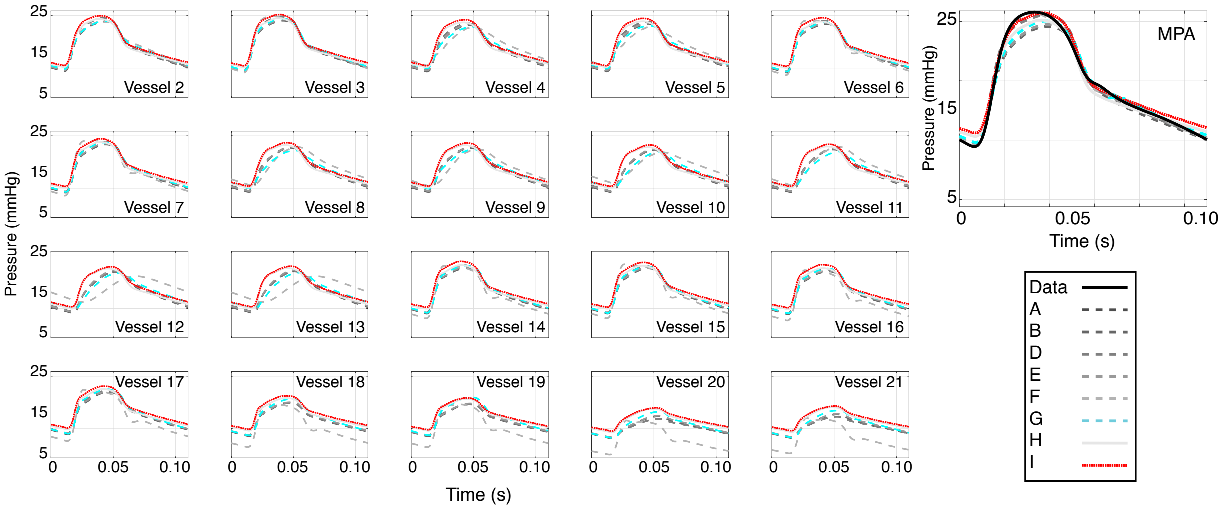


Figure B.2: Pressure predictions obtained using the MCMC posterior samples for the parameters from all the models described in Chapter 4, which are denoted by A-I in the figure legend. The median pressure signal for 21 blood vessels in time is shown. The measured pressure data in the MPA is superimposed. This figure corresponds to Figure 4.7 in Chapter 4.

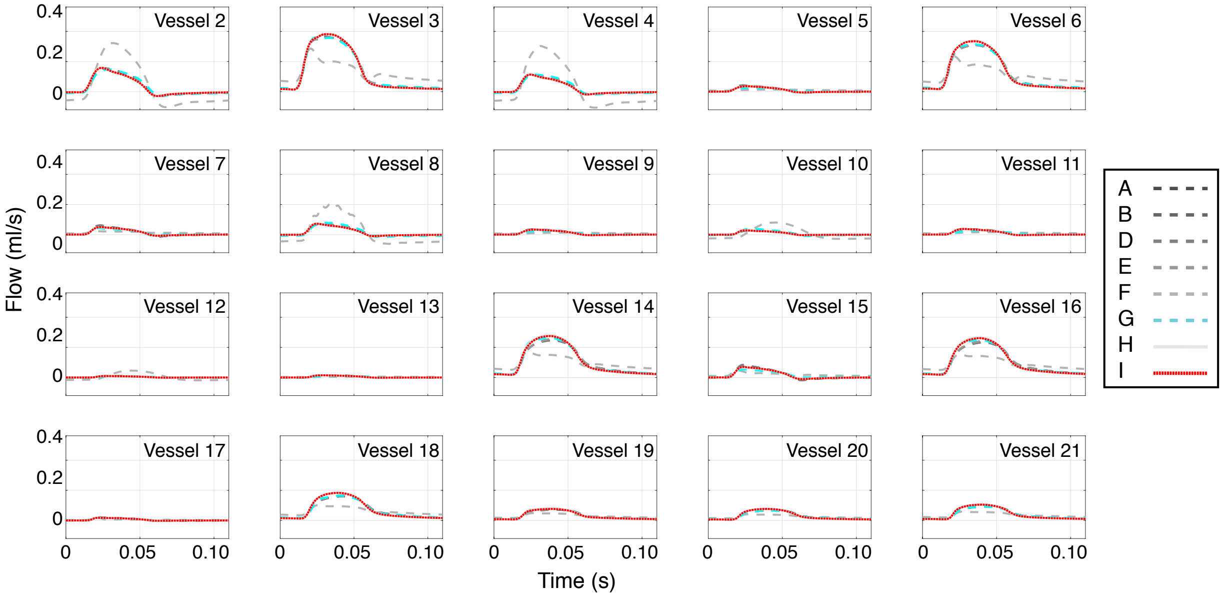


Figure B.3: Flow predictions obtained using the MCMC posterior samples for the parameters from all the models described in Chapter 4, which are denoted by A-I in the figure legend. The median flow signal for blood vessels 2-21 in time is shown. The MPA flow is used as inflow boundary condition for the PDEs. This figure corresponds to the left side subplots in Figure 4.8 in Chapter 4.

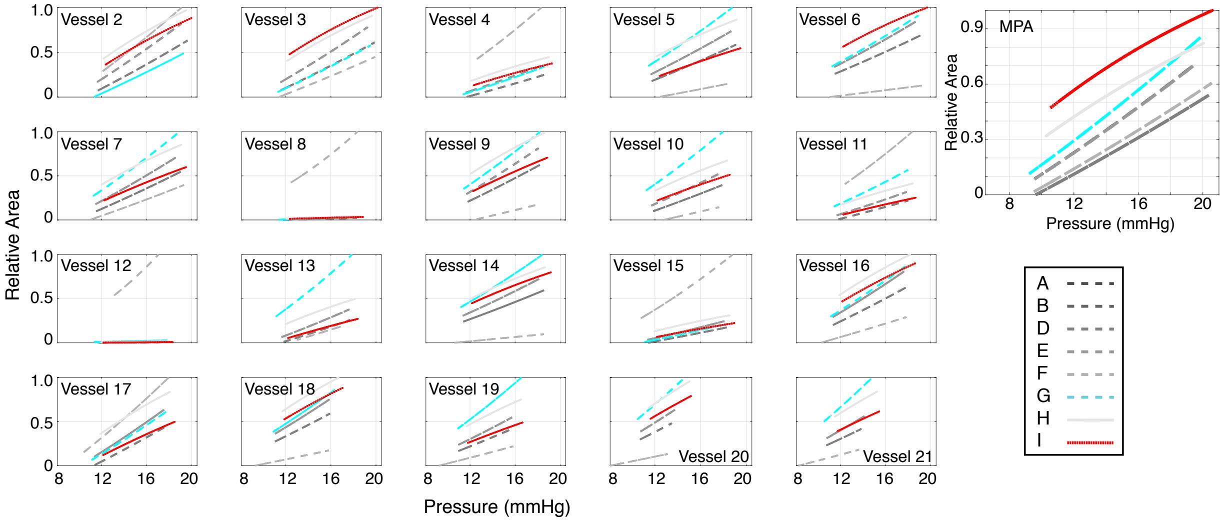


Figure B.4: Pressure-Area predictions obtained using the MCMC posterior sample for the parameters from all the models described in Chapter 4, which are denoted by A-I in the figure legend. The median pressure prediction versus standardised cross-sectional area predictions for all 21 blood vessels are shown. The area, A_i is standardised per vessel i to lie between [0,1] using the expression: $\frac{A_i - l_i}{u_i - l_i}$, where l_i, u_i are the maximum and minimum area value for vessel i. $\mathbf{l}$ = [0.010, 0.003, 0.006, 0.003, 0.001, 0.004, 0.001, 0.002, 0.001, 0.002, 0.001, 0.001, 0.001, 0.002, 0.002, 0.002, 0.001, 0.002, 0.001, 0.002, 0.001]; $\mathbf{u}$ = [0.017, 0.005, 0.010, 0.006, 0.001, 0.007, 0.002, 0.032, 0.002, 0.003, 0.003, 0.035, 0.002, 0.005, 0.004, 0.004, 0.002, 0.004, 0.002, 0.003, 0.002]. This figure corresponds to the right side subplots in Figure 4.8 in Chapter 4.

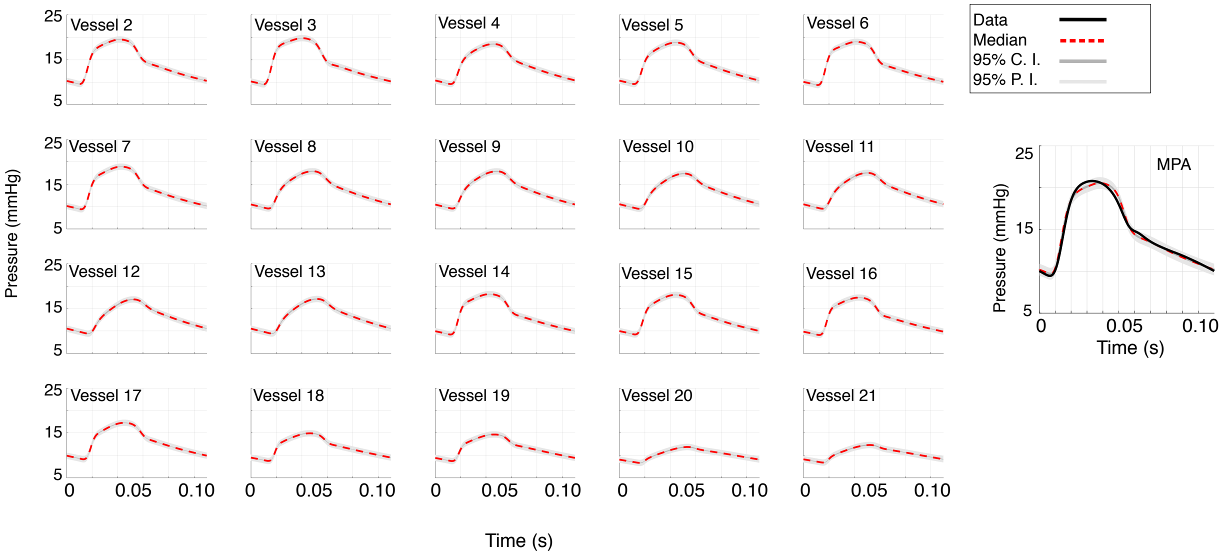


Figure B.5: 95% credible intervals (C.I.) and prediction intervals (P.I.) for the pressure prediction in 21 vessels from the linear model with constant stiffness and no model mismatch (model A in Table 4.2) obtained from MCMC posterior samples. The measured pressure data in the MPA and the median prediction are superimposed. This figure corresponds to the left column subplots in Figure 4.4 in Chapter 4.

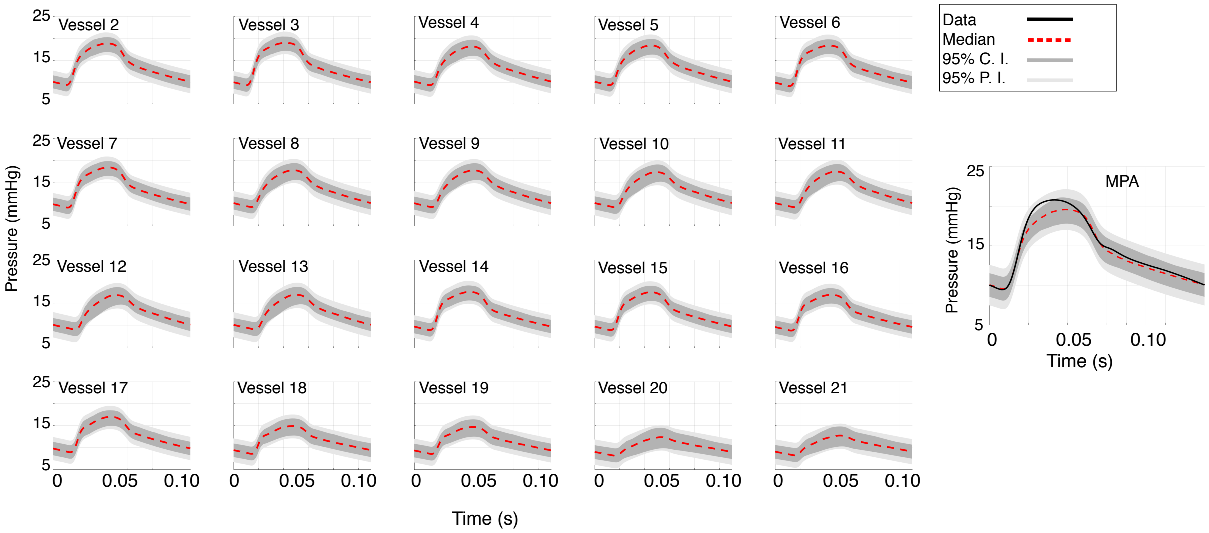


Figure B.6: 95% credible intervals (C.I.) and prediction intervals (P.I.) for the pressure prediction in 21 vessels from the linear model with constant stiffness and model mismatch (model B in Table 4.2) obtained from MCMC posterior samples. The measured pressure data in the MPA and the median prediction are superimposed. This figure corresponds to the centre column subplots in Figure 4.4 in Chapter 4.

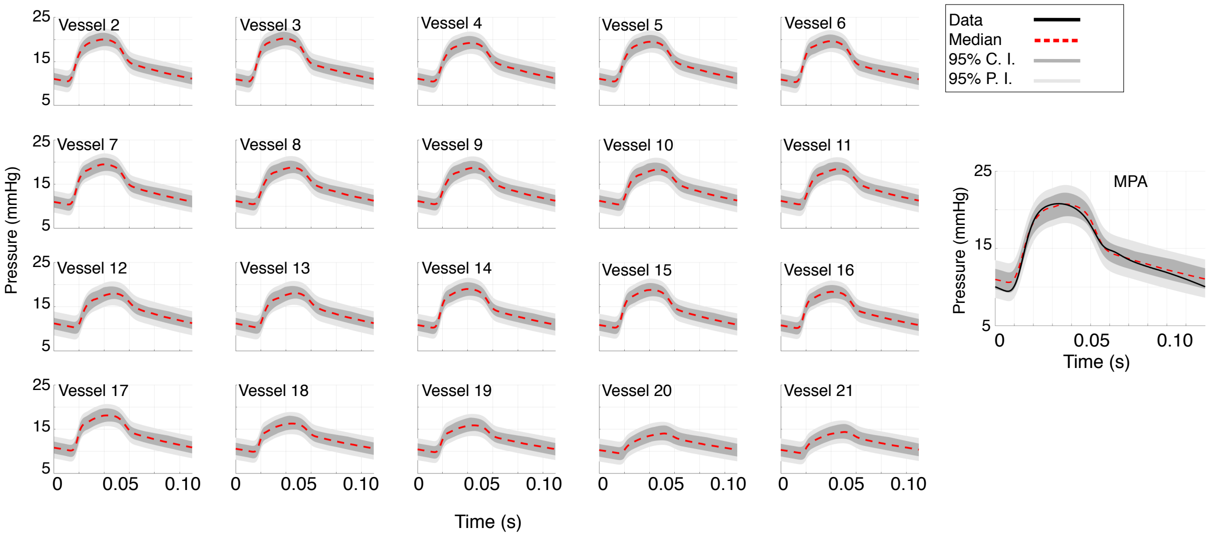


Figure B.7: 95% credible intervals (C.I.) and prediction intervals (P.I.) for the pressure prediction in 21 vessels from the non-linear model with radius-dependent stiffness and model mismatch (model I in Table 4.2) obtained from MCMC posterior samples. The measured pressure data in the MPA and the median prediction are superimposed. This figure corresponds to the right column subplots in Figure 4.4 in Chapter 4.

Vessel	Radius	Model A	Model B	Model C	Model D	Model E	Model F	Model G	Model H	Model I
1	0.05	14.7 (14.6 14.7) (14.0 15.4)	14.2 (12.6 15.7) (11.7 16.7)	14.0 (12.7 15.4) (11.7 16.4)	14.7 (14.6 14.8) (14.0 15.4)	14.4 (12.7 16.0) (11.9 16.8)	14.7 (14.6 14.7) (14.3 15.1)	14.6 (13.0 15.9) (12.1 17.0)	14.5 (13.2 16.1) (12.1 16.9)	15.0 (13.7 16.4) (12.7 17.4)
2	0.03	14.3 (14.2 14.3) (13.5 15.0)	13.8 (12.2 15.4) (11.3 16.3)	13.7 (12.3 15.0) (11.3 16.1)	14.3 (14.2 14.3) (13.6 15.0)	14.0 (12.3 15.6) (11.5 16.5)	14.4 (14.3 14.5) (14.0 14.8)	14.2 (12.5 15.6) (11.7 16.6)	14.2 (12.8 15.9) (11.8 16.6)	14.7 (13.4 16.1) (12.3 17.1)
3	0.04	14.3 (14.2 14.3) (13.6 15.0)	13.9 (12.3 15.4) (11.3 16.3)	13.7 (12.3 15.0) (11.3 16.1)	14.3 (14.2 14.3) (13.6 15.0)	14.0 (12.4 15.7) (11.5 16.5)	14.2 (14.2 14.3) (13.8 14.7)	14.2 (12.6 15.5) (11.7 16.7)	14.2 (12.9 15.9) (11.8 16.6)	14.7 (13.4 16.1) (12.4 17.1)
4	0.02	13.9 (13.8 13.9) (13.1 14.6)	13.5 (11.9 15.1) (11.0 16.0)	13.3 (11.9 14.7) (11.0 15.8)	13.9 (13.8 13.9) (13.2 14.6)	13.7 (12.0 15.3) (11.1 16.2)	14.2 (14.1 14.3) (13.7 14.6)	13.8 (11.9 15.4) (11.1 16.3)	14.0 (12.6 15.6) (11.6 16.4)	14.4 (13.1 15.8) (12.0 16.8)
5	0.01	14.0 (13.9 14.0) (13.3 14.7)	13.6 (12.0 15.2) (11.1 16.1)	13.4 (12.0 14.8) (11.1 15.9)	14.0 (13.9 14.0) (13.3 14.7)	13.8 (12.1 15.4) (11.2 16.3)	14.2 (14.1 14.3) (13.8 14.6)	13.9 (12.1 15.4) (11.3 16.4)	14.0 (12.7 15.7) (11.6 16.5)	14.5 (13.2 15.9) (12.1 16.9)
6	0.03	13.9 (13.8 13.9) (13.1 14.6)	13.5 (11.9 15.0) (11.0 16.0)	13.3 (11.9 14.7) (11.0 15.7)	13.8 (13.8 13.9) (13.1 14.5)	13.7 (12.0 15.3) (11.1 16.1)	13.6 (13.5 13.7) (13.2 14.0)	13.8 (12.1 15.1) (11.3 16.3)	13.9 (12.5 15.6) (11.5 16.3)	14.5 (13.1 15.8) (12.1 16.9)
7	0.02	13.8 (13.8 13.9) (13.1 14.5)	13.5 (11.9 15.0) (10.9 16.0)	13.3 (11.9 14.7) (10.9 15.7)	13.8 (13.8 13.9) (13.1 14.5)	13.7 (12.0 15.3) (11.1 16.1)	13.6 (13.5 13.7) (13.2 14.1)	13.8 (12.1 15.1) (11.3 16.3)	13.9 (12.5 15.6) (11.5 16.3)	14.4 (13.1 15.8) (12.0 16.8)
8	0.02	13.6 (13.5 13.6) (12.9 14.3)	13.3 (11.6 14.9) (10.7 15.8)	13.1 (11.6 14.5) (10.7 15.5)	13.6 (13.5 13.6) (12.9 14.3)	13.4 (11.7 15.1) (10.9 15.9)	14.1 (14.0 14.2) (13.7 14.5)	13.6 (11.4 15.2) (10.8 16.2)	13.8 (12.4 15.5) (11.4 16.2)	14.2 (12.8 15.6) (11.8 16.6)
9	0.02	13.5 (13.5 13.6) (12.8 14.3)	13.3 (11.5 14.9) (10.7 15.8)	13.1 (11.6 14.5) (10.7 15.5)	13.5 (13.5 13.6) (12.8 14.2)	13.4 (11.7 15.1) (10.8 15.9)	13.8 (13.7 14.0) (13.4 14.3)	13.6 (11.6 15.2) (10.8 16.2)	13.8 (12.3 15.5) (11.3 16.2)	14.2 (12.8 15.6) (11.8 16.6)
10	0.02	13.3 (13.2 13.4) (12.6 14.0)	13.0 (11.3 14.7) (10.4 15.6)	12.9 (11.4 14.3) (10.4 15.3)	13.3 (13.2 13.4) (12.6 14.0)	13.2 (11.5 14.9) (10.6 15.7)	14.0 (13.8 14.2) (13.6 14.5)	13.4 (10.9 15.1) (10.4 16.1)	13.6 (12.2 15.3) (11.2 16.1)	14.0 (12.6 15.5) (11.6 16.4)
11	0.02	13.4 (13.3 13.4) (12.7 14.1)	13.1 (11.4 14.7) (10.5 15.6)	12.9 (11.4 14.4) (10.5 15.4)	13.4 (13.3 13.5) (12.7 14.1)	13.3 (11.5 15.0) (10.7 15.8)	14.1 (14.0 14.2) (13.7 14.5)	13.4 (10.9 15.2) (10.5 16.1)	13.7 (12.2 15.4) (11.2 16.1)	14.1 (12.7 15.5) (11.7 16.5)
12	0.02	13.1 (13.0 13.2) (12.4 13.8)	12.9 (11.1 14.6) (10.3 15.5)	12.7 (11.2 14.2) (10.3 15.2)	13.1 (13.0 13.2) (12.4 13.8)	13.1 (11.3 14.8) (10.5 15.6)	13.8 (13.5 14.2) (13.3 14.4)	13.3 (10.7 15.0) (10.3 16.0)	13.5 (12.0 15.2) (11.1 16.0)	13.9 (12.5 15.3) (11.5 16.3)
13	0.01	13.2 (13.1 13.3) (12.5 13.9)	12.9 (11.2 14.6) (10.3 15.5)	12.8 (11.2 14.2) (10.3 15.2)	13.2 (13.1 13.3) (12.5 13.9)	13.1 (11.3 14.8) (10.5 15.7)	13.8 (13.5 14.2) (13.3 14.4)	13.3 (10.7 15.1) (10.3 16.1)	13.5 (12.1 15.3) (11.1 16.0)	13.9 (12.6 15.4) (11.5 16.4)
14	0.03	13.4 (13.4 13.5) (12.7 14.1)	13.1 (11.5 14.7) (10.6 15.6)	13.0 (11.6 14.3) (10.6 15.4)	13.4 (13.4 13.5) (12.7 14.1)	13.3 (11.6 15.0) (10.8 15.8)	12.8 (12.6 13.0) (12.3 13.3)	13.5 (11.8 14.9) (11.0 15.9)	13.7 (12.3 15.3) (11.2 16.1)	14.2 (12.8 15.6) (11.8 16.6)
15	0.02	13.4 (13.3 13.4) (12.7 14.1)	13.1 (11.4 14.6) (10.5 15.6)	12.9 (11.5 14.3) (10.5 15.3)	13.4 (13.3 13.4) (12.7 14.1)	13.3 (11.6 14.9) (10.7 15.7)	13.0 (12.8 13.1) (12.5 13.4)	13.4 (11.7 14.8) (10.9 15.9)	13.6 (12.2 15.3) (11.2 16.0)	14.1 (12.7 15.5) (11.7 16.5)
16	0.03	13.1 (13.0 13.1) (12.3 13.8)	12.8 (11.1 14.4) (10.2 15.3)	12.6 (11.2 14.0) (10.2 15.1)	13.0 (13.0 13.1) (12.3 13.7)	13.0 (11.2 14.6) (10.4 15.4)	12.2 (12.0 12.4) (11.7 12.6)	13.2 (11.3 14.6) (10.6 15.7)	13.4 (12.0 15.1) (11.0 15.8)	13.9 (12.5 15.3) (11.5 16.3)
17	0.02	13.0 (12.9 13.1) (12.3 13.7)	12.8 (11.1 14.3) (10.2 15.3)	12.6 (11.2 14.0) (10.2 15.0)	13.0 (12.9 13.1) (12.3 13.7)	12.9 (11.2 14.6) (10.4 15.4)	12.2 (11.8 12.4) (11.6 12.7)	13.1 (11.3 14.6) (10.5 15.6)	13.4 (11.9 15.1) (10.9 15.8)	13.8 (12.4 15.2) (11.4 16.2)
18	0.02	11.8 (11.7 11.9) (11.1 12.5)	11.7 (9.9 13.3) (9.1 14.2)	11.5 (10.1 13.0) (9.1 14.0)	11.8 (11.7 11.9) (11.1 12.5)	11.9 (10.0 13.6) (9.3 14.4)	10.3 (10.0 10.7) (9.7 10.9)	12.1 (10.0 13.9) (9.3 14.8)	12.5 (11.0 14.3) (10.1 15.0)	12.9 (11.5 14.4) (10.5 15.4)
19	0.02	11.6 (11.6 11.7) (10.9 12.4)	11.6 (9.8 13.2) (9.0 14.1)	11.4 (10.0 12.9) (9.0 13.9)	11.6 (11.6 11.7) (10.9 12.3)	11.8 (9.9 13.5) (9.1 14.3)	10.4 (9.9 10.7) (9.7 11.0)	12.2 (10.0 14.0) (9.4 14.9)	12.4 (10.9 14.2) (10.0 14.9)	12.7 (11.3 14.2) (10.3 15.1)
20	0.02	10.2 (10.1 10.3) (9.5 10.9)	10.3 (8.4 12.1) (7.6 12.9)	10.1 (8.6 11.7) (7.7 12.7)	10.2 (10.1 10.3) (9.5 10.9)	10.5 (8.5 12.3) (7.8 13.1)	8.1 (7.5 8.8) (7.2 8.9)	10.7 (8.3 13.2) (7.5 14.0)	11.5 (9.8 13.4) (9.0 14.0)	11.8 (10.3 13.3) (9.4 14.3)
21	0.02	10.4 (10.4 10.5) (9.7 11.1)	10.5 (8.6 12.3) (7.8 13.1)	10.3 (8.8 11.9) (7.9 12.9)	10.4 (10.4 10.5) (9.7 11.1)	10.7 (8.8 12.5) (8.0 13.3)	8.4 (7.8 9.1) (7.6 9.2)	10.9 (8.5 13.3) (7.7 14.1)	11.7 (10.0 13.5) (9.2 14.2)	12.0 (10.5 13.5) (9.5 14.5)

Table B.1: Summary of the MCMC simulation results on measured data for each of the models considered, see Table 4.2 also for a summary. For each of the 21 blood vessels the average value over time of the median pressure waveform is shown, as well as the average value over time of the 2.5th and 97.5th noise-free pressure waveform, which is the average 95% explanatory credible interval (CI) for the pressure data, and the 2.5th and 97.5th noisy pressure waveform, which is the average 95% predictive CI for the pressure data. While the explanatory CI is calculated based on the PDE model predictions, the predictive CI includes the error.

Vessel-specific stiffness values The vessel-specific stiffness values (median value and 95% credible interval) are provided here, which were obtained from running the Bayesian Hierarchical model in Chapter 4 on the physiological data. These values complement Table 4.5 (rows 3 and 4 from bottom to top, column 5 from left to right, marked by '*').

For iid errors, $f_3 : (\times 10^4)$ 4.93 (4.71, 5.10), 3.84 (3.61, 4.15), 6.10 (5.53, 7.10), 2.34 (2.10, 2.64), 10.4 (2.51, 18.1), 16.0 (9.16, 23.2), 6.27 (3.45, 10.9), 0.53 (0.51, 0.55), 9.95 (1.95, 17.00), 7.98 (6.36, 11.4), 2.29 (0.59, 7.00), 0.27 (0.23, 0.30), 5.54 (1.19, 14.36), 17.8 (9.77, 24.0), 1.94 (1.46, 2.85), 8.16 (4.67, 13.4), 3.47 (2.17, 11.3), 11.4 (6.30, 22.4), 7.63 (4.08, 12.5), 11.6 (5.51, 20.0), 8.44 (2.61,17.3).

For correlated errors, $f_3 : (\times 10^4)$ 3.96 (3.57, 4.24), 5.76 (3.55, 9.32), 5.22 (4.27, 6.77), 4.46 (1.55, 12.9), 3.51 (0.52, 9.11), 4.13 (2.77, 6.01), 3.56 (1.86, 8.49), 4.47 (1.22, 12.24), 3.79 (0.73, 9.93), 2.94 (0.12, 11.9), 3.17 (0.26, 11.3), 2.62 (0.35, 11.2), 2.52 (0.23, 7.40), 3.43 (1.79, 8.15), 4.70 (2.28, 11.3), 4.15 (2.04, 11.6), 4.45 (2.21, 10.7), 4.05 (2.10, 12.9), 3.03 (1.23, 10.4), 3.30 (0.58, 10.6), 3.02 (0.41, 8.66).

The posterior median and 95% credible interval for the hyperpriors, m_s and σ_s^2 are also provided here. For iid errors, $m_s : (\times 10^3)$ 4.29 (-0.31, 10.9), and $\sigma_s^2 : (\times 10^9)$ 5.81 (2.80, 13.9). For correlated errors, $m_s : (\times 10^3)$ 0.79 (-1.40, 3.22), and $\sigma_s^2 : (\times 10^9)$ 1.8 (0.88, 4.10).

Emulator efficiency Table B.2 compares the efficiency of the standard MCMC sampler to that of MCMC with emulation using GPs (N-steps ahead Adaptive Metropolis with emulation, see Algorithm 1g), for the linear wall model with constant vessel stiffness (models B and C in Table 4.2). Efficiency is quantified using the effective sample size (ESS) [95], and it can be seen that while the emulation method requires much fewer PDE evaluations (compare 5000 to 150,000), it also registers a much higher acceptance rate (compare 89% to 24%) and highly increased efficiency, as given by the median ESS (across all parameters) normalised by the number of PDEs evaluated (compare 0.35 to 0.04).

Emulator	Acceptance rate	$\frac{\text{Median ESS}}{\text{no of PDEs}}$
yes	89%	0.35
no	24%	0.04

Table B.2: Comparison of efficiency for models B and C in Table 4.2 obtained with standard MCMC (model C) and MCMC with emulation – N-steps ahead Adaptive Metropolis with emulation, see Algorithm 1g (model B). Results for model B are based on 5000 iterations (i.e. PDE evaluations), and 150,000 for model C. The acceptance rate and the median ESS (across all parameters) normalised by the number of PDEs evaluated are shown.

Appendix C

Appendix for Chapter 5

C.1 Cross-validation results

To select the kernel for the noise-dependent GP model employed in Chapter 5, cross-validation was performed. One input-output data point $(x_i, y_i),\ i = 1 \ldots n$ (where x: standardised vessel radius or length defined in eq (5.1) and y: coefficient of variation, CV defined in eq (5.20)) at a time was removed from the data set, a GP noise-dependent model was fitted to the rest of the data points, and the GP model thus fitted was used to predict the data point y_i left out. The log predictive density (given in eq (C.1)) of each held-out point was recorded, and the distribution of the log predictive densities over all points was constructed. This process was repeated for four GP models, each GP employing different kernels: squared exponential, Matérn 3/2, Matérn 5/2 and neural network. The four distributions for the radius measurements are shown in Figure C.1. The GP kernel giving the highest log predictive densities is preferred. It can be noticed that the performances of the squared exponential, Matérn 3/2 and Matérn 5/2 kernels are similar, and they appear better than that of the neural network. Hypothesis testing reveals no significant difference between the stationary kernels, hence Matérn 5/2 was arbitrarily selected. For consistency, the same kernel was used for the length measurements.

$$\log p(y_i|\mathbf{y}_{-i}) = -\log\left(\sqrt{2\pi k_p(x_i,x_i)}\right) - \frac{(y_i - m_p(x_i))^2}{2k_p(x_i,x_i)}, \tag{C.1}$$

where x_i, is the input point excluded, y_i is the output point excluded, $\mathbf{y}_{-i}$ is the data set with the i^{th} point removed, and the predictive distribution is given by:

$$y_i \sim \mathcal{N}(m_p(x_i), k_p(x_i, x_i^{'})). \tag{C.2}$$

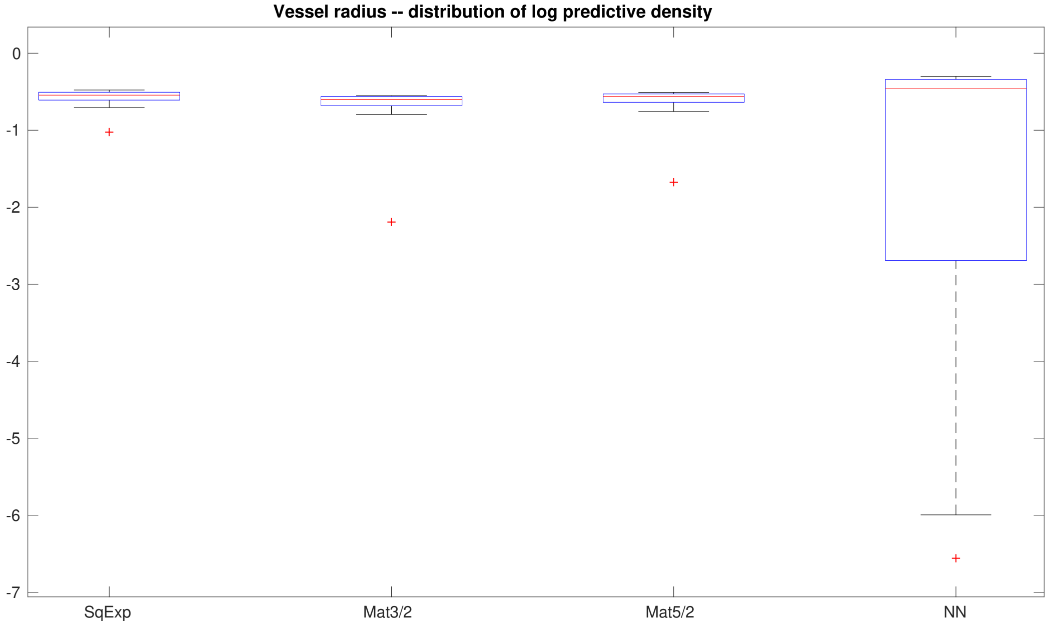


Figure C.1: Cross validation results for the vessel radius measurements. One radius measurement point (input) and the corresponding coefficient of variation point (output) was removed at a time from the data set, the GP noise-dependent model was fitted to the rest of the (input, output) points, and the GP model thus fitted was used to predict the coefficient of variation point for the radius measurement left out. The log predictive density (given in eq (C.1)) for each held-out point was recorded, and the distribution of the log predictive densities for all points is shown here. This process was repeated for four GP models, each GP employing a different kernel: squared exponential (SqExp), Matérn 3/2 (Mat3/2), Matérn 5/2 (Mat5/2) and neural network (NN).

Appendix D

Appendix for Chapter 6

D.1 GP derivatives of predictive mean and variance

Derivatives of the predictive equations with respect to a new input $\tilde{\mathbf{x}}$ may be needed if GPs are coupled with methods requiring knowledge of the derivatives, e.g. with Hamiltonian/Lagrangian Monte Carlo algorithms [156]. These derivatives can be computed analytically.

D.1.1 Zero mean GP

The first order partial derivatives of the posterior predictive mean and variance in eqns (2.66) and (2.67) with respect to a new input $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots \tilde{x}_d)$ are:

$$\frac{\partial m_{\mathrm{p}}(\tilde{\mathbf{x}})}{\partial \tilde{x}_j} = \frac{\partial k(\tilde{\mathbf{x}}, \mathbf{X}|\gamma)}{\partial \tilde{x}_j}(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{y}, \tag{D.1}$$

$$\frac{\partial k_{\mathrm{p}}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}|\gamma)}{\partial \tilde{x}_j} = \frac{\partial k(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}|\gamma)}{\partial \tilde{x}_j} - \frac{\partial k(\tilde{\mathbf{x}}, \mathbf{X}|\gamma)}{\partial \tilde{x}_j}(\mathbf{K}+\sigma^2\mathbf{I})^{-1}k(\mathbf{X}, \tilde{\mathbf{x}}|\gamma) - k(\tilde{\mathbf{x}}, \mathbf{X}|\gamma)(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\frac{\partial k(\mathbf{X}, \tilde{\mathbf{x}}|\gamma)}{\partial \tilde{x}_j}, \tag{D.2}$$

where

$$\left(\frac{\partial k(\tilde{\mathbf{x}}, \mathbf{X}|\gamma)}{\partial \tilde{x}_j}\right)^{\mathrm{T}} = \begin{bmatrix} \frac{\partial k(\tilde{\mathbf{x}}, \mathbf{x}_1|\gamma)}{\partial \tilde{x}_j} \\ \vdots \\ \frac{\partial k(\tilde{\mathbf{x}}, \mathbf{x}_n|\gamma)}{\partial \tilde{x}_j} \end{bmatrix} \quad \text{and} \quad \frac{\partial k(\mathbf{X}, \tilde{\mathbf{x}}|\gamma)}{\partial \tilde{x}_j} = \begin{bmatrix} \frac{\partial k(\mathbf{x}_1, \tilde{\mathbf{x}}|\gamma)}{\partial \tilde{x}_j} \\ \vdots \\ \frac{\partial k(\mathbf{x}_n, \tilde{\mathbf{x}}|\gamma)}{\partial \tilde{x}_j} \end{bmatrix}. \tag{D.3}$$

In addition, the second and third order derivatives of the posterior predictive mean, needed for the RMHMC or LDMC algorithms applied in the sampling phase of the GP HMC algorithm, are:

$$\frac{\partial^2 m_{\mathrm{p}}(\tilde{\mathbf{x}})}{\partial \tilde{x}_k \partial \tilde{x}_j} = \frac{\partial^2 k(\tilde{\mathbf{x}}, \mathbf{X}|\gamma)}{\partial \tilde{x}_k \partial \tilde{x}_j} (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}, \tag{D.4}$$

$$\frac{\partial^3 m_{\mathrm{p}}(\tilde{\mathbf{x}})}{\partial \tilde{x}_l \partial \tilde{x}_k \partial \tilde{x}_j} = \frac{\partial^3 k(\tilde{\mathbf{x}}, \mathbf{X}|\gamma)}{\partial \tilde{x}_l \partial \tilde{x}_k \partial \tilde{x}_j} (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}. \tag{D.5}$$

D.1.2 Mean functions GP

Derivatives of the predictive equations (2.74) and (2.75) with respect to a new input $\tilde{\mathbf{x}}$ make use of the derivatives of the zero mean GP predictive equations (see eqns (D.1) and (D.2)) as follows:

$$\frac{\partial m^*_{\mathrm{p}}(\tilde{\mathbf{x}})}{\partial \tilde{x}_j} = \frac{\partial m_{\mathrm{p}}(\tilde{\mathbf{x}})}{\partial \tilde{x}_j} + \left(\frac{\partial \mathbf{R}^{\mathrm{T}}}{\partial \tilde{x}_j} \right) \bar{\beta}, \tag{D.6}$$

$$\begin{split} \frac{\partial k^*_{\mathrm{p}}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}|\gamma)}{\partial \tilde{x}_j} = \frac{\partial k_{\mathrm{p}}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}|\gamma)}{\partial \tilde{x}_j} + \left(\frac{\partial \mathbf{R}^{\mathrm{T}}}{\partial \tilde{x}_j} \right) (\mathbf{B}^{-1} + \mathbf{H}(\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{H}^{\mathrm{T}})^{-1} \mathbf{R} + \\ \mathbf{R}^{\mathrm{T}} (\mathbf{B}^{-1} + \mathbf{H}(\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{H}^{\mathrm{T}})^{-1} \left(\frac{\partial \mathbf{R}}{\partial \tilde{x}_j} \right), \end{split} \tag{D.7}$$

$$\frac{\partial \mathbf{R}}{\partial \tilde{x}_j} = \frac{\partial \mathbf{h}(\tilde{\mathbf{x}})}{\partial \tilde{x}_j} - \mathbf{H}(\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \frac{\partial k(\mathbf{X}, \tilde{\mathbf{x}})}{\partial \tilde{x}_j}. \tag{D.8}$$

In addition, the second and third order derivatives of the posterior predictive mean are needed for the RMHMC or LDMC algorithms applied in the sampling phase of GP HMC algorithm. These derivatives make use of the derivatives of the zero mean GP (see eqns (D.4) and (D.5)), and are given by:

$$\frac{\partial^2 m^*_{\mathrm{p}}(\tilde{\mathbf{x}})}{\partial \tilde{x}_k \partial \tilde{x}_j} = \frac{\partial^2 m_{\mathrm{p}}(\tilde{\mathbf{x}})}{\partial \tilde{x}_k \partial \tilde{x}_j} + \left(\frac{\partial^2 \mathbf{R}^{\mathrm{T}}}{\partial \tilde{x}_k \partial \tilde{x}_j} \right) \bar{\beta}, \tag{D.9}$$

$$\frac{\partial^2 \mathbf{R}}{\partial \tilde{x}_k \partial \tilde{x}_j} = \frac{\partial^2 \mathbf{h}(\tilde{\mathbf{x}})}{\partial \tilde{x}_k \partial \tilde{x}_j} - \mathbf{H}(\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \frac{\partial^2 k(\mathbf{X}, \tilde{\mathbf{x}})}{\partial \tilde{x}_k \partial \tilde{x}_j}, \tag{D.10}$$

$$\frac{\partial^3 m^*_{\mathrm{p}}(\tilde{\mathbf{x}})}{\partial \tilde{x}_l \tilde{x}_k \partial \tilde{x}_j} = \frac{\partial^3 m_{\mathrm{p}}(\tilde{\mathbf{x}})}{\partial \tilde{x}_l \partial \tilde{x}_k \partial \tilde{x}_j} + \left(\frac{\partial^3 \mathbf{R}^{\mathrm{T}}}{\partial \tilde{x}_l \partial \tilde{x}_k \partial \tilde{x}_j} \right) \bar{\beta}, \tag{D.11}$$

$$\frac{\partial^3 \mathbf{R}}{\partial \tilde{x}_l \partial \tilde{x}_k \partial \tilde{x}_j} = \frac{\partial^3 \mathbf{h}(\tilde{\mathbf{x}})}{\partial \tilde{x}_l \partial \tilde{x}_k \partial \tilde{x}_j} - \mathbf{H}(\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \frac{\partial^3 k(\mathbf{X}, \tilde{\mathbf{x}})}{\partial \tilde{x}_l \partial \tilde{x}_k \partial \tilde{x}_j}, \tag{D.12}$$

D.1.3 Squared exponential kernel derivatives

For a squared exponential kernel (defined in eq (2.54)), the first order partial derivatives with respect to a new input $\tilde{\mathbf{x}}$ are:

$$\frac{\partial k(\tilde{\mathbf{x}}, \mathbf{x}_i)}{\partial \tilde{x}_j} = -k(\tilde{\mathbf{x}}, \mathbf{x}_i) \left(\frac{1}{l_j^2} (\tilde{x}_j - x_{i,j}) \right), \tag{D.13}$$

where $i = 1 \ldots n$ and $j = 1 \ldots d$.

Note that

$$\frac{\partial k(\tilde{\mathbf{x}}, \mathbf{x}_i)}{\partial \tilde{x}_j} = \frac{\partial k(\mathbf{x}_i, \tilde{\mathbf{x}})}{\partial \tilde{x}_j}. \tag{D.14}$$

In addition, the second and third order partial derivatives of the squared exponential kernel with respect to a new input $\tilde{\mathbf{x}}$ are:

$$\frac{\partial^2 k(\tilde{\mathbf{x}}, \mathbf{x}_i)}{\partial \tilde{x}_k \partial \tilde{x}_j} = -\left(\frac{\partial k(\tilde{\mathbf{x}}, \mathbf{x}_i)}{\partial \tilde{x}_k} \Big(\frac{1}{l_j^2} (\tilde{x}_j - x_{i,j}) \Big) + k(\tilde{\mathbf{x}}, \mathbf{x}_i) \frac{1}{l_j^2} \mathbb{I}(k = j) \right), \tag{D.15}$$

$$\frac{\partial^3 k(\tilde{\mathbf{x}}, \mathbf{x}_i)}{\partial \tilde{x}_l \partial \tilde{x}_k \partial \tilde{x}_j} = -\left(\frac{\partial^2 k(\tilde{\mathbf{x}}, \mathbf{x}_i)}{\partial \tilde{x}_l \partial \tilde{x}_k} \Big(\frac{1}{l_j^2} (\tilde{x}_j - x_{i,j}) \Big) + \frac{\partial k(\tilde{\mathbf{x}}, \mathbf{x}_i)}{\partial \tilde{x}_k} \frac{1}{l_j^2} \mathbb{I}(l = j) + \frac{\partial k(\tilde{\mathbf{x}}, \mathbf{x}_i)}{\partial \tilde{x}_l} \frac{1}{l_j^2} \mathbb{I}(k = j) \right), \tag{D.16}$$

where $\mathbb{I}$ is the indicator function, i.e. $\mathbb{I}(k = j) = 1$ if $k = j$ and 0 otherwise.

D.2 Pseudocode

Algorithm 1h GP HMC algorithm with DA (no classifier) – initial and exploratory phase

```
 1: Let $D = \{(\mathbf{X}, \mathscr{S}) : \lambda = 1\}$: training set for the GP emulator, where $\mathbf{X}$: $n \times d$ matrix of input parameter vectors $\theta$ and $\mathscr{S}$: vector of residual sum-of-square (RSS) values. Denote $S$: number of HMC samples, L: number of HMC trajectory steps, $\varepsilon$: step size, $\mathscr{S}(\theta)$: simulator RSS, $\tilde{\mathscr{S}}(\theta)$: emulator RSS, $\mathbf{M}$: mass matrix, $f(.)$: emulated RSS function, and $p(\mathbf{y}|\theta)$: simulator data likelihood, $\tilde{p}(\mathbf{y}|\theta)$: emulator data likelihood, $E(\theta)$: true potential function, $\tilde{E}(\theta)$: surrogate potential function, $K(\mathbf{r})$: kinetic energy for the momentum variable $\mathbf{r}$.
 2: $\log p(\mathbf{y}|\theta) = -\frac{\mathscr{S}(\theta)}{2\sigma^2} - \frac{n}{2}\log(2\pi\sigma^2)$ and $\log \tilde{p}(\mathbf{y}|\theta) = -\frac{\tilde{\mathscr{S}}(\theta)}{2\sigma^2} - \frac{n}{2}\log(2\pi\sigma^2)$ for $\varepsilon \overset{iid}{\sim} \mathscr{MVN}(\mathbf{0}, \sigma^2\mathbf{I})$; $E(\theta) = -(\log p(\mathbf{y}|\theta) + \log p(\theta))$, where $p(\theta)$: prior distribution; $\tilde{E}(\theta) = -(\log \tilde{p}(\mathbf{y}|\theta) + \log p(\theta))$.
 3: INITIAL DESIGN STAGE: Build the GP emulator.
 4: EXPLORATORY PHASE: Set $\theta^0$ and $l = 1$, where $l$ marks the $l^{\text{th}}$ point being deleted
 5: for i = 1 : S do                                                ▷ loop over HMC samples
 6:     Draw $\mathbf{r} \sim \exp(-K(\mathbf{r}))$ and let $\theta_0 = \theta^{i-1}$ and $\mathbf{r}_0 = \mathbf{r} + \frac{\varepsilon}{2}\frac{\partial \tilde{E}}{\partial \theta}\Big|_{\theta_0}$
 7:     for j = 1 : L do                                            ▷ loop over HMC steps
 8:         $\theta_j = \theta_{j-1} + \varepsilon \mathbf{M}^{-1}\mathbf{r}_{j-1}$;    $\mathbf{r}_j = \mathbf{r}_{j-1} + \varepsilon \frac{\partial \tilde{E}}{\partial \theta}\Big|_{\theta_j}$
 9:     end for
10:     $\mathbf{r}_L = \mathbf{r}_{L-1} + \frac{\varepsilon}{2}\frac{\partial \tilde{E}}{\partial \theta}\Big|_{\theta_L}$
11:     Set $(\theta^*, \mathbf{r}^*) = (\theta_L, \mathbf{r}_L)$, $\tilde{\mathscr{S}}(\theta^*) = \mathbb{E}(f(\theta^*)|D) - \sqrt{\text{var}(f(\theta^*)|D)}$, where $\mathbb{E}(f(\theta^*)|D)$ & $\sqrt{\text{var}(f(\theta^*)|D)}$: GP predictive mean & standard deviation ((2.66), (2.74), (2.67), (2.75))
12:     M-H accept/reject step with 1st stage acceptance probability (emulator based):

        $\alpha_1(\theta^*, \mathbf{r}^*|\theta^{i-1}, \mathbf{r}) = 1 \wedge \frac{\exp(-\tilde{E}(\theta^*))}{\exp(-\tilde{E}(\theta^{i-1}))}\frac{\exp(-K(\mathbf{r}^*))}{\exp(-K(\mathbf{r}))}$, with $\tilde{E}$ computed from $\tilde{\mathscr{S}}$ as in line 2

13:     if $\alpha_1 \geq v_1$,   $v_1 \sim U(0,1)$ then, solve ODEs/PDEs for $\theta^* = \theta_L$ to get $(\theta^*, \mathscr{S}(\theta^*))$
14:         Calculate the 2nd stage acceptance probability (simulator based):

        $\alpha_2(\theta^*|\theta^{i-1}) = 1 \wedge \frac{\exp(-E(\theta^*))}{\exp(-E(\theta^{i-1}))}\frac{\exp(-\tilde{E}(\theta^{i-1}))}{\exp(-\tilde{E}(\theta^*))}$, with $E$ computed from $\mathscr{S}$ as in line 2

15:         if $\alpha_2 \geq v_2$,   $v_2 \sim U(0,1)$ then, set $\theta^i = \theta^*$
16:             if $\mathscr{S}(\theta^l) > T$, where $T$: threshold value chosen based on the $\mathscr{S}$ values from the initial design stage (e.g. $T = 10^{\text{th}}$ percentile) then update $D = D \setminus (\theta^l, \mathscr{S}(\theta^l))$
17:             else set l = l + 1
18:             end if
19:             Re-train GP emulator with $D = D \cup (\theta^*, \mathscr{S}(\theta^*))$
20:         else
21:             $\theta^i = \theta^{i-1}$
22:         end if
23:     else
24:         $\theta^i = \theta^{i-1}$
25:     end if
26: end for
27: Re-train GP with $D = D \setminus (\theta, \mathscr{S}(\theta))$ for $\theta$: $\mathscr{S}(\theta) > T$; enter the SAMPLING PHASE
```

Algorithm 1i GP HMC algorithm with DA (no classifier) – sampling phase

```
 1: Denote $\theta$: input parameter vector, $S$: number of HMC samples, L: number of HMC trajectory steps, $\varepsilon$: step size, $\mathscr{S}(\theta)$: simulator residual sum-of-square (RSS) value, $\tilde{\mathscr{S}}(\theta)$: emulator RSS, $\mathbf{M}$: mass matrix, $f(.)$: emulated RSS function, $p(\mathbf{y}|\theta)$: simulator data likelihood, $\tilde{p}(\mathbf{y}|\theta)$: emulator data likelihood, $E(\theta)$: true potential function, $\tilde{E}(\theta)$: surrogate potential function, $K(\mathbf{r})$: kinetic energy for the momentum variable $\mathbf{r}$.
 2: $\log p(\mathbf{y}|\theta) = -\frac{\mathscr{S}(\theta)}{2\sigma^2} - \frac{n}{2}\log(2\pi\sigma^2)$ and $\log \tilde{p}(\mathbf{y}|\theta) = -\frac{\tilde{\mathscr{S}}(\theta)}{2\sigma^2} - \frac{n}{2}\log(2\pi\sigma^2)$ for $\varepsilon \overset{iid}{\sim} \mathscr{MVN}(\mathbf{0},\sigma^2\mathbf{I})$; $E(\theta) = -(\log p(\mathbf{y}|\theta) + \log p(\theta))$, where $p(\theta)$: prior distribution; $\tilde{E}(\theta) = -(\log \tilde{p}(\mathbf{y}|\theta) + \log p(\theta))$.
 3: Initialise $\theta^0$
 4: for i = 1 : S do                                        ▷ loop over HMC samples
 5:     Draw $\mathbf{r} \sim \exp(-K(\mathbf{r}))$
 6:     Let $\theta_0 = \theta^{i-1}$ and $\mathbf{r}_0 = \mathbf{r} + \frac{\varepsilon}{2}\frac{\partial \tilde{E}}{\partial \theta}\Big|_{\theta_0}$
 7:     for j = 1 : L do                                    ▷ loop over HMC steps
 8:         $\theta_j = \theta_{j-1} + \varepsilon \mathbf{M}^{-1}\mathbf{r}_{j-1}$
 9:         $\mathbf{r}_j = \mathbf{r}_{j-1} + \varepsilon \frac{\partial \tilde{E}}{\partial \theta}\Big|_{\theta_j}$
10:     end for
11:     $\mathbf{r}_L = \mathbf{r}_{L-1} + \frac{\varepsilon}{2}\frac{\partial \tilde{E}}{\partial \theta}\Big|_{\theta_L}$
12:     Set proposed points $(\theta^*, \mathbf{r}^*) = (\theta_L, \mathbf{r}_L)$ and $\tilde{\mathscr{S}}(\theta^*) = \mathbb{E}(f(\theta^*)|D)$, where $\mathbb{E}(f(\theta^*)|D)$ is the GP posterior predictive mean (see eq (2.66) and (2.74))
13:     Compute $\tilde{E}$ from $\tilde{\mathscr{S}}$ as explained in line 2
14:     M-H accept/reject step with 1st stage acceptance probability (emulator based):
            $$\alpha_1(\theta^*, \mathbf{r}^*|\theta^{i-1}, \mathbf{r}) \; = \; 1 \wedge \frac{\exp(-\tilde{E}(\theta^*))}{\exp(-\tilde{E}(\theta^{i-1}))}\frac{\exp(-K(\mathbf{r}^*))}{\exp(-K(\mathbf{r}))}$$
15:     if $\alpha_1 \geq v_1, \quad v_1 \sim U(0,1)$ then
16:         Solve ODEs/PDEs for $\theta^*$ to get $(\theta^*, \mathscr{S}(\theta^*))$
17:         Compute $E$ from $\mathscr{S}$ as explained in line 2
18:         Calculate the 2nd stage acceptance probability (simulator based):
                $$\alpha_2(\theta^*|\theta^{i-1}) \; = \; 1 \wedge \frac{\exp(-E(\theta^*))}{\exp(-E(\theta^{i-1}))}\frac{\exp(-\tilde{E}(\theta^{i-1}))}{\exp(-\tilde{E}(\theta^*))}$$
19:         if $\alpha_2 \geq v_2, \quad v_2 \sim U(0,1)$ then
20:             Set $\theta^i = \theta^*$
21:         else
22:             $\theta^i = \theta^{i-1}$
23:         end if
24:     else
25:         $\theta^i = \theta^{i-1}$
26:     end if
27: end for
```

Algorithm 1j GP HMC algorithm without DA (no classifier) – initial and exploratory phase

```
Lines 1-11 are the same as in Algorithm 1h
Solve ODEs/PDEs for proposed point $\theta^* = \theta_L$ to get $(\theta^*, \mathscr{S}(\theta^*))$
Compute $E$ from $\mathscr{S}$ as explained in line 2 of Algorithm 1h
M-H accept/reject step with acceptance probability :
    $$\alpha = 1 \wedge \frac{\exp(-E(\theta^*))}{\exp(-E(\theta^{i-1}))} \frac{\exp(-K(\mathbf{r}^*))}{\exp(-K(\mathbf{r}))}$$
if $\alpha \geq v, \quad v \sim U(0,1)$ then
    Set $\theta^{(i)} = \theta^*$
    Lines (16)-(19) are the same as in Algorithm 1h
else
    $\theta^{(i)} = \theta^{(i-1)}$
end if
Line (27) is the same as in Algorithm 1h
```

Algorithm 1k GP HMC algorithm without DA (no classifier) – sampling phase

```
Lines 1-12 are the same as in Algorithm 1i
Solve ODEs/PDEs for proposed point $\theta^* = \theta_L$ to get $(\theta^*, \mathscr{S}(\theta^*))$
Compute $E$ from $\mathscr{S}$ as explained in line 2 of Algorithm 1i
M-H accept/reject step with acceptance probability :
    $$\alpha = 1 \wedge \frac{\exp(-E(\theta^*))}{\exp(-E(\theta^{i-1}))} \frac{\exp(-K(\mathbf{r}^*))}{\exp(-K(\mathbf{r}))}$$
if $\alpha \geq v, \quad v \sim U(0,1)$ then
    Set $\theta^{(i)} = \theta^*$
else
    $\theta^{(i)} = \theta^{(i-1)}$
end if
end for
```

Algorithm 1l GP HMC algorithm with DA and a classifier – initial, exploratory and sampling phase

```
For the initial and exploratory phase, modify Algorithm 1h as follows:
Line 2: Replace $p(\theta)$ by $\tilde{p}(\theta)$ (eq (6.5))
Line 3: Build the GP emulator and classifier.
Line 13: Solve ODEs/PDEs for proposed point $\theta^* = \theta_L$ to get $(\theta^*, \mathscr{S}(\theta^*), \lambda^*)$
Line 16: if $\mathscr{S}(\theta^l) > T$, where $T$: threshold value chosen based on the $\mathscr{S}$ values from
    the initial design stage (e.g. $T = 10^{\text{th}}$ percentile), update $D = D \setminus (\theta^l, \mathscr{S}(\theta^l))$, $\mathscr{H} =
    \mathscr{H} \setminus (\theta^l, \lambda^l)$
Line 19: Re-train GP emulator with $D = D \cup (\theta^*, \mathscr{S}(\theta^*))$ and GP classifier with $\mathscr{H} =
    \mathscr{H} \cup (\theta^*, \lambda^*)$
Line 27: Update $D = D \setminus (\theta, \mathscr{S}(\theta))$, $\mathscr{H} = \mathscr{H} \setminus (\theta, \lambda)$ for the remaining $\theta$ for which
    $\mathscr{S}(\theta) > T$; Re-train GP emulator with new $D$ and GP classifier with new $\mathscr{H}$ and enter the
    SAMPLING PHASE
For the sampling phase, modify Algorithm 1i as follows:
Line 2: Replace $p(\theta)$ by $\tilde{p}(\theta)$ (eq (6.5))
Line 16: Solve ODEs/PDEs for $\theta^*$ to get $(\theta^*, \mathscr{S}(\theta^*), \lambda^*)$
```

Bibliography

[1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Proc. 2nd International Symposium on Information Theory, Tsahkadsor, Armenia, USSR*, pages 267–281. Budapest: Akadémiai Kiadó, 1971.

[2] J. Alastruey, N. Xiao, H. Fok, T. Schaeffter, and C.A. Figueroa. On the impact of modelling assumptions in multi-scale, subject-specific models of aortic haemodynamics. *Journal of the Royal Society Interface*, 13(119):20160073, 2016.

[3] K. Alden, J. Cosgrove, M. Coles, and J. Timmis. Using emulation to engineer and understand simulations of biological systems. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 17(1):302–315, 2020.

[4] C. Andrieu and A. Doucet. Joint Bayesian Model Selection and Estimation of Noisy Sinusoids via Reversible Jump MCMC. *IEEE Transactions on Signal Processing*, 47:2667 – 2676, 1999.

[5] L. Antiga, M. Piccinelli, L. Botti, B. Ene-Iordache, A. Remuzzi, and D.A. Steinman. An image-based modeling framework for patient-specific computational hemodynamics. *Med Biol Eng Comput*, 46(1097), 2008.

[6] M. J. Asher, B. F. W. Croke, A. J. Jakeman, and L. J. M. Peeters. A review of surrogate models and their application to groundwater modeling. *Water Resources Research*, 51(8):5957–5973, 2015.

[7] M. Banterle, C. Grazian, A. Lee, and C.P. Robert. Accelerating metropolis-hastings algorithms by delayed acceptance. *Foundations of Data Science*, 1(2):103–128, 2019.

[8] L.S. Bastos and A. O'Hagan. Diagnostics for Gaussian Process Emulators. *Technometrics*, 51(4):425–438, 2009.

[9] M. Betancourt. The Fundamental Incompatibility of Scalable Hamiltonian Monte Carlo and Naive Data Subsampling. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 533–540, Lille, France, 2015. PMLR.

[10] N. Bliznyuk, D. Ruppert, C. Shoemaker, R. Regis, S. Wild, and P. Mugunthan. Bayesian Calibration and Uncertainty Analysis for Computationally Expensive Models Using Optimization and Radial Basis Function Approximation. *Journal of Computational and Graphical Statistics*, 17:270–294, 2008.

[11] P.T. Boggs and J.W. Tolle. Sequential quadratic programming for large-scale nonlinear optimization. *Journal of Computational and Applied Mathematics*, 124(1–2):123 – 137, 2000. Numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations.

[12] W.F. Boron and E.L. Boulpaep. *Medical Physiology*. Elsevier, Philadelpha, PA, 3rd edition, 2016.

[13] C.S. Bos. A comparison of marginal likelihood computation methods. In *COMPSTAT 2002: Proceedings in Computational Statistics*, pages 111–117, 2002.

[14] A. W. Bowman and A. Azzalini. *Applied Smoothing Techniques for Data Analysis*. New York: Oxford University Press Inc., 1997.

[15] G.E.P. Box and G. Jenkins. *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated, 1990.

[16] P. Bratley and B.L. Fox. Algorithm 659: Implementing Sobol's Quasirandom Sequence Generator. *ACM Trans. Math. Softw.*, 14(1):88–100, 1988.

[17] S. Brooks, A. Gelman, G.L. Jones, and X.L. Meng. *Handbook of Markov Chain Monte Carlo*. Handbooks of Modern Statistical Methods. Chapman and Hall, 2011.

[18] S.P. Brooks and A. Gelman. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7(4):434–455, 1998.

[19] C. G. Broyden. *Quasi-Newton Methods*. London: Academic Press, 1972.

[20] J. Brynjarsdóttir and A. O'Hagan. Learning about physical parameters: The importance of model discrepancy. *Inverse Problems*, 30(11):114007, 2014.

[21] B. Brüggemeier, C. Schusterreiter, H. Pavlou, N. Jenkins, S. Lynch, A. Bianchi, and X. Cai. Improving the utility of Drosophila melanogaster for neurodegenerative disease research by modelling courtship behaviour patterns. 2014.

[22] T. Bui-Thanh and M. Girolami. Solving large-scale PDE-constrained Bayesian inverse problems with Riemann Manifold Hamiltonian Monte Carlo. *Inverse Problems*, 30(11):114014, 2014.

[23] K.P. Burnham and D.R. Anderson. *Model Selection and Multimodel Inference: A practical information-theoretic approach*. Springer-Verlag, second edition, 2002.

[24] B. Calderhead. *Differential geometric MCMC methods and applications*. PhD thesis, University of Glasgow, 2012.

[25] D. Campbell and R.J. Steele. Smooth functional tempering for nonlinear differential equation models. *Statistics and Computing*, 22:429–443, 2012.

[26] G. Casella and E.I. George. Explaining the Gibbs Sampler. *The American Statistician*, 46(3):167–174, 1992.

[27] G. Casella, C.P. Robert, and M.T. Wells. *Generalized Accept-Reject sampling schemes*, volume 45 of *Lecture Notes–Monograph Series*, pages 342–347. Institute of Mathematical Statistics, 2004.

[28] J.E. Cavanaugh and A.A. Neath. The Akaike information criterion: Background, derivation, properties, application, interpretation, and refinements. *WIREs Computational Statistics*, 11(3):e1460, 2019.

[29] M. J. Chambers, M. J. Colebank, M. U. Qureshi, R. Clipp, and M. S. Olufsen. Structural and hemodynamic properties of murine pulmonary arterial networks under hypoxia-induced pulmonary hypertension. In *Proceedings of the Institution of Mechanical Engineers, Part H*, 2020.

[30] R. Chandra, D. Azam, A. Kapoor, and R. Mulller. Surrogate-assisted Bayesian inversion for landscape and basin evolution models. *Geoscientific Model Development Discussions*, pages 1–28, 2019.

[31] T. Chen, E. Fox, and C. Guestrin. Stochastic Gradient Hamiltonian Monte Carlo. *31st International Conference on Machine Learning, ICML*, 32, 2014.

[32] S. Chib and I. Jeliazkov. Marginal Likelihood From the Metropolis–Hastings Output. *Journal of the American Statistical Association*, 96(453):270–281, 2001.

[33] J.A. Christen and C. Fox. Markov Chain Monte Carlo using an approximation. *Journal of Computational and Graphical Statistics*, 14(4):795–810, 2005.

[34] J.A. Christen and C. Fox. A general purpose sampling algorithm for continuous distributions (the t-walk). *Bayesian Analysis*, 5(2):263–282, 2010.

[35] R. Clipp and B. Steele. An evaluation of dynamic outlet boundary conditions in a 1D fluid dynamics model. *Math Biosci Eng*, 9(1):61–74, 2012.

[36] Mitchel J. Colebank, M. Umar Qureshi, and Mette S. Olufsen. Sensitivity analysis and uncertainty quantification of 1-D models of pulmonary hemodynamics in mice under control and hypertensive conditions. *International Journal for Numerical Methods in Biomedical Engineering*, n/a(n/a):e3242, 2019.

[37] M.J. Colebank, L.M. Paun, M.U. Qureshi, N. Chesler, D. Husmeier, M.S. Olufsen, and L. Ellwein. Influence of image segmentation on one-dimensional fluid dynamics predictions in the mouse pulmonary arteries. *Journal of the Royal Society Interface*, 16(159):20190284, 2019.

[38] M.J. Colebank, U.M. Qureshi, and M.S. Olufsen. Sensitivity analysis and uncertainty quantification of 1-D models of pulmonary hemodynamics in mice under control and hypertensive conditions. *International Journal for Numerical Methods in Biomedical Engineering*, page e3242, 2019.

[39] P.R. Conrad, A.D. Davis, Y.M. Marzouk, N.S. Pillai, and A. Smith. Parallel Local Approximation MCMC for Expensive Models. *SIAM/ASA Journal on Uncertainty Quantification*, 6(1):339–373, 2018.

[40] P.R. Conrad, Y.M. Marzouk, N.S. Pillai, and A. Smith. Accelerating Asymptotically Exact MCMC for Computationally Intensive Models via Local Approximations. *Journal of the American Statistical Association*, 111(516):1591–1607, 2016.

[41] S. Conti, J. P. Gosling, J. E. Oakley, and A. O'Hagan. Gaussian process emulation of dynamic computer codes. *Biometrika*, 96(3):663–676, 2009.

[42] S. Conti and A. O'Hagan. Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, 140(3):640 – 651, 2010.

[43] F.S. Costabal, K. Matsuno, J. Yao, P. Perdikaris, and E. Kuhl. Machine learning in drug development: Characterizing the effect of 30 drugs on the QT interval using Gaussian process regression, sensitivity analysis, and uncertainty quantification. *Computer Methods in Applied Mechanics and Engineering*, 348:313 – 333, 2019.

[44] S. L. Cotter, M. Dashti, and A. M. Stuart. Approximation of Bayesian Inverse Problems for PDEs. *SIAM Journal on Numerical Analysis*, 48(1):322–345, 2010.

[45] S. Coveney and R.H. Clayton. Fitting two human atrial cell models to experimental data using Bayesian history matching. *Progress in Biophysics and Molecular Biology*, 139:43 – 58, 2018.

[46] M.K. Cowles and B.P. Carlin. Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review. *Journal of the American Statistical Association*, 91(434):883–904, 1996.

[47] T. Cui, C. Fox, and M. O'Sullivan. Bayesian calibration of a large-scale geothermal reservoir model by a new adaptive delayed acceptance Metropolis Hastings algorithm. *Water Resources Research*, 47(10):W10521, 2011.

[48] J. Dahlin, F. Lindsten, and T.B. Schön. Particle Metropolis–Hastings using gradient and Hessian information. *Statistics and Computing*, 25:81–92, 2015.

[49] V. Davidoiu, L. Hadjilucas, I. Teh, N.P. Smith, J.E. Schneider, and J. Lee. Evaluation of noise removal algorithms for imaging and reconstruction of vascular networks using micro-CT. *Biomedical Physics & Engineering Express*, 2(4):045015, 2016.

[50] J. Dhamala, H.J. Arevalo, J. Sapp, B.M. Horácek, K.C. Wu, N.A. Trayanova, and L. Wang. Quantifying the uncertainty in model parameters using Gaussian process-based Markov chain Monte Carlo in cardiac electrophysiology. *Medical Image Analysis*, 48:43 – 57, 2018.

[51] A. Dietzel and P. Reichert. Bayesian inference of a lake water quality model by emulating its posterior density. *Water Resources Research*, 50(10):7626–7647, 2014.

[52] J. Doherty and D. Welter. A short exploration of structural noise. *Water Resources Research*, 46(5):W05525, 2010.

[53] C.C. Drovandi, M.T. Moores, and R.J. Boys. Accelerating pseudo-marginal MCMC using Gaussian processes. 118:1–17, 2018.

[54] M. Ebden. Gaussian Processes: A Quick Introduction. *ArXiv e-prints,*, 2015.

[55] V.G. Eck, W.P. Donders, J. Sturdy, J. Feinberg, T. Delhaas, L.R. Hellevik, and W. Huberts. A guide to uncertainty quantification and sensitivity analysis for cardiovascular applications. *International Journal for Numerical Methods in Biomedical Engineering*, 32(8):e02755, 2016.

[56] L. Ellwein, D.S. Marks, R.Q. Migrino, W.D. Foley, S. Sherman, and J.F. LaDisa Jr. Image-based quantification of 3D morphology for bifurcations in the left coronary artery: Application to stent design. *Catheterization and Cardiovascular Interventions*, 87(7):1244–1255, 2016.

[57] M. Fielding, D. Nott, and S.Y. Liong. Efficient MCMC schemes for computationally expensive posterior distributions. *Technometrics*, 53(1):16–28, 2011.

[58] R. Fitzhugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal*, 1(6):445–466, 1961.

[59] E. Fong and C.C. Holmes. On the marginal likelihood and cross-validation. *Biometrika*, 107(2):489–496, 2020.

[60] F.E. Fossan, J. Mariscal-Harana, J. Alastruey, and L.R. Hellevik. Optimization of topological complexity for one-dimensional arterial blood flow models. *Journal of the Royal Society Interface*, 15(149):20180546, 2018.

[61] M. Frangos, Y. Marzouk, K. Willcox, and B. van Bloemen Waanders. *Surrogate and Reduced-Order Modeling: A Comparison of Approaches for Large-Scale Statistical Inverse Problems*, chapter 7, pages 123–149. John Wiley & Sons, Ltd, 2010.

[62] N. Friel and A. N. Pettitt. Marginal likelihood estimation via power posteriors. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(3):589–607, 2008.

[63] Y. Fukumoto. *Diagnosis and treatment of pulmonary hypertension*, volume 101, pages 311–319. Singapore: Springer, 2015.

[64] H. Gao, A. Aderhold, K. Mangion, X. Luo, D. Husmeier, and C. Berry. Changes and classification in myocardial contractile function in the left ventricle following acute myocardial infarction. *Journal of the Royal Society Interface*, 14(132):20170203, 2017.

[65] M. Gelbart, J. Snoek, and R. Adams. Bayesian optimization with unknown constraints. In *Proceedings of the Thirtieth Annual Conference on Uncertainty in Artificial Intelligence (UAI-14)*, pages 250–259, Corvallis, Oregon, 2014. AUAI Press.

[66] A. Gelman, J. Carlin, H. Stern, D. Rubin, D. Dunson, and A. Vehtari. *Bayesian Data Analysis*. Chapman & Hall/CRC Texts in Statistical Science. 3rd edition, 2013.

[67] A. Gelman, J. Hwang, and A. Vehtari. Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, 24(6):997–1016, 2014.

[68] A. Gelman, G. O. Roberts, and W. R. Gilks. Efficient Metropolis jumping rules. *Bayesian Statistics*, 5:599 – 608, 1996.

[69] A. Gelman and D.B. Rubin. Inference from iterative simulation using multiple sequences. *Statist. Sci.*, 7(4):457–472, 1992.

[70] J. Geweke. Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. In *Bayesian Statistics*, pages 169–193. University Press, 1992.

[71] J. Geweke. Getting it right: Joint distribution tests of posterior simulators. *Journal of the American Statistical Association*, 99(467):799–804, 2004.

[72] C.J. Geyer. Practical Markov Chain Monte Carlo. *Statist. Sci.*, 7(4):473–483, 11 1992.

[73] C.J. Geyer. *Introduction to Markov Chain Monte Carlo*. Chapman & Hall/CRC. In Handbook of Markov Chain Monte Carlo, 2011.

[74] M. Girolami and B. Calderhead. Riemann Manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

[75] P.W. Goldberg, C.K.I. Williams, and C.M. Bishop. Regression with Input-dependent Noise: A Gaussian Process Treatment. In *Advances in Neural Information Processing Systems 10*, pages 493–499. MIT Press, 1998.

[76] A. Golightly, D.A. Henderson, and C. Sherlock. Delayed acceptance particle MCMC for exact inference in stochastic kinetic models. *Statistics and Computing*, 25(5):1039–1055, 2015.

[77] W. Gong and Q. Duan. An adaptive surrogate modeling-based sampling strategy for parameter optimization and distribution estimation (ASMO-PODE). *Environmental Modelling & Software*, 95:61 – 75, 2017.

[78] P.J. Green and A. Mira. Delayed rejection in reversible jump Metropolis–Hastings. *Biometrika*, 88(4):1035–1053, 2001.

[79] A.C. Guidoum. Kernel estimator and bandwidth selection for density and its derivatives. *The R Journal*, 2015.

[80] S. Göktepe and E. Kuhl. Computational modeling of cardiac electrophysiology: A novel finite element approach. *International Journal for Numerical Methods in Engineering*, 79(2):156–178, 2009.

[81] H. Haario, M. Laine, A. Mira, and E. Saksman. DRAM: Efficient adaptive MCMC. *Statistics and Computing*, 16(4):339–354, 2006.

[82] H. Haario, E. Saksman, and J. Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242, 2001.

[83] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[84] M. Helmberger, M. Pienn, M. Urschler, P. Kullnig, R. Stollberger, G. Kovacs, A. Olschewski, H. Olschewski, and Z. Bálint. Quantification of tortuosity and fractal dimension of the lung vessels in pulmonary hypertension patients. *PloS one*, 9(1):e87515, 2014.

[85] D. Higdon, C.S. Reese, J. Moulton, J. Vrugt, and C. Fox. *Posterior exploration for computationally intensive forward models*, pages 401–418. 05 2011.

[86] J.A. Hoeting, D. Madigan, A.E. Raftery, and C.T. Volinsky. Bayesian model averaging: a tutorial. *Statist. Sci.*, 14(4):382–417, 1999.

[87] M.D. Hoffman and A. Gelman. The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *The Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

[88] C.C. Holmes and L. Held. Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Anal.*, 1(1):145–168, 2006.

[89] D.R. Hose, P.V. Lawford, W. Huberts, L.R. Hellevik, S.W. Omholt, and F.N. van de Vosse. Cardiovascular models for personalised medicine: Where now and where next? *Medical Engineering & Physics*, 72:38 – 48, 2019.

[90] W. Huberts, W.P. Donders, T. Delhaas, and F.N. van de Vosse. Applicability of the polynomial chaos expansion method for personalization of a cardiovascular pulse wave propagation model. *International Journal for Numerical Methods in Biomedical Engineering*, 30(12):1679–1704, 2014.

[91] W. Huberts, S.G.H. Heinen, N. Zonnebeld, D.A.F. van den Heuvel, J.P.M. de Vries, J.H.M. Tordoir, D.R. Hose, T. Delhaas, and F.N. van de Vosse. What is needed to make cardiovascular models suitable for clinical decision support? a viewpoint paper. *Journal of Computational Science*, 24:68 – 84, 2018.

[92] D. Husmeier and L.M. Paun. Closed-loop effects in coupling cardiac physiological models to clinical interventions. In *Proceedings of the 35th IWSM*, pages 120–125, 2020.

[93] R.H. Johnstone, E.T.Y. Chang, R. Bardenet, T.P. de Boer, D.J. Gavaghan, P. Pathmanathan, R.H. Clayton, and Mirams G.R. Uncertainty and variability in models of the cardiac action potential: Can we build trustworthy models? *Journal of Molecular and Cellular Cardiology*, 96:49 – 62, 2016. Special Issue: Computational Modelling of the Heart.

[94] D. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive blackbox functions. *Journal of Global Optimization*, 13:455–492, 1998.

[95] R.E. Kass, B.P. Carlin, A. Gelman, and R.M. Neal. Markov Chain Monte Carlo in Practice: A Roundtable Discussion. *The American Statistician*, 52(2):93–100, 1998.

[96] M.C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

[97] V. O. Kheyfets, W. O'Dell, T. Smith, J. J. Reilly, and E. A. Finol. Considerations for Numerical Modeling of the Pulmonary Circulation—A Review With a Focus on Pulmonary Hypertension. *Journal of Biomechanical Engineering*, 135(6):061011, 2013.

[98] V.O. Kheyfets, L. Rios, T. Smith, T. Schroeder, J. Mueller, S. Murali, D. Lasorda, A. Zikos, J. Spotti, J.J. Reilly, and E.A. Finol. Patient-specific computational modeling of blood flow in the pulmonary arterial circulation. *Computer Methods and Programs in Biomedicine*, 120(2):88 – 101, 2015.

[99] E.Y. Kim and E.J. Choi. Pathological roles of mapk signaling pathways in human diseases. *Biochimica et Biophysica Acta (BBA) - Molecular Basis of Disease*, 1802(4):396 – 405, 2010.

[100] T. Kind, T.J. Faes, A. Vonk-Noordegraaf, and N. Westerhof. Proportional relations between systolic, diastolic and mean pulmonary artery pressure are explained by vascular properties. *Cardiovasc Eng Technol.*, 2(1):15–23, 2011.

[101] E. Konukoglu et al. Efficient Probabilistic Model Personalization Integrating Uncertainty on Data and Parameters: Application to Eikonal-Diffusion Models in Cardiac Electrophysiology. *Progress in biophysics and molecular biology*, 107(1):134–46, 2011.

[102] A. Kramer, B. Calderhead, and N. Radde. Hamiltonian Monte Carlo methods for efficient parameter estimation in steady state dynamical systems. *BMC Bioinformatics*, 15(1):253, 2014.

[103] G.S. Krenz and A.D. Dawson. Flow and pressure distributions in vascular networks consisting of distensible vessels. *American Journal of Physiology-Heart and Circulatory Physiology*, 284(6):H2192–H2203, 2003.

[104] C. Kreutz, A. Raue, D. Kaschek, and J. Timmer. Profile likelihood in systems biology. *The FEBS Journal*, 280(11):2564–2571, 2013.

[105] M. Kuss. *Gaussian Process Models for Robust Regression, Classification, and Reinforcement Learning*. PhD thesis, Technische Universität Darmstadt, 2006.

[106] D. Kwiatkowski, P. Phillips, P. Schmidt, and Y. Shin. Testing the null hypothesis of stationarity against the alternative of a unit root. How sure are we that economic time series have a unit root? *Journal of Econometrics*, 54(1-3):159–178, 1992.

[107] M. Laine. MCMC Toolbox for Matlab. http://helios.fmi.fi/ lainema/dram/, 2007.

[108] S. Lan, T. Bui-Thanh, M. Christie, and M. Girolami. Emulation of higher-order tensors in manifold Monte Carlo methods for Bayesian Inverse Problems. *Journal of Computational Physics*, 308:81 – 101, 2016.

[109] S. Lan, V. Stathopoulos, B. Shahbaba, and M. Girolami. Markov Chain Monte Carlo from Lagrangian Dynamics. *Journal of Computational and Graphical Statistics*, 24(2):357–378, 2015.

[110] I.M. Lang, C. Plank, S.K. Roela, J. Jakowitsch, W. Klepetko, and G. Maurer. Imaging in pulmonary hypertension. *JACC: Cardiovascular Imaging*, 3(12):1287–1295, 2010.

[111] N. Lartillot and H. Philippe. Computing Bayes Factors Using Thermodynamic Integration. *Systematic biology*, 55:195–207, 2006.

[112] N. D. Lawrence, M. Girolami, M. Rattray, and G. Sanguinetti. *Learning and Inference in Computational Systems Biology*. MIT Press, 2010.

[113] P. Lax and B. Wendroff. Systems of conservation laws. *Communications on Pure and Applied Mathematics*, 13(2):217–237, 1960.

[114] P. Lee, B.E. Carlson, N. Chesler, M.S. Olufsen, M. U. Qureshi, N.P. Smith, T. Sochi, and D.A. Beard. Heterogeneous mechanics of the mouse pulmonary arterial network. *Biomechanics and Modelling in Mechanobiology*, 15(5):1245–1261, 2017.

[115] C.L. Lei et al. Considering discrepancy when calibrating a mechanistic electrophysiology model. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 378(2173):20190349, 2020.

[116] A. Lungu, J.M. Wild, D. Capener, D.G Kiely, A.J Swift, and D.R. Hose. MRI model-based non-invasive differential diagnosis in pulmonary hypertension. *Journal of Biomechanics*, 47(12):2941 – 2947, 2014.

[117] M. Lê, H. Delingette, J. Kalpathy-Cramer, E. R. Gerstner, T. Batchelor, J. Unkelbach, and N. Ayache. MRI Based Bayesian Personalization of a Tumor Growth Model. *IEEE Transactions on Medical Imaging*, 35(10):2329–2339, 2016.

[118] G.S. Martin. Cell signaling and cancer. *Cancer Cell*, 4(3):167–174, 2004.

[119] M. D. McKay, R. J. Beckman, and W. J. Conover. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[120] A.I. McLeod. Derivation of the theoretical autocovariance function of autoregressive-moving average time series. *Appl. Statist.*, 24(2):255, 1975.

[121] A.I. McLeod. *FitARMA: Fit ARMA or ARIMA using fast MLE algorithm*, 2013.

[122] A. Melis, R.H. Clayton, and A. Marzo. Bayesian sensitivity analysis of a 1D vascular model with Gaussian process emulators. *International Journal for Numerical Methods in Biomedical Engineering*, 33(12):e2882, 2017.

[123] A. Mira. On Metropolis–Hastings algorithm with Delayed Rejection. *Metron*, 59(3):231–241, 2001a.

[124] G.R. Mirams, P. Pathmanathan, R.A. Gray, P. Challenor, and R.H. Clayton. Uncertainty and variability in computational and mathematical models of cardiac physiology. *The Journal of Physiology*, 594(23):6833–6847, 2016.

[125] J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2:117–129, 1978.

[126] P. Moreno-Muñoz, A. Artés, and M. Álvarez. Heterogeneous Multi-output Gaussian Process Prediction. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 6711–6720. Curran Associates, Inc., 2018.

[127] de Leval M.R., Dubini G., Migliavacca F., Jalali H., Camporini G., Redington A., and Pietrabissa R. Use of computational fluid dynamics in the design of surgical procedures: Application to the study of competitive flows in cavopulmonary connections. *The Journal of Thoracic and Cardiovascular Surgery*, 111(3):502 – 513, 1996.

[128] K.P. Murphy. *Machine learning: a probabilistic perspective*. MIT Press, 2013.

[129] I. Murray and M. Graham. Pseudo-marginal slice sampling. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *JMLR: W&CP*, pages 911–919, Cadiz, Spain, 2016.

[130] J. P. Mynard, M. R. Davidson, D. J. Penny, and J. J. Smolich. A simple, versatile valve model for use in lumped parameter and one-dimensional cardiovascular models. *International Journal for Numerical Methods in Biomedical Engineering*, 28(6-7):626–641, 2012.

[131] J. Nagumo, S. Arimoto, and S. Yoshizawa. An active pulse transmission line simulating nerve axon. *Proceeding IRE*, 50:2061–2070, 1962.

[132] R.M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2, 2011.

[133] M.G. Newberry, D.B. Ennis, and V.M. Savage. Testing foundations of biological scaling theory using automated measurements of vascular networks. *PLoS Comput Biol*, 11(8):e1004455, 2015.

[134] T.V. Nguyen and E.V. Bonilla. Automated Variational Inference for Gaussian Process Models. In *Advances in Neural Information Processing Systems 27*, pages 1404–1412. Curran Associates, Inc., 2014.

[135] U. Noè, W. Chen, M. Filippone, N. Hill, and D. Husmeier. Inference in a partial differential equations model of pulmonary arterial and venous blood circulation using statistical

emulation. In *Computational Intelligence Methods for Bioinformatics and Biostatistics*, pages 184–198, Cham, 2017. Springer International Publishing.

[136] A. Noordegraaf and N. Gali. The role of right ventricle in pulmonary arterial hypertension. *Eur Respir Rev*, 20(122):243–253, 2011.

[137] A. O'Hagan. Bayesian analysis of computer code outputs: A tutorial. *Reliability Engineering & System Safety*, 91(10):1290 – 1300, 2006.

[138] M.S. Olufsen, N. A. Hill, G.D.A. Vaughan, C. Sainsbury, and M. Johnson. Rarefaction and blood pressure in systemic and pulmonary arteries. *Journal of Fluid Mechanics*, 705:280–305, 2012.

[139] M.S. Olufsen, C. S. Peskin, W.Y. Kim, E. M. Pedersen, A. Nadim, and J. Larsen. Numerical simulation and experimental validation of blood flow in arteries with structured-tree outflow conditions. *Annals of Biomedical Engineering*, 28:1281–1299, 2000.

[140] L.M. Paun, M. Colebank, M.U. Qureshi, M. Olufsen, N. Hill, and D. Husmeier. MCMC with delayed acceptance using a surrogate model with an application to cardiovascular fluid dynamics. In *Proceedings of the International Conference on Statistics: Theory and Applications (ICSTA'19)*, 2019.

[141] L.M. Paun, M.U. Qureshi, M. Colebank, N.A. Hill, M.S. Olufsen, M.A. Haider, and D. Husmeier. MCMC methods for inference in a mathematical model of pulmonary circulation. *Statistica Neerlandica*, 72(3):306–338, 2018.

[142] C. Payer et al. Automated integer programming based separation of arteries and veins from thoracic CT images. *Medical Image Analysis*, 34:109 – 122, 2016.

[143] M. Peirlinck, F. Sahli Costabal, K. L. Sack, J. S. Choy, G. S. Kassab, J. M. Guccione, M. De Beule, P. Segers, and E. Kuhl. Using machine learning to characterize heart failure across the scales. *Biomechanics and Modeling in Mechanobiology*, 18(6):1987–2001, 2019.

[144] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Chapter 9. Root Finding and Nonlinear Sets of Equations*. Numerical Recipes: The Art of Scientific Computing (3rd ed.). New York: Cambridge University Press, 2007.

[145] W. Pruett and R. Hester. The creation of surrogate models for fast estimation of complex model outcomes. *PLOS ONE*, 11(6):e0156574, 2016.

[146] Y. Qi and T. Minka. Hessian-based Markov Chain Monte-Carlo Algorithms. In *1st Cape Cod Workshop on Monte Carlo Methods*, 2002.

[147] A. Quaglino, S. Pezzuto, P. Koutsourelakis, A. Auricchio, and R. Krause. Fast uncertainty quantification of activation sequences in patient-specific cardiac electrophysiology meeting clinical time constraints. *International Journal for Numerical Methods in Biomedical Engineering*, 34(7):e2985, 2018.

[148] A. Quaglino, S. Pezzuto, and R. Krause. High-dimensional and higher-order multifidelity Monte Carlo estimators. *Journal of Computational Physics*, 388:300 – 315, 2019.

[149] M. Quiroz, M.N. Tran, M. Villani, and R. Kohn. Speeding up MCMC by Delayed Acceptance and Data Subsampling. *Journal of Computational and Graphical Statistics*, 27:12–22, 2017.

[150] M.U. Qureshi. *Simulating the pulse wave in the human pulmonary circulation.* PhD thesis, University of Glasgow, 2013.

[151] M.U. Qureshi, M. Colebank, L.M. Paun, N. Chesler, M.A. Haider, N.A. Hill, D. Husmeier, and M.S. Olufsen. A computational study of pulmonary hemodynamics in healthy and hypoxic mice. *Biomechanics and Modelling in Mechanobiology*, 18(1):219–243, 2018.

[152] M.U. Qureshi, G.D.A. Vaughan, C. Sainsbury, M. Johnson, C.S. Peskin, M.S. Olufsen, and N.A. Hill. Numerical simulation of blood flow and pressure drop in the pulmonary arterial and venous circulation. *Biomechanics and modeling in mechanobiology*, 13(5):1137–1154, 2014.

[153] A.E. Raftery, M. Karny, and P. Ettler. Online Prediction Under Model Uncertainty via Dynamic Model Averaging: Application to a Cold Rolling Mill. *Technometrics: a journal of statistics for the physical, chemical, and engineering sciences*, 52(1):52–66, 2010.

[154] A.E. Raftery, M.A. Newton, J.M. Satagopan, and P.N. Krivitsky. Estimating the Integrated Likelihood via Posterior Simulation Using the Harmonic Mean Identity. In *Bayesian Statistics*, volume 8, pages 1–45. University Press, 2007.

[155] J. O. Ramsay, G. Hooker, D. Campbell, and J. Cao. Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society Series B*, 69(5):741–796, 2007.

[156] C.E. Rasmussen. Gaussian Processes to Speed up Hybrid Monte Carlo for Expensive Bayesian Integrals. *Bayesian Statistics*, 7:651–659, 01 2003.

[157] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning).* The MIT Press, 2005.

[158] S. Razavi, B. Tolson, and D. Burn. Review of surrogate modeling in water resources. *Water Resources Research*, 48(7):W07401, 2012.

[159] A. C. Riches, J. G. Sharp, D. Brynmor Thomas, and S. Vaughan Smith. Blood volume determination in the mouse. *The Journal of Physiology*, 228(2):279–284, 1973.

[160] J. Riihimäki and A. Vehtari. Laplace Approximation for Logistic Gaussian Process Density Estimation and Regression. *Bayesian Anal.*, 9(2):425–448, 06 2014.

[161] S. Rivolo et al. Impact of coronary bifurcation morphology on wave propagation. *American Journal of Physiology-Heart and Circulatory Physiology*, 311(4):H855–H870, 2016.

[162] C. Robert and J.M. Marin. *Bayesian Core: A Practical Approach to Computational Bayesian Statistics*. Springer Science & Business Media, 2007.

[163] J.C. Robinson. *An Introduction to Ordinary Differential Equations*. Cambridge University Press, 2004.

[164] J.A.J. Rynn, S.L. Cotter, C.E. Powell, and L.E. Wright. Surrogate accelerated Bayesian inversion for the determination of the thermal diffusivity of a material. *Metrologia*, 56(1):015018, 2019.

[165] M.D. Ryser, R. Gulati, M.C. Eisenberg, Y. Shen, E.S Hwang, and R.B. Etzioni. Identification of the Fraction of Indolent Tumors and Associated Overdiagnosis in Breast Cancer Screening Trials. *American journal of epidemiology*, 188(1):197—205, 2019.

[166] M.P. Saccomani and K. Thomaseth. The union between structural and practical identifiability makes strength in reducing oncological model complexity: a case study. *Complexity*, 2018, 2018.

[167] S. Safaei, C.P. Bradley, V. Suresh, K. Mithraratne, A. Muller, H. Ho, D. Ladd, L. Hellevik, S.W. Omholt, J.G. Chase, L.O. Mller, S.M. Watanabe, P.J. Blanco, B. De Bono, and P.J. Hunter. Roadmap for cardiovascular circulation model. *J Physiol*, 594(23):6909–6928, 2016.

[168] S. Sankaran, L. Grady, and C. A. Taylor. Fast computation of hemodynamic sensitivity to lumen segmentation uncertainty. *IEEE Transactions on Medical Imaging*, 34(12):2562–2571, 2015.

[169] T. Schaul. *Optimization with Surrogate Models*, pages 55–70. Springer Netherlands, 2013.

[170] D. Schiavazzi, G. Arbia, C. Baker, A. Hlavacek, T. Hsia, A. Marsden, and I. Vignon-Clementel. Uncertainty quantification in virtual surgery hemodynamics predictions for

single ventricle palliation. *International Journal for Numerical Methods in Biomedical Engineering*, 32(3):e02737, 2016.

[171] D.E. Schiavazzi, A. Baretta, G. Pennati, T.Y. Hsia, and A.L. Marsden. Patient-specific parameter estimation in single-ventricle lumped circulation models under uncertainty. *Int J Numer Method Biomed Eng*, 33(3):e02799, 2017.

[172] G. Schwarz. Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464, 03 1978.

[173] B. Sengupta, K.J. Friston, and W.D. Penny. Efficient gradient computation for dynamical models. *NeuroImage*, 98:521–527, 2014.

[174] B. Sengupta, K.J. Friston, and W.D. Penny. Gradient-based MCMC samplers for dynamic causal modelling. *NeuroImage*, 125:1107–1118, 2016.

[175] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, and N. de Freitas. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104:148–175, 2016.

[176] D. Shahsavani, S. Tarantola, and M. Ratto. Evaluation of MARS modeling technique for sensitivity analysis of model output. *Procedia - Social and Behavioral Sciences*, 2(6):7737 – 7738, 2010. Sixth International Conference on Sensitivity Analysis of Model Output.

[177] C. Sherlock, A. Golightly, and D.A. Henderson. Adaptive, Delayed-Acceptance MCMC for Targets With Expensive Likelihoods. *Journal of Computational and Graphical Statistics*, 26(2):434–444, 2017.

[178] Y. Shi, P. Lawford, and R. Hose. Review of Zero-D and 1-D Models of Blood Flow in the Cardiovascular System. *Biomed Eng Online*, 10(33), 2011.

[179] R. Shibata. Statistical aspects of model selection. pages 215–240, 1989.

[180] B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. London: Chapman & Hall/CRC. Chapman and Hall, 1986.

[181] G. Simonneau et al. Updated clinical classification of pulmonary hypertension. *Journal of the American College of Cardiology*, 62(25):D34–D41, 2013.

[182] U. Simsekli, R. Badeau, A.T. Cemgil, and R. Gaël. Stochastic Quasi-Newton Langevin Monte Carlo. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 642–651. JMLR.org, 2016.

[183] R.C. Smith. *Uncertainty Quantification: Theory, Implementation, and Applications*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2013.

[184] E. Snelson and Z. Ghahramani. Sparse Gaussian Processes using Pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1257–1264. MIT Press, 2006.

[185] D.J. Spiegelhalter, N.G. Best, B.P. Carlin, and A. Van Der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4):583–639, 2002.

[186] M. Stone. An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):44–47, 1977.

[187] Z. Su, K. Hunter, and R. Shandas. Impact of pulmonary vascular stiffness and vasodilator treatment in pediatric pulmonary hypertension: 21 patient-specific fluid-structure interaction studies. *Computer methods and programs in biomedicine*, 108(2):617–28, 2011.

[188] D.M. Tabima, A. Roldan-Alzate, Z. Wang, T.A. Hacker, R.C. Molther, and N.C. Chesler. Persistent vascular collagen accumulation alters hemodynamic recovery from chronic hypoxia. *J. Biomech*, 45(5):799–804, 2012.

[189] A.A. Taflanidis, J. Zhang, and D. Patsialis. Applications of reduced order and surrogate modeling in structural dynamics. In *Model Validation and Uncertainty Quantification, Volume 3*, pages 297–299. Springer International Publishing, 2020.

[190] M.H. Tawhai, A.R. Clark, and K.S. Burrowes. Computational models of the pulmonary circulation: Insights and the move towards clinically directed studies. *Pulmonary Circulation*, 1(2):224–238, 2011.

[191] C. Taylor and D. Steinman. Image-based modeling of blood flow and vessel wall dynamics: Applications, methods and future directions. *Annals of biomedical engineering*, 38(3):1188–1203, 2010.

[192] L. Tian and N. Chesler. In vivo and in vitro measurements of pulmonary arterial stiffness: A brief review. *Pulmonary circulation*, 2(4):505–517, 2012.

[193] L. Tierney and A. Mira. Some adaptive Monte Carlo methods for Bayesian inference. *Statistics in Medicine*, 18(17-18):2507–2515, 1999.

[194] C. Triantafyllou, J. Polimeni, and L. Wald. Physiological noise and signal-to-noise ratio in fMRI with multi-channel array coils. *NeuroImage*, 55(2):597–606, 2011.

[195] R.M. Tuder. Pulmonary vascular remodeling in pulmonary hypertension. *Cell Tissue Res*, 367(3):643–649, 2017.

[196] B. Turner, P. Sederberg, S. Brown, and M. Steyvers. A method for efficiently sampling from distributions with correlated dimensions. *Psychological methods*, 18(3):368–384, 2013.

[197] A. Utkarsh. The Paraview guide: a parallel visualization application, 2015.

[198] D. Valdez-Jasso. *Modeling and identification of vascular biomechanical properties in large arteries*. PhD thesis, North Carolina State University, Raleigh, NC, 2010.

[199] D. Valdez-Jasso, H.T. Banks, M.A. Haider, D. Bia, Y. Zocalo, R.L. Armentano, and M.S. Olufsen. Viscoelastic models for passive arterial wall dynamics. *Advances in Applied Mathematics and Mechanics*, 1(2):151–165, 2009.

[200] P. van Horssen, M. G. J. T. B. van Lier, J. P. H. M. van den Wijngaard, E. VanBavel, I. E. Hoefer, J. A. E. Spaan, and M. Siebes. Influence of segmented vessel size due to limited imaging resolution on coronary hyperemic flow prediction from arterial crown volume. *American Journal of Physiology-Heart and Circulatory Physiology*, 310(7):H839–H846, 2016.

[201] E.M. van Rikxoort and B. van Ginneken. Automated segmentation of pulmonary structures in thoracic computed tomography scans: a review. *Physics in Medicine and Biology*, 58(17):R187–R220, 2013.

[202] R.R. Vanderpool, A.R. Kim, R. Molthen, and N.C. Chesler. Effects of acute rho kinase inhibition on chronic hypoxia-induced changes in proximal and distal pulmonary arterial structure and function. *Journal of Applied Physiology*, 110(1):188–198, 2011.

[203] J. Vanhatalo, J. Riihimäki, J. Hartikainen, P. Jylänki, V. Tolvanen, and A. Vehtari. GPstuff: Bayesian Modeling with Gaussian Processes. *J. Mach. Learn. Res.*, 14(1):1175–1179, 2013.

[204] V. Vyshemirsky and M. Girolami. Bayesian ranking of biochemical system models. *Bioinformatics (Oxford, England)*, 24:833–839, 2008.

[205] J.X. Wang, C. Roy, and H. Xiao. Propagation of Input Uncertainty in Presence of Model-Form Uncertainty: A Multi-fidelity Approach for CFD Applications. *ASCE-ASME J. Risk and Uncert. in Engrg. Sys., Part B: Mech. Engrg.*, 4(1):011002, 2017.

[206] Z. Wang and N.C. Chesler. Pulmonary vascular wall stiffness: An important contributor to the increased right ventricular afterload with pulmonary hypertension. *Pulm. Circ*, 1(2):212–223, 2011.

[207] Z. Wang, R.S. Lakes, M. Golob, J.C. Eickhoff, and N.C. Chesler. Changes in large pulmonary arterial viscoelasticity in chronic pulmonary hypertension. *PLoS ONE*, 8(11):e78569, 2013.

[208] Z. Wang, S. Mohamed, and N. de Freitas. Adaptive Hamiltonian and Riemann Manifold Monte Carlo Samplers. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages III–1462–III–1470. JMLR.org, 2013.

[209] S. Watanabe. Asymptotic Equivalence of Bayes Cross Validation and Widely Applicable Information Criterion in Singular Learning Theory. *J. Mach. Learn. Res.*, 11:3571–3594, 2010.

[210] S. Watanabe. A Widely Applicable Bayesian Information Criterion. *J. Mach. Learn. Res.*, 14(1):867–897, 2013.

[211] D.G. Whittaker, M. Clerx, C.L. Lei, D.J. Christini, and G.R. Mirams. Calibration of ionic and cellular cardiac electrophysiology models. *WIREs Systems Biology and Medicine*, 12:e1482, 2020.

[212] D.J. Wilkinson. Bayesian methods in bioinformatics and computational systems biology. *Briefings in Bioinformatics*, 8(2):109–116, 2007.

[213] R. Wilkinson. Accelerating ABC methods using Gaussian processes. In Samuel Kaski and Jukka Corander, editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 1015–1023, Reykjavik, Iceland, 2014. PMLR.

[214] C. K. I. Williams and D. Barber. Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351, 1998.

[215] U. Windberger, A. Bartholovitsch, R. Plasenzotti, K. J. Korak, and G. Heinze. Whole blood viscosity, plasma viscosity and erythrocyte aggregation in nine mammalian species: Reference values and comparison of data. *Experimental Physiology*, 88(3):431–440, 2003.

[216] J.L. Wu, J.X. Wang, and H. Xiao. A Bayesian Calibration–Prediction Method for Reducing Model-Form Uncertainties with Application in RANS Simulations. *Flow, Turbulence and Combustion*, 97(3):761–786, 2016.

[217] K. Wu and J. Li. A surrogate accelerated multicanonical Monte Carlo method for uncertainty quantification. *Journal of Computational Physics*, 321:1098 – 1109, 2016.

[218] T. Xu, A.J. Valocchi, M. Ye, and F. Liang. Quantifying model structural error: Efficient Bayesian calibration of a regional groundwater flow model using surrogates and a data-driven error model. *Water Resources Research*, 53(5):4084–4105, 2017.

[219] W. Yang, M. Dong, M. Rabinovitch, F.P. Chan, A.L. Marsden, and J.A. Feinstein. Evolution of hemodynamic forces in the pulmonary tree with progressively worsening pulmonary arterial hypertension in pediatric patients. *Biomechanics and Modeling in Mechanobiology*, 18(3):779–796, 2019.

[220] Z. Yichuan and C.A. Sutton. Quasi-Newton Methods for Markov Chain Monte Carlo. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2393–2401. Curran Associates, Inc., 2011.

[221] P.A. Yushkevich, J. Piven, H.C. Hazlett, R.G. Smith, S. Ho, J.C. Gee, and G. Gerig. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *NeuroImage*, 31(3):1116 – 1128, 2006.

[222] C. Zhang. *Scalable Hamiltonian Monte Carlo via Surrogate Methods*. PhD thesis, University of California, Irvine, 2016.

[223] C. Zhang, B. Shahbaba, and H. Zhao. Hamiltonian Monte Carlo acceleration using surrogate functions with random bases. *Statistics and Computing*, 27(6):1473–1490, 2017.

[224] M. Álvarez, D. Luengo, M. Titsias, and N.D. Lawrence. Efficient Multioutput Gaussian Processes through Variational Inducing Kernels. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 25–32, Chia Laguna Resort, Sardinia, Italy, 2010. PMLR.