Mathematics for Industry Ph.D. Modelling Week

Heart Stream Final Report

Problem brought by Prof. Dirk Husmeier & Prof. Xiaoyu Luo

R. Barry[1], M. Capoccia[2], W. Chen[1], S. Dandapani[3], G. Luraghi[4],
J. Mackenzie[1], P. Mortensen[1], G. Pederzani[3]

Instructors: Hao Gao, Benn Macdonald, Umberto Noè, Xin Zhuan

[1] University of Glasgow, School of Mathematics and Statistics, Glasgow, UK, G12 8QW

[2] University of Strathclyde, Department of Biomedical Engineering, Glasgow, UK, G4 0NW

[3] University of Sheffield, School of Mathematics and Statistics, Sheffield, UK, S3 7RH.

[4] Università degli Studi di Milano, Dipartimento di Matematica, Milan, Italy

October 13, 2016

Contents

1 Introduction

Mathematical modelling and simulation have the potential to become clinically applicable tools for detailed evaluation of the cardiovascular system and clinical decision-making in relation to therapeutic intervention [11]. The increasingly sophisticated mathematical frameworks and related computational simulations are in fact causing a major paradigm shift in many medical disciplines, transforming them more and more from qualitative to quantitative sciences. These mathematical models usually consist of systems of partial differential equations, which sometimes admit an analytical solution, but more often require numerical methods to be solved. Models of this kind have been developed to describe many biomechanical processes, such as the pressure-structure relationship in blood vessels or the physiology and time-varying deformations of entire organs. During the modelling week, we focused on a well-established constitutive model of the heart kinematics [15].

As is usual for these types of model, they depend on a number of patient-specific biomechanical parameters, for example contractility of the ventricles or myocardial stiffness, in the case of the heart. These parameters can be associated with disease status and may therefore be essential for disease prognosis. The problem arises when these parameters cannot be measured in vivo and need to be inferred in some way. What is the best way to infer them?

2 An Introduction to Clinical Applications

The mathematical models discussed in this report haves great potential for application in heart failure where the impact of ventricular assist devices (VADs) has played a significant role as a bridge to transplant and more recently as a long-term solution for non-eligible candidates. Continuous flow rotary blood pumps are currently the most popular devices in view of their smaller size, increased reliability, and higher durability compared to pulsatile flow devices. The trend towards their use is increasing. Mathematical modelling and computer simulation are invaluable tools to investigate the interactions between VADs and the cardiovascular system ([8], [9]). Simulations for continuous flow VADs are usually performed under steady flow conditions where the inlet boundary is set to a specific steady velocity profile while the outlet boundary is fixed at a steady pressure.

The unsteady boundary conditions will be dependent on the flow from the heart and the aortic pressure, which are the result of the interactions between the device and the cardiovascular system. The study of the interaction between left ventricular assist devices (LVADs) and the whole cardiovascular system with a 3D CFD model is highly demanding although limited parts of the assisted circulation have been developed with this method [4]. A more simplified approach consists of using one-dimensional (1D) or lumped parameter (0D) models. In a 1D model space dependence is confined to the axial coordinate while in a 0D model space dependence is addressed by splitting the cardiovascular system in compartments. According to Euler's equation, the pump pressure head can be estimated by a linear relationship with the flow rate and a quadratic relationship with the rotational speed as follows:

$$H = a_1 Q + c_1 \omega^2. \tag{1}$$

A basic model assumes the right ventricle and the pulmonary circulation to be normal; therefore, their effect on the LVAD is negligible. The left ventricle is modelled according to the time-varying elastance theory ([41], [43], [42]) as follows:

$$E(t) = \frac{P_{LV}(t)}{V_{LV}(t) - V_0}, \tag{2}$$

where $E(t)$ is the left ventricular elastance, $P_{LV}(t)$ is the left ventricular pressure, $V_{LV}(t)$ is the left ventricular volume, and V_0 is the theoretical ventricular volume at zero pressure.

In order to improve the accuracy of pressure head estimation, additional parameters can be added such as a quadratic function of the flow rate to account for viscous loss across the pump, the time derivative of flow rate for energy loss secondary to fluid inertia and time derivative of rotational speed for pulsating operation mode ([46], [47], [21], [32], [40]). The outcome is consistent with improved accuracy as shown by curve fitting and statistical analysis [36] although complexity is increased. An axial flow pump can be characterised by the relation between pressure difference, flow and speed ([7], [17]). Under steady state conditions, a linear relationship with the flow and a quadratic relationship with the pump speed are observed. In mathematical terms, we have:

$$H = P_o - P_i = b_0 Q + b_1 \frac{\mathrm{d}Q}{\mathrm{d}t} + b_2 \omega^2. \tag{3}$$

Where P_o and P_i are pressures at the outlet and the inlet of the pump; $\mathrm{d}Q/\mathrm{d}t$ is the time derivative of flow; $b_0,\ b_1$, and b_2 are pump-dependent constant coefficients. The pressure difference is linearly dependent on flow derivative. The model identifies the characteristics of the pump with the state variables related to its operation and provides estimates of flow and pressure difference of the axial pump.

VADs operate in the transitional region at the boundary of laminar and turbulent flow (low Reynolds number). If the pump flow pattern is expected to be turbulent, then the pump pressure loss is related to the square of the flow. The LVAD governing equation becomes:

$$P_{Ao} - P_{LV} = R \cdot Q^2 + L \frac{\mathrm{d}Q}{\mathrm{d}t} + \beta \omega^2. \tag{4}$$

Continuous flow blood pumps operate at constant speed with a flow waveform related to the waveform of the pressure difference between the pump inlet (left ventricular pressure) and outlet (aortic pressure) ([32], [20]). Therefore, the pump flow is affected by the pulsatility of the left ventricular pressure and the features of the pump itself. Although the time-varying elastance theory has been very successful with the modelling and simulation of left ventricular function, it may not be completely suitable for the modelling of the complex interactions between the left ventricle and a LVAD [48]. A nonlinear time-varying lumped parameter model of the cardiovascular system [44] can be modified to include the interventricular septum and a rotodynamic, continuous flow LVAD [49]. This is a more accurate heart failure model where the ESPVR is a unimodal function that takes into account the descending limb of the Frank- Starling curve. In addition, it is particularly suitable to study ventricular interactions and the leftward septal shift secondary to left ventricular decompression following LVAD insertion. Outlet tube pump modelling with a linearly flow-dependent resistance is also used to study pulsatile and continuous flow LVADs. The resistance consists of a flow-proportional and a constant components in the context of a steady state environment but dynamic modelling with a time-varying resistance has been considered more recently ([22], [18]). A Lagrange multiplier coupling approach [35] can be applied to LVAD modelling [30] using fictitious domain methods ([3], [45]) to address the interactions between the LVAD cannula and the ventricle at the expense of significant increase in computational time and instability in specific regions of interface between the fluid and solid meshes. To overcome these limitations, a fluid-solid left ventricular model coupled with a 0D Windkessel model ([25], [26]) already successfully applied to different types of LVAD ([38], [39], [37]) can be used where optimisation with high order interpolation at the fluid-solid boundary allows simulations of fluid-solid interaction over a complete cardiac cycle during LVAD support [31].

A simulation based approach as a potential preoperative strategy in the context of patient-specific modelling may be an additional tool to obtain accurate predictions of the performance of these devices in a clinical setting. Besides, it would be a valuable training opportunity for medical and nursing staff involved with the care of this complex and challenging group of patients [14]. Interactive software like CARDIOSIM© [10], CircAdapt ([1], [29]), Harvi and HemoLab ([27], [6]) can reproduce physiological and pathological conditions for educational purposes and clinical decision-making in a controlled environment. Examples are the study of the interactions with pulsatile VADs, continuous flow VADs and the intra-aortic balloon pump (IABP); simulation of the effects of ageing, vasodilatation, vasoconstriction, rest and exercise and calculation of characteristic impedance of the arterial network ([27], [6]).

A combination of models can be coupled locally and globally in order to obtain the systemic response of the cardiovascular system: the so called 3D-1D-0D models. The propagation of the arterial pulse can be represented with a 1D-0D model, which describe the behaviour of the flow rate, mean pressure and cross-sectional area as a function of time. Typical governing equations are [27]:

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial z}\left(\frac{Q^2}{A}\right) = -\frac{A}{\rho}\frac{\partial P}{\partial z} - \frac{2\pi R \tau_0}{\rho} \tag{5a}$$

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial z} = 0 \tag{5b}$$

$$P - P_0 = \frac{E\pi R_0 h}{A}\left(\sqrt{\frac{A}{A_0}} - 1\right) + \frac{k\pi R_0 h}{A}\frac{1}{2\sqrt{A_0 A}}\frac{\partial A}{\partial t} \tag{5c}$$

where Q is the flow rate, P the mean pressure and $A = \pi R^2$ is the cross sectional area with radius R. The parameters E and k are the Young modulus and fluidity of the arterial wall. Mass conservation and pressure continuity are imposed at the bifurcations while inflow/outflow boundary conditions for the arterial system are modelled with 0D models. The governing equations are [27]:

$$P = P_t \tag{6a}$$

$$Q + Q_t = 0 \tag{6b}$$

$$\frac{\mathrm{d}Q_t}{\mathrm{d}t} = \frac{1}{R_1} + \frac{\mathrm{d}P_t}{\mathrm{d}t} + \frac{P_t}{R_1 R_2 C} - \frac{R_1 + R_2}{R_1 R_2 C} Q_t \tag{6c}$$

where R_1, R_2, and C are lumped parameters for the peripheral circulation.

Numerical approximations of the 3D Navier-Stokes equations allow the study of the complex blood flow in a chosen arterial vessel. The governing equations are [27]:

$$\rho\frac{\partial v}{\partial t} + \rho(\nabla \mathbf{v})(\mathbf{v} - \mathbf{u}) = -\nabla p + \mu\Delta\mathbf{v} + \rho\mathbf{g} \qquad \text{in } \Omega \tag{7a}$$

$$\text{div } \mathbf{v} = 0 \qquad \text{in } \Omega \tag{7b}$$

$$\mathbf{v}_{|\Gamma_w} = \mathbf{v}_w \qquad \text{on } \Gamma_w \tag{7c}$$

$$p - p_0 = \frac{Eh}{R_0^2}\delta + \frac{kh}{R_0^2}\frac{\partial \delta}{\partial t} \qquad \text{on } \Gamma_w \tag{7d}$$

where $\mathbf{v}$ and p are the blood velocity and pressure, ρ is is the density, and μ the dynamic viscosity. Ω is the 3D spatial domain moving with velocity $\mathbf{u}$, lateral boundary Γ_w (arterial wall) and inflow/outflow boundaries $\Gamma_{i/o}$. All of these equations allow the mathematical setting for 3D-1D-0D models in order to simulate the interaction between local and global events in the circulation under accurate physiological conditions.

3 Aims

The aim of the project is to find an accurate and computationally efficient way to infer patient-specific parameters, given the information provided by patient-specific data (e.g. MRI scans) and constitutive mathematical models of the processes or organs of interest (e.g. electrophysiological and mechanical models of the heart) that depend on those parameters. The usual strategy is to measure other quantities of interest described by the model and find the parameter values that minimize the mismatch between the data and the output predicted by the mathematical model. This approach, however, has several drawbacks. Firstly, the number of parameters can be very high and their interdependence complex, which can cause the inference to be inaccurate. Secondly, and this is the biggest obstacle, every new parameter adaptation requires a new associated numerical solution of the PDEs describing the model, which can be computationally very expensive thus rendering this strategy futile when an immediate decision needs to be made. Therefore, although mathematical models are successful in solving the forward problem (given the parameters, find the output), the backward problem (where we must use given the data to find the parameters) is much harder and requires advanced statistical techniques to provide a valuable clinical decision support system. During the modelling week we explored state-of-the-art parameter inference techniques that could provide a viable solution to the problem of fast and accurate parameter estimation.

4 Methodology

The main problem with the strategy mentioned before is the extremely high computational cost of iteratively solving a system of PDEs for a sequence of parameter values, that will hopefully converge to the parameter vector that minimizes the mismatch between the data and the simulation. In order to overcome this problem, we used a state-of-the-art technique in statistical inference called adaptive gradient matching.

Adaptive gradient matching is a powerful technique that completely bypasses the phase of numerically solving the system of PDEs, thus drastically decreasing computational costs while still yielding accurate solutions. In a nutshell, instead of minimizing the distance between the data and the numerical solution, gradient matching aims at minimizing the *gradients* of the two. In principle, this would still require knowing the solution to the mathematical model, but, as we will see in the next sections, our approach made use of an *emulator*, which in essence is a proxy of the numerical solution and is obtained by computing the solutions of the model for a grid of values in the parameter space and interpolating them.

In this Section we will explain in more detail how this algorithm works. First, we will introduce Gaussian processes and explain why we think they are the best choice for our problem. Next, we will introduce and explain the concept of an emulator, and finally, we will present the method of adaptive gradient matching. As an example, we applied these methods to two models of the heart: one describing its kinematics, namely its deformation when subject to varying pressure, and one describing its electrophysiology, namely the voltage and recovery variable as functions of time. The last subsection therefore presents these models.

4.1 Non-Parametric Modelling with Gaussian Processes

We begin our discussion of nonparametric modelling with Gaussian processes by giving a brief overview of the differences between a classical frequentist approach and a Bayseian approach to statistical interference. For this we follow the structure of [23]. We then move on to consider linear regression in an abstract setting, using this to motivate the Gaussian perspective. In this we will largely follow the formulation found in [5]. We then move onto Gaussian processes in a more concrete setting, illustrating their advantages over other curve fitting techniques and some variations upon this method that we can use to model different data. Finally we move onto a toy model of the heart using Gaussian processes and present some results thereof.

Consider a situation in which you toss a coin several times. This is an experiment with two distinct outcomes: heads (which we denote by 1) and tails (denoted by 0). We can record the result of this experiment and call the recorded results data $\mathcal{D}$. One possible example of the data with seven tosses is $\mathcal{D} = \{1, 0, 0, 1, 1, 1, 1\}$. The probability of observing the data from our experiment, $P(\mathcal{D}|\theta)$, is

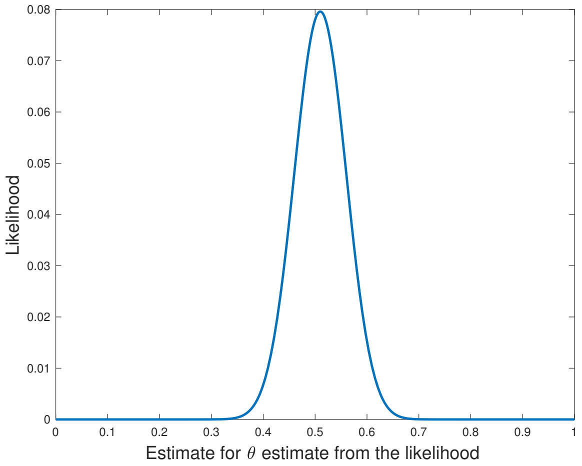


Figure 1: Parameter estimate versus the likelihood for $\hat{\theta} = 0.5$.

called the *likelihood*, where θ is the probability of observing a head. The likelihood is given by

$$P(\mathcal{D}|\theta) = \binom{N}{k} \theta^k (1-\theta)^{N-k}, \tag{8}$$

where N is the number of observations made, i.e. the number of elements of $\mathcal{D}$, and k is the number of observed heads, which is a random variable. A plot of equation 8 can be found in Figure (1) for 100 pseudorandom draws from a Gaussian distribution. Since the true value of θ is not usually known, we need to develop some way of estimating this. We denote the best estimate by $\hat{\theta}(\mathcal{D})$. In this context, "best" refers to the value for $\hat{\theta}$ that allows us to best model the data. A standard approach is to choose the value of $\hat{\theta}$ that maximises the likelihood, this is known as the *maximum likelihood*. It is worth noting that, for large sample sizes, this estimate is unbiased; however, this result breaks down for small sample sizes. The essential idea of the classical frequentist approach – that we're currently interested in – is that the value of θ (although it may be unknown) is fixed, and that we can seek it using, for example, a maximum likelihood approach. We can do this in the following way. Begin by taking a log transformation:

$$\log\left(P(\mathcal{D}|\theta)\right) = \log \binom{N}{k} + k \log\theta + (N-k)\log(1-\theta). \tag{9}$$

Given an elementary knowledge of differential calculus, we can see that the maximum of this will occur when $\mathrm{d}\log\left(P(\mathcal{D}|\theta)\right)/\mathrm{d}\theta = 0$:

$$\begin{aligned}\frac{\mathrm{d}\log\left(P(\mathcal{D}|\theta)\right)}{\mathrm{d}\theta} &= k\frac{\mathrm{d}\log\theta}{\mathrm{d}\theta} + (N-k)\frac{\mathrm{d}\log(1-\theta)}{\mathrm{d}\theta} \\ &= \frac{k}{\theta} - \frac{N-k}{1-\theta} = 0.\end{aligned}$$

Rearranging this gives

$$\theta = \frac{k}{N}.$$

This tells us that the best estimator for θ, the probability of observing a head, is given by the ratio of the number of heads tossed (k) to the total number of tosses (N).

The point that I wish to illustrate with the example of the coin flip is the difference between the perspective of a pure frequentist approach compared to that of Bayesian inference. To a frequentist, the only meaningful definition of probability of an event is the frequency with which it occurs over a long sequence of independent trials. However, arbitrarily many independent trials are not always possible or practical. A frequentist would accept the probability of tossing a heads with a fair coin (being $\frac{1}{2}$) as a true probability, but would not recognise "the probability of voting Labour in the last election" as a probability [16]. Most people would agree that this probability is meaningful, since it expresses the degree of belief in a proposition regarding the world – these are known as subjective uncertainties.

In Bayesian inference we consider the probability of an event as a random variable complete with some distribution. This is not possible in the frequentist framework, and it affords us some key advantages – for instance, we can now discuss unique and non-repeatable events. As part and parcel of a Bayesian approach we require prior distributions for input data (discussed shortly), which represent our preconceived notions of an event.

Armed with a Bayesian perspective, we can motivate the Gaussian perspective by considering linear regression, in so doing deriving the predictive distribution. We wish to be able to model some, as yet unknown data, using a predictor $y(\mathbf{x})$ where $\mathbf{x} \in \mathcal{D}$ is a vector of length m, and $\mathcal{D}$ is a set of data from, for example, experiments. We can do this with a linear predictor, namely

$$y(\mathbf{x}) = \boldsymbol{\beta}^T \mathbf{x}$$

where $\boldsymbol{\beta}$ is also a vector of parameters of length m and is called the weight. Obviously this is a linear function of x, thus we need to modify our linear predictor if we need it to better model nonlinear data. This is done simply by replacing $\mathbf{x}$ with a vector of basis functions $\phi(\mathbf{x})$. We can use, for example, polynomials in x:

$$\phi(x) = \begin{pmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^n \end{pmatrix}.$$

So we can write

$$y(\mathbf{x}) = \boldsymbol{\beta}^T \phi(\mathbf{x}), \tag{10}$$

for some $\phi(\cdot)$ and $\boldsymbol{\beta}$ which I assume are given. We consider a prior distribution over $\boldsymbol{\beta}$ given by an isotropic Gaussian of the form

$$\boldsymbol{\beta} \sim \mathcal{N}\left(\boldsymbol{\beta} | \mathbf{0}, \sigma^{-1}\mathbf{I}\right), \tag{11}$$

governed by a hyperparameter σ which represents the variance of the distribution [5]. For any value of $\boldsymbol{\beta}$, equation (10) returns a function of $\mathbf{x}$. The probability distribution over β thus induces the same distribution over the function $y(\mathbf{x})$. In practice we evaluate this at specific values of $\mathbf{x}$, such as those we gain from experimental data, we call these $\mathbf{x}_1, \ldots, \mathbf{x}_n$. We can then evaluate y at these to get $y(\mathbf{x}_1), \ldots, y(\mathbf{x}_n)$. This is written in matrix form as

$$\begin{aligned}\mathbf{y} &= \begin{pmatrix} y(\mathbf{x}_1) \\ \vdots \\ y(\mathbf{x}_n) \end{pmatrix} = \begin{pmatrix} \boldsymbol{\beta}^T \phi(\mathbf{x}_1) \\ \vdots \\ \boldsymbol{\beta}^T \phi(\mathbf{x}_n) \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^m \beta_i \phi(x_{1i}) \\ \vdots \\ \sum_{i=1}^m \beta_i \phi(x_{ni}) \end{pmatrix} \\ &= \begin{pmatrix} \phi(x_{11}) & \cdots & \phi(x_{1m}) \\ \vdots & & \vdots \\ \phi(x_{n1}) & \cdots & \phi(x_{nm}) \end{pmatrix} \boldsymbol{\beta} = \Phi \boldsymbol{\beta} \end{aligned} \tag{12}$$

Φ is called the design matrix with elements $\Phi_{nk} = \phi_k(\mathbf{x}_n)$. Since $\mathbf{y}$ is a linear combination of normally distributed elements with mean 0 and variance σ, $\mathbf{y}$ itself is normally distributed , i.e.

$$\mathbb{E}\left[\mathbf{y}\right] = \mathbb{E}\left[\Phi\boldsymbol{\beta}\right] = \Phi\mathbb{E}\left[\boldsymbol{\beta}\right] = 0 \qquad \text{by eqn. (11)} \tag{13a}$$

$$\mathrm{cov}[\mathbf{y},\mathbf{y}] = \mathbb{E}\left[\left(\mathbf{y} - \mathbb{E}\left[\mathbf{y}\right]\right)\left(\mathbf{y} - \mathbb{E}\left[\mathbf{y}\right]\right)^{\mathbf{T}}\right] = \mathbb{E}\left[\mathbf{y}\mathbf{y}^{\mathbf{T}}\right] = \mathbf{\Phi}\mathbb{E}\left[\boldsymbol{\beta}\boldsymbol{\beta}^{\mathbf{T}}\right]\mathbf{\Phi}^{\mathbf{T}} = \frac{\mathbf{1}}{\sigma}\mathbf{\Phi}\mathbf{\Phi}^{\mathbf{T}} =: \mathbf{K}. \tag{13b}$$

$\mathbf{K}$ is called the *gram matrix* where $\mathbf{K} = [k_{ij}] = [k(\mathbf{x}_i, \mathbf{x}_j)]$, k is a kernel function. K is obviously a symmetric matrix whose entires are scaled with σ, the covariance of the entries of $\boldsymbol{\beta}$, so it is easy to see that the functions $k(\cdot)$ are also scaled with this variance, hence the function $k(\cdot)$ is inextricably linked to the covarience of some multivariate Gaussian process (GP). By Gaussian process we mean a stochastic process $\{f(\mathbf{x}), \mathbf{x} \in \mathcal{D}\}$ where each $f(\mathbf{x}_i)$ is normally distributed with mean μ and covariance matrix $\mathbf{K}$ [34]. A Gaussian stochastic process is completely specificed by its mean and covariance matrix [5], we usually have no knowledge about the mean of $y(\mathbf{x})$, but without loss of generality we can take this to be 0. We finish specifying the GP by noting that

$$\mathrm{cov}[y_n, y_m] = k(\mathbf{x}_n, \mathbf{x}_m) \tag{14}$$

and computing this at any two values of $\mathbf{x}$ gives the covariance of $y(\mathbf{x})$.

To return to the issue of interpolation, suppose that we wish to use a GP to predict y from some given predictor $\mathbf{x}$. A predictor is any function $\eta(\mathbf{x})$. We wish to find the difference between y and the predictor η, which we measure through the mean squared error (MSE), defined in the usual way:

$$\mathbb{E}\left[||y - \eta(\mathbf{x})||^2\right]. \tag{15}$$

Our goal is to find the *best predictor* of y, namely the function $\mu(\mathbf{x})$ that minimises the MSE. We can rewrite the MSE as follows:

$$\begin{aligned}\mathbb{E}_y\left[\left(y-\eta(\mathbf{x})\right)^2\right] &= \mathbb{E}_y\left[\eta(\mathbf{x})^2\right] - 2\mathbb{E}_y\left[y\eta(\mathbf{x})\right] + \mathbb{E}_y\left[y^2\right] \\ &= \mathbb{E}_y\left[\eta(\mathbf{x})^2\right] - \mathbb{E}_y\left[\eta(\mathbf{x})\right]^2 + \mathbb{E}_y\left[\eta(\mathbf{x})\right]^2 - 2\mathbb{E}_y\left[y\eta(\mathbf{x})\right] + \mathbb{E}_y\left[y^2\right] \\ &= \left(\mathbb{E}_y\left[\eta(\mathbf{x})^2\right] - \mathbb{E}_y\left[\eta(\mathbf{x})\right]^2\right) + \left(\mathbb{E}_y\left[\eta(\mathbf{x})\right] - \mathbb{E}_y\left[y\right]\right)^2.\end{aligned}$$

Here it is crucial to note that subscript y denotes that the sample in $\mathbf{x}$ comes from the distribution with true y i.e. $\mathbb{E}_y[y] = y$ which allows us to write $\mathbb{E}_y[y\eta(\mathbf{x})] = y\mathbb{E}_y[\eta(\mathbf{x})]$. And we can finally write

$$\mathbb{E}_y\left[\left(y-\eta(\mathbf{x})\right)^2\right] = \text{var}\left(\eta(\mathbf{x})\right) + \mathbb{E}_y\left[\eta(\mathbf{x}) - y\right]^2. \tag{16}$$

This is minimised when $\eta(\mathbf{x}) = \mu(\mathbf{x}) = \mathbb{E}[y|\mathbf{x}]$, $\mu(\mathbf{x})$ is the conditional mean of y given $\mathbf{x}$ [28].

Linear regression allows us to approximate the conditional mean as either a linear (when $\phi(\mathbf{x}) = \mathbf{x}$) or nonlinear function of $\mathbf{x}$ (when $\phi(\mathbf{x}) \neq \mathbf{x}$) as per equation 10 [34]. In both cases we assume independent and identically distributed (iid) errors:

$$y(x_i) = \phi(x_i)^{\mathrm{T}}\boldsymbol{\beta} + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}\left(0, \omega^2\right), \quad i = 1, \ldots, n.$$

This assumption does not hold in the setting of deterministic simulators, where the errors need not be independent. Conceptually it makes sense that errors (ϵ_i) from measurements (x_i) that occur close together (in the sense of the L2 norm) should be dependent on one another in some way; similarly, errors in measurements that occur far apart can bear little relation. That is, we can more realistically model the outcome y using correlated errors. We represent this correlation of data in input space using the kernel function k [34]:

$$y(\mathbf{x}) = \mu(\mathbf{x}) + \epsilon(\mathbf{x}), \qquad \epsilon(\mathbf{x}) \sim \mathcal{N}\left(0, k(\mathbf{x}, \mathbf{x})\right). \tag{17}$$

We are relatively free to choose the kernel with which we want to work, and there are many to choose from. This is what we explore in the next section.

We begin by considering the Squared Exponential (SE) kernel, as it is simple and widely used. The SE kernel is given in [34] as

$$k_{SE}\left(\mathbf{x}, \mathbf{x}'\right) = a^2 \exp\left\{-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\lambda^2}\right\} \tag{18}$$

with positive hyperparameters $\theta = \{a, \lambda\}$. The length scale of our interpolant is given by λ, and a the amplitude.

It is now straightforward to plot the relationship between the distance $||\mathbf{x} - \mathbf{x}'||$ and the covariance k_{SE}. The result of this (with $a = \lambda = 1$) is shown in Figure 2.

With similar ease, we can visualise the effect that varying the hyperparameters has on the interpolant given some normally distributed, multivariate noise. This is illustrated in Figures 3 and 4

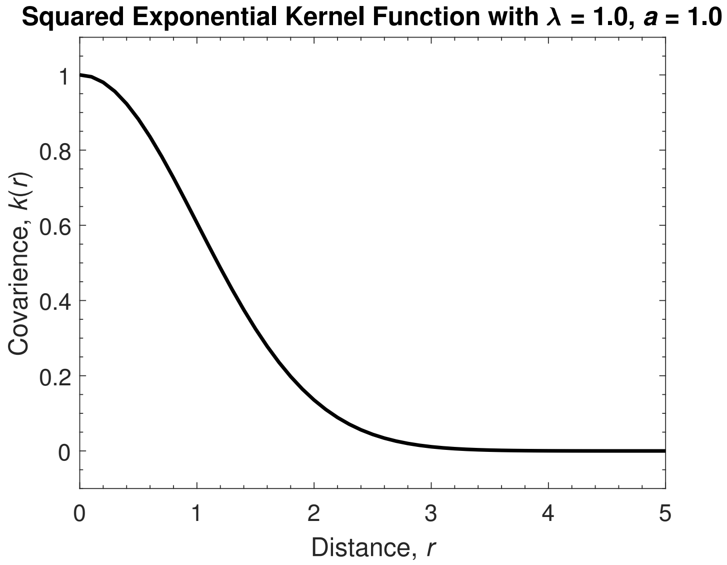


Figure 2: Covariance of the squared exponential kernel for $\lambda = a = 1$.

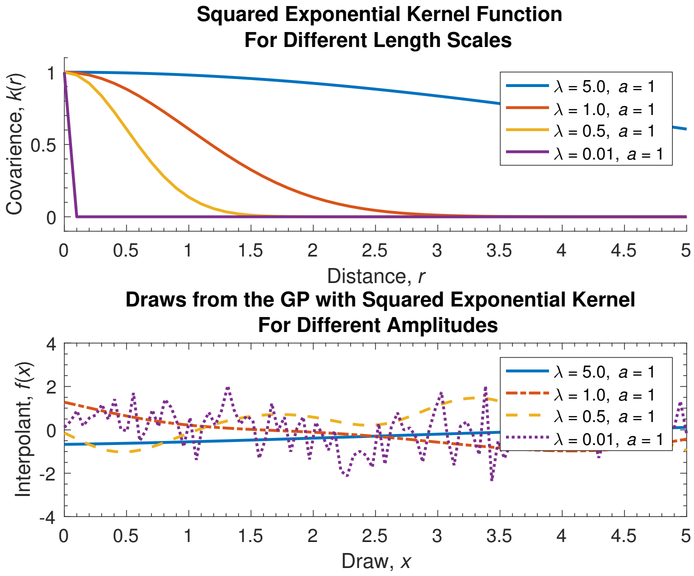
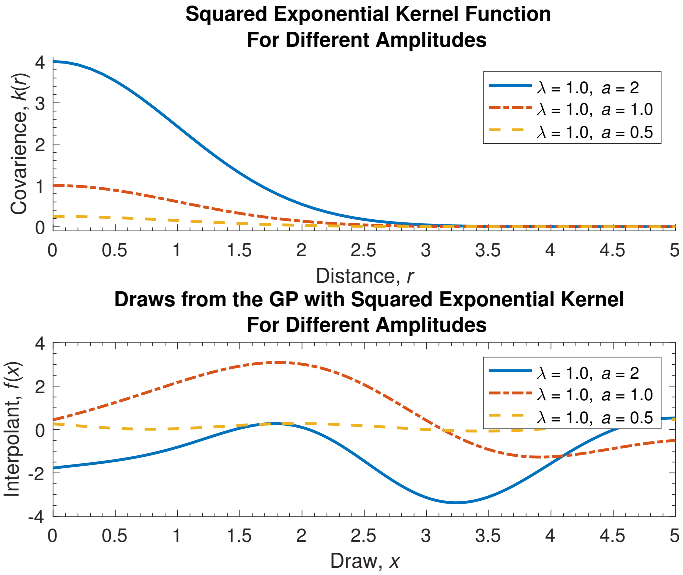


Figure 3: Covariance versus distance (upper) and an interpolant (lower) for different length scales ($\lambda = 5,\ 1,\ 0.5,\ 0.01$).


Squared Exponential Kernel Function For Different Amplitudes

Draws from the GP with Squared Exponential Kernel For Different Amplitudes

Figure 4: Covariance versus distance and an interpolant for different amplitudes ($a = 2,\ 1,\ 0.5$).

for a few different length scales ($\lambda = 5,\ 1,\ 0.5,\ 0.01$ with constant amplitude $a = 1$) and amplitudes ($a = 2,\ 1,\ 0.5$ with constant length scale $\lambda = 1$) respectively.

The behaviour of the interpolants subject to changes in the hyperparameter values is what we expect, given the form of the squared exponential kernel. That is, it is natural to expect that increasing the amplitude a will lead to an interpolant with increased amplitude, this can be seen in the lower part of Figure 4 where there is a marked increase in amplitude of the interpolant when comparing the curves with, for example $a = 2$ and $a = 0.5$. There is an analogous observation to be made between the interpolants and covarience plots for different length scales: the covarience becomes tail heavy (i.e. the covarience decreases increasingly quickly with increasing distance) as the length scale decreases. This property gives rise to different appearances in the interpolants: for small length scales, for example $\lambda = 0.01$, the interpolant looks similar to white noise or a random walk, whereas for $\lambda = 5.0$ the interpolant is nearly linear. This ability to choose amplitude and length scale allows us to model and interpolate different data that exhibits different behaviours with ease, as we need not come up with a new interpolation scheme for each new data set. Herein lies part of the power of Gaussian processes.

So far in our discussion of kernel methods we have only considered the squared exponential kernel, but other kernels exist and are widely employed. An example of this is kernels of the Matérn class:

$$k_\nu(\mathbf{x}, \mathbf{x}') = a^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left[\frac{\sqrt{2\nu}||\mathbf{x} - \mathbf{x}'||}{\lambda} \right]^\nu K_\nu \left(\frac{\sqrt{2\nu}||\mathbf{x} - \mathbf{x}'||}{\lambda} \right) \tag{19}$$

where ν is the *degree* of the kernel, and K_ν is a modified Bessel function. This gives rise to a wide

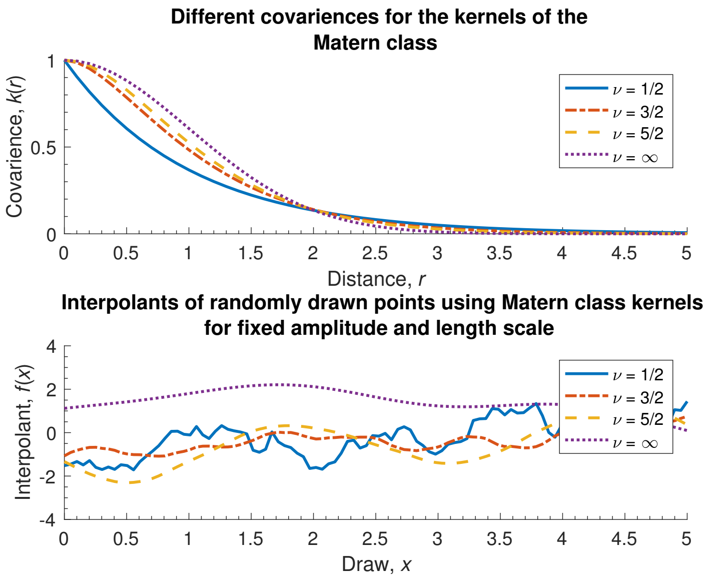


Figure 5: Covariance versus distance for different values of ν in the Matérn class (upper), and interpolants for random draws using the same kernels.

class of functions, for example: $\nu = \frac{1}{2}$ gives the Ornstein-Uhlenbeck process [34]:

$$k_{\nu=\frac{1}{2}}(\mathbf{x}, \mathbf{x}') = a^2 \exp\left\{-\frac{||\mathbf{x}-\mathbf{x}'||}{\lambda}\right\}.$$

The kernels of the Matérn class can be deployed in similar ways to the squared exponential kernel. For instance we can visualise the how the covarience changes with the distance between points, as in Figure 2. However it is more interesting to compare the covarience versus distance curves of each kernel to each other, thereby illustrating the effect of ν in the kernel. This is shown in Figure 5. It is interesting to note that as ν tends to infinity, the kernels of the Matérn class tend to the squared exponential kernel.

The final class of kernels that we wish to discuss are those of the rational quadratic class. These, too, take positive hyperparameters $\theta = \{\lambda, a\}$, but also α. In general, a rational quadratic kernel is defined as follows [34]:

$$k_{RQ}(\mathbf{x}, \mathbf{x}') = a^2 \left(1 + \frac{||\mathbf{x}-\mathbf{x}'||^2}{2\alpha\lambda^2}\right)^{-\alpha}. \qquad (20)$$

In the same vein as the other kernels we have seen, we can plot the covariance, k_{RQ}, as a function of distance r in order to compare the behaviour of the kernel given different values of α. This is shown in Figure 6. It is interesting to note that as α increases, the kernels of this class tend to the Squared Exponential kernel (see Figure 7).

Given that we have examined some kernel functions and are interested in using Gaussian processes to emulate data, it is natural to wonder why we use different kernels at all. The reason is a

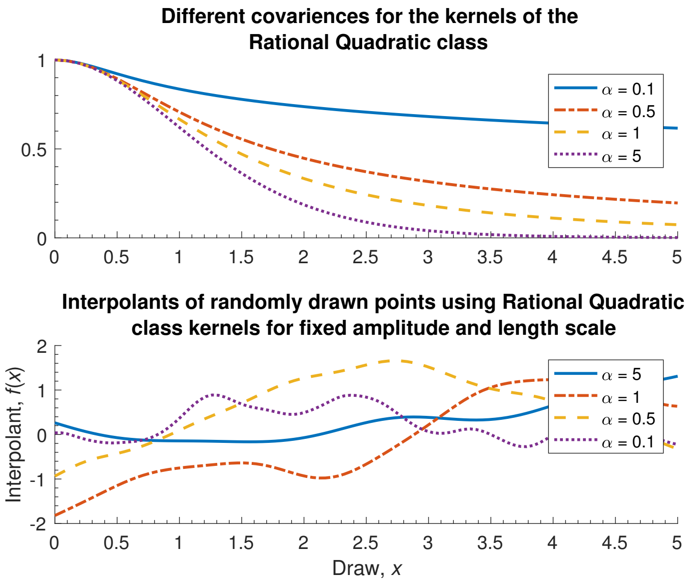


Figure 6: Covariance versus distance for different values of α in the Rational Quadratic class.

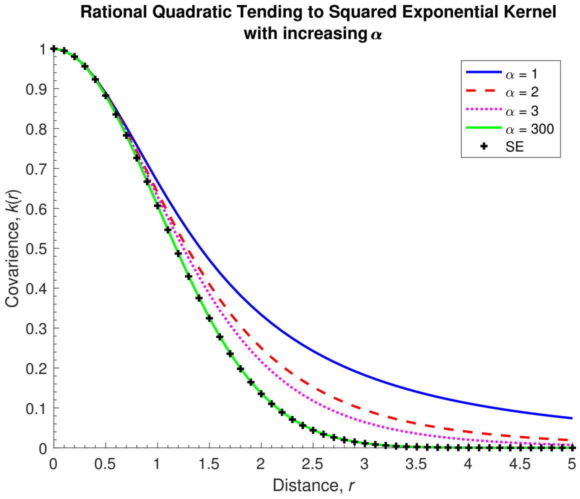


Figure 7: Covariance versus distance for different values of α in the Rational Quadratic class.

relatively simple one and is illustrated (at least in part) in Figures 3 – 6 above, i.e. we can model different behaviours with ease and efficiency simply by choosing an appropriate kernel class and hyperparameters. We do this by understanding the influence of the hyperparameters on the interpolant that is obtained. Consider figure 8, from this and previous figure we can see that the length scale λ controls the "wiggliness" of the interpolant that is obtained, which makes intuitive sense given both the position of λ in the equations of the various kernels that we have seen, and its name. Similarly, we can control the amplitude of the interpolant using the hyperparameter a, this can be seen from Figure 4 and is an ultimately obvious given the formulae for the kernels. We are also able to compose kernels with a binary operator (e.g. addition or multiplication) to form a new composite kernel, as shown in Figure 10 where the covariances of the squared exponential kernel (solid black) and the Matérn class kernel with $\nu = \frac{1}{2}$ (dash/dot blue) both with $a = \lambda = 1$. The other curves shown are the covariances of the sum and product of these, given by k_{sum} and k_{prod} respectively:

$$k_{sum}(\mathbf{x}, \mathbf{x}') = a^2 \left(\exp\left\{ -\frac{||\mathbf{x} - \mathbf{x}'||}{\lambda} \right\} + \exp\left\{ -\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\lambda^2} \right\} \right) \tag{21a}$$

$$k_{prod}(\mathbf{x}, \mathbf{x}') = a^4 \exp\left\{ -\frac{||\mathbf{x} - \mathbf{x}'||}{\lambda} \left(1 + \frac{||\mathbf{x} - \mathbf{x}'||}{2\lambda} \right) \right\} \tag{21b}$$

Indeed, we are not restricted to only one set of hyperparameters when composing kernels, we are free to take entirely different sets θ_i, $i = 1, ..., n$ where n is the number of kernels in the composition.

We have now discussed kernels, Gaussian processes, and interpolation using these in some depth, but have yet to apply them to any Heart model. This will come shortly, but first we wish to revisit curve fitting using Gaussian processes with given data. During the lab session that covered this area of material, we were provided a selection of MATLAB files and code that were used to produce the following figures by the instructor Umberto Noè. All of what follows is effectively a reproduction of what is found in his notes: [34]. He also provided us with a toy simulator, to us this was a black box and we do not discuss its contents.

From the toy simulator, we obtain the scatter graph shown in Figure 11a. We can then perform Gaussian regression on this data to get a curve of best fit with a 95% confidence interval, this is shown in Figure 11b.

This was achieved by using the inbuilt MATLAB function "fitrgp" which fits a Gaussian process to some given data x and y. MATLAB has a number of inbuilt kernel functions, we opted for the Squared exponential kernel that was discussed previously. The choice of kernel is an important one as different kernels give rise to different fittings. The choice of a particular kernel reflects your beliefs about how the data behave, this is the introduction of prior knowledge. This point is easily illustrated by choosing a (few) different kernel and comparing the fit that we get with each, as shown in Figure 12. Here we are comparing the fit and confidence intervals between the Matérn 3/2, Matérn 5/2, and squared exponential. We can employ analytical methods to determine which kernel best suits our purpose, this is done with a combination of, for example, the log likelihood

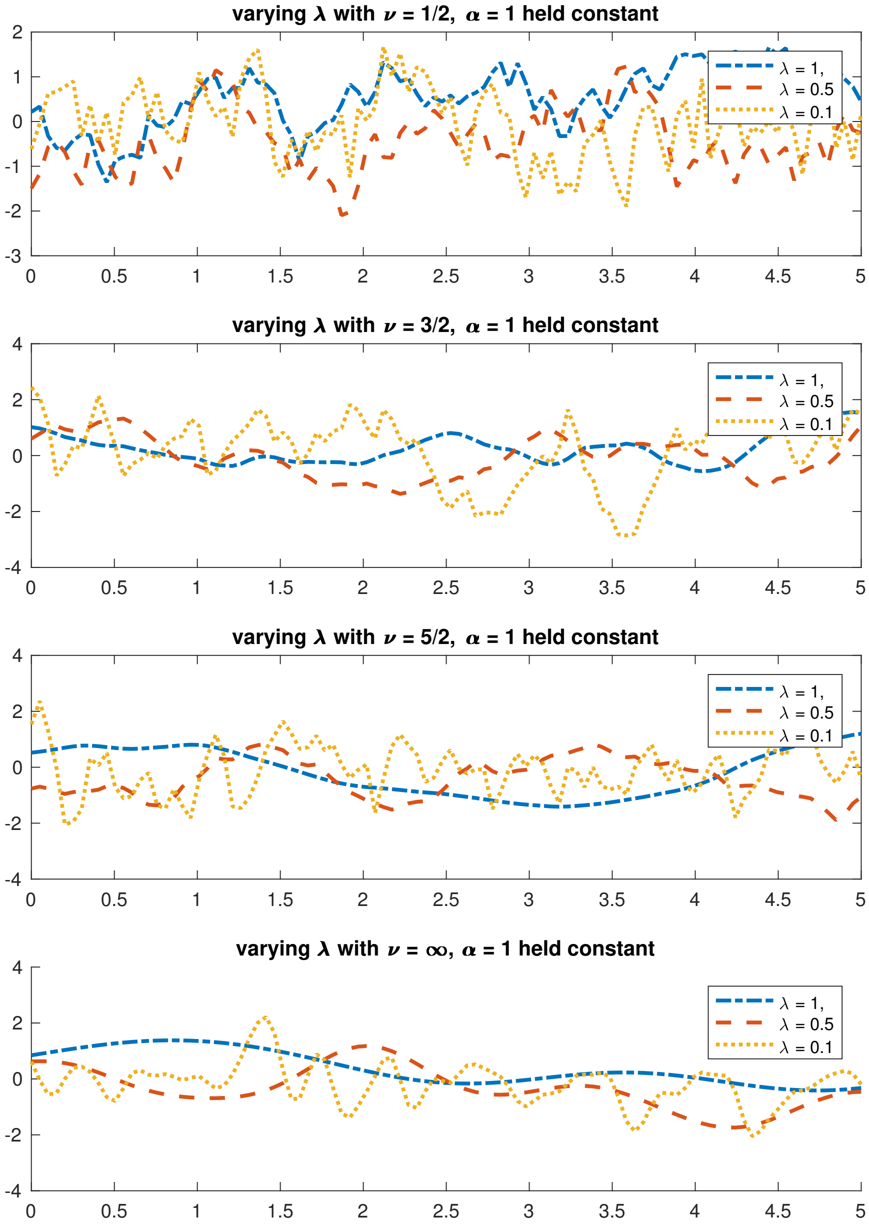


Figure 8: Matern class kernel interpolations for decreasing ν and λ.

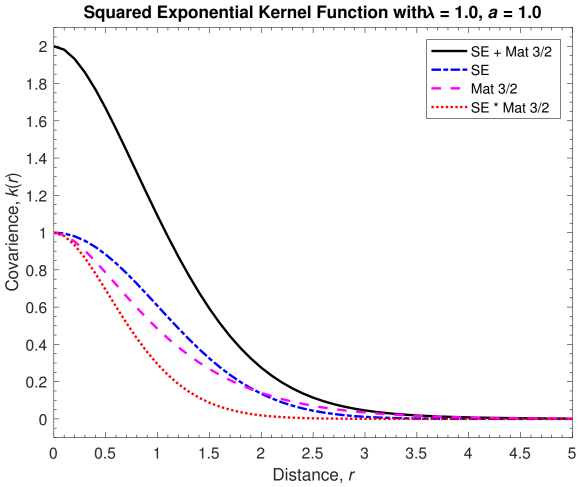


Figure 9: Matern class kernel interpolations for decreasing ν and λ.

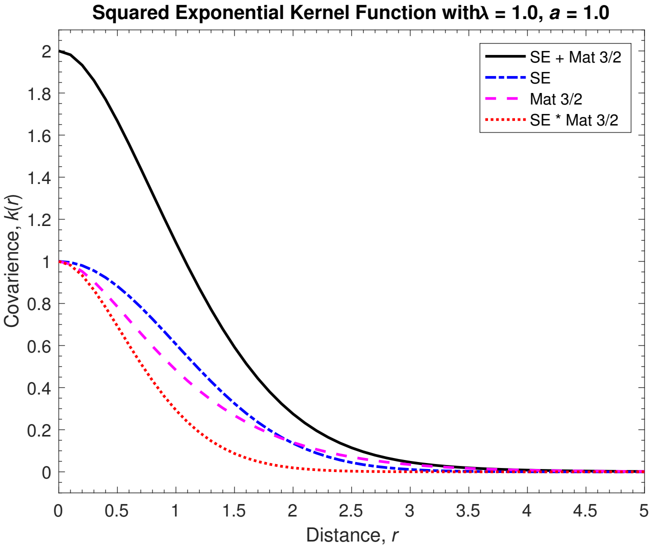
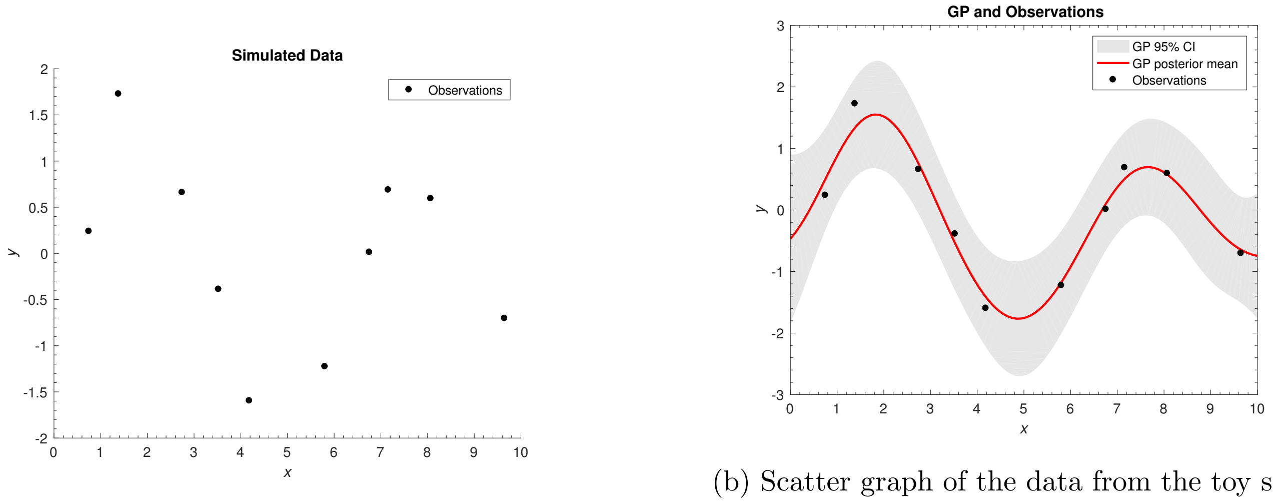


Figure 10: Matern class kernel interpolations for decreasing ν and λ.



(a) Observed data from the simulator.

(b) Scatter graph of the data from the toy simulator with a fitted curve and 95%CI .

Figure 11

(the 1D analogue of this is given in equation 9) and an estimation of the re-substitution loss which is a function of the predictor $\mathbf{y}$ and posterior mean $\mathbf{m}_|$. The posterior mean is a quantity in Bayesian statistics where an event is assigned a conditional probability after it has transpired, this depends on the context of the event.

A similar procedure can be performed on data in 2 dimensions. For this we were not provided data, rather used the inbuilt MATLAB 2D function “peaks” (hereafter called the simulator). The points from peaks are shown in Figure 13a. Figure 13b shows a GP that has been fitted using an ARD squared exponential kernel, we plot the posterior mean and upper and lower confidence intervals. We note that the upper and lower CIs are very small at the data points, this makes sense given that at a data point we need not estimate the surface. This is perhaps easier to visualise using a contour plot, we provide these in Figures 14a and 14b. Figure 14a shows the the GP posterior, where blue represents low values, and yellow high values. This is complemented by Figure 14b that shows the standard deviation of the GP posterior mean from the data points. We immediately notice that the regions around the given data have very low standard deviations, i.e. the uncertainty of our estimate of the surface is very low when we have measured data. This makes intuitive sense and we make use of this in GP emulation in section 5.

4.2 Gradient Matching

4.2.1 What is gradient matching?

Differential equations (DEs) are a powerful way of describing physical processes, providing insight into how a system changes with respect to certain variables. For the proposed problem, we are

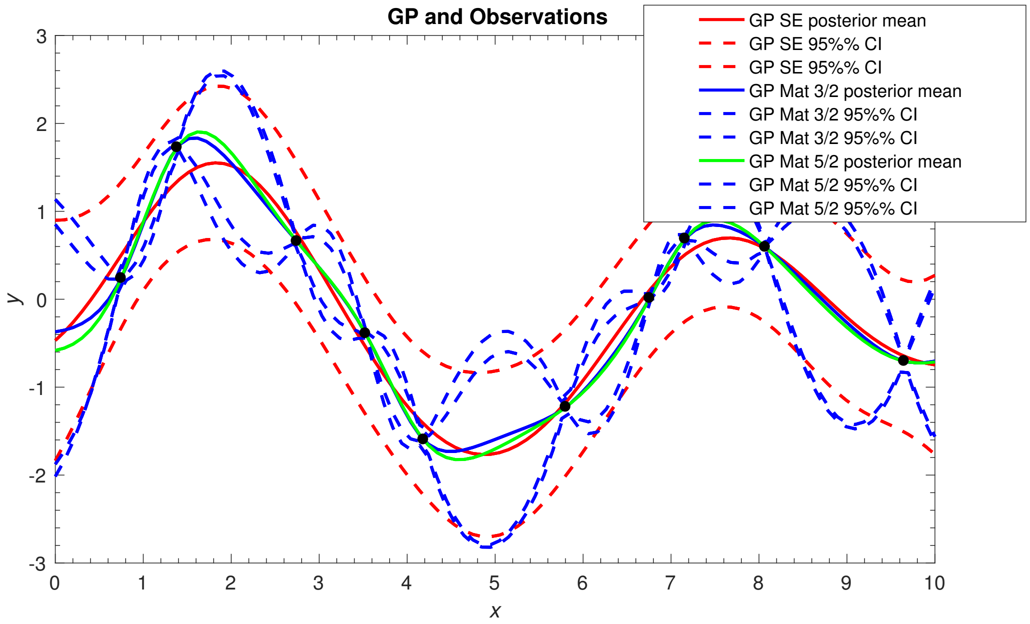
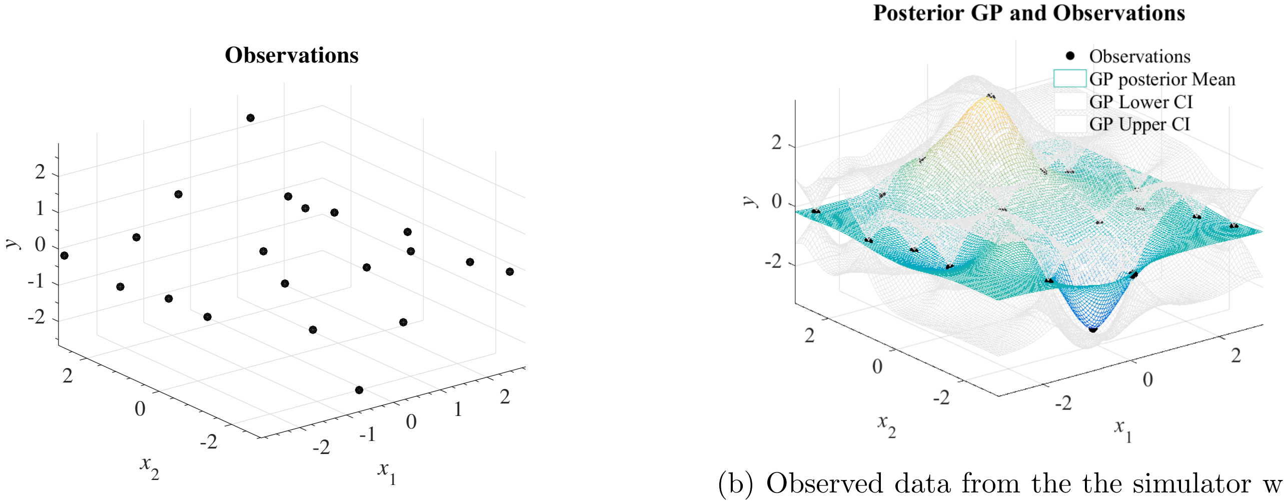


Figure 12: Comparison of fit and 95% CI for

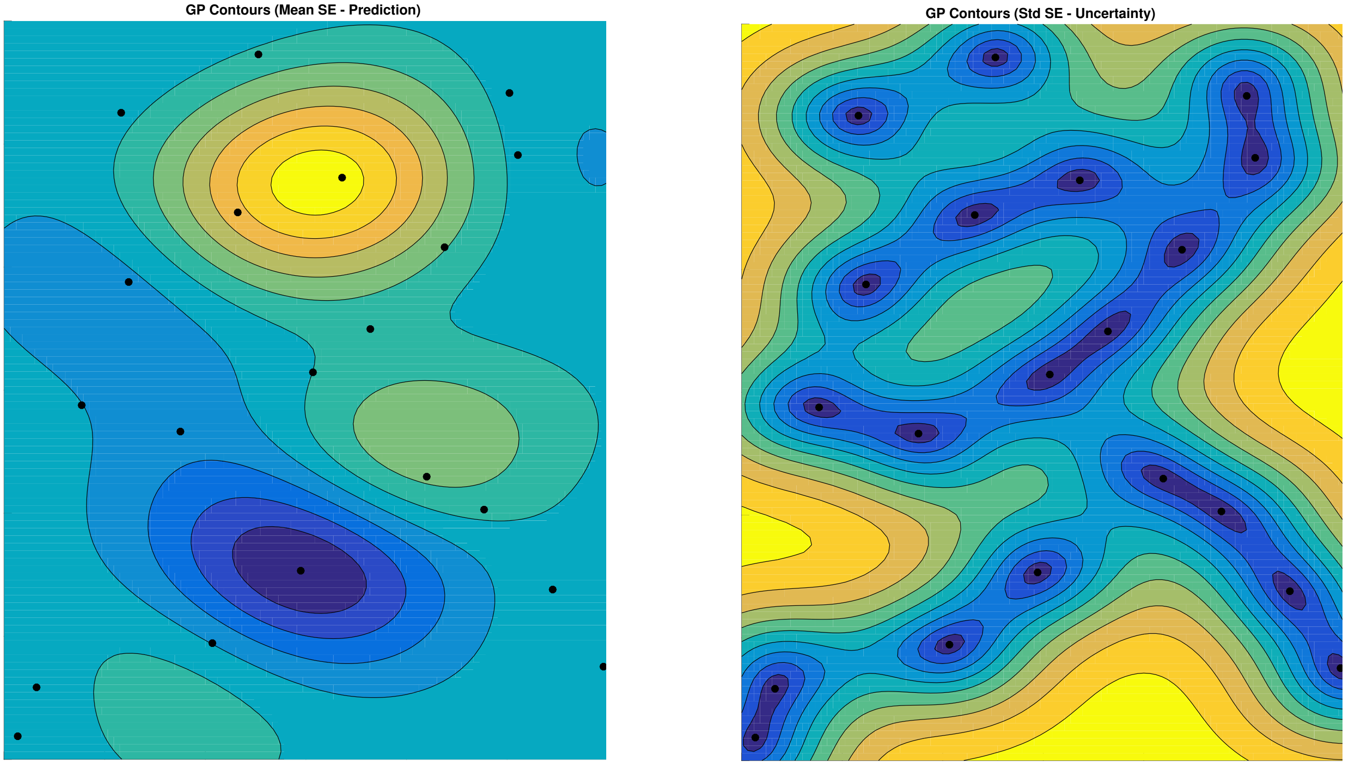


(a) Observed data from the the simulator.

(b) Observed data from the the simulator with a surface fitted and 95%CIs shown.

Figure 13

particularly interested in DEs modelling the heart. The simple cylindrical heart model considered in this report can be described by a partial differential equation (PDE). However, analytical solutions to DEs can be difficult to find and if this is the case, or if no analytical solution exists, one turns to numerical solutions, which can be very computationally expensive. For example, for a partial differential equation (PDE) describing the simple cylinder model considered in this report, one numerical integration for the inference procedure can take around one day to complete, meaning one full inference procedure could take years, which is highly impractical.



(a) Contour plot of the surface fit to the simulator data.

(b) Contour plot of the mean confidence intervals of the fit to the simulator.

Figure 14

Explicit solution of a DE system concerns integrating the DEs to produce a signal (given a set of initial conditions and DE parameters, θ), which is compared to the data signal using a metric defined by the chosen noise model. Here, we consider the signal we are interested in, $\mathbf{X}$ (with initial value $\mathbf{X}(0)$), the observation of this signal with $\mathbf{Y}$ (the noisy version of $\mathbf{X}$ described by the initial value). The observational error has standard deviation σ and interactions in processes described by DEs depend on a vector of parameters, $\boldsymbol{\theta}_s$. Typically, inference methods search the space of parameter values and numerically solve the DEs for each parameter choice, comparing the output with the observed output. Once an appropriate noise model is chosen, data signals and signals described by the current set of DE parameters can be calculated. This is repeated either as an iterative optimisation scheme or sampling procedure in order to estimate the parameters. However this method takes too long, so we need a faster and more accurate technique.

This is where gradient matching comes into play. Alternative approaches using gradient matching have been founded to reduce the computational expense of inference procedures for differential equations. These newer approaches follow a two-step procedure;

- Step 1: Interpolation is utilised to smooth time series data to avoid modelling noisy observations;

- Step 2: The parameters of the DEs, $\boldsymbol{\theta}$, are either optimised or sampled, whilst minimising a metric which measures the difference between the gradients of the tangents to the interpolants, and the $\boldsymbol{\theta}$-dependent derivatives in the DEs. This step is visualised in Figure 15.

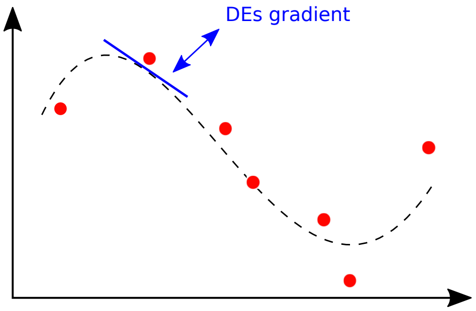


Figure 15: Step 2 in the gradient matching process.

This two-step method bypasses the numerical integration, which, as discussed above, is computationally expensive, as the DEs never have to be numerically integrated. Hence the problem of inferring the typically unknown initial conditions of the system is removed, as this step isn't required for matching gradients. In step 2, once an interpolant is fitted to the data, the gradient of this interpolant is compared to the gradient given by the DEs being considered (Figure 15), and the mismatch of these gradients is minimised. Since the DEs of a problem are defined, the gradient of the DEs are easily obtained.

For the process of gradient matching (as proposed by Calderheat, Girolami and Lawrence [2] and Dondelinger, Filippone, Rogers and Husmeier [13]), we compare the gradients, $\dot{\mathbf{X}}$, given by the two modelling approaches; the interpolant and the DEs themselves. We must consider hyperparameters for the interpolant, ϕ, and a slack parameter, γ, which governs how closely the gradients need to match . For example, for an exact match, $\gamma = 0$, however this would incorporate noise from the data, which is undesirable. Here, the hyperparameters are different from the usual parameters. These can be chosen and 'tuned' to get the system to work optimally, however don't necessarily represent physical parameters.

One disadvantage to this two-step scheme is that the results of parameter inference are critically dependent on the quality of the initial interpolant. Alternatively, the DEs can be allowed to regularise the interpolant, as first suggested by Ramsay et al. [24]. Dondelinger et al. [13] applied this to the nonparametric Bayesian approach in Calderhead et al. [2], which uses Gaussian processes (GPs), demonstrating that it significantly improves the parameter inference accuracy and robustness with respect to noise. Here, unlike in Ramsay et al. [24], all hyperparameters that control the interpolant smoothness are consistently inferred in the framework of nonparametric Bayesian statistics. This dispenses the need of heuristics and approximations when configuring the interpolation function.

During the modelling week, we worked with the method put forth by Dondelinger et al. [13]. Gaussian processes were used to determine the interpolant, as this has a number of benefits. GPs are able to model non-linear processes and the hyperparameters controlling the shape of the interpolant can be inferred from the data. Additionally, a priori knowledge can be utilised by specifying the kernel controlling the covariance structure of the GP. Furthermore, the derivative of a GP is also a GP - a nice property which is highly relevant to gradient matching. This means that after a GP is fitted, derivatives for the process can be immediately calculated in closed form. These can then be easily compared to the derivatives predicted by the DE.

4.2.2 Adaptive Gradient Matching

There are a lot of different gradient matching methods, but we will only consider adaptive gradient matching (AGM)here, with Bayesian inference. In classical inference, there is considered to be one fixed true parameter for everything, with an interval of reasonable parameters. In Bayesian inference, the parameters can have multiple values depending on the set up (or, if considered to only have one value, the uncertainty can be modelled as a random variable). In this case, we are interested in the posterior probability of the parameters (the probability of the parameters given the data). Bayes theorem is given by

$$p(\theta|Y) = \frac{p(\theta)p(Y|\theta)}{p(Y)} = \frac{p(\theta, Y)}{p(Y)} \tag{22}$$

where $p(\theta)$ is the prior distribution over the parameters, $p(Y|\theta)$ is the likelihood, $p(Y)$ is the marginal likelihood and $p(\theta, Y)$ is the joint distribution (probability density of parameter given the data set).

The joint distribution for the AGM method of Dondelinger et al. [13] is, by the chain rule of probability;

$$p(Y, X, \sigma, \theta, \gamma, \phi) = p(Y|X, \sigma)p(X|\theta, \gamma, \phi)p(\theta)p(\gamma)p(\phi)p(\sigma). \tag{23}$$

Here, $p(Y|X, \sigma)$ is the likelihood. Assuming a Gaussian noise model, this can be expressed as a normal distribution, centred on the data;

$$p(Y|X, \sigma) = N(Y|X, \sigma^2\mathbb{I}) \tag{24}$$

where $\mathbb{I}$ is the identity matrix. Additionally, in (23), $p(\theta)$ is the prior distribution over the differential equation parameters, $p(\gamma)$ is the prior distribution over the gradient mismatch parameter(s), $p(\phi)$ is the prior distribution over the Gaussian process hyperparameters and $p(\sigma)$ is the prior distribution over the standard deviation of the error in the system. These four all need to be specified based on a priori knowledge. Lastly, the 'crux' of the method (containing the gradient matching part) is formally defined by

$$p(X|\theta, \gamma, \phi) \propto \frac{1}{\prod |2\pi(A + \gamma\mathbb{I})|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\sum \left(X^T K X + (f - m)^T (A + \gamma\mathbb{I})^{-1}(f - m)\right)\right), \tag{25}$$

where K is the covariance matrix of the Gaussian process, f are the derivatives predicted by the differential equations, m are the Gaussian process (interpolant) derivatives and A is a covariance matrix of the Gaussian process derivatives. With a defined GP, both m and A are computable in closed form. Notice that γ is the variance parameter - as γ gets larger, f and m have more freedom to deviate from one another - meaning γ controls how closely the gradients from the GP need to match the DE predictions.

It is important to note that there will naturally be weak identifiability in equation (23), due to there being more than one 'error' component; σ and γ, where σ governs how closely X must match the data, and γ controls how closely the derivatives must match. This is important as it isn't useful to determine a GP where the derivatives match the DE well but where the Xs don't match the data. In principle, the term $p(Y|X, \sigma)$ should penalise this. However it is possible to infer a large value for σ, effectively ignoring the contribution of the likelihood. To avoid this, σ should be inferred before the AGM method is implemented and then fixed. For our workshop, σ was be fixed at it's true value to allow us to implement this method successfully. Hence, we need only the marginal likelihood from equation (22) to compute the posterior probability of the parameters. The marginal likelihood is the likelihood function after the parameters have been marginalised over, for which we rely on the Markov chain Monte Carlo (MCMC) methods, specifically Gibbs sampling (which was not explored in this workshop).

4.2.3 Example Application of Gradient Matching: The Fitz-Hugh Nagumo System

The Fitz-Hugh Nagumo System is given by the coupled ODEs,

$$[\dot{V}] = \psi \left([V] - \frac{[V]^3}{3} + [R] \right); \tag{26}$$

$$[\dot{R}] = -\frac{1}{\psi} \left([V] - \alpha + \beta * [R] \right), \tag{27}$$

where the dot notation represents the temporal derivative. The parameters are $\alpha =: \theta_1$, $\beta =: \theta_2$ and $\psi =: \theta_3$, with components of Voltage, V, and a Recovery variable, R. The square brackets here denote the time-dependent concentration for each component.

This system can be used to model cardiac conditions in Biomedical Engineering. For example, electrical excitation-conduction in cardiac tissue [33], cardiac action potentials [12] and arrhythmias [19].

During the workshop, the 'Heart' group carried out a parameter inference on this Fitz-Hugh Nagumo system. It is also possible to carry out inference on systems describing the heart, starting with the Fitz-Hugh Nagumo equations and moving on to more complex models. However the workshop did not allow time for us to extend to these more complicated systems.

Figures 17 and 16 shows results from MATLAB code provided by Benn Macdonald during the Modelling week. The group chose initial ranges (and step sizes) of θ_1, θ_2 and θ_3 to be considered to obtain parameter estimates for these three parameters in the Fitz-Hugh Nagumo system. These three parameter estimates are outlined in figure 16 and the signals produced are outlined in figure 17.

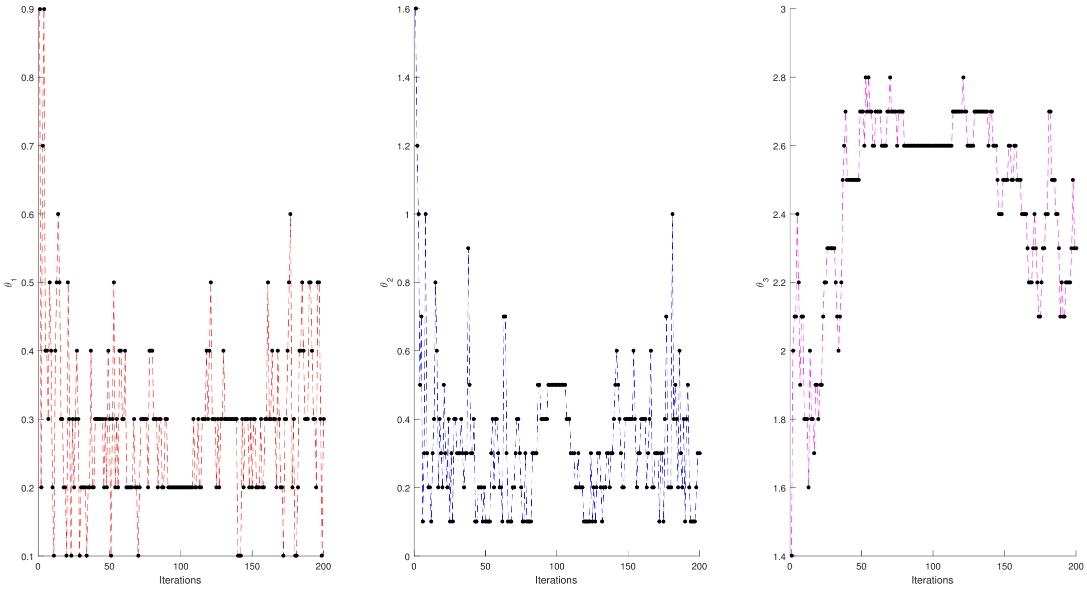


Figure 16: Parameter estimates for θ_1, θ_2 and θ_3.

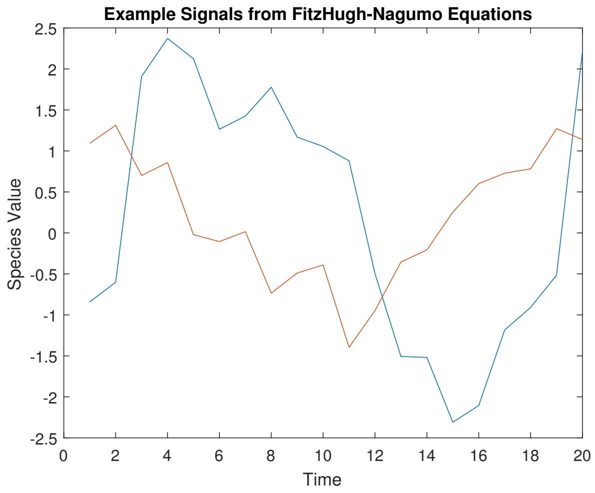


Figure 17: Example of signals produced using the fitz-Hugh Nagumo ODEs in (26) and (27). The blue line represents species V and the red line represents species R.

Figures 17 and 16 shows results from MATLAB code provided by Benn Macdonald during the Modelling week. The group chose initial ranges (and step sizes) of θ_1, θ_2 and θ_3 to be considered

to obtain parameter estimates for these three parameters in the Fitz-Hugh Nagumo system. These three parameter estimates are outlined in figure 16 and the signals produced are outlined in figure 17.

As iterations progress, the parameter estimates fluctuate in a smaller range, suggesting they will converge towards the optimal values. However, we ran these estimates over 200 iterations and, typically, many more than that are required, hence significant convergence cannot be seen in Figure 16. We did not conduct further iterations as the current results were very computationally expensive, taking around 2 hours. Time restraints did not allow for further investigation.

4.3 Continuum Mechanical Heart Model

4.3.1 Mathematical Description of the Heart Kinematics

The continuum model uses a pair of partial differential equations to describe the heart deformation in response to varying pressures. It assumes the heart tissue to be anisotropic hyperelastic material and depends on four parameters related to the contractility of the ventricles and the myocardial stiffness, among others. It is possible to numerically solve the model using finite element methods.

Basic Kinematics The equations to solve the kinematics of a generic body B with a certain geometry (in the reference configuration) loaded by the bulk forces ($\mathbf{b}$), surface forces ($\mathbf{s}$) on $\mathbf{s}_f$ and kinematic constrains on the boundary $\mathbf{s}_u$ are:

- Three scalar equilibrium equations:

$$\frac{\partial \sigma_{ij}}{\partial x_i} + b_j = 0; \qquad i, j = 1, \ldots, 3 \tag{28}$$

- Six Compatibility equations with differential relation between displacements and strains:

$$\epsilon_{ij} = \frac{1}{2}\left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right] \tag{29}$$

- Six constitutive equations with relation between stresses and the strain energy:

$$\sigma_{ij} = pI_{ij} + \frac{\partial \Phi}{\partial F}; \qquad i, j = 1, \ldots, 3 \tag{30a}$$

$$\sigma_{ij} n_i = f_j \text{ on } \mathbf{s}_f \tag{30b}$$

$$u_i = \bar{u}_i \text{ on } \mathbf{s}_u \tag{30c}$$

An illustration of this and the geometry that we are assuming for the heart can be found in Figure 18.

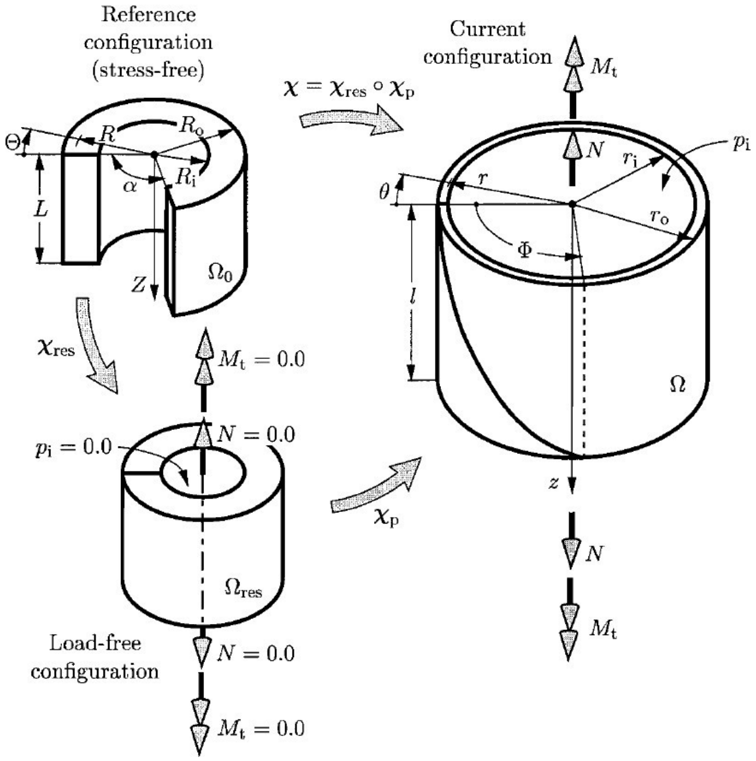


Figure 18

Cylindrical Model of the Left Ventricle The left ventricle was simplified into a thick-walled single-layered circular cylindrical tube under a pressure. The left ventricular was simplified by a cylindrical model; the geometry at a stress-free configuration Ω_0 was defined in cylindrical polar coordinates: $R_i \leqslant R \leqslant R_0$ as the radial coordinate with $\mathbf{E}_R$ radial base, $0 \leqslant \Theta \leqslant (2\pi - \alpha)$ circumferential coordinate with $\mathbf{E}_\Phi$ circumferential base, $0 \leqslant Z \leqslant L$ axial coordinate with $\mathbf{E}_Z$ axial base.

In the intact ring configuration Ω_{res} there was a residual stress only if $\alpha \neq 0$. In case of loading, as an internal pressure, a isochoric deformation caused a deformed configuration Ω with:

- $r_i \leq r \leq r_0$ radial coordinate with $\mathbf{e}_r$ radial base.
- $0 \leq \theta \leq 2\pi$ circumferential coordinate with $\mathbf{e}_\theta$ circumferential base
- $0 \leq z \leq l$ axial coordinate with $\mathbf{e}_z$ axial base
- $\lambda_z = \frac{l}{L}$ constant axial stretch
- $k = \frac{2\pi}{2\pi - a}$ measure of the opening angle in Ω_0.

The deformation gradient was given as

$$F_1 = \lambda_1 \mathbf{e}_r \otimes \mathbf{E}_R + \lambda_2 \mathbf{e}_\theta \otimes \mathbf{E}_\Phi + \lambda_z \mathbf{e}_z \otimes \mathbf{E}_Z \tag{31}$$

The Momentum equilibrium (equation 28) for this problem was reduced into a simple ODE (in cylindrical polar coordinates) with boundary conditions respected by the physiological loading:

$$\frac{\partial \sigma_{rr}}{\partial r} + \frac{\sigma_{rr} - \sigma_{\theta\theta}}{r} = 0, \tag{32a}$$

$$\sigma_{rr}\Big|_{r=r_0} = 0, \tag{32b}$$

$$\sigma_{rr}\Big|_{r=r_i} = -p^r. \tag{32c}$$

The solution was

$$\sigma_{rr}\Big|_{r=r_0} - \sigma_{rr}\Big|_{r=r_i} = \int_{r_i}^{r_0} \frac{\sigma_{rr} - \sigma_{\theta\theta}}{r} \mathrm{d}r = p^r \tag{33}$$

4.3.2 Constitutive Law

The anisotropy of the cardiac tissue represents a key feature of the heart mechanics. The organisation of the myocardium at a macroscopic level in a unique ventricular muscle bans revealed the unavoidable coherence and mutual coupling of anatomy and function in the ventricular myocardium, making it possible to understand the principles governing development, electrical, mechanical and energetic events occurring in the human heart. In addition, the comprehension of the influence of this feature on heart response is an essential prerequisite in clinical evaluation and treatment of different heart diseases. In this work we used a reduced version of the invariant-based constitutive Holzapel-Ogden law to model the ventricular material, which defined the strain energy as:

$$\Phi = \frac{a}{2b}\Big(\exp\big\{b(I_1 - 3)\big\}\Big) + \frac{a_f}{2b_f}\Big(\exp\big\{b_f(I_4 - 1)^2\big\}\Big) \tag{34}$$

where $I_1 = \mathbf{C} : \mathbf{I}$, matrix invariant, $\mathbf{I}$ unit matrix; $\mathbf{C} = \mathbf{F}^{\mathrm{T}}\mathbf{F}$, right Cauchy-Green, $\mathbf{F}$ deformation tensor; $I_4 = \mathbf{f}_0 \cdot (\mathbf{C}\mathbf{f_0})$, fibres invariant, $\mathbf{f}$ direction of the fibres.

The four parameters a, b, a_f, and b_f are achievable from experimental data and aim of this work was to estimate these parameters.

In this case, the constitutive equation of the material (equations 30) becomes

$$\boldsymbol{\sigma} = -p\mathbf{I} + a\left(e^{b(I_1-3)}\right)\mathbf{B} + H(I_4 - 1)2a_f(1_4 - 1)\left(e^{b_f(I_4-1)^2}\right)\mathbf{f}\otimes\mathbf{f} \tag{35}$$

with $\mathbf{B} = \mathbf{F}^{\mathrm{T}}\mathbf{F}$ left Cauchy Green.

The ODE equation 32a becomes

$$\frac{\partial \sigma_{rr}}{\partial r} = \frac{1}{r}\left(2\frac{\partial \phi_m}{\partial I_1}\left(\lambda_2^2 - \lambda_1^2\right) + 2H(I_4 - 1)\frac{\partial \phi_f}{\partial I_4}\lambda_2^2 \cos^2\gamma\right) \tag{36}$$

with

$$\frac{\partial \phi_m}{\partial I_1} = \frac{a}{2}\left(e^{b(I_1-3)}\right) \tag{37a}$$

$$\frac{\partial \phi_f}{\partial I_4} = a_f(I_4 - 1)\left(e^{b_f(I_4-1)^2}\right) \tag{37b}$$

$$\gamma = \frac{2\pi}{3\,(R_0 - R_i)}(R - R_i) - \frac{\pi}{3} \tag{37c}$$

5 Results: Emulation of the Heart

5.1 Heart Model Toy Simulator

A toy simulator has been created to test the methods being discussed in this report. The parameters of interest are the pressure and also a, b, a_f & b_f. These parameters are the stiffness and anisotropy (preferred directions) of the heart tissue and vary with the radius of tube's inner radius, r.

In this section the mentioned methods will be used to build an emulator for the heart model, in 1 dimension, 2 dimensions and 4 dimensions. Firstly in the 1D case, the emulation will be built of the radius, r vs the four parameters of interest. This will involve fixing three of the parameters and varying the fourth. Then emulator will then move onto an emulation with two dimensions (where two of the parameters are fixed) and lastly with four dimensions (where none of the parameters are fixed). However the 4 dimensional emulation presents a difficulty in that 4 dimensions can not be visualised with a plot, how this is overcome will be discussed later.

5.1.1 1D Heart Model Emulation

Figure 19 shows how the radius of the heart varies as pressure is loaded. In figure 20 are four plots of the four parameters plotted against the radius.

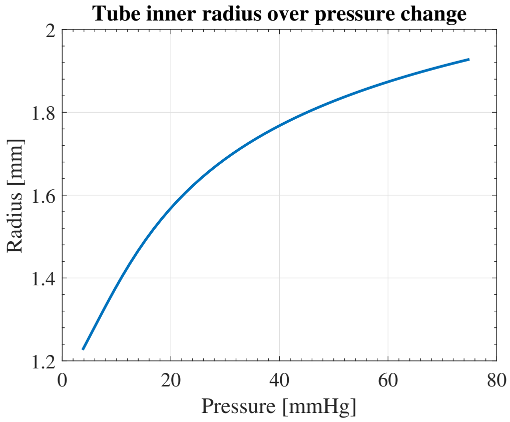


Figure 19: The plot of the radius against the pressure.

As discussed earlier, using Gaussian processes can also give us a measure of the uncertainty of the interpolation, thus confidence intervals can be included in the plots, which can be seen on the figures 21a and 21b. It is clear that there is more uncertainty in figure 21a, this is due to the non-linear nature of the curve.

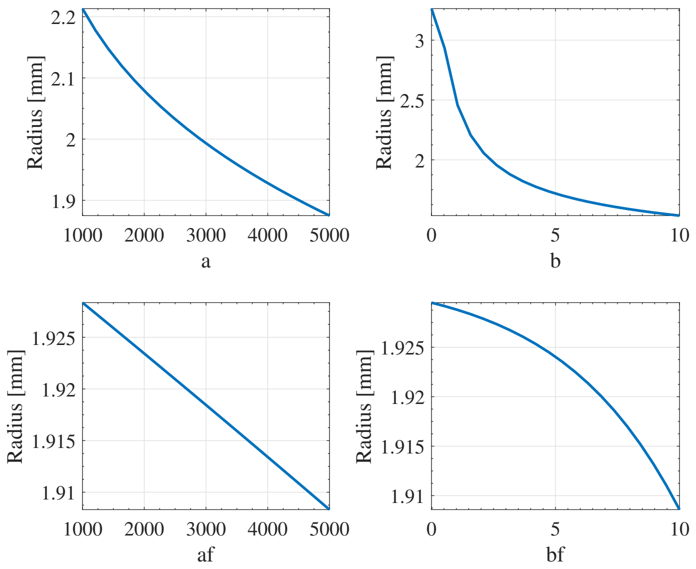
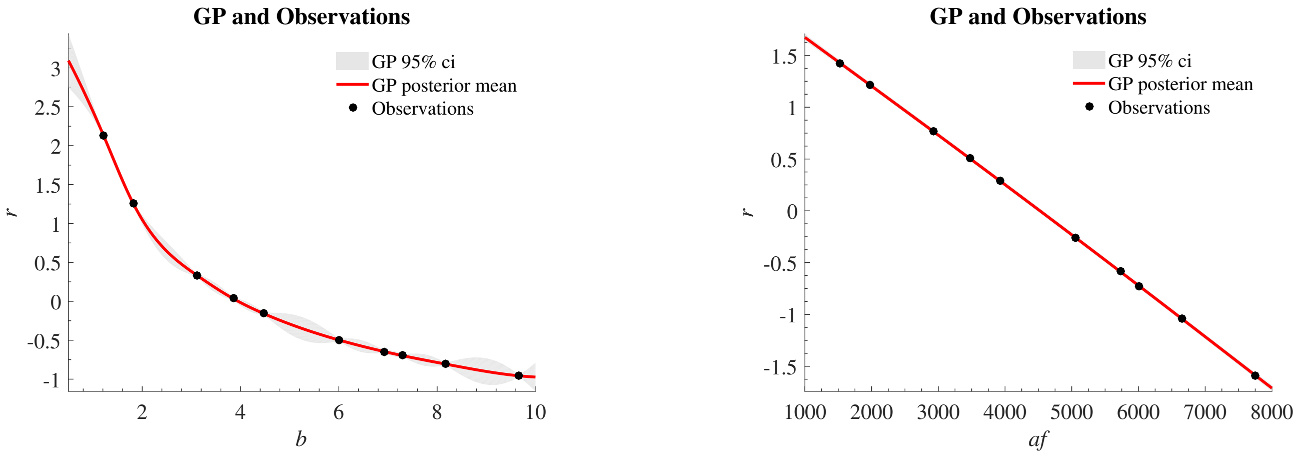


Figure 20: The plots of the four parameters against the radius.



(a) The plot of the radius against b, with 95% confidence intervals.

(b) The plot of the radius against a_f, with 95% confidence intervals.

5.1.2 2D Heart Model Emulation

The process was then related for two parameters which were then plotted against the radius. This results in three dimensional plots, with six possible pairs of the parameters. These can be seen in figure 22. Due to the high computational expense, producing these plots took hours, despite being five discretised points. However one of the plots was created with 20 discretised points, giving an example of the quality that is possible if there had been more time, this is shown in figure 23.

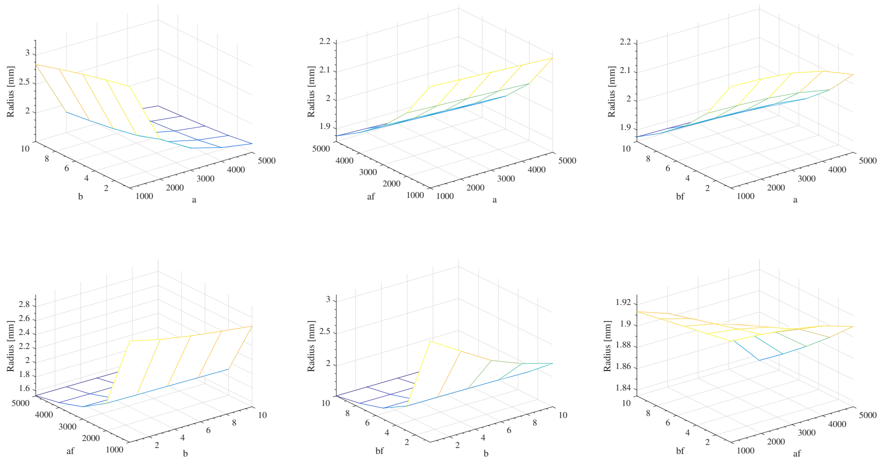


Figure 22: This plot shows the plots of parameter pairs against radius.

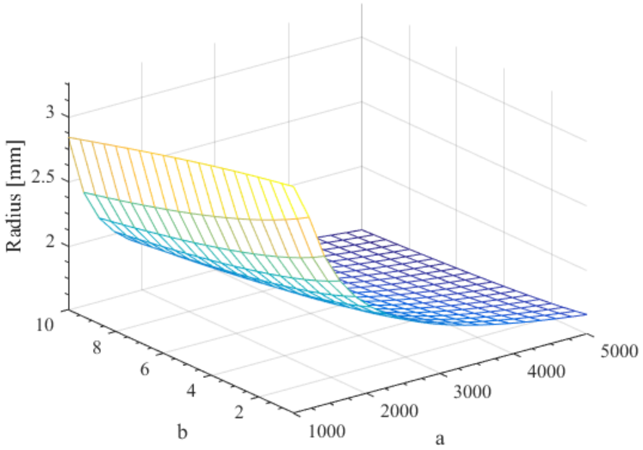


Figure 23: This plot shows a and b plotted against the radius.

5.2 4D Heart Model Emulation

During the modelling week the ultimate aim was to use Gaussian regression to perform multivariate regression for all four parameters simultaneously. This was successfully completed, however all the data could not be plotted simultaneously (as this would obviously require a five dimensional plot). Thus plots were created by using a similar method as with the 2D emulator, in that six pairs of the parameters can be plotted against each other.

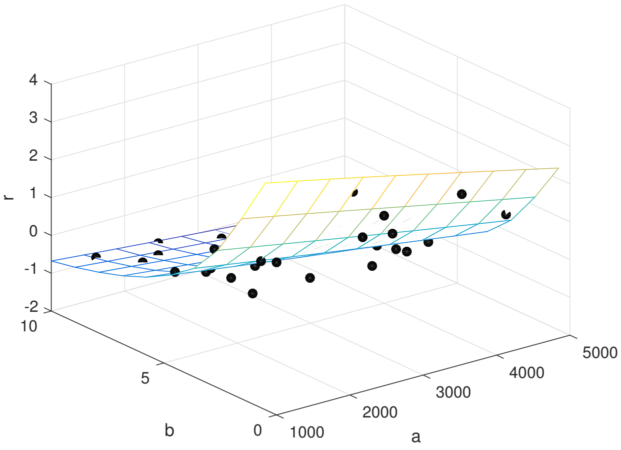


Figure 24: Plot of the parameters a and b from a 4 dimensional emulation, with the associated data

6 Conclusions and Future Work

The techniques of emulation and gradient matching that we explored during the modelling week have proven to be extremely powerful and valuable in the attempt of solving the problem of parameter inference given the information derived from patient-specific data and a parameter-dependent model of a process of interest. Our example application of the use of adaptive gradient matching was a rather simple model with a small number of equations and of parameters and it still required hours to run and converge to an informative solution. A more realistic model would probably take days to solve, but this still represents a drastic improvement compared to the months or even years of computational time that previous methods would require. The use of emulation and gradient matching therefore brings the use of mathematical models as aids in the process of clinical decision back into the realm of practical feasibility. These models also have the advantage of depending on a number of hyperparameters and slack parameters that allow the clinical practitioner to exploit their knowledge of the expected data (e.g. how rough or smooth the signal is) or of the accuracy of the mathematical model. Although we focused on the heart for our example application, these techniques can be used for a very wide range of problems and therefore have extremely wide applicability.

Despite the limited amount of time, we have been able to explore some of the techniques that may be suitable for heart modelling. Bayesian optimisation is worth considering and may well deserve further study and development. The heart cylinder model has been our starting point with application of 4-D emulation plots and gradient matching. Further development is needed and other geometries should be considered in order to find out the most suitable model for simulations in a clinical setting.

Future work would certainly include allowing a higher number of iterations in our gradient matching code in order to improve the results. From a theoretical point of view, there could be room for exploration of how the hyperparameters (regulating the shape of the data interpolant) and the slack parameters (governing how wide the mismatch is allowed to be between the data and the interpolant or between the gradients of the emulator versus those of the model) interact with each other and can be fine tuned to increase efficiency in the application of this method. Moreover, the gradient matching algorithm requires the mathematical model of reference to be quite precise and accurate. This is not always the case, since computational restrictions often require significant simplifications in the description of biomechanical processes of interest, which can limit the accuracy or the model or its applicability only to a small range of clinical applications. For example, our model of the heart idealised the organ as a simple cylinder, which is quite a simplification from the actual shape, but can still be a valid reference for some specific clinical applications. Improvements can then certainly be done from the modelling viewpoint, even though feasibility of computational implementation often remains a significant restraint to their accuracy.

The PhD modelling week was an attractive opportunity to bring together students and professionals with different background in an attempt to learn from each other and link current thinking to future developments with potential for clinical application. Patient-specific modelling may become a daily approach for clinical management and training of medical and nursing staff. All of this is available but requires willingness to change attitude towards a different type of approach, which may become the key to successful outcome and the beginning of a complete revolution in patient care. No magic bullets, only a different way of thinking.

References

[1] T Arts, T Delhaas, P Bovendeerd, X Verbeek, and F W Prinzen. Adaptation to mechanical load determines shape and properties of heart and circulation: the CircAdapt model. *Am J Physiol Heart Circ Physiol*, 288:H1943–H1954, 2005.

[2] M. A. Girolami B. Calderhead and N. D. Lawrence. Accelerating bayesian inference over non-linear differential equations with gaussian processes. *Neural Information Processing Systems (NIPS)*, 22, 2008.

[3] FP Baaijens. A fictitious domain/mortar element method for fluid-structure interaction. *Int J Numer Meth Fl*, 35:743–761, 2001.

[4] Y Bazilevs, J R Gohean, T J R Hughes, R D Moser, and Y Zhang. Patient-specific isogeometric fluid-structure interaction analysis of thoracic aortic blood flow due to implantation of the Jarvik 2000 left ventricular assist device. *Comput Methods Appl Mech Eng*, 198:3534–3550, 2009.

[5] C Bishop. *Pattern Recognition and Machine Learning.* Springer Science+Business Media, 2006.

[6] P J Blanco and R A Feijóo. A dimensionally-heterogeneous closed-loop model for the cardiovascular system and its applications. *Med Eng & Phys*, 35:652–667, 2013.

[7] S Choi, JR Boston, D Thomas, and JF Antaki. Modelling and Identification of an Axial Flow Pump. *Proceedings of the American Control Conference, Albuquerque, New Mexico*, pages 3714–3715, 1997.

[8] C De Lazzari, M Darowski, G Ferrari, D M Pisanelli, and G Tosti. The impact of rotary blood pump in conjunction with mechanical ventilation on ventricular energetic parameters: Numerical Simulation. *Methods Inf Med*, 45:574–583, 2006.

[9] C De Lazzari and G Ferrari. Right Ventricular Assistance by Continuous Flow Device. A Numerical Simulation. *Methods Inf Med*, 46:530–537, 2007.

[10] C De Lazzari, I Genuini, D M Pisanelli, A D'Ambrosi, and F Fedele. Interactive simulator for e-Learning environments: a teaching software for health care professionals. *BioMedical Engineering OnLine*, pages 13–172, 2014.

[11] D Doshi and D Burkhoff. Cardiovascular Simulation of Heart Failure. *Pathophysiology and Therapeutics. J. Card. Fail.*, 22(4):303–311, 2016.

[12] G. Duckett and D. Barkley. Modelling the dynamics of cardiac action potentials. *Physical Review Letters*, 85(4), 1999.

[13] S. Rogers F. Dondelinger, M. Filippone and D. Husmeier. Ode parameter inference using adaptive gradient matching with gaussian processes. *Journal of Machine Learning Research*, 31:216–228, 2013.

[14] L Fresiello, G Ferrari, A Di Molfetta, K Zieliński, A Tzallas, S Jacobs, M Darowski, M Kozarski, B Meyns, N S Katertsidis, E C Karvounis, M G Tsipouras, and M G Trivella. A cardiovascular simulator tailored for training and clinical uses. *Journal of Biomedical Informatics*, 57(100-112), 2015.

[15] X. Gao, H. Zhaun. Parameter inference for a simple left ventricle model. Lecture Notes, Glasgow Mathematics for Industry Modelling Week, 2016, August 2016.

[16] G Gibson. Bayesian inference and computational bayesian inference and computational methods. Lecture Notes, Heriot-Watt University, 2016.

[17] GA Giridharan, M Skliar, DB Olsen, and GM Pantalos. Modelling and Control of a Brushless DC Axial Flow Ventricular Assist Device. *ASAIO J*, 48:272–289, 2002.

[18] JR Gohean, MJ George, TD Pate, M Kurusz, RG Longoria, and RW Smalling. Verification of a computational cardiovascular system model comparing the hemodynamics of a continuous flow to a synchronous valveless pulsatile flow left ventricular assist device. *ASAIO J*, 59(2):107–116, 2013.

[19] S. Goktepe and E. Kuhl. Computation modelling of cardiac electrophysiology: A novel finite element approach. *International journal for numerical methods in engineering*, 2009.

[20] M Granegger, F Moscato, F Casas, G Wieselthaler, and H Schima. Development of a Pump Flow Estimator for Rotary Blood Pumps to Enhance Monitoring of Ventricular Function. *Artificial Organs*, 36:691–699, 2012.

[21] K-W Gwak, M Ricci, S Snyder, B E Paden, J R Boston, M A Simaan, and J F Antaki. In Vitro Evaluation of Multiobjective Hemodynamic Control of a Heart-Assist Pump. *ASAIO J*, 51:329–335, 2005.

[22] H Huang, M Yang, S Wu, and H Liao. Dynamic modelling of the outlet of a pulsatile pump incorporating a flow-dependent resistance. *Medical Engineering & Physics*, 35:1097–1104, 2013.

[23] D. Husmeier. A leisurely look at statistical inference. *In: Probabilistic Modeling in Bioinformatics and Medical Informatics*, 2005.

[24] D. Campbell J. O. Ramsay, G. Hooker and J. Cao. Parameter estimation for differential equations: a generalized smoothing approach. *Journal Rankings on Statistics*, pages 741–796, 2007.

[25] T Korakianitis and Y Shi. A concentrated parameter model for the human cardiovascular system including heart valve dynamics and atrioventricular interaction. *Med Eng & Phys*, 28:613–628, 2006.

[26] T Korakianitis and Y Shi. Numerical simulation of cardiovascular dynamics with healthy and diseased heart valves. *J Biomech*, 39:1964–1982, 2006.

[27] I Larrabide, P J Blanco, S A Urquiza, E A Dari, M J Vénere, and N A Feijóo RA de Souza e Silva. HeMoLab - Haemodynamics Modelling Laboratory: An application for modelling the human cardiovascular system. *Computers in Biology and Medicine*, 42:993–1004, 2012.

[28] G Lord. Stochastic Simulation. Lecture Notes, Heriot-Watt University, Edinburgh, 2016.

[29] J Lumens, T Delhaas, B Kirn, and T Arts. Three-Wall Segment (TriSeg) Model Describing Mechanics and Haemodynamics of Ventricular Interaction. *Ann Biomed Eng*, 37(11):2234–2255, 2009.

[30] M McCormick, D Nordsletten, D Kay, and N Smith. Modelling left ventricular function under assist device support. *Int J Numer Method Biomed Eng*, 27:1073–1095, 2011.

[31] M McCormick, DA Nordsletten, D Kay, and NP Smith. Simulating left ventricular fluid-solid mechanics through the cardiac cycle under LVAD support. *J Comput Phys*, 244:80–96, 2013.

[32] F Moscato, GA Danieli, and H Schima. Dynamic modelling and identification of an axial flow ventricular assist device. *International Journal of Artificial Organs*, 32:336–343, 2009.

[33] M. H. Jabbar N. A. Adon and F. Mahmud. Fpga implementation for cardiac excitation-conduction simulation based on fitzhugh-nagumo model. *5th International Conference on Biomedical Engineering in Vietnam*, 2015.

[34] U Noè. Emulation using gps. Lecture Notes, Glasgow Mathematics for Industry Modelling Week, 2016.

[35] DA Nordsletten, D Kay, and N Smith. A non-conforming monolithic finite element method for problems of coupled mechanics. *J Comput Phys*, 229:7571–7593, 2010.

[36] T Pirbodaghi, A Weber, T Carrel, and S Vandenberghe. Effect of Pulsatility on the Mathematical Modelling of Rotary Blood Pumps. *Artificial Organs*, 35(8):825–832, 2011.

[37] Y Shi, AG Brown, PV Lawford, A Arndt, P Nuesser, and DR Hose. Computational modelling and evaluation of cardiovascular response under pulsatile impeller pump support. *Interface Focus*, 1:320–337, 2011.

[38] Y Shi and T Korakianitis. Numerical Simulation of Cardiovascular Dynamics With Left Heart Failure and In-series Pulsatile Ventricular Assist Device. *Artificial Organs*, 30(12):929–948, 2006.

[39] Y Shi, T Korakianitis, and C Bowles. Numerical simulation of cardiovascular dynamics with different types of VAD assistance. *J Biomech*, 40:2919–2933, 2007.

[40] F Shu, S Vandenberghe, and J F Antaki. The Importance of dQ/dt on the Flow Field in a Turbodynamic Pump With Pulsatile Flow. *Artificial Organs*, 33:757–762, 2009.

[41] H Suga and K Sagawa. Mathematical Interrelationship Between Instantaneous Ventricular Pressure-Volume Ratio and Myocardial Force-Velocity Relation. *Ann Biomed Eng*, 1:160–182, 1972.

[42] H Suga and K Sagawa. Instantaneous Pressure-Volume Relationships and Their Ratio in the Excised, Supported Canine Left Ventricle. *Circ Res*, 35:117–126, 1974.

[43] H Suga, K Sagawa, and A A Shoukas. Load Independence of the Instantaneous Pressure-Volume Ratio of the Canine Left Ventricle and Effects of Epinephrine and Heart Rate on the Ratio. *Circ Res*, 32(314-322), 1973.

[44] M Ursino. Interaction between carotid baroregulation and the pulsating heart: a mathematical model. *Am J Physiol Heart Circ Physiol*, 275:H1733–H1747, 1998.

[45] R van Loon, PD Anderson, and FN van de Vosse. A fluid-structure interaction method with solid-rigid contact for heart valve dynamics. *J Comput Phys*, 217:806–823, 2006.

[46] S Vandenberghe, P Segers, B Meyns, and P R Verdonck. Effect of Rotary Blood Pump Failure on Left Ventricular Energetics Assessed by Mathematical Modelling. *Artificial Organs*, 26(12):1032–1039, 2002.

[47] S Vandenberghe, P Segers, B Meyns, and P R Verdonck. Unloading Effect of a Rotary Blood Pump Assessed by Mathematical Modelling. *Artificial Organs*, 27(12):1094–1101, 2003.

[48] S Vandenberghe, P Segers, P Steendijk, B Meyns, RAE Dion, JF Antaki, and P Verdonck. Modelling Ventricular Function during Cardiac Assist: Does Time-Varying Elastance Work? *ASAIO J*, 52:4–8, 2006.

[49] Y Wang, N Loghmanpour, S Vandenberghe, A Ferreira, B Keller, J Gorcsan, and J Antaki. Simulation of Dilated Heart Failure with Continuous Flow Circulatory Support. *PLoS ONE*, 9(1):e85234, 2014.