Emulation and calibration with smoothed system and simulator data

Benedict Powell

Emulation and calibration with smoothed system and simulator data

Benedict Powell

A thesis presented for the degree of
Doctor of Philosophy

Statistics
Department of Mathematical Sciences
Durham University

2013

Abstract

Emulation and calibration with smoothed system and simulator data

This thesis is concerned with structuring the statistical model with which we relate physical systems and computer simulators. The novelty of the work lies in the fact that we relate them via imagined smoothed versions of themselves, reflecting the belief that they are similar on large scales but discrepant when in comes to small scale details. Our central, paradigmatic example involves relating the planet's climate to a climate simulator. Here the simulator is suspected to be incapable of faithfully reproducing changes in the system as time or certain physical parameters are changed by a small amount, but is still considered informative for the changes in the system over long time scales and large parameter changes.

Declaration

The work in this thesis is based on research carried out in the Statistics Group at the Department of Mathematical Sciences, Durham University. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Acknowledgements

I would like to thank Michael Goldstein for his guidance, encouragement and patience. All our meetings and discussions over the past few years have been valuable to me but I was especially grateful for the opportunity to attend the 2012 ISBA World Meeting in Japan together.

I am also grateful for the support and camaraderie provided by the staff and postgraduate students of the Maths department. To name them all would be indulgent, but I hope I can repay their efforts when such opportunities arise.

Lastly, I must thank Lucy and my parents for tolerating my occasionally obsessive approach to completing this work, for looking after me so well and for proofreading this document.

Contents

Chapter 1

Introduction

Deterministic computer models, which will be referred to as simulators, now play a key role in almost every scientific field in exploring the implications of theories for physical systems that are too complex to handle ourselves. The application that motivates this thesis is the simulation of the Earth's climate or, more precisely, the statistical analysis of simulated data and those observed in the real world.

While there is a significant body of literature on the analysis and emulation of computer simulators, with notable contributions from Kennedy and O'Hagan [25] and Santner [49] for example, there is less work that has focused on simulators with high-dimensional or time series outputs. Higdon et al. [21], Bayarri et al. [3] and Rougier [48] are among those who have taken on this challenge, and their work informs ours by inspiring dimension reduction strategies relying on basis representations, and computation strategies relying on the decomposition of otherwise intractable calculations. It is the simulation of the planet's climate that necessitates our own contributions to the field by presenting us with huge arrays of outputs and a context for understanding the importance of inferences arising from the analysis.

A central theme to our work is the decomposition of the simulator and system signals into climate and weather components. This partition serves two purposes: firstly, it allows us to model the belief that the simulator may reproduce long-term, low-resolution variability but not high-resolution variability; secondly, the relative smoothness of the climate component means that there are fewer effective degrees of freedom in which it can vary. By focusing only on the dominant degrees of freedom we will be able to render otherwise unmanageably large data sets intelligible and pliable.

This chapter continues with a discussion of issues and objects relevant to the statistical modelling of physical and synthetic systems, and to the content of the thesis.

1.1 Introduction to the field

1.1.1 Data, models and knowledge

We approach this project as Bayesians and anti-realists, and it is in the light of these philosophies that we understand and prioritize the tasks ahead. There is a huge amount of material available, such as [5], introducing Bayesian statistics and exploring its deeper implications, but the central notion to the subjective Bayesian paradigm we will adopt is that statistical models are quantitative representations of reasoning processes: they are descriptions and extrapolations of inherently subjective mental constructs rather than objective physical ones.

The Bayesian perspective complements an anti-realist view of science, a rejection of the existence of universal truths and unobservable entities. From this point of view scientific theory loses some of the prestige it might otherwise have claimed, but is still vitally important. Its role is now to summarise, organize and structure our experiences of various phenomena. Scientific theory becomes our scientific heritage, the culmination of vast numbers of experiments condensed neatly and elegantly into convenient structures.

While we deny the ultimate truth of our scientific and statistical assumptions, in order to reason coherently from one belief to another we act as though they were true. For example, we act as though there is a polynomial curve underlying a series of data because we are forced to make a decision or inference from those data and the curve makes the inference practical. Statistical likelihood serves as a measure of the curve's empirical adequacy to preserve certain information within a data set and a prior may be understood as encoding information from outside the data set. Adopting the curve as truth means adopting the parameters that define it too. We say that they exist because otherwise we could not proceed in our reasoning process, and we say that there is a correct parameter in so far as there is a most successful decision or inference.

'Success' or 'surprise' can be given a mathematical definition in terms of a scoring function or loss function, but it is also important to consider ‘effort’. Our scepticism cushions us from the shocks and revelations that can follow from taking statistical models too literally, and cautions us from investing disproportionate effort in the hunt for parameters that are only tentatively attributed physical significance. The effort involved in an analysis can be hard to talk about. Considerable effort may need to be invested to compensate for inefficient coding, mathematical naivety or a lack of hardware, for example. Whatever the cause, it is worth taking seriously because it is certain to influence our work. Our analyses are likely to involve linear algebra on large matrices, multiple passes over the data and searches through high-dimensional spaces, all of which can be computationally burdensome. The avoidance of excessive computational effort is sometimes equated with the adoption of the principle of parsimony. However, we do not take the view that parsimony is somehow natural: that the laws governing the universe, or describing our statistical models are required to be profoundly elegant. We see the laws as our creations and elegance as a matter of taste. In the work that follows, there are frequently occasions when we must strike a compromise between adopting a model that is easily interpretable and one that is computationally practical. Ideally we will develop models that achieve both, but when that is not possible we make our compromise on a case by case basis rather than letting the principle of parsimony make the decision for us.

Expert knowledge, communicated through parameter values and model choices, and encoded in prior specifications, is the lens through which data is seen and so deserves significant attention in any statistical exercise. In the work that follows we will be particularly interested in parameterisations that focus our attention on quantities we will eventually learn about and those that might help us structure our thoughts in a natural way. These ideas are especially relevant when specifying notions of correlation and smoothness.

Sometimes it is natural to question or re-evaluate our understanding of the data. We can try to model this by building hierarchies of models but we also recognize the need for diagnostics that provide an opportunity for model criticism. Therefore, the identification of informative diagnostics will also be an important part of the methodologies we develop.

1.1.2 Climate and weather

An important notion in our work with simulators is the decomposition of a signal into climate and weather components. This decomposition could be justified on the basis that there are certain features in the data from the system that we hope or believe can be reproduced by a simulator, and others that cannot. In particular we believe that climate simulators, despite not being able to follow weather patterns for more than a few days, can mimic the larger, slower trends in climatological variables. Alternatively, the decomposition could be made on the grounds that we are only interested in the signal at well-spaced intervals, or in its convolution with a smooth function, in which case the correlation between very closely-spaced points could be regarded as irrelevant.

We will therefore frequently introduce statistical models by restating the notional decomposition,

$$y(t) = c(t) \oplus w(t), \tag{1.1}$$

where the notation $y(t)$ is used to describe a physically meaningful scalar function of time, t. The components of (1.1), c and w, represent the climate and weather quantities respectively, and the $\oplus$ symbol is used to express the addition of independent quantities. Of course, additional notation will be required as we progress through the thesis; this is introduced as needed and may be found in the notational glossary in appendix A. A glossary of acronyms and abbreviations is also included in appendix A.2, while appendix D gives brief descriptions of the probability density functions referenced in the course of our research.

We will also explore the consequences of similarly decomposing $y(t)$ seen as a function of a vector of physical variables x that contribute to the system, such as diffusion coefficients and coupling strengths. We will find that doing so provides an interesting way to structure our beliefs for the differences between systems and their simulators.

$$y(t, x) = c(t, x) \oplus w(t, x).$$

We treat the climate and weather variables as constructs that we are free to define. In chapter 2 we discuss routes to making suitable definitions and the identification of the variables given those definitions.

While we employ the climate/weather terminology, we also maintain a certain degree of abstraction from the climatological application so that our ideas might be applicable to the simulation of other systems. It is not uncommon for scientists to consider their simulators, for biological, social, geophysical, mechanical or cosmological processes for example, to be reliable on larger spatial and temporal scales but not on smaller scales. As such, a collection of statistical tools for relating large-scale trends in simulator and system data is likely to be useful to the wider scientific community.

1.1.3 Emulation

The concept of emulation will appear often in this thesis. It refers to the statistical modelling of a quantity that is, in principle, computable but not evaluated, most often because we lack the resources to make the necessary calculations. The randomness of the modelled quantity arises solely from our ignorance of it and not from an intrinsic stochasticity in its behaviour. The output of a complex simulator and the conclusions of an MCMC algorithm are two examples of quantities we might usefully emulate.

1.1.4 Calibration

Calibration, as a concept, is easiest to understand when we have a simulator that is capable of exactly reproducing the mechanisms of the physical system of interest. In this case, to calibrate the simulator is to find the inputs that lead to a simulation which reproduces the observed system output. The equality of the system and simulator outputs defines for us an equivalence for the inputs. So, to calibrate the simulator is to infer the set of parameters, or hidden state variables, of the systems which could have produced the phenomena we experience.

When we assert that the parameter dependence of the system and simulator are different, we need to be more careful about what we mean by calibration. It is no longer the search for inputs that leads to the simulator reproducing the system behaviour. We will think of calibration as the search for the parameters that would reproduce the observed system output if we could perturb and evolve the system through time as though it were another simulator.

In chapter 3 we will discuss our preferred model structure for relating a system and its simulator or simulators. It is only in the context of this structure that we can really make sense of what we are doing when we calibrate. A key aspect of our model is that it admits a spatial analogy for describing the system/simulator relationship. The analogy allows for a parallel interpretation of the model; one in which the parameters index locations in an imaginary space. The distances between two random quantities in that space encode the similarity between them. In this context, calibration is a more like the positioning or arrangement of our beliefs relative to each other.

1.1.5 Climate simulation

We now take a look at the sort of models we will use later on to guide our work and test our ideas. The section also provides some context to climate simulation.

1.1.5.1 General Circulation Models

Twenty-three General Circulation Models (GCM) from research institutes around the world formed a significant proportion of the evidence used to inform the Intergovernmental Panel on Climate Change's (IPCC) fourth assessment report. Given the report's implications for global development policy, the models have a great deal of influence on our future. It is currently possible to browse the code of models from the Institut Pierre Simon Laplace, the Max Planck Institute and NASA's Goddard Institute for Space Studies, but the majority are not publicly available. Simulated data from a small selection of standard experiments are also accessible for many of the models. A common feature of all the GCMs is the computational power they require. Even on the fastest computers, simulations are generally too slow for a standard exploration of the input space that would serve to calibrate or validate the models. Considerable thought is clearly needed to develop an intelligent strategy, or range of strategies, for exploration.

Our closest experience with a GCM is with FAMOUS, a variant on the UK Met Office's HadCM3, which contributed to the findings of the IPCC's Third, Fourth and Fifth Assessment Reports. FAMOUS is a low-resolution ocean-atmosphere GCM whose output was processed and provided to us by members of the RAPID RAPIT research program working at Durham University. Their goal was to explore the behaviour of the AMOC, a powerful ocean current driven by heat and salinity gradients and wind stresses, across a large perturbed physics ensemble of simulations. The climatological relevance of the AMOC lies in the belief that it is responsible for bringing heat at a rate of approximately 1.27 ± 0.15 10^{15}W to northern Europe[13]. A flavour of the data produced by FAMOUS is given by a small sample of time series plotted in figure 1.1. The time series run over a period of about 150000 simulator days, approximately 400 simulator years, and quantify the AMOC in Sverdrups. A flux of one Sverdrup is equivalent to one million cubic metres of water passing through a surface every second. Calculation of the AMOC is not straightforward; it is computed as the sum of three sub-fluxes whose definitions are also quite involved, so we will defer further unwrapping of the quantity and focus instead on its behaviour. On first inspection the variation in the time series of flux strengths produced by FAMOUS is dominated by the differences between groups of simulations that were run under different forcing scenarios. There are eight scenarios in which the amount of CO2 released into the simulated world differed significantly. In the dataset made available to us, three physical parameters are also varied; two continuously and one that takes one of three levels. The dataset provides an additional challenge by being ragged or incomplete, a feature due to many simulations crashing.

1.1.5.2 Earth system Models of Intermediate Complexity

This category of simulator is not defined precisely, but is often used to refer to those run on desktops and small clusters rather than supercomputers. Typically they will exclude certain physical processes like the biosphere and discretise the solution domain into large compartments in order to avoid the computational demands of the GCMs. While it is possible to access the source code for several EMICs (GENIE and PUMA for example), they tend to be specialised research tools rather than robust, user-friendly public products. Compiling, executing and postprocessing stages are time-consuming and fragile. So while experimenting with these models is possible, because our interest is primarily in statistical methodology rather than the Earth system itself, we will not present any examples using EMICs. Instead we will concentrate on synthetic examples produced by statistical models and random number generators until we reach chapter 5, in which we will jump straight to FAMOUS.

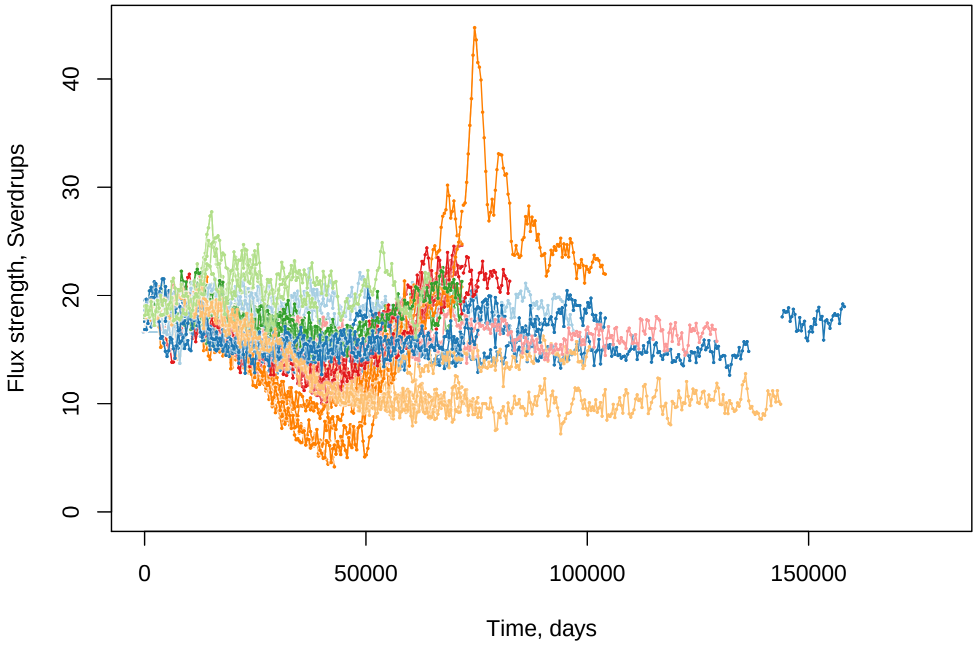


Figure 1.1: A selection of postprocessed time series from FAMOUS.

1.2 Chapter summary

In this chapter we have introduced the key concepts and objects necessary to describe the goal of this thesis: to emulate jointly the smooth part of a climate simulator and the smooth part of a function describing the real world's climate; and to calibrate the latter.

The calibration process will involve using the emulator to estimate observed system values conditional on specific input parameters and comparing the estimates with the observations. Prediction of the system is the natural destination of this work but it is not a topic that we will be able to cover within the scope of the thesis. In principle, our emulators will be capable of providing estimates for system values that are not observed, and these values may be allocated probabilities or plausibilities in the same way as input parameters. This statement is not intended to play down the technical and conceptual difficulties of prediction, rather we wish to acknowledge that it is a very important issue that, although not addressed directly here, is anticipated as a natural continuation of our work.

Our first task is to provide mathematical descriptions for the climate and weather terms and to explore how those descriptions relate to each other, to our understanding of the system, and to methods for inference. In chapter 2 we investigate two complementary routes to the linear smoothing methodology that will allow us to make appropriate variance specifications for climate and weather. We also present a procedure for smoothing large arrays of data and an examination of the inferability of smoothness parameters.

In chapter 3 we formulate a model structure for expressing beliefs for the expected degree and type of correspondence between a system and its simulator, commenting on the implications for emulation and calibration procedures. The model we arrive at supports a integrated interpretation of output, input and discrepancy parameters, and provides a context for understanding the climate not as a type of average or transformation of weather but as a device to link theory and reality.

In chapter 4, we present novel modelling techniques developed in order to process data from a real climate model. These techniques rely on the smoothness of the climate term, allowing it to be well approximated with a moderate number of basis functions, and on the roughness of the weather term, allowing many of its values to be considered uncorrelated. Such features make for a light-weight model that may be quickly and efficiently manipulated, which is crucial for the assimilation of large data sets and for the numerical investigation of the model's fitted parameters. The most significant product of our work in this area is a construct we call the Cholesky emulator. The key innovation behind the Cholesky emulator is an algorithm for thinning out a set of candidate basis functions, coupled with an intuitive interpretation of those basis functions, which helps us to identify correlations as being either important as expressions of belief, or negligible when weighed against the computational demands they introduce. The algorithm is essentially a modification of the standard Cholesky algorithm for matrix decomposition, thus inheriting some of speed and stability properties that have made this latter algorithm such an asset to statisticians and applied mathematicians. Chapter 4 also contains our proposed strategy for using a fitted emulator to infer plausible system parameters from system observations. While deliberation of the role of the simulator in this inference is dealt with in chapter 3, here we consider the practicality of such inferences in regard to their tractability and comprehensibility, concluding that the parallel, yet distinct, notions of ‘likelihood’ and ‘plausibility’ can help us identify parameters at least deserving further investigation, in spite of our reluctance to designate parameters with the labels ‘true’ or ‘false’.

In chapter 5 we demonstrate the calculations involved in the application of our emulation and calibration methodology to the FAMOUS data. The chapter serves as an opportunity to prove the value of the previous chapter’s modelling techniques and to examine how the type of simulator discrepancy described in chapter 3 determines the precision of the inferences we can reasonably make for the system’s parameters.

Finally, in chapter 6 we comment on the issues raised during the course of our research and on avenues of investigation that could not be explored properly here.

Chapter 2

Smoothing

In a wide range of contexts, scientists and statisticians seek to make inferences for smooth trends from noisy data. The trend may represent a physically meaningful process, believed to be part of the mechanism that generated the data, or it may be a device for parsimoniously describing the data. Either way, in order to make such inferences we need to be able to describe quantitatively what we mean by smooth. In this chapter we will investigate two classes of description and discuss their merits in the context of the simulation of physical systems.

Smoothing is particularly relevant to climate modelling because of the belief that, despite not being able to reproduce observed high-frequency weather patterns, climate simulators may still be informative for slower, large-scale climatological trends. To test this belief we will need to extract the smooth trends from the real world climate data and from files of simulated data so that we can compare them. In this way, the smooth plays another role, by constructing metrics between data sets.

Before we progress any further, we need to pin down exactly what we mean by the smooth of a set of points. Given the model

$$y(t) = c(t) \oplus w(t),$$

where all t, $c(t)$, $w(t)$ and $y(t)$ are scalars; we define the smooth to be our expectation of the function $c(t)$ given t. Most frequently we will adjust our expectation by a linear combination of observed data values, in which case we will use the Bayes linear formula (2.1). A full description of Bayes linear methodology is given in [17]; we choose to adopt it here for the geometric clarity of meaning it can provide and for its intrinsic reluctance to commit to probability density functions as statements of belief, which accords with our general scepticism regarding the literal interpretation of physical or statistical models. Using $\mathbf{T}$ and $\mathbf{T}'$ to denote column vectors of observation times we write

$$\mathbb{E}_{y(\mathbf{T})}\left(c(t)\right) = \mathbb{E}\left(c(t)\right) + \mathrm{Cov}\left(c(t)\,,\; c(\mathbf{T})\right)\left(\mathrm{Var}\left(c(\mathbf{T})\right) + \mathrm{Var}\left(w(\mathbf{T})\right)\right)^{-1}\left(y(\mathbf{T}) - \mathbb{E}\left(y(\mathbf{T})\right)\right), \tag{2.1}$$

$$\mathrm{Cov}\left(c(t)\,,\; c(t')\right) = k_c(t,t'), \qquad\qquad \mathrm{Cov}\left(w(t)\,,\; w(t')\right) = k_w(t,t'), \tag{2.2}$$

where the output, climate and weather functions produce column vectors from column vector arguments

$$\begin{aligned} y(\mathbf{T}) &= (y([\mathbf{T}]_1), y([\mathbf{T}]_2), \ldots, y([\mathbf{T}]_N))^T, \\ c(\mathbf{T}) &= (c([\mathbf{T}]_1), c([\mathbf{T}]_2), \ldots, c([\mathbf{T}]_N))^T, \\ w(\mathbf{T}) &= (w([\mathbf{T}]_1), w([\mathbf{T}]_2), \ldots, w([\mathbf{T}]_N))^T. \end{aligned}$$

The smooth is thus also a function with the same domain as $y(t)$, $c(t)$ and $w(t)$. It is not the true function $c(t)$, so we can talk about different smooths of the same series arising from different covariance specifications and about the residual variance for $c(t)$ given its smooth. The adjusted variance for a particular set of values $c(\mathbf{T}')$ is calculated as

$$\begin{aligned} \mathrm{Var}_{y(\mathbf{T})}\left(c(\mathbf{T}')\right) =& \mathrm{Var}\left(c(\mathbf{T}')\right) \\ &- \mathrm{Cov}\left(c(\mathbf{T}')\,,\; c(\mathbf{T})\right)\left(\mathrm{Var}\left(c(\mathbf{T})\right) + \mathrm{Var}\left(w(\mathbf{T})\right)\right)^{-1}\mathrm{Cov}\left(c(\mathbf{T})\,,\; c(\mathbf{T}')\right). \end{aligned} \tag{2.3}$$

For the time being, the subscripted expectation and variance quantities are understood only in the Bayes linear context: as simple estimates based on a linear geometry for belief quantities. As we progress however, especially in chapter 4, we will find it useful to employ likelihoods and full probabilistic specifications. When this happens we will need to start using conditional expectations and variances, which we write with the usual bar notation as $\mathbb{E}\left(\cdot \mid \cdot\right)$ and $\mathrm{Var}\left(\cdot \mid \cdot\right)$. Under this statistical paradigm many of the Bayes linear quantities may be reinterpreted as descriptions of Normal approximations.

In the following subsections we will essentially look at ways to define the covariance functions in (2.2) that determine the elements of the variance matrices in (2.1) and so determine the value of the smooth.

2.1 Basis functions and roughness penalties

Our treatment of basis functions and penalties is informed primarily by the work of Ramsay and Silverman [43] under the title of functional data analysis (FDA). Their work is appealing because they endeavour to define smoothness in terms of quantities with physical relevance. The standard example when introducing penalty approaches is for the inference of the location of a flexible bar that is distorted by attractive forces originating from certain points. Here, the bar is our smoother and the points are observed data quantities. They argue that the smoothest, most natural position of the bar is the one that minimises an expression for the elastic energy within it. More generally though they call such a quantity to be minimised a roughness penalty.

The first steps to thinking about smoothness under the FDA paradigm are to define a linear differential operator (LDO), L, that approximately describes a conserved quantity for the climate system so that

$$Lc(t) = \sum_{m=0}^{M} \tau_m(t) \frac{\partial^m c(t)}{\partial t^m} \approx 0,$$

and then to model $c(t)$ as the sum of a finite set of known basis functions, collected into the column vector $\phi(t)$, with weights given by the column vector of coefficients β,

$$c(t) = \beta^T \phi(t) = \sum_{i=0}^{p} [\beta]_i [\phi(t)]_i. \tag{2.4}$$

The functions $\tau_m(t)$, which may be constant, fulfil the role of covariance parameters and the parameter M encodes the highest penalized derivative. For now we consider only the calculation of smooths conditional on the values of $\tau_m(t)$ being known.

So here we are defining smoothness from two directions that will result in soft and hard constraints on the space of functions we consider smooth. The finite basis means that climate trends must exist inside the space spanned by the basis functions, and within that space we intend to favour functions that almost satisfy the linear differential operator. To do this we let the operator define a penalty as the norm formed by integrating the square of $Lc(t)$ over a period, Ω, of time. The linearity of L means that it acts on each of the basis functions separately, and the penalty may be computed as a quadratic form in the basis coefficients:

$$||c||_L^2 = \int_\Omega (Lc(t))^2 dt, \tag{2.5}$$

$$= \int_\Omega \left(L \sum_{i=0}^{p} [\beta]_i [\phi(t)]_i \right)^2 dt, \tag{2.6}$$

$$= \sum_{i,j=0}^{p} [\beta]_i [\beta]_j \int_\Omega (L[\phi(t)]_i)(L[\phi(t)]_j) dt, \tag{2.7}$$

$$= \sum_{i,j=0} [\beta]_i [\beta]_j [\mathbf{P}]_{i,j}, \tag{2.8}$$

where the integrals have been arranged as a matrix $\mathbf{P}$ satisfying

$$[\mathbf{P}]_{i,j} = \int_\Omega (L[\phi(t)]_i)(L[\phi(t)]_j) dt.$$

The FDA smooth $\hat{c}(t)$ is then defined as the function whose coefficients, gathered into the vector $\hat{\beta}$, minimise the function $\mathcal{L}_{(\lambda)}(\beta)$, which is a weighted sum of the roughness penalty and a squared distance from a set of observations $y(\mathbf{T})$:

$$\mathcal{L}_{(\lambda)}(\beta) = \lambda \beta^T \mathbf{P} \beta + (y(\mathbf{T}) - \boldsymbol{\phi}\beta)^T \mathbf{D}^{-1} (y(\mathbf{T}) - \boldsymbol{\phi}\beta). \tag{2.9}$$

The matrix of constants $\mathbf{D}$ has been introduced here to measure the distance between the observations and the smooth, and the rows of the matrix $\boldsymbol{\phi}$ are comprised of transposed basis function vectors at the observation times:

$$[\boldsymbol{\phi}]_{i,\cdot} = \phi([\mathbf{T}]_i)^T.$$

The minimiser of (2.9), which is found by differentiating $\mathcal{L}_{(\lambda)}(\beta)$ and solving for zero, is given by

$$\hat{\beta}_{(\lambda)} = \left[\lambda \mathbf{P} + \boldsymbol{\phi}^T \mathbf{D}^{-1} \boldsymbol{\phi} \right]^{-1} \boldsymbol{\phi}^T \mathbf{D}^{-1} y(\mathbf{T}), \tag{2.10}$$

or, equivalently,

$$\hat{\beta}_{(\lambda)} = (\lambda \mathbf{P})^\dagger \boldsymbol{\phi}^T \left[\boldsymbol{\phi} (\lambda \mathbf{P})^\dagger \boldsymbol{\phi}^T + \mathbf{D} \right]^{-1} y(\mathbf{T}). \tag{2.11}$$

The † symbol in (2.11) denotes the Moore-Penrose generalised inverse, which we will expand upon shortly. We identify (2.11) with the linearly adjusted expectation $\mathbb{E}_{\mathbf{y}}(c(\mathbf{T}))$ that would have arisen from the linear Bayes belief specification:

$$y(\mathbf{T}) = c(\mathbf{T}) \oplus w(\mathbf{T}) = \boldsymbol{\phi}\beta \oplus w(\mathbf{T}),$$

where β and $\mathbf{w}$ have prior moments:

$$\mathbb{E}(\beta) = \mathbf{0}, \qquad \mathbb{E}(w(\mathbf{T})) = \mathbf{0}, \tag{2.12}$$

$$\mathrm{Var}(\beta) = \sigma^2 \times (\lambda\mathbf{P})^\dagger, \qquad \mathrm{Var}(w(\mathbf{T})) = \sigma^2 \times \mathbf{D}, \tag{2.13}$$

and σ^2 is any non-zero constant. The adjusted variance for β resulting from these specifications is

$$\mathrm{Var}_{\mathrm{y}}(\beta) = \sigma^2 \times \left[\lambda\mathbf{P} + \boldsymbol{\phi}^T\mathbf{D}^{-1}\boldsymbol{\phi}\right]^\dagger .$$

While the differential operator defines a quantity believed to be small, exactly how small, absolutely and relative to the cost of deviating from observations, is determined by the two constants σ^2 and λ respectively. Since σ^2 does not appear in (2.11), the adjusted expectation for β is invariant to σ^2, but not to λ. Thus, λ is the relevant variance quantity to study when the smooth is required simply as a summary of the data, but thought and meaning needs to be to attributed to σ^2 in order to render the smooth a quantifiably approximate estimate of climate. Eliciting a specification for λ may be difficult on physical grounds and, in practice, is often made by choosing a value based on trial and error and visual inspection of the resulting smooths, or on a cross-validation criterion. In section 4.1.1, in which we introduce a full probabilistic model for the c and w terms, we will revisit the quantities λ and σ^2 and demonstrate how they can be estimated with likelihood-based methods.

The term $(\lambda\mathbf{P})^\dagger$ in (2.11) and (2.13) denotes the generalised inverse of $\lambda\mathbf{P}$. It is required because of the possibility that certain linear combinations of coefficients could correspond to functions within the null space of L. These combinations do not contribute to the roughness penalty; they are a priori unconstrained, with the effect that $\mathbf{P}$ possesses zero eigenvalues in certain directions. As long as $\boldsymbol{\phi}^T\mathbf{D}^{-1}\boldsymbol{\phi}$ does not also have zero eigenvalues in these directions, the equivalent adjusted expectation and variance still exist. If $\boldsymbol{\phi}^T\mathbf{D}^{-1}\boldsymbol{\phi}$ does have zero eigenvalues for these directions then there are some linear combinations of β which are constrained neither by the penalty nor the observations; their equivalent prior and adjusted variances are effectively infinite.

These penalty methods are useful to us because with them we can define the climate $c(t)$ as the component of a function $y(t)$ almost within the null space of a differential operator, the excursion from the null space being measured by the penalty. The weather is then the residual variation of $y(t)$ that is not described by $c(t)$.

Example 2.1.1 (Smoothing FAMOUS simulations with an LDO)**.** Here we demonstrate smoothing a time series of a simulated ocean flux from the FAMOUS climate model. Calling the simulator output $y(t)$ and suppressing its dependence on input parameters while we consider just one run, we choose to represent it as the sum of a smooth part, $c(t)$, that we think of as the climate trend and a rough part, $w(t)$, that we think of as the weather trend. We then specify that $c(t)$ exists within the space spanned by 50 b-splines, basis functions that consist of piecewise polynomials joined at equally-spaced knots across a specified interval, and employ a penalty on $c(t)$'s first derivative:

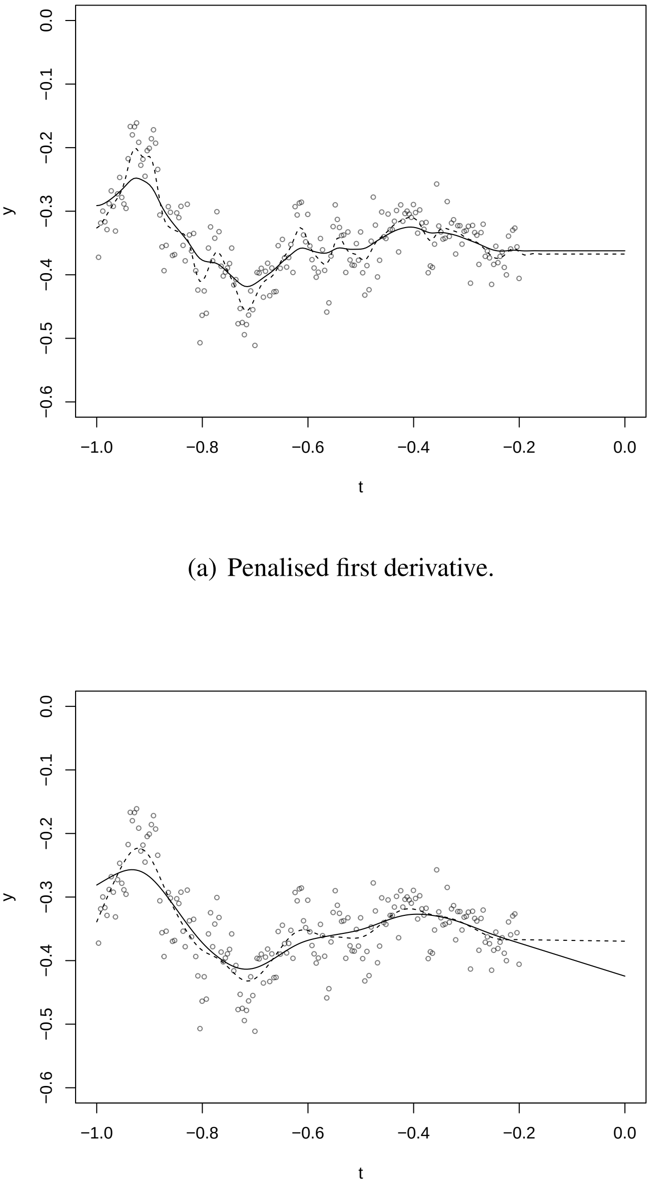
$$y(t) = c(t) \oplus w(t), \qquad\qquad c(t) = \sum_{j=1}^{50} [\beta]_j [\phi(t)]_j.$$

We set $\mathbf{D} = \mathbf{I}$, the identity, so that the deviation of each observation from $c(t)$ contributes independently to the cost function $\mathcal{L}_{(\lambda)}(\beta)$. To produce the smooths in figure 2.1 we do not need to specify σ^2 and we choose two values of λ based on inspection of the curves they produce.

The quantity we are smoothing here is an ocean flux. We assume, for the purposes of the example, that work is required to slow down or speed up the flux but that if left alone it will remain constant, hence the choice of the first derivative as the linear differential operator L. The most important large-scale implication of this choice is that beyond the range of the data, the smooths in figure 2.1(a) level out to a weighted average of the observed values. If we choose the second derivative, as we do to produce figure 2.1(b), the smooths continue beyond the data range on linearly increasing or decreasing paths. Within the data range the smooths generally appear to fulfil our expectations, following the long- and medium-term trends while ignoring very high-frequency variation.

2.2 Stationary fields and autocovariance functions

Inducing smoothness through a differential operator may feel like a circuitous route to take, especially if we lack intuition for an appropriate quantity to penalise. Instead we



(a) Penalised first derivative.

(b) Penalised second derivative.

Figure 2.1: Output of the MOC from one simulation with the climate model FAMOUS. The y axis describes the MOC flux and the t axis describes time, but both variables have been normalised, removing their units. Overlaid are two smooths arising from a set of b-spline basis functions, with a penalty on the curve's first derivative (subfigure 2.1(a)) and a penalty on the curve's second derivative (subfigure 2.1(b)). The smoother curves correspond to a λ penalty multiplier that is ten times greater than that for the rougher curves.

may prefer to directly specify correlations or covariances between $c(t)$ at different times. To do so we define the covariance function

$$k(s,t) = \mathrm{Cov}\left(c(s)\ ,\ c(t)\right).$$

However, the covariance function cannot take arbitrary form. It must lead to coherent belief specifications; specifically, it must produce covariance matrices for sets of values of the climate trend that are positive definite. The greatest part of the literature for covariance functions focuses on the case in which the expectation for $c(t)$ is constant over the time domain, possibly after trends have been removed as a preprocessing step, and the covariance between values of the function at different times is a function only of their separation. In this case we call our beliefs for $c(t)$ weakly stationary, and the covariance function, which now takes the separation between points as its only argument, an autocovariance function. So we define $k(\cdot)$ with one argument as

$$k(|s-t|) = k(s,t) = \mathrm{Cov}\left(c(s)\ ,\ c(t)\right).$$

In the work to come, it will also be convenient to describe matrices of covariance function values. In anticipation of this, we lay out the notation we will use to describe how a covariance function can take two matrix arguments and produce a matrix of values: if we have two matrices of p-dimensional coordinates,

$$\mathbf{X} \in \mathbb{M}(n,p), \qquad\qquad \mathbf{X}' \in \mathbb{M}(m,p),$$

and k is an autocovariance function for a field over p dimensions, then we write the matrix of covariances between the rows of $\mathbf{X}$ and $\mathbf{X}'$ as

$$k(\mathbf{X},\mathbf{X}') \in \mathbb{M}(n,m), \qquad\qquad [k(\mathbf{X},\mathbf{X}')]_{ij} = k(\mathbf{X}_{i,\cdot},\mathbf{X}'_{j,\cdot}),$$

where $\mathbf{X}_{i,\cdot}$ and $\mathbf{X}'_{j,\cdot}$ are p-dimensional coordinate vectors.

The centrepiece to the theory regarding weakly stationary fields is Bochner's theorem [54, p. 24], which states that a complex-valued function on $\Re^D$ is a coherent autocovariance function if and only if it is the Fourier transform of a positive finite measure on $\Re^D$,

$$k(x) = \int_{\Re^D} \exp(i\omega^T x) F(\,\mathrm{d}\omega),$$

where $F(d\omega)$ is that positive finite measure. If F admits a Lebesgue measurable density, we denote it $f(\omega)$ and refer to it as the field's spectral density. When $f(\omega)$ exists we can write

$$k(x) = \int_{\Re^D} \exp(i\omega^T x) f(\omega)\,\mathrm{d}\omega, \tag{2.14}$$

$$f(\omega) = \frac{1}{(2\pi)^D} \int_{\Re^D} k(x) \exp(-i\omega^T x)\,\mathrm{d}x. \tag{2.15}$$

Decomposing the exponential into real and imaginary parts, we can see that in one dimension, spectral densities that are real and symmetric about the origin result in real autocovariance functions, because the imaginary contributions to integral (2.14) cancel out.

Bochner's theorem means that we can think of a function like $c(t)$ as an infinite weighted sum of sinusoidal basis functions, the infinitesimal variance for each of their coefficients being given by the spectral density. This is apparent upon looking at $c(t)$'s inverse Fourier transform as an unknown function. Reducing our scope to one dimension we write

$$c(t) = \int_{\Re} \tilde{c}(\omega) \exp(i\omega t) d\omega, \tag{2.16}$$

$$\tilde{c}(\omega) = \frac{1}{2\pi} \int_{\Re} c(t) \exp(-i\omega t) dt. \tag{2.17}$$

Integral (2.16) is analogous to the sum of basis functions in (2.4) and $\tilde{c}(\omega)$ is analogous to the vector of unknown basis coefficients. For a mean zero field we have

$$\mathbb{E}\left(\tilde{c}(\omega)\right) = \frac{1}{2\pi} \int_{\Re} \mathbb{E}\left(c(t)\right) \exp(-i\omega t) dt = 0,$$

and,

$$\begin{aligned}
\operatorname{Cov}\left(\tilde{c}(\omega)\,,\,\overline{\tilde{c}(\omega')}\right) &= \frac{1}{(2\pi)^2} \int_{\Re} c(s) \exp(-i\omega s) ds \int_{\Re} c(t) \exp(i\omega' t) dt && (2.18)\\
&= \frac{1}{(2\pi)^2} \int_{\Re} \int_{\Re} k(s-t) \exp(-i\omega(s-t)) \exp(i(\omega' - \omega)t) ds dt && (2.19)\\
&= \frac{1}{2\pi} \int_{\Re} f(\omega) \exp(i(\omega' - \omega)t) dt && (2.20)\\
&= \delta(\omega - \omega') f(\omega). && (2.21)
\end{aligned}$$

The delta function in (2.21) implies that each of the frequency components of $c(t)$ is uncorrelated to all others while their variances are given by the function f, the field's spectral density.

Consequently, we can use the notion of the stationary field to define smoothness in terms of the similarity of separated evaluations of $c(t)$ or in terms of a distribution of frequencies that characterise it. In fact, Bochner's theorem also leads to a third way of interpreting the smoothness of a stationary field, which becomes apparent upon the realisation that complex exponentials are eigenfunctions of the differential operator:

$$\frac{\partial \exp(i\omega t)}{\partial t} = i\omega \exp(i\omega t).$$

The equations below, described in [44] but given in full here, show that a roughness penalty comprised of a sum of M squared derivatives weighted by the constants ζ_j can be computed as a particular norm of a curve's spectral density:

$$\begin{align}
\|c(t)\|_P^2 &= \sum_{j=0}^{M} \zeta_j \int_{\Re} \left(\frac{\partial^j c(t)}{\partial t^j}\right)\left(\frac{\partial^j c(t)}{\partial t^j}\right) dt \tag{2.22}\\
&= \sum_{j=0}^{M} \zeta_j \int_{\Re} \left(\frac{\partial^j}{\partial t^j}\frac{1}{2\pi}\int_{\Re} \tilde{c}(\omega)\exp(i\omega t)d\omega\right)\left(\frac{\partial^j}{\partial t^j}\frac{1}{2\pi}\int_{\Re} \overline{\tilde{c}(\omega')}\exp(-i\omega' t)d\omega'\right) dt \notag\\
&= \sum_{j=0}^{M} \zeta_j \int_{\Re} \frac{1}{(2\pi)^2}\left(\int_{\Re} \tilde{c}(\omega)(i\omega)^j\exp(i\omega t)d\omega\right)\left(\int_{\Re} \overline{\tilde{c}(\omega')}(-i\omega')^j\exp(-i\omega' t)d\omega'\right) dt \notag\\
&= \sum_{j=0}^{M} \zeta_j \int_{\Re} \frac{1}{(2\pi)^2}\int_{\Re}\int_{\Re} \tilde{c}(\omega)\overline{\tilde{c}(\omega')}(\omega\omega')^j\exp(i(\omega-\omega')t)d\omega d\omega' dt \notag\\
&= \sum_{j=0}^{M} \zeta_j \int_{\Re} \frac{1}{(2\pi)^2}\int_{\Re}\int_{\Re} \tilde{c}(\omega)\overline{\tilde{c}(\omega')}(\omega\omega')^j\exp(i(\omega-\omega')t)dt d\omega d\omega' \notag\\
&= \sum_{j=0}^{M} \zeta_j \frac{1}{2\pi}\int_{\Re}\int_{\Re} \tilde{c}(\omega)\overline{\tilde{c}(\omega')}(\omega\omega')^j\delta(\omega-\omega')d\omega d\omega' \notag\\
&= \int_{\Re} |\tilde{c}(\omega)|^2\left(\frac{1}{2\pi}\sum_{j=0}^{M}\zeta_j\omega^{2j}\right)d\omega \tag{2.23}\\
&= \int_{\Re} |\tilde{c}(\omega)|^2 g(\omega)d\omega, \tag{2.24}
\end{align}$$

where

$$g(\omega) = \left(\frac{1}{2\pi}\sum_{j=0}^{M}\zeta_j\omega^{2j}\right). \tag{2.25}$$

The first and last expressions (2.22) and (2.24) are, like (2.16), viewed as being analogous to the finite basis case: specifically, as being analogous to (2.5) and (2.8) of the previous section. Both expressions say that the roughness penalty is computable as a quadratic form in the basis coefficients. Previously, we made the connection between the penalty matrix **P** and the inverse prior variance for the basis coefficients. Here, the function $g(\omega)$ acts like an infinite-dimensional diagonal matrix and we make the same connection between $g(\omega)$ and the reciprocal of the spectral density $f(\omega)$. The equivalence between smooths from the penalty and the Bayesian perspectives has been established by Kimmeldorf and Wahba [26] amongst others, and may be further demonstrated using calculus of variations. The technique enables us to show that a function $\hat{c}$ consisting of a sum of autocovariance kernels centred on data points, which is the smooth that results from the autocovariance approach, is a stationary point of the penalised loss function:

$$\int_{\Re} \mathcal{L}\, \mathrm{d}t,$$

where

$$\mathcal{L} = \sum_{j=0}^{\infty} \zeta_j \left(\frac{\mathrm{d}^j c}{\mathrm{d}t^j}\right)^2 + \sigma^{-2} \sum_{i=1}^{N} \delta(t - t_i)(y(t) - c(t))^2.$$

In the lines below we present a heuristic argument in support of this claim, omitting technical issues regarding the behaviour of $\hat{c}$ as its argument tends to the extremes of the domain. The result comes from inserting the loss function into the Euler-Lagrange formula for the functional derivative, which is given by

$$\sum_{j=0}^{\infty} (-1)^j \frac{\mathrm{d}^j}{\mathrm{d}x^j} \left(\frac{\partial \mathcal{L}}{\partial \hat{c}^{(j)}}\right) = 0, \tag{2.26}$$

where $\hat{c}^{(j)}$ is short-hand for the jth derivative of $\hat{c}$ with respect to its argument. We now introduce the ansatz that the smooth is a sum of kernel functions with spectral density, $f(\omega)$, inversely proportional to $g(\omega)$,

$$\hat{c}(t) = \sum_{i=1}^{N} [\hat{\beta}]_i k(t - t_i) = \hat{\beta}^T k(\mathbf{T}, t).$$

With $\hat{c}$ taking this form, the part of (2.26) concerning the roughness penalty becomes,

$$\sum_{i=1}^{N} \sum_{j=0}^{\infty} (-1)^j \frac{\mathrm{d}^j}{\mathrm{d}t^j} \frac{\partial}{\partial \hat{c}^{(j)}} \sum_{n=0}^{\infty} \zeta_n \left(\frac{\mathrm{d}^n \hat{c}}{\mathrm{d}t^n}\right)^2 = \sum_{i=1}^{N} \sum_{j=0}^{\infty} (-1)^j \frac{\mathrm{d}^j}{\mathrm{d}t^j} 2\zeta_j \frac{\mathrm{d}^j \hat{c}}{\mathrm{d}t^j} \tag{2.27}$$

$$= \sum_{i=1}^{N} \sum_{j=0}^{\infty} (-1)^j 2\zeta_n \frac{\mathrm{d}^{2n} \hat{c}}{\mathrm{d}t^{2n}} \tag{2.28}$$

$$= \sum_{i=1}^{N} \sum_{j=0}^{\infty} 2(-1)^j \zeta_j \frac{\mathrm{d}^{2j}}{\mathrm{d}t^{2j}} [\hat{\beta}]_i k(t - t_i). \tag{2.29}$$

Then, expressing the kernel as the inverse Fourier transform of its spectral density we have

$$\sum_{i=1}^{N}[\hat{\beta}]_i \sum_{j=0}^{\infty} 2(-1)^j \zeta_j \frac{\mathrm{d}^{2j}}{\mathrm{d}t^{2j}} \int_{\Re} f(\omega)\exp(i\omega(t-t_i))\,\mathrm{d}\omega. \tag{2.30}$$

The differential operators act only on the exponential term so that (2.30) becomes

$$\begin{aligned}\sum_{i=1}^{N}[\hat{\beta}]_i \sum_{j=0}^{\infty} 2(-1)^j \zeta_j \int_{\Re} (i\omega)^{2j} f(\omega)\exp(i\omega(t-t_i))\,\mathrm{d}\omega \qquad (2.31)\\ = \sum_{i=1}^{N}[\hat{\beta}]_i \int_{\Re} \sum_{j=0}^{\infty} 2\zeta_j\omega^{2j} f(\omega)\exp(i\omega(t-t_i))\,\mathrm{d}\omega. \qquad (2.32)\end{aligned}$$

Now, because we use the spectral density $f(\omega) = 1/g(\omega)$, a whole sum of terms cancel out, meaning that (2.31) is equal to

$$\sum_{i=1}^{N}[\hat{\beta}]_i \int_{\Re} \left(\sum_{j=0}^{\infty} 2\zeta_j\omega^{2j}\right)\left(\frac{1}{2\pi}\sum_{j=0}^{\infty}\zeta_j\omega^{2j}\right)^{-1} \exp(i\omega(t-t_i))\,\mathrm{d}\omega,$$

leaving the exponential, which integrates to leave a sum of Dirac delta functions. This is the key to satisfying the Euler-Lagrange equation,

$$2\int_{\Re}\sum_{i=1}^{N}[\hat{\beta}]_i \frac{1}{2\pi}\exp(i\omega(t-t_i))\,\mathrm{d}\omega = 2\sum_{i=1}^{N}[\hat{\beta}]_i\delta(t-t_i).$$

Reintroducing the part of the Euler Lagrange equation that includes the data, we have

$$2\sum_{i=1}^{N}[\hat{\beta}]_i\delta(t-t_i) + 2\sigma^{-2}\sum_{i=1}^{N}\delta(t-t_i)\left(\sum_{j=1}^{N}[\hat{\beta}]_j k(t-t_j) - y(t)\right) = 0. \tag{2.33}$$

Integrating (2.33) with respect to t leaves us with

$$\sigma^2[\hat{\beta}]_i + \sum_{j=1}^{N}\big([\hat{\beta}]_j k(t_i - t_j) - y(t_i)\big) = 0,$$

which means that the coefficients satisfy

$$\hat{\beta} = (k(\mathbf{T},\mathbf{T}) + \mathbf{I}\sigma^2)^{-1}y(\mathbf{T}),$$

and the values of smooth $\hat{c}$ take the same form as the Bayes linear adjusted expectation derived from a zero prior expectation for the climate and weather, from climate covariances specified by the function k, and weather terms treated as white noise deviates with variance σ^2:

$$\begin{aligned}\hat{c(t)} &= k(t,\mathbf{T})(k(\mathbf{T},\mathbf{T}) + \mathbf{I}\sigma^2)^{-1}y(\mathbf{T})\\ &= \mathrm{Cov}\,(c(t)\,,\ y(\mathbf{T}))\,\mathrm{Var}\,(y(\mathbf{T}))^{-1}\,(y(\mathbf{T}) - \mathbb{E}\,(y(\mathbf{T}))).\end{aligned}$$

In section 2.1 we showed that a curve within a finite-dimensional space spanned by a set of basis functions, which minimises a particular type of physically informed penalty, could also be arrived at as the linear Bayes adjusted expectation given certain prior belief specifications. Here we have, informally, made the case for the equivalent result in which the space of functions is the infinite-dimensional space of smooth functions. The device that allows us to make the connection between the linear Bayes solution and the penalty-based solution is the spectral density. In doing so, it shows itself to be a key conceptual and mathematical instrument for understanding the behaviour of a field.

Tracing the argument backwards (from an autocovariance k, to its spectral density f, to the spectral density's reciprocal g in (2.25), to the norm in (2.22)), we see that we can derive a penalty from an autocovariance function, with the implication that we can reverse engineer a conserved quantity, interpretable as a physical law or theory, that would produce the sort of behaviour the autocovariance describes. The precise form of the relationship between the autocovariance and the penalty identifies the particular link between penalties consisting of sums of squares of derivatives and weakly stationary fields. One notable aspect of the link is that the tail behaviour of the spectral density is determined by the highest power of ω in the sum (2.25) with a non-zero coefficient, which corresponds to the highest derivative that is penalised in (2.22). This provides insight for how the penalisation of higher derivatives makes for smoother smooths. In the next section we will see that the insight can made more explicit by associating the parameter M in (2.25) with the parameter m in (2.43), which we use to parameterise the Matérn autocovariance function. Among the benefits of the association is the way it suggests how we can understand the penalisation of non-integer derivatives.

2.2.1 The Matérn autocovariance function

There are a significant number of autocovariances in popular use, but one that is consistently recommended in the relevant literature, by Stein [54] in particular, is known as the Matérn autocovariance. The common form is derived by taking a non-standardised

Students t-distribution centred on zero, denoted π_{St}, as the field's spectral density:

$$k_{Mat}(t; u, \nu) = \frac{1}{2^{\nu-1}\Gamma(\nu)}(t/u)^{\nu}\mathcal{K}_{\nu}(t/u), \tag{2.34}$$

$$k_{Mat}(t; u, \nu) = \int_{\Re} f_{Mat}(\omega)\exp(i\omega t)d\omega, \tag{2.35}$$

$$f_{Mat}(\omega; u, \nu) = \frac{\sqrt{2\nu}}{u} \times \frac{\Gamma(\nu + 1/2)u}{\Gamma(\nu)\sqrt{\pi}}\left(1 + \omega^2 u^2\right)^{-\nu-1/2}, \tag{2.36}$$

$$\pi_{St}(\omega; u, \nu) = \frac{\Gamma(\nu + 1/2)u}{\Gamma(\nu)\sqrt{\pi}}\left(1 + \omega^2 u^2\right)^{-\nu-1/2}, \tag{2.37}$$

$$\mathbb{E}\,(\omega) = 0, \tag{2.38}$$

$$\text{Var}\,(\omega) = u^{-2}(2\nu - 2)^{-1}, \tag{2.39}$$

where $\mathcal{K}_{\nu}$ is a modified Bessel function of the third kind. Note from (2.36) that the spectral density f_{Mat} is the t-distribution π_{St} with 2ν degrees of freedom scaled by $\sqrt{2\nu}u$ so that the covariance function tends to one as t is taken to zero, making it an autocorrelation function. Specifying a small value for ν makes for a spectral density with flat tails, describing a spiky field in which high frequency terms are to be expected. The parameter u, meanwhile, is a more straightforward scaling parameter, which we will refer to as a correlation length. Two other important autocovariance functions can be shown to arise from particular cases of the Matérn function: when $\nu = 1/2$ the autocovariance decays exponentially, resulting in (2.40), and the spectral density takes the form of a Cauchy distribution; as ν tends to infinity, while $u^2(\nu + 1/2)$ is held fixed, the spectral density tends towards a normal distribution and the autocovariance tends towards what is known as the squared exponential autocovariance (2.41):

$$k_{Mat}(t; u, 1/2) = k_{Exp}(t; u) = \exp(-t/u), \tag{2.40}$$

$$\lim_{\nu\to\infty} k_{Mat}(t; u, \nu) = k_{SE}(t; l_{SE} = 4u(\nu + 1/2)^{1/2}) = \exp\left(-\frac{t^2}{l_{SE}^2}\right). \tag{2.41}$$

For intermediate values such that $m = (\nu - 1/2) \in \mathbb{N}^0$, the Matérn autocovariance takes the form of a polynomial multiplied by an exponential. This observation is useful because these special cases are around an order of magnitude faster to compute than the general case. Specific expressions for them are derivable from from an identity for the modified Bessel function found in the Wolfram functions library[59],

$$\mathcal{K}_{m+1/2}(z) = \sqrt{\frac{\pi}{2}}\frac{\exp(-z)}{\sqrt{z}}\sum_{j=0}^{m}\frac{(m + j)!}{j!(m - j)!}(2z)^{-j} \qquad \text{for} \qquad m \in \mathbb{N}^0. \tag{2.42}$$

Plugging (2.42) directly into (2.34) results in

$$k_{Mat}(t;u,m+1/2) = \frac{1}{2^{\nu-1}\Gamma(m+1/2)}(t/u)^{m+1/2}\sqrt{\frac{\pi}{2}}\frac{\exp(-t/u)}{\sqrt{t/u}}\sum_{j=0}^{m}\frac{(m+j)!}{j!(m-j)!}(2t/u)^{-j},$$

then, collecting powers of $2t/u$ outside of the sum leads to

$$k_{Mat}(t;u,m+1/2) = \frac{1}{2^{\nu-1}}\frac{1}{\Gamma(m+1/2)}\frac{1}{2^m}(2t/u)^m\sqrt{\frac{\pi}{2}}\exp(-t/u)\sum_{j=0}^{m}\frac{(m+j)!}{j!(m-j)!}(2t/u)^{-j}.$$

We then move these powers inside the sum,

$$k_{Mat}(t;u,m+1/2) = \frac{1}{2^{\nu-1}}\frac{1}{\Gamma(m+1/2)}\frac{1}{2^m}\sqrt{\frac{\pi}{2}}\exp(-t/u)\sum_{j=0}^{m}\frac{(m+j)!}{j!(m-j)!}(2t/u)^{m-j},$$

and collect powers of 2 and π,

$$k_{Mat}(t;u,m+1/2) = \frac{1}{\Gamma(m+1/2)}\frac{\sqrt{\pi}}{4^m}\exp(-t/u)\sum_{j=0}^{m}\frac{(m+j)!}{j!(m-j)!}(2t/u)^{m-j}.$$

For non-negative integer m the Gamma function also takes a special form, which allows for some cancellations,

$$k_{Mat}(t;u,m+1/2) = \frac{4^m m!}{\sqrt{\pi}(2m)!}\frac{\sqrt{\pi}}{4^m}\exp(-t/u)\sum_{j=0}^{m}\frac{(m+j)!}{j!(m-j)!}(2t/u)^{m-j},$$

so that we are left with the following expression for the Matérn function,

$$k_{Mat}(t;u,m+1/2) = \frac{m!}{(2m)!}\exp(-t/u)\sum_{j=0}^{m}\frac{(m+j)!}{j!(m-j)!}(2t/u)^{m-j} \quad \text{for } m \in \mathbb{N}^0. \tag{2.43}$$

The Matérn autocovariance functions for $3/2$ and $5/2$, for example, are thus,

$$k_{Mat}(t;u,3/2) = \exp(-t/u)(1+t/u),$$
$$k_{Mat}(t;u,5/2) = \frac{1}{3}\exp(-t/u)(3+3t/u+(t/u)^2).$$

We can also use (2.25) to uncover the derivative penalty equivalent to the Matérn autocovariance. We do so by expanding the reciprocal of its spectral density using the binomial theorem,

$$g(\omega) = \frac{1}{f(\omega)} \propto (u^{-2}+\omega^2)^{\nu+1/2} = \sum_{j=0}^{\infty}\binom{m+1}{j}u^{-2(m+1-j)}\omega^{2j},$$

and matching powers of ω to the term $g(\omega)$ in (2.25). This reveals the coefficients of the equivalent penalty expression to be,

$$\zeta_j \propto 2\pi \binom{m+1}{j} u^{-2(m+1-j)}. \tag{2.44}$$

From (2.44), we can see that another consequence of $m = \nu - 1/2 \in \mathbb{N}^0$ is that the series of implicitly penalised derivatives terminates after finitely many terms.

For our purposes it also proves useful to be able to describe curves with non-zero dominant frequencies because we anticipate needing to smooth objects such as populations and climatological variables, whose medium term oscillations are one of their most interesting features. This can be achieved with the application of the following theorem.

Theorem 2.2.1. *If $k(t)$ is an autocovariance function then $k(t)\cos(\omega_0 t)$, for real ω_0, is also an autocovariance function.*

Proof. For an autocovariance function $k(t)$ with spectral density $f(\omega)$,

$$k(t) = \int_{\Re} f(\omega)\exp(i\omega t)d\omega,$$

we can construct another one, $k_{\omega_0}(t)$, by taking a spectral density composed of a shifted version of $f(\omega)$ and its reflection about zero:

$$\begin{aligned}
k_{\omega_0}(t) = &\int_{\Re}\left(\frac{1}{2}f(\omega-\omega_0)+\frac{1}{2}f(\omega+\omega_0)\right)\exp(i\omega t)d\omega,\\
=&\frac{1}{2}\exp(i\omega_0 t)\int_{\Re} f(\omega-\omega_0)\exp(i(\omega-\omega_0)t)d\omega\\
&+\frac{1}{2}\exp(-i\omega_0 t)\int_{\Re} f(\omega+\omega_0)\exp(i(\omega+\omega_0)t)d\omega,\\
=&\frac{1}{2}(\exp(i\omega_0 t)+\exp(-i\omega_0 t))k(t),\\
=&\cos(\omega_0 t)k(t).
\end{aligned}$$

□

In particular, the autocovariance function corresponding to a pair of t-distributions centred on ω_0 and $-\omega_0$ is just the standard Matérn autocovariance multiplied by $\cos(\omega_0 t)$. More generally, we can make real autocovariance functions by taking the real part of any probability distribution's characteristic function. The act of taking the real part is equivalent to the reflection of the spectral density. The spectral representation also allows us to use metrics for measuring distances between distributions to measure distances between weakly stationary processes.

A decomposition of a field's spectral density represents another natural way to think about climate and weather. We can describe the spectral density as a mixture of distributions and associate one mixture component, or set of components, with climate and the remainder as weather.

Example 2.2.2 (Smoothing noisy Van der Pol observations with an autocovariance function)**.** The point of this example is to demonstrate smoothing with a Matérn autocovariance function, and also to introduce the role a simulator can play in informing a smooth. In example 2.1.1 we approached the task of inferring a climate signal with the belief that its first derivative was small. In this example we imagine that we are given more information; we are told that the climate behaves like a certain simple dynamical system. We will look at the Van der Pol oscillator, but intend for our findings to be more broadly applicable to systems with stable solutions that we can simulate easily.

We imagine that we receive observations of $y(t)$, over an interval of 20 time units, that we consider decomposable as

$$y(t) = c(t) \oplus w(t),$$

and we are told that the function $c(t)$ approximately satisfies the Van der Pol equation with known parameter $x = 2$ so that

$$\ddot{c}(t) - x(1 - c^2(t))\dot{c}(t) + c(t) \approx 0.$$

To calculate the smooth, which is our expectation for $c(t)$ given a set of observations of $y(t)$, we suggest three different autocovariance specifications. The first is a standard Matérn function with correlation length $u = 2$ and spikiness parameter $\nu = 2$. The choice of parameters here reflects the notion that the climate is relatively smooth with a correlation length that is smaller than, but of the same order as, the interval of observations.

Next, we try to exploit insight for the system derived from theoretical consideration of the system equation. Following a re-parameterisation to spherical coordinates for displacement and velocity; for $x > 0$, the Van der Pol oscillator can be shown through theoretical means, namely the Hartman-Grobman theorem and Liénard's theorem, to exhibit a repelling critical point at its origin and a limit cycle around it with a frequency of approximately one. In an attempt to take these findings into account, we specify the second autocovariance as a standard Matérn function, with a longer correlation length $u = 6$, multiplied by $\cos(t)$. We will refer to this product as the resonant Matérn function.

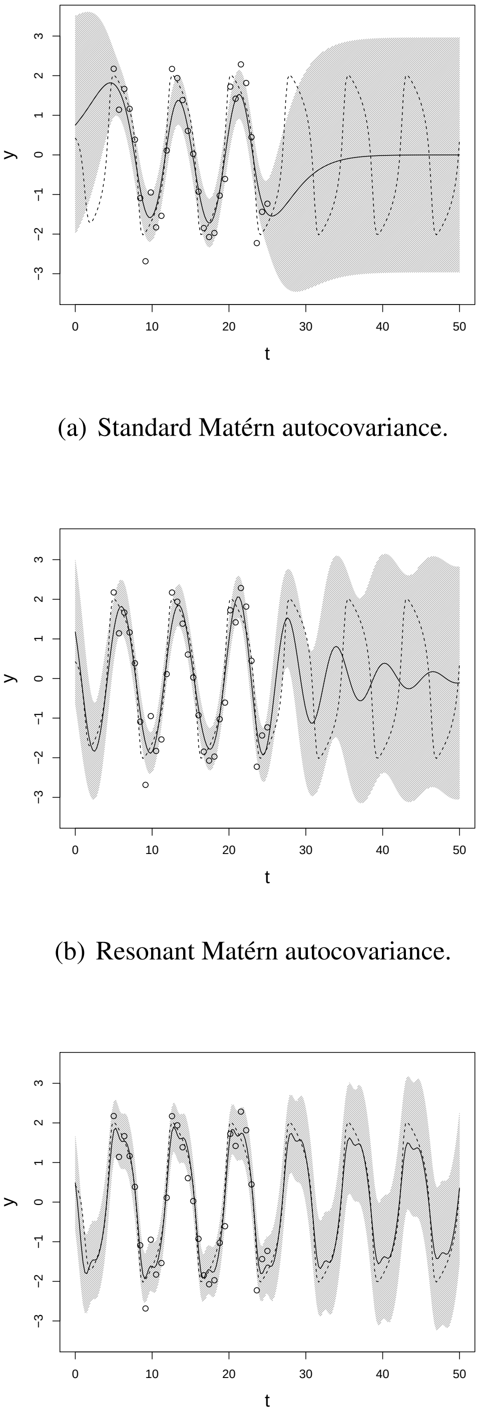
Beyond the existence of the critical point and the cycle, analysis of the Van der Pol equation yields little further information for the behaviour of the system's solutions. Our response is to simulate a long trajectory of the Van der Pol system, using the R [42] package of numerical solvers deSolve [52]. From the simulation of $N_t = 1000$ equally spaced trajectory values over the interval $[0, 40\pi]$ we calculate the following estimates for the autocovariances,

$$\tilde{k}(t_0 - t_j) = \frac{1}{N_t}\sum_{i=1}^{N_t - j}[c(t_i) - \bar{c}][c(t_{i+j}) - \bar{c}], \qquad \text{where} \qquad \bar{c} = \sum_{i=1}^{N_t} c(t_i).$$

Note how dividing by N_t rather than $N_t - t_j$ stabilises and biases our estimates of correlations at great separations towards zero. This choice is recommended by Jenkins [23] on the grounds of reducing mean squared error. We then employ a simple linear interpolator to turn the estimates into a function, which we refer to as the empirical autocovariance function.

To test the autocovariances we use the numerical solver to generate a set of points from an approximate solution of the Van der Pol equation. We then select a subset of those points and add *iid* normal deviates with standard deviation 0.6 to each one. Figure 2.2 shows these observations, the smooths and the pointwise standard deviations around them calculated according to (2.1) and (2.3).

We see that the first autocovariance function is reasonably successful within the data range but rapidly becomes uninformative beyond it. The second one shows a greater ability to extrapolate, but only for a short time. The theory-informed estimate of the resonant frequency is not precise enough for the smooth to stay in phase with the system trajectory for very long. Also, higher frequencies in the system output produce a shape, a sawtooth-type oscillation rather than a clean sinusoid, that the resonant Matérn cannot anticipate. The third subfigure in 2.2 provides the most noticeable conclusion to be taken from the all the plots. The autocovariance function informed by the simulation produces a smooth and variance that clearly capture the true system values, both within and beyond the range of observations, better than the other two.



(a) Standard Matérn autocovariance.

(b) Resonant Matérn autocovariance.

(c) Empirical autocovariance.

Figure 2.2: Noisy observations from the Van der Pol system overlaid with the smooths arising from a range of autocovariance functions (solid lines) and the clean Van der Pol trajectory (dashed lines). The shaded regions mark out intervals with width two times the pointwise standard deviation for c, centred on its expectation.

One may ask why we do not choose to use the simulator in an MCMC scheme; repeatedly reinitialising the simulator and treating the simulations, weighted by their proximity to the observations, as an approximate population of climate values. Our answer is that such a procedure is slow even for a very simple system. Furthermore, given that one of our principal assumptions is that the system that provides us with data is only approximately described by the system that can be simulated, the time and effort required by the MCMC scheme is unlikely to be rewarded with usable information.

This example suggests to us that the smoothness parameters of an autocovariance function, as well as relating to small or penalised derivatives as described in section 2.2, may be seen as describing the attractor of a dynamical system. This is another way in which they may be endowed with physical or mechanistic relevance as well as having relevance as statements of belief for a curve's similarity over time. The example also demonstrates that, even for structurally simple systems, theoretical insight may not be able to lead us very far in terms of quantitatively informing smoothing methods.

The autocovariance function we derive from the long simulated Van der Pol trajectory leads to a highly satisfactory smooth and adjusted variance. Interestingly, the standard time series estimates we use to build the function include a biasing factor that shrinks the covariance estimates for large lags to zero. We note that this bias may be construed as a prior on the autocovariance that steers us away from concluding that high correlations at large lags are appropriate. The prior's sentiment is consistent with uncertainties arising from doubts for the simulator/system correspondence, which we will consider properly in chapter 3.

We leave the example with the view that if we were presented with a data set from another complex simulator or physical system whose governing equations were hidden or intractable, we ought to ask whether they almost behave like a simpler system. We would then investigate the simpler system numerically, as we have done with the Van der Pol system, and use our findings to infer an appropriate covariance function with which to smooth the data.

2.3 Smoothing over many dimensions

In the preceding sections we focused on smoothing functions with one variable, namely time. Now we broaden our attention to multiple dimensions with the intention of smoothing a simulator's output over both its output space and the space of its input parameters x. As before, we choose to think of the output as a sum of independent parts,

$$y(t, x) = c(t, x) \oplus w(t, x),$$

and aim to make inferences for the function $c(t, x)$. The equation for calculating the linearly adjusted expectation for $c(t, x)$, (2.1), does not change in the higher-dimensional setting, but the range of possible covariance functions becomes broader. We also associate the higher-dimensional setting with a significant increase in the size of the data set of observations. This is the case when we start to look at climate models that either produce full spatio-temporal grids of output or that are evaluated on complete grids over their input space.

For now, we will concentrate on the case in which we define covariances across multiple dimensions as the product of covariances in each one so that, in two dimensions for example,

$$\operatorname{Cov}\left(c(t', x')\,,\ c(t'', x'')\right) = k_{ct}(t', t'')k_{cx}(x', x''), \tag{2.45}$$

$$\operatorname{Cov}\left(w(t', x')\,,\ w(t'', x'')\right) = k_{wt}(t', t'')k_{wx}(x', x''). \tag{2.46}$$

Note that there is an obvious over-parameterisation if we allow both factors of the products in (2.45) and (2.46) to vary by a scalar multiplier. How best to deal with this will depend on the form of the factors and for now will not be an issue since we treat the covariance functions as known.

A ramification of the factorisable variance structure is that it can only ascribe different expected modes of variation in the directions of the inputs. It cannot, for example, describe a prior variance for a surface that is expected to exhibit ridges that are not aligned to the coordinate axes.

Computationally, application of the factorisable structure to data that form a full gridded design, leads to major savings when computing a smooth, as it enables us to perform operations mostly on the factors of the covariance matrices involved rather than their products. To show why this is the case we now walk through a two-dimensional example. Our progress relies heavily on the vec $(\cdot)$ notation that unfolds a matrix of numbers, such as a full grid of simulated data $\mathbf{Y}$ whose entry in the ith row and jth column is

$$[\mathbf{Y}]_{i,j} = y(t_j, x_i),$$

into a column vector,

$$\operatorname{vec}(\mathbf{Y}) = \left([\mathbf{Y}]_{\cdot,1}^T, \quad [\mathbf{Y}]_{\cdot,2}^T, \quad \ldots \quad [\mathbf{Y}]_{\cdot,N_t}^T\right)^T, \tag{2.47}$$

by concatenating its columns, where the object $[\mathbf{Y}]_{\cdot,j}$ represents the jth column of $\mathbf{Y}$. This re-shaping of the matrix can also be understood as the creation of a single index, labelled h in (2.49), that runs through all the rows and columns of $\mathbf{Y}$:

$$[\operatorname{vec}(\mathbf{Y})]_{i+(j-1)N_x} = [\mathbf{Y}]_{i,j}, \qquad i = 1, \ldots, N_x, j = 1, \ldots, N_t, \tag{2.48}$$

$$[\operatorname{vec}(\mathbf{Y})]_h = [\mathbf{Y}]_{(h-1)\%N_x+1,\ (h-1)\backslash N_x+1}, \qquad h = 1, \ldots, N_t N_x, \tag{2.49}$$

where $\%$ and $\backslash$ are the standard mod and integer division operators, which are complementary in the sense that,

$$x = (x\backslash y) \times y + x\%y \qquad \text{for } y \neq 0.$$

Example 2.3.1. We would, for example, index the entries of a matrix $\mathbf{Y}$ with the subscripts i_1 and i_2, while indexing the entries of the vector vec $(\mathbf{Y})$ with the subscript i. In the case of $\mathbf{Y}$ being a 2×2 matrix we have

$$\mathbf{Y} = \begin{pmatrix} [\mathbf{Y}]_{1,1} & [\mathbf{Y}]_{1,2} \\ [\mathbf{Y}]_{2,1} & [\mathbf{Y}]_{2,2} \end{pmatrix}, \qquad \operatorname{vec}(\mathbf{Y}) = \begin{pmatrix} [\mathbf{Y}]_{1,1} \\ [\mathbf{Y}]_{2,1} \\ [\mathbf{Y}]_{1,2} \\ [\mathbf{Y}]_{2,2} \end{pmatrix}.$$

where

i	i_1	i_2
1	1	1
2	2	1
3	1	2
4	2	2

We denote the number of simulations performed as N_x, and the number of observations in each simulated time series as N_t, so that $\mathbf{Y}$ and $\text{vec}\,(\mathbf{Y})$ are $N_t \times N_x$ and $N_t N_x \times 1$ matrices respectively. We can then express the prior variance for the outputs as a sum of Kronecker products of matrices,

$$\text{Var}\,(\text{vec}\,(\mathbf{Y})) = \mathbf{V} = \mathbf{A} + \mathbf{B}, \tag{2.50}$$

where

$$\mathbf{A} = (\mathbf{A}_t \otimes \mathbf{A}_x), \qquad \mathbf{B} = (\mathbf{B}_t \otimes \mathbf{B}_x),$$

so that

$$\begin{aligned} [\mathbf{A}_x]_{i,i'}[\mathbf{A}_t]_{j,j'} &= \text{Cov}\left(c(t_j, x_i)\ ,\ c(t_{j'}, x_{i'})\right), \\ [\mathbf{B}_x]_{i,i'}[\mathbf{B}_t]_{j,j'} &= \text{Cov}\left(w(t_j, x_i)\ ,\ w(t_{j'}, x_{i'})\right). \end{aligned}$$

The linear adjustment for values of the climate function requires the inversion of the matrix $\text{Var}\,(\text{vec}\,(\mathbf{Y}))$ which, if performed directly, will become extremely demanding as its size increases. However, the efficiency of the inversion calculation can be vastly improved when we utilise certain algebraic tricks. In the next few paragraphs we walk through our recommended inversion procedure.

The first property of the problem that we can exploit is the smoothness of c, which may render $\mathbf{A}$ almost or exactly rank deficient. For example, if the climate term is specified as a sum of multivariate basis functions derived from products of M_t basis functions of time and M_x basis functions of the simulator's input parameter, we may write

$$c(t_j, x_i) = \sum_{k=1}^{M_t}\sum_{l=1}^{M_x}[\boldsymbol{\beta}]_{k,l}[\boldsymbol{\phi}_t]_{j,k}[\boldsymbol{\phi}_x]_{i,l} = \sum_{k=1}^{M_t}\sum_{l=1}^{M_x}[\boldsymbol{\beta}]_{k,l}[\phi_t(t_j)]_k[\phi_x(x_i)]_l.$$

And if we specify that the covariance matrix for the basis coefficients is factorisable so that

$$\text{Cov}\left([\boldsymbol{\beta}]_{i,j}\ ,\ [\boldsymbol{\beta}]_{i',j'}\right) = [\mathbf{V}_t]_{i,i'}[\mathbf{V}_x]_{j,j'},$$

where $\mathbf{V}_t$ and $\mathbf{V}_x$ are positive definite $M_t \times M_t$ and $M_x \times M_x$ matrices respectively, we can write

$$\mathbf{A}_t = \boldsymbol{\phi}_t \mathbf{V}_t \boldsymbol{\phi}_t^T, \qquad \mathbf{A}_x = \boldsymbol{\phi}_x \mathbf{V}_x \boldsymbol{\phi}_x^T.$$

From here we compute the QR decompositions of $\boldsymbol{\phi}_t$ and $\boldsymbol{\phi}_x$:

$$\boldsymbol{\phi}_t = \mathbf{Q}_t\mathbf{R}_{\phi_t}, \qquad \boldsymbol{\phi}_x = \mathbf{Q}_x\mathbf{R}_{\phi_x}.$$

Most implementations of the QR decomposition also allow us to easily compute $\mathbf{Q}_t^c$ and $\mathbf{Q}_x^c$, the orthogonal completions of $\mathbf{Q}_t$ and $\mathbf{Q}_x$. These two sets of matrices define a partition of the space of linear combinations of data quantities into subspaces inside and outside the span of the columns of $\mathbf{A}$, as well as orthonormal coordinate systems for each one.

If we also compute the Cholesky decompositions of $\mathbf{V}_t$ and $\mathbf{V}_x$, we can write

$$\mathbf{A}_t = \mathbf{Q}_t\mathbf{R}_{\phi_t}\mathbf{R}_{\mathbf{V}_t}^T\mathbf{R}_{\mathbf{V}_t}\mathbf{R}_{\phi_t}^T\mathbf{Q}_t^T, \qquad \mathbf{A}_x = \mathbf{Q}_x\mathbf{R}_{\phi_x}\mathbf{R}_{\mathbf{V}_x}^T\mathbf{R}_{\mathbf{V}_x}\mathbf{R}_{\phi_x}^T\mathbf{Q}_x^T,$$

from which we take the products of the triangular matrices and call them

$$\mathbf{R}_t = \mathbf{R}_{\mathbf{V}_t}\mathbf{R}_{\phi_t}^T, \qquad \mathbf{R}_x = \mathbf{R}_{\mathbf{V}_x}\mathbf{R}_{\phi_x}^T,$$

so that,

$$\mathbf{A}_t = \mathbf{Q}_t\mathbf{R}_t^T\mathbf{R}_t\mathbf{Q}_t^T, \qquad \mathbf{A}_x = \mathbf{Q}_x\mathbf{R}_x^T\mathbf{R}_x\mathbf{Q}_x^T.$$

This reparameterisation of the matrices is equivalent to redefining the basis functions so that the matrices formed by their values at the grid points consist of orthonormal columns. Equivalent expressions can be derived when $\mathbf{A}_t$ and $\mathbf{A}_x$ are derived directly using an autocovariance function. In this case we can either calculate or approximate their eigendecompositions. Eigenvectors corresponding to zero, or machine-zero, eigenvalues are identified with the columns of $\mathbf{Q}_t^c$ and $\mathbf{Q}_x^c$, while the remaining eigenvectors are identified with $\mathbf{Q}_t$ and $\mathbf{Q}_x$.

Defining

$$\mathbf{Q} = \mathbf{Q}_t \otimes \mathbf{Q}_x, \qquad \mathbf{R} = \mathbf{R}_t \otimes \mathbf{R}_x,$$

for convenience, we now partition $\mathbf{B}$ and $\mathbf{V}$ into components that describe variance within and outside the column space of $\mathbf{A}$,

$$\mathbf{V} = \mathbf{Q}\mathbf{R}^T\mathbf{R}\mathbf{Q}^T + \mathbf{B} \tag{2.51}$$

$$= \mathbf{Q}\mathbf{R}^T\mathbf{R}\mathbf{Q}^T + \mathbf{Q}\mathbf{Q}^T\mathbf{B}\mathbf{Q}\mathbf{Q}^T + \mathbf{Q}^c\mathbf{Q}^{cT}\mathbf{B}\mathbf{Q}^c\mathbf{Q}^{cT} \tag{2.52}$$

$$= \mathbf{Q}\mathbf{R}^T(\mathbf{I} + \mathbf{R}^{-T}\mathbf{Q}^T\mathbf{B}\mathbf{Q}\mathbf{R}^{-1})\mathbf{R}\mathbf{Q}^T + \mathbf{Q}^c\mathbf{Q}^{cT}\mathbf{B}\mathbf{Q}^c\mathbf{Q}^{cT}. \tag{2.53}$$

To be clear, the left-hand term in line (2.53) describes the variance of $\operatorname{vec}(\mathbf{Y})$ in the column space of $\mathbf{A}$ and the right-hand term describes the variance orthogonal to it.

As long as the two terms of the sum (2.53) do not share zero eigenvalue eigenvectors, which is guaranteed by $\mathbf{V}$ being positive definite, we can compute the inverse of $\mathbf{V}$ as the sum of the generalised inverses of the summands. In practice, $\mathbf{B}$, the weather variance matrix, is more often than not full rank, meaning that $\mathbf{V}$ is full rank too. The generalised inverse of the second term in (2.53) is easy to compute since the generalised inverse of a Kronecker product is the Kronecker product of generalised inverses:

$$\begin{aligned}(\mathbf{Q}_t^c\mathbf{Q}_t^{cT}\mathbf{B}_t\mathbf{Q}_t^c\mathbf{Q}_t^{cT} \otimes \mathbf{Q}_x^c\mathbf{Q}_x^{cT}\mathbf{B}_x\mathbf{Q}_x^c\mathbf{Q}_x^{cT})^\dagger &= ((\mathbf{Q}_t^c\mathbf{Q}_t^{cT}\mathbf{B}_t\mathbf{Q}_t^c\mathbf{Q}_t^{cT})^\dagger \otimes (\mathbf{Q}_x^c\mathbf{Q}_x^{cT}\mathbf{B}_x\mathbf{Q}_x^c\mathbf{Q}_x^{cT})^\dagger)\\ &= (\mathbf{Q}_t^c\mathbf{Q}_t^{cT}\mathbf{B}_t^{-1}\mathbf{Q}_t^c\mathbf{Q}_t^{cT} \otimes \mathbf{Q}_x^c\mathbf{Q}_x^{cT}\mathbf{B}_x^{-1}\mathbf{Q}_x^c\mathbf{Q}_x^{cT})\end{aligned}$$

The first term of (2.53) is not a straightforward Kronecker product because its middle factor is a sum of Kronecker products. We can handle the problem, however, by invoking theorem B.0.10, which tells us that the eigenvectors of

$$\mathbf{I} + \mathbf{R}^{-T}\mathbf{Q}^T\mathbf{B}\mathbf{Q}\mathbf{R}^{-1} \tag{2.54}$$

are the same as those of

$$\mathbf{R}^{-T}\mathbf{Q}^T\mathbf{B}\mathbf{Q}\mathbf{R}^{-1}. \tag{2.55}$$

We now call upon theorem B.0.12, which tells us that the eigenvectors and eigenvalues of a Kronecker product are given by the Kronecker products of the eigenvectors and eigenvalues of its factors. The theorem allows us to write (2.55) as

$$\begin{aligned}(\mathbf{R}_t^{-T}\mathbf{Q}_t^T\mathbf{B}_t\mathbf{Q}_t\mathbf{R}_t^{-1} \otimes \mathbf{R}_x^{-T}\mathbf{Q}_x^T\mathbf{B}_x\mathbf{Q}_x\mathbf{R}_x^{-1}) &= \mathbf{U}_t\boldsymbol{\lambda}_t\mathbf{U}_t^T \otimes \mathbf{U}_x\boldsymbol{\lambda}_x\mathbf{U}_x^T,\\ &= (\mathbf{U}_t \otimes \mathbf{U}_x)(\boldsymbol{\lambda}_t \otimes \boldsymbol{\lambda}_x)(\mathbf{U}_t^T \otimes \mathbf{U}_x^T).\end{aligned}$$

Together, theorems B.0.10 and B.0.12 show us how the generalised inverse of (2.54) can be derived via the eigen-structure of the factors of (2.55):

$$\begin{aligned}\mathbf{I} + \mathbf{R}^{-T}\mathbf{Q}^T\mathbf{B}\mathbf{Q}\mathbf{R}^{-1} &= (\mathbf{I}_{M_t} \otimes \mathbf{I}_{M_x}) + (\mathbf{U}_t \otimes \mathbf{U}_x)(\boldsymbol{\lambda}_t \otimes \boldsymbol{\lambda}_x)(\mathbf{U}_t^T \otimes \mathbf{U}_x^T), \qquad (2.56)\\ &= (\mathbf{U}_t \otimes \mathbf{U}_x)(\mathbf{I}_{M_t} \otimes \mathbf{I}_{M_x} + \boldsymbol{\lambda}_t \otimes \boldsymbol{\lambda}_x)(\mathbf{U}_t^T \otimes \mathbf{U}_x^T), \qquad (2.57)\\ &= (\mathbf{U}_t \otimes \mathbf{U}_x)\boldsymbol{\Lambda}(\mathbf{U}_t^T \otimes \mathbf{U}_x^T). \qquad (2.58)\end{aligned}$$

Expression (2.58) is easy to invert because inversion of the outer two factors only requires that we invert their factors, and inversion of the central term only requires that we invert the $N_t N_x$ scalars on its diagonal.

Putting together our results, we produce an expression for the inverse of **V**:

$$\begin{aligned}
\mathbf{V}^{-1} =& \mathbf{Q}\mathbf{R}^{-1}\mathbf{U}\mathbf{\Lambda}^{-1}\mathbf{U}^{T}\mathbf{R}^{-T}\mathbf{Q}^{T} + \mathbf{Q}^{c}\mathbf{Q}^{cT}\mathbf{B}^{-1}\mathbf{Q}^{c}\mathbf{Q}^{cT}, && (2.59)\\
=& (\mathbf{Q}_t\mathbf{R}_t^{-1}\mathbf{U}_t \otimes \mathbf{Q}_x\mathbf{R}_x^{-1}\mathbf{U}_x)\mathbf{\Lambda}^{-1}(\mathbf{U}_t^{T}\mathbf{R}_t^{-T}\mathbf{Q}_t^{T} \otimes \mathbf{U}_x^{T}\mathbf{R}_x^{-T}\mathbf{Q}_x^{T}) && (2.60)\\
& + (\mathbf{Q}_t^{c}\mathbf{Q}_t^{cT}\mathbf{B}_t^{-1}\mathbf{Q}_t^{c}\mathbf{Q}_t^{cT} \otimes \mathbf{Q}_x^{c}\mathbf{Q}_x^{cT}\mathbf{B}_x^{-1}\mathbf{Q}_x^{c}\mathbf{Q}_x^{cT}). && (2.61)
\end{aligned}$$

We note that Rougier[48] has also produced work capitalising on the special features of the variance expressions occurring here in order to construct emulators efficiently. In that paper, matrix multiplication is facilitated by the Kronecker factorisation properties of certain matrices, and the low rank of **A** is used in the application of the Sherbury-Morrison-Woodbury inversion formula B.0.6. Rougier does not, however, use our eigen structure argument, namely (2.58), to reduce the cost of the inversion even further. The result is that, given that the weather matrices are structured so that they are easy to invert, the most demanding operation of Rougier's construction scales with the cube of the rank of **A**, or the total number of multivariate basis functions. With our construction it scales with the cube of the highest rank of **A**'s factors, or the size of the largest of the univariate bases that are tensored to make the full multivariate basis. In one dimension these are the same, in two dimensions the difference is already substantial, and in higher dimensions the difference is likely to be great enough for there to be no real competition between the two methods.

We also note that although many of the expressions in this section make use of the vector vec (**Y**), in practice we keep its elements in matrix form, and never actually construct the Kronecker products. When we are required to pre-multiply the vector by a Kronecker product we pre- and post-multiply the matrix by the factors of the multiplier,

$$(\mathbf{B}_t^{T} \otimes \mathbf{B}_x^{T})\operatorname{vec}(\mathbf{Y}) = \operatorname{vec}\left(\mathbf{B}_x^{T}\mathbf{Y}\mathbf{B}_t\right), \tag{2.62}$$

and when the expressions require us to pre-multiply the vector by a diagonal matrix we entrywise multiply by another matrix, which is constructed by folding up the diagonal matrix's diagonal elements in an action that reverses the vec (·) operation:

$$\mathbf{\Lambda}\mathrm{vec}\,(\mathbf{Y}) = \mathrm{vec}\,(\mathbf{D} \circ \mathbf{Y})\,, \tag{2.63}$$

$$[\mathbf{\Lambda}]_{i,i} = [\mathbf{D}]_{(i-1)\%N_x+1,(i-1)\backslash N_x+1}\,. \tag{2.64}$$

If the set of points at which we want to calculate the adjusted expectation of $c(x,t)$ also lies on an axis-aligned grid, then the covariance between the values of $c(x,t)$ here and the observations vec ($\mathbf{Y}$) is also factorisable. This means that we can carry on using identities 2.62 and 2.63 to compute the adjustment for the grid of smooth values.

Exploitation of the structure of $\mathbf{V}$ can also be used to efficiently calculate its determinant, which will be helpful in evaluating probability densities. Specifically, we can write the determinant as any one of:

$$\begin{aligned}
|\mathbf{V}| = |\mathbf{\Lambda}| \quad & \times|\mathbf{R}_t|^{2M_x}|\mathbf{R}_x|^{2M_t} && \times|\mathbf{Q}_t^T\mathbf{B}_t\mathbf{Q}_t|^{N_x-M_x}|\mathbf{Q}_x^T\mathbf{B}_x\mathbf{Q}_x|^{N_t-M_t},\\
= |\mathbf{\Lambda}| \quad & \times|\mathbf{A}_t|_+^{M_x}|\mathbf{A}_x|_+^{M_t} && \times|\mathbf{Q}_t^T\mathbf{B}_t\mathbf{Q}_t|^{N_x-M_x}|\mathbf{Q}_x^T\mathbf{B}_x\mathbf{Q}_x|^{N_t-M_t},\\
= |\mathbf{\Lambda}| \quad & \times|\mathbf{R}_{\phi_t}|^{2M_x}|\mathbf{R}_{\phi_t}|^{2M_t}|\mathbf{V}_t|^{M_x}|\mathbf{V}_x|^{M_t} && \times|\mathbf{Q}_t^T\mathbf{B}_t\mathbf{Q}_t|^{N_x-M_x}|\mathbf{Q}_x^T\mathbf{B}_x\mathbf{Q}_x|^{N_t-M_t},
\end{aligned}$$

where

$$|\mathbf{\Lambda}| = \prod_{i=1}^{M_x}\prod_{j=1}^{M_t}\Big([\boldsymbol{\lambda}_x]_{i,i}[\boldsymbol{\lambda}_t]_{j,j} + 1\Big)\,.$$

All these tricks for factorisable matrices can be generalised up into higher dimensions and allow us to perform calculations that would quickly fill the RAM of a desktop computer if we simply built the full variance matrices. Using the index d to identify particular dimensions, we redefine $\mathbf{Y}$ to be a D-dimensional array with extents N_d. So, to clarify, $\mathbf{Y}$ is constructed by tensoring a grid of N_1 points in the first dimension with a grid of N_2 points in the second dimension, producing an array that is tensored with a grid of N_3 points in the third dimension and so on until the Dth dimension. From this definition, we define the generalisation of the vec (·) operator as,

$$[\mathrm{vec}\,(\mathbf{Y})]_i = [\mathbf{Y}]_{i_1,\ldots,i_d,\ldots,i_D}, \tag{2.65}$$

where the correspondence between the indices is given by

$$
\begin{aligned}
i_1 &= (i-1)\%N_1 + 1, &\quad (2.66)\\
i_d &= ((i-1)\backslash N_1 \ldots N_{d-1})\%N_d + 1, &\quad (2.67)\\
i_D &= (i-1)\backslash N_1 N_2 \ldots N_{D-1} + 1, &\quad (2.68)\\
i &= i_1 + \ldots + (i_d - 1)N_1 \ldots N_{d-1} + \ldots + (i_D - 1)N_1 \ldots N_{D-1}. &\quad (2.69)
\end{aligned}
$$

As in the two-dimensional context, the vec $(\cdot)$ operator still produces a one-dimensional array or vector from its argument. Indices (2.66)-(2.69) are also used to describe the Kronecker product of D matrices:

$$
[\mathbf{A}_D \otimes \ldots \otimes \mathbf{A}_d \otimes \ldots \otimes \mathbf{A}_1] = \left[\bigotimes_{d=1}^{D} \mathbf{A}_d\right]_{i,j} = \prod_{d=1}^{D} [\mathbf{A}_d]_{(i_d, j_d)}, \tag{2.70}
$$

where the $\mathbf{A}_d$ are matrices of size $N_d \times N_d$ and the order of the product is important. The indices we use here are consistent with the indexing convention in R, so that the leftmost index cycles fastest. Having defined the higher-dimensional Kronecker product, we can generalise (2.61) in order to compute the variance for a D-dimensional grid of data,

$$
\begin{aligned}
\mathbf{V} &= \left(\bigotimes_{d=1}^{D} \mathbf{A}_d\right) + \left(\bigotimes_{d=1}^{D} \mathbf{B}_d\right),\\
&= \left(\bigotimes_{d=1}^{D} \mathbf{Q}_d \mathbf{R}_d^T \mathbf{U}_d\right) \mathbf{\Lambda} \left(\bigotimes_{d=1}^{D} \mathbf{U}_d^T \mathbf{R}_d \mathbf{Q}_d^T\right) + \left(\bigotimes_{d=1}^{D} \mathbf{Q}_d \mathbf{Q}_d^T \mathbf{B}_d \mathbf{Q}_d \mathbf{Q}_d^T\right),
\end{aligned}
$$

and its inverse,

$$
\mathbf{V}^{-1} = \left(\bigotimes_{d=1}^{D} \mathbf{Q}_d \mathbf{R}_d^{-1} \mathbf{U}_d\right) \mathbf{\Lambda}^{-1} \left(\bigotimes_{d=1}^{D} \mathbf{U}_d^T \mathbf{R}_d^{-T} \mathbf{Q}_d^T\right) + \left(\bigotimes_{d=1}^{D} \mathbf{Q}_d \mathbf{Q}_d^T \mathbf{B}_d^{-1} \mathbf{Q}_d \mathbf{Q}_d^T\right).
$$

Just as in the two-dimensional setting, although we use the expanded Kronecker notation to derive the result for the inverse, construction of the resulting matrices is avoided by performing calculations on their components. Specifically, we follow rules analogous to (2.62) and (2.63) relating to the multiplication of higher-dimensional arrays. In fact (2.62) and (2.63) are arguably easier to understand when we see them as special cases of the generalised identities.

The generalisation of (2.62) requires that we extend the notion of matrix multiplication beyond just left and right matrix multiplication; we imagine that we can multiply the array $\mathbf{Y}$ from as many directions as it has dimensions. In fact, we define an operator that does just that: $\mathcal{M}(\cdot)$ takes a D-dimensional array $\mathbf{Y}$ and a set of D matrices $\mathbf{B}_d$ and returns another D-dimensional array with elements

$$[\mathcal{M}(\mathbf{B}_1, \ldots, \mathbf{B}_D, \mathbf{Y})]_{j_1,\ldots,j_D} = \sum_{i_1=1}^{N_1} [\mathbf{B}_1]_{j_1,i_1} \ldots \sum_{i_D=1}^{N_D} [\mathbf{B}_D]_{j_D,i_D} [\mathbf{Y}]_{i_1,\ldots,i_D}.$$

Pre-multiplication of $\text{vec}(\mathbf{Y})$ by a Kronecker product is equivalent to sequentially tensor multiplying $\mathbf{Y}$ by the product's factors along each of its extents,

$$\left(\bigotimes_{d=1}^{D} \mathbf{B}_d\right) \text{vec}(\mathbf{Y}) = \text{vec}(\mathcal{M}(\mathbf{B}_1, \ldots, \mathbf{B}_D, \mathbf{Y})).$$

The generalisation of (2.63) is more straightforward as it is more obvious how the relevant operations scale up with the dimension. Just as in the two-dimensional case, all we need to do is fold up the diagonal of the multiplying matrix into a D-dimensional array by reversing the $\text{vec}(\cdot)$ operator, and entrywise multiply it with $\mathbf{Y}$

$$\begin{aligned} \mathbf{\Lambda}\text{vec}(\mathbf{Y}) &= \text{vec}(\mathbf{D} \circ \mathbf{Y}), \\ [\mathbf{D} \circ \mathbf{Y}]_{i_1,\ldots,i_D} &= [\mathbf{D}]_{i_1,\ldots,i_D} [\mathbf{Y}]_{i_1,\ldots,i_D}, \\ [\mathbf{\Lambda}]_{i,i} &= [\mathbf{D}]_{i_1,\ldots,i_d,\ldots,i_D}, \end{aligned}$$

So in practice, we store factorisable matrices in terms of their factors, and diagonal matrices as arrays of the same size as $\mathbf{Y}$. Multiplication of arrays takes place over one index at a time or by performing entrywise multiplication. At this point, it is not the inversion of large variance matrices that limits the scale of our computation, nor is it their construction and storage, rather it is the manipulation of arrays like $\mathbf{Y}$ of size $\prod_{d=1}^{D} N_d$.

For sophisticated simulators of physical systems with grid-based solvers, we anticipate being faced with complete four-dimensional grids of outputs. Large complex simulators tend to have a great number of variable parameters and are slow to run, meaning that although we may have access to large ensembles of simulations it is likely to be too expensive to produce full grids of simulator data across the input space. The full-gridded structure may become relevant again if we consider a small number of forcing scenarios for the system, under which all simulations are repeated.

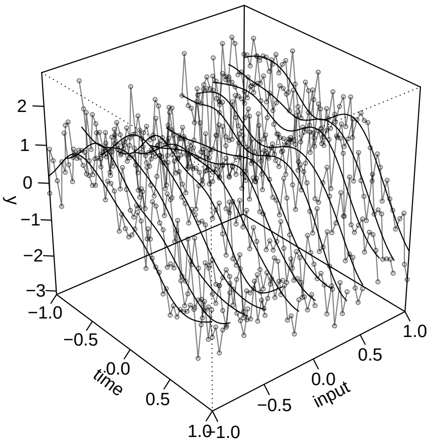
Example 2.3.2 (Smoothing over time and input dimensions)**.** Let us consider a simulator that takes in one input $x \in [-1, 1]$ and returns a time series of data points at equal intervals also over $[-1, 1]$. We specify the variance structure of the output in the way described in (2.50), and choose Matérn autocovariance functions of the form

$$k_{wt}(t'; \sigma, u, \nu, \omega) = \sigma^2 k_{Mat}(t'; u, \nu) \cos(\omega t'),$$

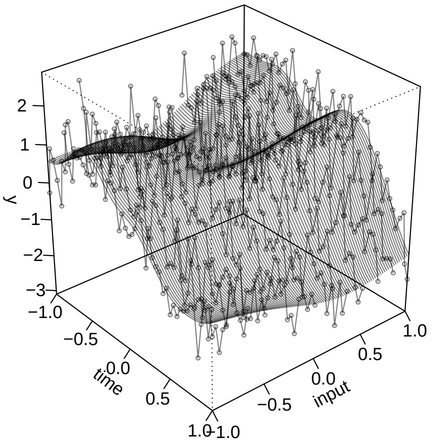
equipped with the parameters given in table 2.1, to define particular covariance values. Figure 2.3(a) shows the $N_x = 13$ interpolated series of $N_t = 31$ points drawn from a multivariate normal distribution with a variance matrix determined by the autocovariance functions. Overlaid are the individually smoothed series, that is to say each smooth is an adjusted expectation given only the data from the corresponding time series. Figure 2.3(b) shows the data again and the joint smooth, which is the expectation for c given all the series simultaneously. In both plots we can see how the expectation for $c(x, t)$ is being informed by the observations $y(x, t)$. The surface being traced out by the individual smooths is not too dissimilar to the surface of the joint smooth. The difference between the individual and joint smooths becomes clearer in figures 2.4(a) and 2.4(b); these are projections of the three-dimensional plots, which also include shaded regions around the expectations with height equal to four times the pointwise standard deviation for $c(x, t)$. We can see that the precision of the smooths is considerably degraded when we ignore their correlations across the input space. These more subtle differences have the potential to be magnified when we progress to calibration, as we will see in the continuation of this example in 3.1.2.

	σ	u	ν	ω
k_{cx}	1	0.5	3	0
k_{ct}	1	0.2	3	0
k_{wx}	1	0.02	1	0
k_{wt}	0.5	0.02	1	30

Table 2.1: Variance parameters for components of example 2.3.2's simulated data.

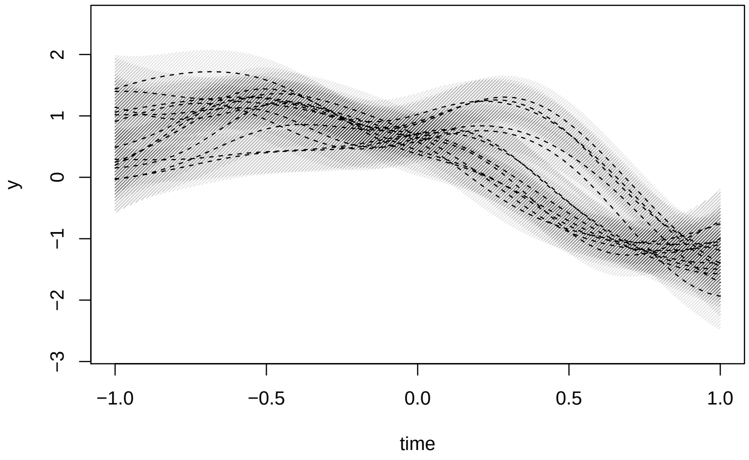


(a) Individual smooths.

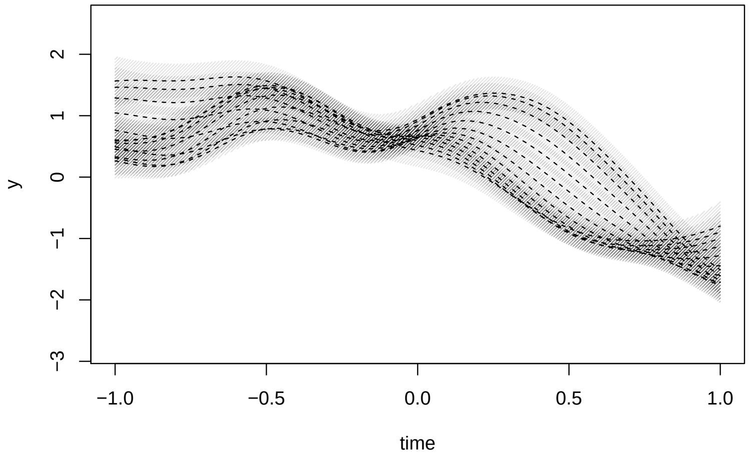


(b) Joint smooths.

Figure 2.3: Subfigure 2.3(a) shows a plot of the individually smoothed simulated time series from example 2.3.2. Subfigure 2.3(b) shows a plot of the joint smooths at the input coordinates corresponding to observed simulations as well as others at which the join smooths are not defined. The spiky lines are interpolations of the simulated series themselves $y(x_i, t)$.



(a) Individual smooths.



(b) Joint smooths.

Figure 2.4: Projections of the individual and joint smooths of the simulated data, corresponding to figures 2.3(a) and 2.3(b), with shaded regions indicating plus and minus two standard deviations for each estimate of $c(x_i, t)$.

2.4 Inference for smoothness parameters

The identification of appropriate smoothness parameters, such as the weighting parameter λ in the roughness penalty context or the correlation length parameter u in the autocovariance context, has a reputation for being difficult. A priori specification of these parameters is hard because the implications for the type of smooth that results from them are often unclear. Furthermore, learning about them from data has the potential to become highly demanding in terms of computation since evaluation of the likelihoods involved tends to require the inversion of large matrices, and is something we will need to repeat many times. Even when we can manage the computation, we may also find that the likelihoods we employ are particularly sensitive, perhaps inappropriately so, to certain data points, normally those that are very close together. Such difficulties are enough for Ramsay and Silverman [43] to suggest picking smoothness parameters by eye, and developing interactive elicitation tools. Nevertheless, we would like to pursue the inference of smoothness parameters from data primarily because we are interested in identifying interesting simulator input parameters on the basis of whether the smoothness of the resulting output is similar to that of the relevant physical system.

With regard to computational demand, the matrix inversion bottlenecks may be ameliorated if the matrices involved are characterised by certain structures that can be exploited by specialised algorithms. In particular, if the covariance matrix is defined using a covariance function with compact support, as advocated by Kaufman in [24], and we can arrange the non-zero entries around the main diagonal, we can use the Thomas algorithm for block tridiagonal matrices, which we describe in appendix B.1. Another special case occurs when the time points corresponding to the rows of the variance matrix are equally spaced and the covariance function is stationary; in this case the matrix is a symmetric Toeplitz matrix, which may be solved for using the Levinson-Durbin algorithm. This algorithm is also described in appendix B.2.

Additionally we ought to ask ourselves whether evaluating the likelihood very many times with only slightly different covariance parameters is really worthwhile. Evaluating the likelihood insufficiently many times will contribute numerical approximation error to our optimisation or integration calculations, but is this error really significant? We ought to acknowledge that synthetic examples are likely to be misleading in helping us answer this question, because when we have created an example in which the covariance values really do exist, and could be used to make predictions for further data, then it does make sense to invest in precise estimates. When the data come from a physical system, whose values only approximately fit the moments described by a stationary field, then moderate numerical error is likely to be dwarfed by the error arising from model misspecification.

If we specify a prior and likelihood function in order to produce a posterior density over the covariance parameters, an understandable reaction is to attempt to maximise it. We can use off-the-shelf optimisation routines to locate modes and approximate the Hessian there as a measure of the precision of the estimate the mode represents. Informally, in practice, we have found this type of optimisation to be problematic. Without adequate customisation, the routines are prone to instability and the erroneous identification of optima, diagnosable from the indefiniteness of the Hessian there. Even with the necessary fine-tuning, the usefulness of the mode as an indicator of the location of the most interesting parameters is questionable since it does not take into account the location of the bulk of the 'good' parameters, just the single 'best' one. Mackay [29] provides an interesting discussion of this point, illustrating it with a calculation based on a multivariate normal distribution, and points out that a mode typically becomes even less representative of a distribution as its dimension increases. We agree that, for uni-modal distributions, better descriptions of the posterior are its mean and variance as they do take into account the location of the posterior's mass rather than its height. We then need to estimate these statistics via numerical integration, which is normally an unattractive prospect since it usually involves a very high number of density evaluations. To render the integrations tractable we turn to adaptive Gaussian quadrature, a technique expounded by Naylor and Smith in [33] to cut down the number of evaluations by utilising knowledge of the integrand's functional form. The Gaussian quadrature method is thus particularly efficient when a prior, whose form is well known, is the dominant force in the posterior. The simple extension of the work in [33] to higher dimensions via the tensoring of quadrature grids again leads to impractically high number evaluations. This problem is reduced with the introduction of sparse grid methodology which involves building high-dimensional grids as sums of tensor products only of subsets of grid points in each dimension. We can utilise sparse grids for our integration here with the help of Jelmer Ympa's R package

SparseGrid [62].

Vihola's [57] Robust Adaptive MCMC algorithm has also proven to be an extremely useful tool for estimating integrals over posterior densities. This Metropolis-Hastings algorithm tunes its proposal step size by targeting an acceptance rate, bypassing the estimation of the posterior's variance or roughness, which are quantities that might otherwise be used to inform the proposal. This algorithm has been prepared for use in R by Andreas Scheidegger [50].

Example 2.4.1 (Looking at the identifiability of Matérn autocovariance parameters)**.** In this example we briefly examine the type of likelihood surface across the space of covariance parameters that we can expect to see upon the observation of a time series.

In figure 2.5 we have plotted two sets of $N_t = 100$ synthetic training data we will use to estimate the covariance parameters. The two interpolating curves here are almost indistinguishable: the first shows the interpolation of points drawn from a multivariate normal distribution whose covariance matrix was specified using the Matérn function with $\sigma^2 = 1$, $u = 0.5$, $\nu = 2.5$; the second shows the interpolation of those points with the addition of a very small error term consisting of *iid* $\mathrm{N}\left(0, 0.03^2\right)$ deviates.

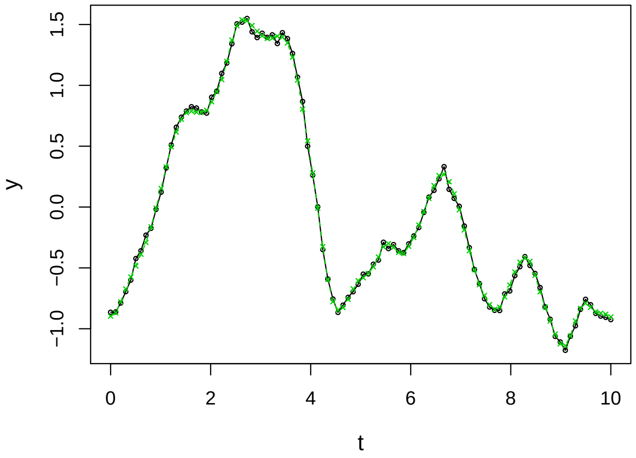


Figure 2.5: Plots of the synthetic training data. The dashed green line shows the interpolation of the points arising from the Matérn covariance, while the solid black line shows the superposition of these points with a small white noise contribution.

The point we wish to highlight here is that with the addition of even a very small white noise error term the inferability of the covariance parameters u and v is significantly degraded. In particular, the likelihood is often only very weakly informative for an upper bound on the spikiness or differentiability parameter v. Figures 2.6(a)-2.6(d) show contour plots of the likelihood of the training data, conditional on $\sigma^2 = 1$, for the cases that do and do not include the white noise terms. The plots were produced using complete two-dimensional grids on which multivariate normal likelihoods were calculated with the help of the Levinson-Durbin algorithm.

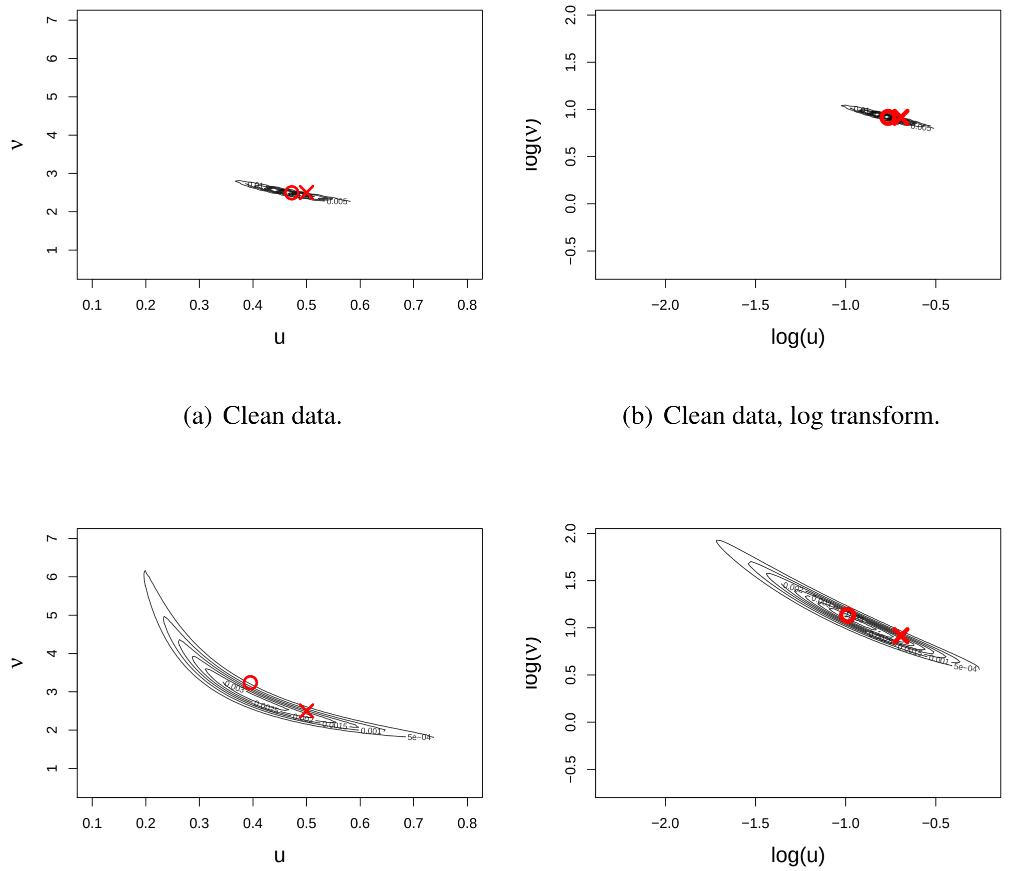
An implication of the reduced precision, signified by the looser contours in figures 2.6(c) and 2.6(d), is that the Matérn autocovariance function's ability to discern between fields with different degrees of differentiability is mostly wasted in a model that includes a white noise error term. Thus we ought to lose little by using the powered exponential covariance function instead, or by considering only one of the special cases of the Matérn autocovariance, described in (2.43), whose computation is considerably cheaper. We also see that the log scale is the natural setting for the numerical exploration of the likelihood. This is particularly noticeable in the way the bent contours of figure 2.6(c) are straightened in figure 2.6(d).

2.4.1 Integrating over smooths

In this section we look at more complicated ways to arrive at a smooth for a time series. In doing so we move closer to the model structure of chapter 3 while developing our understanding for the potential ambiguity between climate and weather, and for some of the numerical techniques described in section 2.4.

We have seen, in example 2.4.1, how observations from a time series are informative for covariance parameters without identifying them exactly. We will shortly look at how the residual uncertainty for the covariance parameters translates into uncertainty for the climate term, but before doing so we consider the significance of covariance parameter uncertainty in terms of the ideas it implies.

When we loosen our precise specification of the covariance parameters, the question of the dependence between those describing climate and those describing weather is raised. If we parameterise the covariance functions for $c(t)$ and $w(t)$ identically, and specify identical priors for them as well, the climate and weather trends may exchange



(a) Clean data.

(b) Clean data, log transform.

(c) Noisy data.

(d) Noisy data, log transform.

Figure 2.6: Here we show contour plots of the likelihood for the clean and noisy time series of example 2.4.1 over a range of values for the Matérn function's correlation length variable, u, and its spikiness parameter, ν. In the right-hand plots, the likelihood domain is transformed to the log scale. The red crosses mark the location of the true covariance parameters used to generate the training data, while the red circles mark the locations of the posterior means that the likelihoods would produce in conjunction with improper uniform priors over their respective domains.

places, leading to symmetry in the posterior, which corresponds to both confounding in our estimates and to redundancy in calculations involving the posterior. These problems are alleviated when we break the symmetry between the components by allocating them distinct priors, which constitute a mathematical formulation for our understanding of the distinction between climate and weather. It may be argued that the natural destination of the argument for asymmetry is to model the climate and weather terms relative to one another, that is, to model the covariance parameters jointly with scaling factors so that, if we specify covariances for climate and weather as

$$\mathrm{Cov}\left(c(t)\ ,\ c(t')\right) = \sigma_c^2 k_{Mat}(|t-t'|; u_c, \nu_c), \quad \mathrm{Cov}\left(w(t)\ ,\ w(t')\right) = \sigma_w^2 k_{Mat}(|t-t'|; u_w, \nu_w),$$

for example, then we should induce priors for the weather parameters via priors on the climate covariance parameters and for the scaling parameters $\gamma_1, \gamma_2, \gamma_3$ such that

$$\sigma_w^2 = \gamma_1 \sigma_c^2, \qquad u_w = \gamma_2 u_c, \qquad \nu_w = \gamma_3 \nu_c.$$

Allocating beta prior distributions to the scaling factors would therefore be one choice for formulating the notion that weather exhibits smaller, rougher variation than climate.

By defining climate and weather relative to each other, or more precisely by including parameters that scale them both simultaneously, we may equate climates on different scales, allowing us, for example, to entertain the idea that time might be effectively running fast or slow in the simulator. This is an intriguing possibility and would be appropriate if the high- and low-frequency components of the output quantity represented subsystems for which the input and output quantities have the same meaning.

The relative prior specification is also likely to be an appropriate modelling choice if we were to produce robust software packages of smoothing functions. For we ought to anticipate that users of the software would have data that exhibits multiscale behaviour, because they are seeking smoothing tools, but we would not be able to anticipate the scales of their input and output variables. Thus the relative size and roughness of the climate and weather terms are predictable whereas their absolute size and roughness are not, this feature may be captured naturally in the relative model.

The relative definitions of climate and weather are less appropriate when, for instance, the rate at which time passes has absolute relevance that extends beyond the observed system, which is the case when we have timed forcings like solar events and greenhouse gas emissions scenarios. The relative model would also be inappropriate if the weather type variation is actually attributable to numerical error and has no appreciable relevance to physical processes involved with climate, like the passage of time.

The climatological simulations we will eventually smooth and emulate in chapter 5 do involve precisely timed emissions scenarios and much of the high-frequency variation is suspected to be produced by non-physical numerical error. So with the benefit of this foresight, in the following example we will allocate independent priors to the covariance parameters for climate and weather.

When the covariance parameters are considered unknown, we define the marginal smooth of the data as the expectation for the climate signal over the covariance parameters for $c(t)$ and $w(t)$, which we collect in the set denoted θ,

$$\mathbb{E}\left(c(t) \mid y(t)\right) = \frac{\int \mathbb{E}\left(c(t) \mid y(t), \theta\right) \pi(y(t)|\theta)\pi(\theta)d\theta}{\int \pi(y(t) \mid \theta)\pi(\theta)d\theta}. \tag{2.71}$$

Example 2.4.2 (Integrating out covariance parameters)**.** In this example we observe how covariance parameter uncertainty is manifested in our estimates for a climate trend given observations of a climate-plus-weather system.

Firstly, we write down the model for the output quantity y,

$$y(t) = c(t) \oplus w(t),$$

and specify Matérn autocovariance functions for climate and weather,

$$k_c(t) = \sigma_c^2 k_{Mat}(t; u_c, \nu_c),$$
$$k_w(t) = \sigma_w^2 k_{Mat}(t; u_w, \nu_w).$$

We also specify ‘true’ values for the covariance parameters,

$$\theta_1 = \sigma_c^2 = 1, \qquad \theta_2 = \sigma_w^2 = 0.5, \tag{2.72}$$

$$\theta_3 = u_c = 0.2, \qquad \theta_4 = u_w = 0.02, \tag{2.73}$$

$$\theta_5 = \nu_c = 2, \qquad \theta_6 = \nu_w = 2. \tag{2.74}$$

With these values we define variances for zero mean multivariate normal distributions, which we use to simulate climate and weather values at 200 equally spaced points in the interval $[-1, 1]$. The sums of the first 100 of these are used as the observed data on which the covariance parameters are conditioned.

We then pretend to doubt the true parameters, and model σ_c^2, σ_w^2, u_c and u_w as independent random variables. These quantities are measurable in the units of the output or the input, making them easier to comprehend than the parameters ν_c and ν_w, which we hold fixed. We use gamma distributions for all our priors, parameterising them in terms of their mean, which we set at the true values (2.72)-(2.74), and their standard deviation as a fraction of the mean. This latter statistic is equal to the shape parameter of a gamma distribution raised to the power minus two.

To explore the posterior for the covariance parameters given the observations, we make use of Scheidegger's RAM code. In fact, to improve the mixing of the algorithm we parameterise it in terms of variables θ', whose elements are a priori unit-normally distributed. We do this by using the transformation resulting from the application of an untransformed variables' prior cumulative distribution functions (CDF) and the inverse CDF of the unit normal, denoted $F_{\pi(\theta_i)}$ and F_N^{-1} respectively,

$$[\theta']_i = F_N^{-1}\left(F_{\pi([\theta]_i)}(\theta_i)\right).$$

The algorithm's random walk now jumps around in an unconstrained space in which all variables are on an equal scale with respect to changes in the prior density. Trace plots of the walk provide no evidence that the algorithm is malfunctioning but are otherwise unenlightening and so are not presented here.

From each member of the posterior sample of covariance parameters we calculate the linear estimate for $c(t)$ using the standard formula (2.1); these are values of the expectation appearing in integral (2.71). In figure 2.7 we show how these samples of smooths vary as the covariance parameter priors become more diffuse while maintaining their mean at the true parameter values. The diffuseness property is quantified by the fraction of the prior standard deviation and the prior mean, and is equal for each of the covariance parameters. We also add to the plots curves corresponding to two standard deviations, again conditional on the sampled values of the covariance parameters, for $c(t)$ above and below its expectation:

$$\mathbb{E}\left(c(t) \mid y(t), \theta^{(i)}\right) + h\mathrm{Var}\left(c(t) \mid y(t), \theta^{(i)}\right)^{1/2}, \qquad i = 1, \ldots, N_{it.}, h = -2, 0, 2, \qquad (2.75)$$

where i indexes the $N_{it.}$ samples from the posterior produced by the RAM algorithm.

From figure 2.7, we can see that, in this example, the expectation for the climate shows itself to be highly robust to uncertainty in the covariance parameters. This is mainly because the Matérn autocovariances all describe spaces of smooths that simply regress to the mean in a fairly uniform way. Of all the values of the smooths, those at the turning points are least robust. At these points the curve can follow a wiggle or smooth it over depending on the time scale parameter u_c. The standard deviation beyond the range of observations also exhibits considerable sensitivity, to the extent that the intervals between the curves marking two standard deviations for the climate mean vary by about fifty percent of the values calculated using the true parameters.

2.4.1.1 Simulator smooth

This type of smooth is really the focus of the next chapter but we present it here preemptively alongside the previous type of smooth, namely (2.71). In sections 2.1 and 2.2 we used roughness penalties and autocovariance functions to describe whole spaces of functions, but we now imagine a function, or a simulator, that describes a set of functions indexed by an input coordinate x:

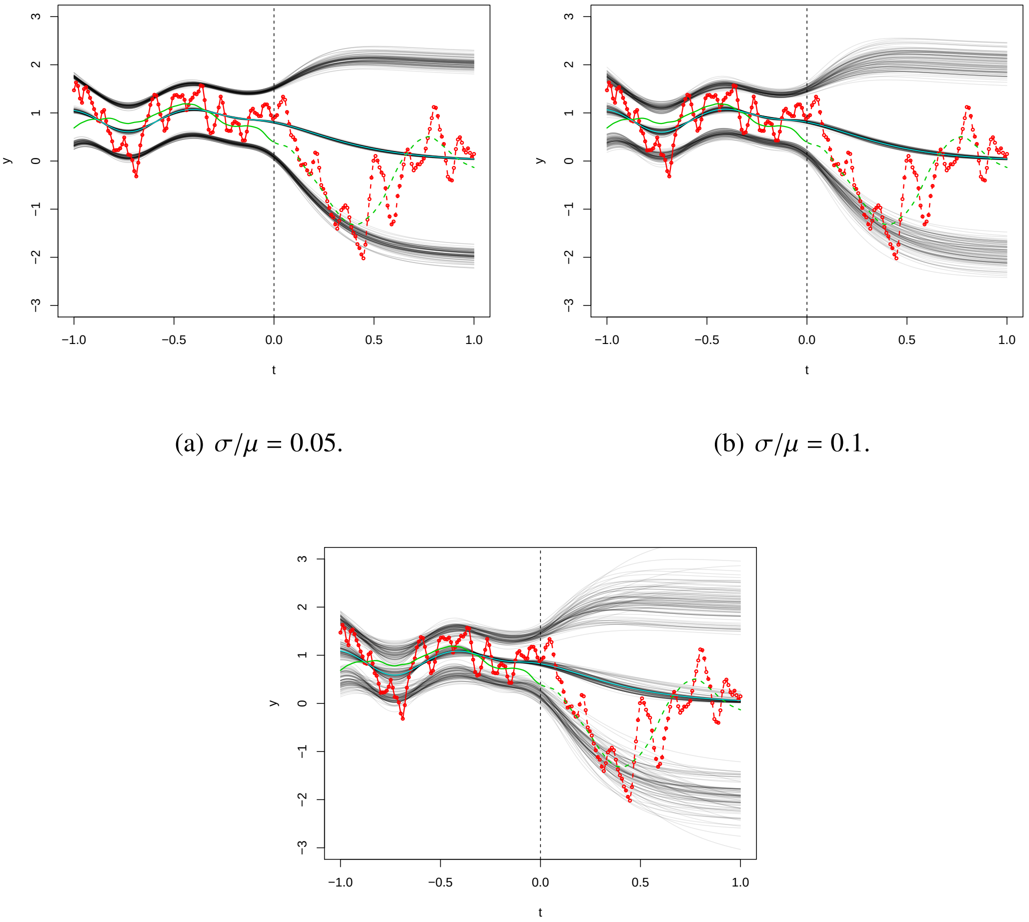
$$y(t, x) = c(t, x) + w(t, x).$$

Our smooth, which is an expectation or integral over functions is now parameterised as an expectation or integral over the input coordinate,

$$\mathbb{E}\left(c(t, x^*) \mid y(t, x^*)\right) = \frac{\int c(t, x)\pi(y(t, x^*) \mid c(t, x))\pi(x)dx}{\int \pi(y(t, x^*) \mid c(t, x))\pi(x)\,\mathrm{d}x}.$$

In practice, it is unusual to think that the simulator is actually simulating just the climate. Perhaps we are only able to approximate solutions to the simulator's equations for the climate, the climate trend we are interested in being obscured by artefacts of numerical approximation procedures or by weather mechanisms that are intentional components of the simulator code. In either case we can treat the simulator output as being only partially informative for the simulator climate surface, in which case $c(t, x)$ remains a random variable and must be integrated out in order to produce the simulator smooth:

$$\mathbb{E}\left(c(t, x^*) \mid y(t, x^*)\right) = \frac{\int\int c(t, x)\pi(y(t, x^*) \mid c(t, x))\pi(c(t, x) \mid x)\pi(x)\,\mathrm{d}c(t, x)\,\mathrm{d}x}{\int\int \pi(y(t, x^*) \mid c(t, x))\pi(c(t, x) \mid x)\pi(x)\,\mathrm{d}c(t, x)\,\mathrm{d}x}. \quad (2.76)$$



(a) $\sigma/\mu = 0.05$.

(b) $\sigma/\mu = 0.1$.

(c) $\sigma/\mu = 0.2$.

Figure 2.7: In this figure we plot quantities (2.75) with solid semi-transparent black lines. The left-hand side of each plot represents the interval on which we observe $y(t)$ at equally spaced intervals. These observations are marked with red dots. The green curve represents the true but unobserved climate trend. The extensions of both $y(t)$ and $c(t)$ into the interval without observations are drawn with dashed lines. The solid pale blue line shows the marginal smooth as described by (2.71).

Example 3.1.2 in the next chapter illustrates the calculation of the simulator smooth. We can see expressions (2.71) and (2.76) in parallel when we interpret the covariance parameters as describing penalties, and their specific values as corresponding to variants of physical laws. The key difference between the expressions is the way climate is effectively implicitly integrated out of the quantities in (2.71).

By noticing this parallel we can appreciate our inference for covariance parameters as signifying a first step towards calibrating a simulator for time series data.

2.5 Chapter summary

In this chapter we discussed the ideas behind particular choices for the covariance structure that defines a smooth, and some practical findings on how to perform the calculations they necessitate. Our first aim was to relate our smooth of some data to the physical or mechanistic properties of the system involved. By using FDA methodology we found that we could produce, up to a multiplicative constant, a covariance structure that describes a space of functions that almost satisfy a linear differential operator. Consequently, if we can elicit such an operator pertaining to a conserved quantity in the subsystem that determines the trend, $c(t)$, underlying the data, $y(t)$, we can incorporate this information into our estimate of $c(t)$. We also traced the link backwards, from a variance specification used to estimate a smooth, to a pseudo-physical quantity that has a stationary point at that estimate. Expression (2.44) gives such a quantity for a Matérn autocovariance function, but more generally, we used the link to identify the correspondence between autocovariances and penalties consisting of sums of squares of derivatives.

We also approached two computational issues for smoothing. Firstly, we saw that large arrays of data need not pose an insurmountable obstacle if we can apply a factorisable covariance structure to the data to be smoothed. In the next section, we looked at making inferences for smoothness parameters from data. The examples there showed that reliable inference is likely to be computationally demanding but that we can reduce the demand by utilising efficient inversion and integration algorithms.

Chapter 3

Relating systems and simulators

In the previous chapter we wrote about smoothing without taking great care to distinguish between real and simulated data. In this chapter we develop our intuition for reconciling the system and the simulator conceptually, and work towards a framework for relating them mathematically.

The emulator is the name given to a statistical model used to describe a distribution for a function's output. The function in question now is the simulator. Crucially the distribution encompasses outputs that have not been observed, making the emulator more than just a look-up table for previous simulations. Although they are just the evaluations of a deterministic function, the outputs are treated as random quantities because their calculation often incurs considerable cost and is therefore not performed. A substantial literature and research community has grown around the field of emulation as the capability, ambition and accessibility of complex simulators have increased. Notable examples of the application of emulators include [9], [4] and [21], which have been motivated by the oil, automotive and military industries respectively.

It is interesting to consider how this meta-modelling conflicts in principle with the sort of scepticism that would have us discard notions of our models being true or untrue. For in the emulation scenario there certainly are parameters that exist. There is a data-generating mechanism that can, in theory, be read precisely line by line, and we can know for certain that our emulator is structurally incorrect. It is also clear that our uncertainty for the simulator output is not really a product of intrinsic variability but of our own unwillingness or inability to pin down a value for it.

While emulators may be known principally for their role in reducing computation, they are also devices for structuring our thoughts concerning the relationship between physical systems and their simulators. Discussion and exposition of the conceptual role of the emulator is given in [15] and [16], and we will contribute to this discussion in section 3.1.

3.1 Relating a system and its simulator

Our focus here is the issue of simulator discrepancy. Our aim is to establish a statistical model, or range of models, with which we feel comfortable as expressions of our beliefs regarding the difference between the outputs of a physical system and those of its simulator.

We start by defining the simulator $y_{sim}(\cdot)$ as a function mapping a vector of inputs $\xi = (t, x) \in \Omega_t \times \Omega_x = \Omega_\xi$ to a real scalar output. If, in practice, our simulator takes a vector of input parameters and returns a vector of outputs in the form of a time series, we can think of $y_{sim}(\cdot)$ as the wrapper function that runs the simulator and returns only one value corresponding to a specific time. In this way, the extra level of detail involved in differentiating output coordinates and input parameters is suppressed so that we can focus on simulator discrepancy.

When the system is the planet, and the inputs represent physical constants, there is no sense in which it is a function that can be evaluated; it is more like a set of random variables waiting to be realised. But we find it useful to think of it as a function, and to think of the values the function takes as hypothetical alternatives to what truly occurs. In order to make sense of the relationship between the system and the simulator we then locate them in a common space, extending their input vector to include a new component, $v \in \Omega_v$, that could be thought of as an additional parameter in the simulator, or more generally as a new dimension in which the separation of two objects can be used to structure the dissimilarity between them. So we write

$$y \colon (\Omega_\xi \times \Omega_v) \mapsto \Re,$$

$$\begin{aligned} y_{sys}(\xi) &= y(\xi, v^*) && \text{for } \xi \in \Omega_\xi, \\ y_{sim}(\xi) &= y(\xi, \hat{v}) && \text{for } \xi \in \Omega_\xi. \end{aligned}$$

The interpretation is that Ω_ξ is a parameter space and Ω_v a discrepancy space, a notional continuum of simulators somewhere in which lies the system. The function y, taking arguments in the product space, reproduces the system or simulator functions at specific values of v, namely v^* and $\hat{v}$. We refer to the system variable we would go outside and measure at time t as $y_{sys}(\xi^*) = y_{sys}(t, x^*)$ and the vector x^* as the physical parameters in the real world we would measure if we could perform a complementary experiment to investigate them directly.

In our application of this discrepancy framework, the system climate function fulfils the role of Goldstein's reified model [16]: an estimator that exhausts our theoretical knowledge regarding the system output and from which further discrepancies, which we understand here as weather, are uncorrelated to every other modelled object.

If the system behaved as a collection of discrete interacting nodes and v were a measure of aggregation of the nodes, then v^* would represent the level of aggregation in the real world while $\hat{v}$ would represent the, probably higher, aggregation necessary for the simulator to run in reasonable time. In this scenario v is a parameter whose significance we can rationalise on a mechanistic level, but we also intend for v to describe simulator discrepancies whose roles are not traced back to specific mathematical contributions to the determination of y; these might be considered vague or external discrepancies. In this way the domain Ω_v can serve to separate systems or simulators on reasoned, or partially reasoned, grounds, or just on the basis of our reluctance to identify them with each other. Understanding the simulator discrepancy from the system in terms of a distance and treating the distance as unknown also allows us to combine calibration and validation procedures into a search through the extended parameter space.

A common choice for modelling covariances or correlations in such a product space, which we have already seen in section 2.3, is to take the product of covariances in each subspace,

$$\mathrm{Cov}\left(y_\Omega(\xi', v'),\ y_\Omega(\xi'', v'')\right) = k_\xi(\xi', \xi'')k_v(v', v''), \tag{3.1}$$

where $k_\xi(\cdot, \cdot)$ and $k_v(\cdot, \cdot)$ are positive definite functions on the products of each space with themselves. The consequence of taking a covariance function of this form is that for fixed v^* and $\hat{v}$ the covariance between the system value $y(\xi^*, v^*)$ and an arbitrary simulation value $y(\xi', \hat{v})$ is, as a function of ξ', proportional to the covariance between that simulation value and simulation $y(\xi^*, \hat{v})$,

$$\begin{aligned} \mathrm{Cov}\left(y_{sys}(\xi^*)\ ,\ y_{sim}(\xi')\right) &= \mathrm{Cov}\left(y_\Omega(\xi^*, v^*)\ ,\ y_\Omega(\xi', \hat{v})\right) = k_\xi(\xi^*, \xi')k_v(v^*, \hat{v}), \\ \mathrm{Cov}\left(y_{sim}(\xi^*)\ ,\ y_{sim}(\xi')\right) &= \mathrm{Cov}\left(y_\Omega(\xi^*, \hat{v})\ ,\ y_\Omega(\xi', \hat{v})\right) = k_\xi(\xi^*, \xi')k_v(\hat{v}, \hat{v}). \end{aligned}$$

The same sort of result arises when we adopt a model that describes the system output as a sum of the simulator output and an additive simulator discrepancy e_{sim}:

$$y_{sys}(\xi^*) = y_{sim}(\xi^*) \oplus e_{sim}.$$

Assuming e_{sim} is also independent from any other simulator value, the same covariances are proportional because they are equal,

$$\mathrm{Cov}\left(y_{sys}(\xi^*)\ ,\ y_{sim}(\xi')\right) = \mathrm{Cov}\left(y_{sim}(\xi^*) \oplus e_{sim}\ ,\ y_{sim}(\xi')\right) = \mathrm{Cov}\left(y_{sim}(\xi^*)\ ,\ y_{sim}(\xi')\right).$$

The significance of these proportionality results lies in the implication that $y_{sim}(\xi^*)$, the value returned by the simulator with the 'true' inputs, separates $y_{sys}(\xi^*)$ from all the other simulations, which we write as

$$\lfloor y_{sys}(\xi^*) \perp\!\!\!\perp \{y_{sim}(\xi')\} \rfloor \mid y_{sim}(\xi^*), \tag{3.2}$$

in the sense that $y_{sim}(\xi^*)$ is Bayes linear sufficient for all the possible simulator outputs for adjusting our beliefs about the system output. That is, for all ξ',

$$\mathbb{E}_{y_{sim}(\xi^*)}\left(y_{sys}(\xi^*)\right) = \mathbb{E}_{y_{sim}(\xi^*)\cup\{y_{sim}(\xi)\}}\left(y_{sys}(\xi^*)\right). \tag{3.3}$$

This assertion is formalized and proven in theorem 3.1.1, while further explanation and discussion of these definitions can be found in [17].

Theorem 3.1.1. *For vector random variables B and D, with finite positive definite variance matrices, the subset of the first n elements of D, denoted D_1, is Bayes linear sufficient for all of D if and only if the rows of the covariance matrix* $\mathrm{Cov}(B\ ,\ D_1)$ *lie within the space spanned by the first n rows of* $\mathrm{Var}(D)$.

Proof. When the covariance is spanned by the first rows of the data variance matrix we can extract $\mathrm{Cov}(B\,,\ D)$ from $\mathrm{Var}(D)$ with the matrix $(\mathbf{A}, \mathbf{0})$:

$$\mathrm{Cov}(B\,,\ D) = \begin{pmatrix}\mathbf{A} & \mathbf{0}\end{pmatrix} \mathrm{Var}(D)\,.$$

Here, $\mathbf{A}^{-1}\mathrm{Cov}(B\,,\ D)$ is the matrix formed by taking the first n rows of $\mathrm{Var}(D)$, but the ability to rearrange the elements of D means that there is no loss of generality. Then, so long as $\mathrm{Var}(D)$ is invertible,

$$\begin{aligned}\mathbb{E}(B) + \mathrm{Cov}(B\,,\ D)\,\mathrm{Var}(D)^{-1}\,[D - \mathbb{E}(D)] =& \mathbb{E}(B) + \begin{pmatrix}\mathbf{A} & \mathbf{0}\end{pmatrix}[D - \mathbb{E}(D)],\\ =& \mathbb{E}(B) + \mathbf{A}[D_1 - \mathbb{E}(D_1)],\end{aligned}$$

showing that only the elements of D_1 contribute to the adjustment. Going in the opposite direction, if D_1 is sufficient for D we must have

$$\mathbb{E}_{D_1}(B) = \mathbb{E}_{D_1 \cup D_2}(B)\,, \tag{3.4}$$

for all possible realisations of D_2. Using the shorthand

$$\mathrm{Cov}\left(B\,,\ (D_1^T, D_2^T)^T\right) = \begin{pmatrix}\mathbf{C}_{11} & \mathbf{C}_{12}\end{pmatrix}, \qquad \mathrm{Var}\left((D_1^T, D_2^T)^T\right) = \begin{pmatrix}\mathbf{V}_{11} & \mathbf{V}_{12}\\ \mathbf{V}_{21} & \mathbf{V}_{22}\end{pmatrix},$$

and defining

$$\mathbf{A} = \mathbf{C}_{11}\mathbf{V}_{11}^{-1},$$

and by using theorem B.0.11 for the inverse of a block-partitioned matrix, the expressions in (3.4) for the adjusted expectation for B are written

$$\mathbb{E}_{D_1}(B) = \mathbf{A}[D_1 - \mathbb{E}(D_1)], \tag{3.5}$$

$$\mathbb{E}_{D_1 \cup D_2}(B) = (\mathbf{A}\mathbf{V}_{11} - \mathbf{C}_{12}\mathbf{V}_{22}^{-1}\mathbf{V}_{21})(\mathbf{V}_{11} - \mathbf{V}_{12}\mathbf{V}_{22}^{-1}\mathbf{V}_{21})^{-1}[D_1 - \mathbb{E}(D_1)] \tag{3.6}$$

$$+ (\mathbf{C}_{12} - \mathbf{A}\mathbf{V}_{12})(\mathbf{V}_{22} - \mathbf{V}_{21}\mathbf{V}_{11}^{-1}\mathbf{V}_{12})^{-1}[D_2 - \mathbb{E}(D_2)]. \tag{3.7}$$

The middle factor in line (3.7) is the adjusted precision for D_2 given D_1; it cannot be zero given our assumptions for D, meaning that the left-hand factor must be zero and

$$\mathbf{C}_{12} = \mathbf{A}V_{12}.$$

Since we also have $\mathbf{C}_{11} = \mathbf{A}\mathbf{V}_{11}$, by definition, we can write

$$\begin{pmatrix}\mathbf{C}_{11} & \mathbf{C}_{12}\end{pmatrix} = \mathbf{A}\begin{pmatrix}\mathbf{V}_{11} & \mathbf{V}_{12}\end{pmatrix},$$

which shows that $\mathrm{Cov}\,(B\ ,\ D)$ must be a linear combination of the first n rows of $\mathrm{Var}\,(D)$.

□

The same phenomenon is apparent if we consider a stationary process in one dimension with an exponentially decaying autocovariance function. The consequence here is that observations of the process at points on either side of a third point are Bayes linear sufficient for all values of the field that are further away, for adjusting the expectation of the third point. It is easiest to see the proportionality condition here when we look at only observations lying to one side of the location of interest, which we label x_0:

$$\begin{aligned}&\text{for } x_0 < x_1 < x_2 < \ldots,\\ \mathrm{Cov}\,(y(x_0)\ ,\ y(x_i)) \propto \exp(-a|x_0 - x_i|) &= \exp(-a|x_0 - x_1|)\exp(-a|x_1 - x_i|),\\ &= \mathrm{Cov}\,(y(x_0)\ ,\ y(x_1))\,\mathrm{Cov}\,(y(x_1)\ ,\ y(x_i))\,.\end{aligned}$$

Consequently,

$$\lfloor y(x_0) \perp\!\!\!\perp \{y(x_i); i = 2, \ldots\}\rfloor \mid y(x_1).$$

The two-sided result comes from realising that the covariance vector in the adjustment formulae is a linear combination of the two rows of the variance matrix corresponding to the nearest observations.

In summary, we must learn to associate factorisable variance structures with linear sufficiency properties. In particular, adopting a model of the form (3.1) for the relationship between the system and the simulator means adopting the belief that a single simulation, $y_{sim}(\xi^*)$, is linearly sufficient for all other simulations. Such a model may be criticised on the grounds that, in practice, we would be uncomfortable with the idea that a single simulation could render all others redundant, at least for the purpose of constructing linear estimates. While x^* is unknown this ought not to be a problem that confronts us because the single sufficient simulation cannot be identified. This is likely to be the case when we want to calibrate climate simulators that require the specification of a large array of initial conditions corresponding to historical data that is not available. The potential sufficiency may become a problem however, if x^* represents a physical quantity that could become known through other means. Our model would then suggest that if we pin down the value we would only need to run the simulator once to learn everything we can about the system.

Another implication of the factorisable structure is that the system and simulator outputs are equally closely correlated everywhere in the Ω_ξ parameter space. We can see this assumption being questioned, for example, if the stability of a simulator's solution method depends on its inputs.

A simple way to compromise the factorisation property, removing the sufficiency phenomenon, is to decompose the function y_Ω into components,

$$y(\xi, v) = c(\xi, v) + w(\xi, v).$$

If we deem the components independent of each other we have

$$\mathrm{Cov}\left(y_{sys}(\xi^*)\ ,\ y_{sim}(\xi')\right) = k_{c\xi}(\xi^*, \xi')k_{cv}(v^*, \hat{v}) + k_{w\xi}(\xi^*, \xi')k_{wv}(v^*, \hat{v}), \tag{3.8}$$

$$\mathrm{Cov}\left(y_{sim}(\xi^*)\ ,\ y_{sim}(\xi')\right) = k_{c\xi}(\xi^*, \xi')k_{cv}(\hat{v}, \hat{v}) + k_{w\xi}(\xi^*, \xi')k_{wv}(\hat{v}, \hat{v}). \tag{3.9}$$

Unless,

$$\frac{k_{cv}(v^*, \hat{v})}{k_{cv}(\hat{v}, \hat{v})} = \frac{k_{wv}(v^*, \hat{v})}{k_{wv}(\hat{v}, \hat{v})}, \tag{3.10}$$

the covariances (3.8) and (3.9) are not proportional functions of ξ'. If we use this model to represent processes corresponding to climate and weather and we judge that the system and simulator weather components are not as strongly correlated as the system and simulator climate components, so that (3.10) does not hold, then there ceases to be one simulation that is Bayes linear sufficient for all the rest. Now the value of finding x^* is altered in that it would no longer serve to unlock all the information for the system that the simulator can provide, and our attention shifts from specific values of the simulator output to its variation over the input space.

Example 3.1.2 (Calibration from a smooth trend)**.** This example provides a context in which to explore some of the ideas raised in the preceding sections; it continues directly from example 2.3.2. There, we looked at the linearly adjusted expectation and variance for a function $c(t, x)$ over the product of input and time spaces. Now we will extend the space by taking its product with the discrepancy variable v. Across this space we define the covariance functions,

$$\mathrm{Cov}\left(c(t',x',v')\ ,\ c(t'',x'',v')\right) = k_{ct}(t',t'')k_{cx}(x',x'')k_{cv}(v',v''),$$
$$\mathrm{Cov}\left(w(t',x',v')\ ,\ w(t'',x'',v'')\right) = k_{wt}(t',t'')k_{wx}(x',x'')k_{wv}(v',v'').$$

Since, for now, we are considering the case in which there is only one simulator and one system, the covariance contributions from separations in v space amount to single numbers, which we denote ρ_c and ρ_w. The other notation and parameter values carry over from example 2.3.2 so that

$$\rho_c = k_{cv}(v^*,\hat{v}), \qquad \rho_w = k_{wv}(v^*,\hat{v}). \tag{3.11}$$
$$[\mathbf{a}_x]_i = k_{cx}(x,[\mathbf{X}]_{i,\cdot}), \qquad [\mathbf{b}_x]_i = k_{wx}(x,[\mathbf{X}]_{i,\cdot}), \tag{3.12}$$
$$[\mathbf{A}_x]_{i,j} = k_{cx}([\mathbf{X}]_{i,\cdot},[\mathbf{X}]_{j,\cdot}), \qquad [\mathbf{B}_x]_{i,j} = k_{wx}([\mathbf{X}]_{i,\cdot},[\mathbf{X}]_{j,\cdot}), \tag{3.13}$$
$$[\mathbf{A}_t]_{i,j} = k_{ct}(t_i,t_j), \qquad [\mathbf{B}_t]_{i,j} = k_{wt}(t_i,t_j). \tag{3.14}$$

The simulator values and covariance parameters for the input and time directions are the same as before, and we specify $\rho_c = 0.95$, $\rho_w = 0.5$ in expression of the notion that the system and simulator climate functions are strongly correlated while their high-frequency weather functions are less strongly correlated.

We can now write the joint variance of the simulated data $\mathbf{Y}$, and the covariance between the system observations $y_{sys}(\mathbf{T},x^*)$ and the simulations as

$$\mathrm{Var}\left(\mathrm{vec}\left(\mathbf{Y}\right)\right) = (\mathbf{A}_t \otimes \mathbf{A}_x) + (\mathbf{B}_t \otimes \mathbf{B}_x), \tag{3.15}$$
$$\mathrm{Cov}\left(y_{sys}(\mathbf{T},x^*)\ ,\ \mathrm{vec}\left(\mathbf{Y}\right)\right) = \rho_c\mathbf{A}_t \otimes \mathbf{a}_{x^*} + \rho_w\mathbf{B}_x \otimes \mathbf{b}_{x^*}. \tag{3.16}$$

Next, we take the pragmatic step of discretising the set of admissible inputs into one hundred equally spaced points over the interval $[-1,1]$. From the unweighted points we sample an x^*, and from this input value, the moments (3.15) and (3.16), and the matrix of simulator values $\mathbf{Y}$, we calculate a conditional mean and variance for $y_{sys}(\mathbf{T},x^*)$. The value of $y_{sys}(\mathbf{T},x^*)$ is produced by sampling from the multivariate normal distribution with these moments. We then partition the synthetic system data into past and future components $y_{sys}(\mathbf{T},x^*) = (y_{sys}(\mathbf{T}_p,x^*), y_{sys}(\mathbf{T}_f,x^*))$, and treat only the past component as having been observed.

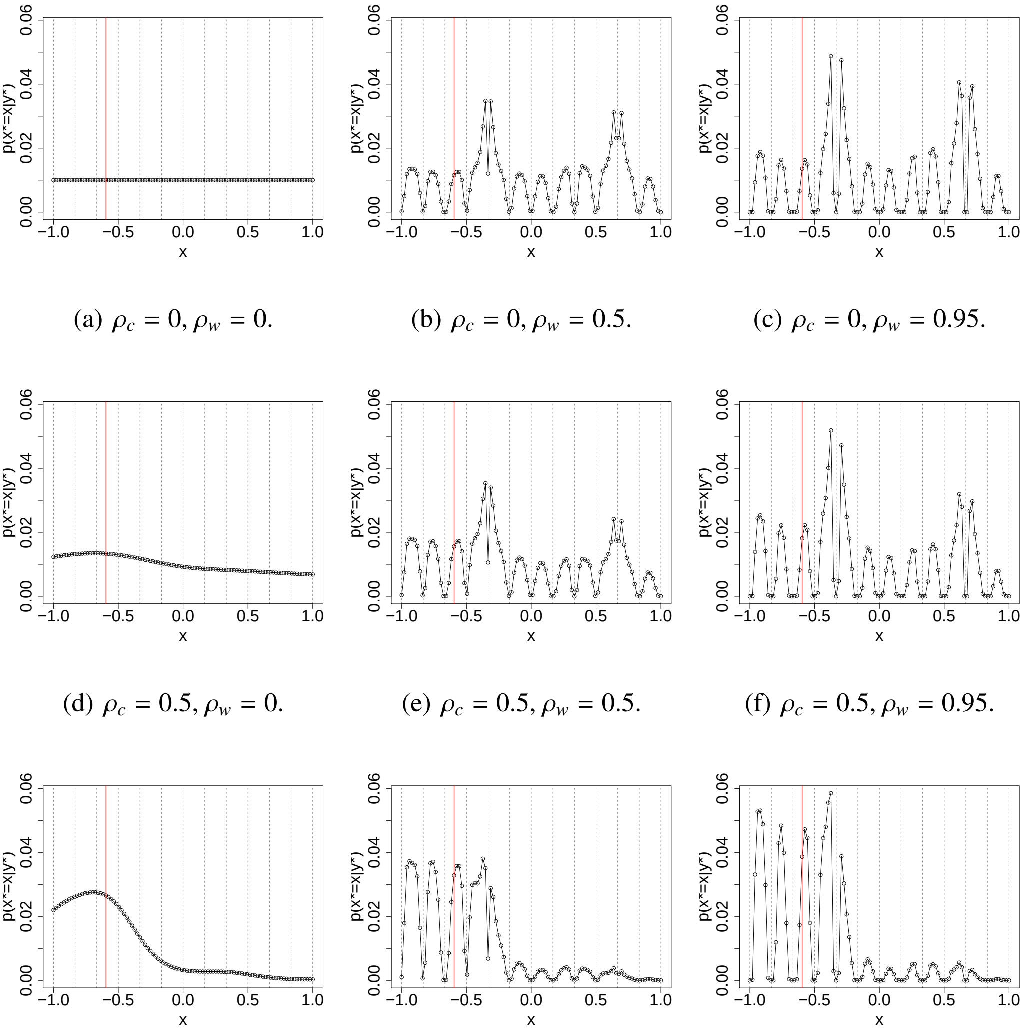
We proceed to make the same calculations for the moments at the other 99 points of the discretised input space, and from each of these we calculate the multivariate normal density at $y_{sys}(\mathbf{T}_p, x^*)$,

$$\pi_{MVN}(y_{sys}(\mathbf{T}_p, x^*) \mid x_i = x^*, \mathbf{Y}, \rho_c, \rho_w), \qquad\qquad i = 1, \ldots, 100. \qquad (3.17)$$

Again, the efficiencies afforded by the factorisable structure described in section 2.3 prove valuable for manipulating arrays so as to calculate means, variances and likelihoods quickly. The results of the calculation of (3.17) are partially illustrated in figure 3.1, where we plot the normalised densities, which are equivalent to the posterior probabilities for x^* if we were to specify equal prior probabilities for each of the 100 points in the discretised space. The subfigure in the middle of the bottom row shows the ‘true’ posterior in the sense that it is calculated using the covariance parameters used to simulate the system and simulator data. The other subfigures are calculated using the wrong ρ_c and ρ_w values so that we may also observe the effect of incorrectly anticipating the similarity between the system and the simulator.

The right-most six plots all feature inverted (downwards) spikes at the input values for the simulator training data. These reflect the fact that none of the weather signals from the performed simulations matches up with the weather signal of the system data. Although we can see here that the likelihood is generally quite evenly spread over the x domain or heaped up at the negative end, these spikes have the potential to mislead the numerical optimisation or integration routines that will guide us when such plots are not practical, such as when the input dimension increases for example. The spikiness problem is avoided if we either calibrate on the basis of the climate signal alone, pretending ρ_w is zero, or if we smooth the likelihood as a post-processing step. The implication is that even if our simulator is deemed capable of reproducing the weather signal that is local both in the time and input directions, it may not be a good idea to take that capability into account when we are calibrating it, because the extra information may trick or stall our inferential calculations. Additionally, if our objective is to make statements about the first few posterior moments of x^* then these spikes are likely to have only a minor influence.

It is also interesting to investigate the extent to which ρ_c and ρ_w are inferable parameters here, because this inference could serve as a validation procedure for the simulator. In table 3.1 we present the normalised sums of the likelihoods (3.17), over the one hundred



(a) $\rho_c = 0, \rho_w = 0$.

(b) $\rho_c = 0, \rho_w = 0.5$.

(c) $\rho_c = 0, \rho_w = 0.95$.

(d) $\rho_c = 0.5, \rho_w = 0$.

(e) $\rho_c = 0.5, \rho_w = 0.5$.

(f) $\rho_c = 0.5, \rho_w = 0.95$.

(g) $\rho_c = 0.95, \rho_w = 0$.

(h) $\rho_c = 0.95, \rho_w = 0.5$.

(i) $\rho_c = 0.95, \rho_w = 0.95$.

Figure 3.1: Interpolations of the posterior for x^* calculated under a selection of choices for the strength of correlation between the system and the simulator. The vertical dashed lines indicate the input values for the simulator time series, while the vertical solid line shows the location of the true x^* value. The solid black curves interpolating the posterior values are plotted only as visual aids. Subfigure 3.1(h) contains the plot produced using the true values for (ρ_c, ρ_w).

candidate input locations, corresponding to the plots in figure 3.1. The precise form of the normalisation for calculating the table's entries is given by (3.18). Again, we identify these quantities with the posterior probabilities that would arise from discretising the space of possible (ρ_c, ρ_w) parameters and allocating them equal prior probabilities. The larger figures in the bottom row of the table indicate that identification of ρ_c is possible, although the same statement cannot be made for ρ_w. In fact the table suggests a higher likelihood of ρ_w being zero than its true value.

$$\pi(y_{sys}(\mathbf{T}_p, x^*) \mid \mathbf{Y}, \rho_c, \rho_w) = \frac{\sum_{i=1}^{100} \pi(y_{sys,p} \mid x_i = x^*, \mathbf{Y}, \rho_c, \rho_w)}{\sum_{\rho_c} \sum_{\rho_w} \sum_{i=1}^{100} \pi(y_{sys} \mid x_i = x^*, \mathbf{Y}, \rho_c, \rho_w)}. \tag{3.18}$$

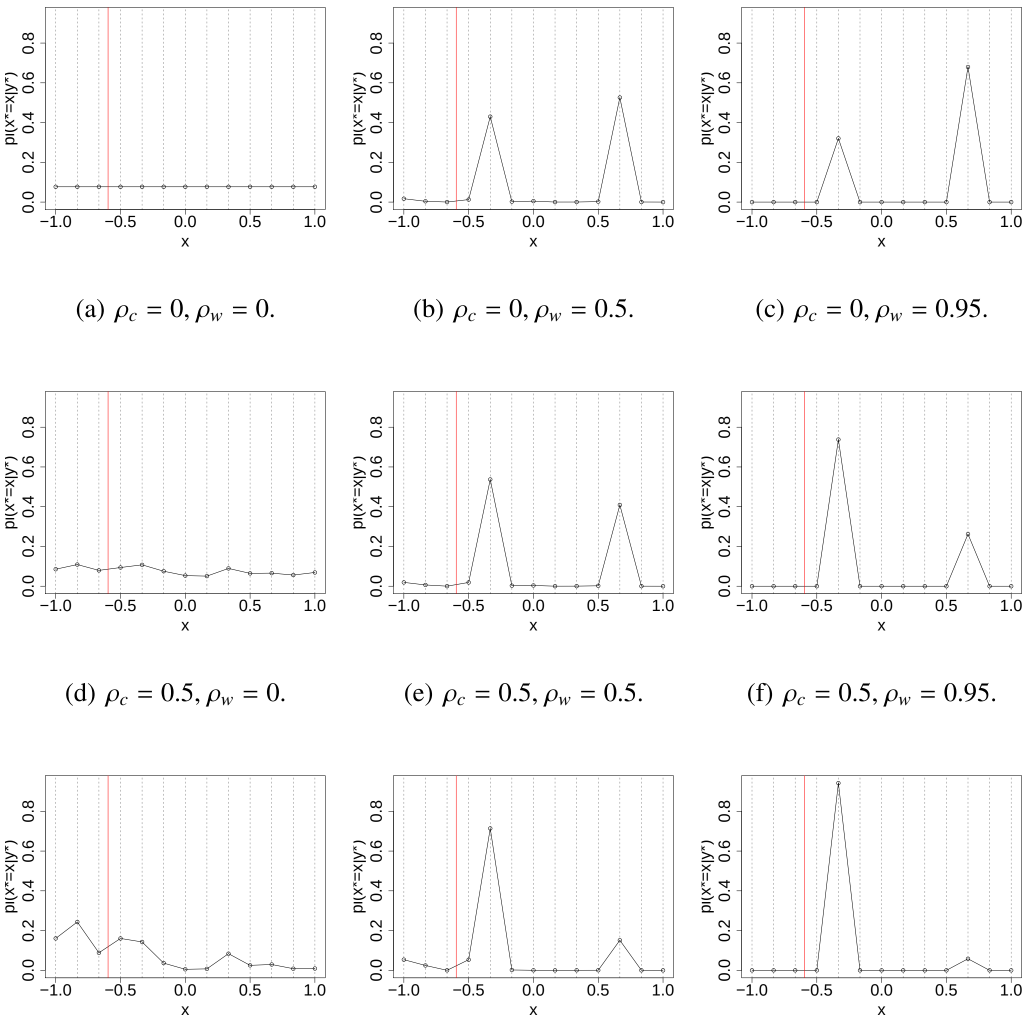
	$\rho_w = 0$	$\rho_w = 0.5$	$\rho_w = 0.95$
$\rho_c = 0$	0.007	0.005	0.004
$\rho_c = 0.5$	0.031	0.023	0.016
$\rho_c = 0.95$	0.417	0.291	0.206

Table 3.1: The normalised sums of likelihoods for each (ρ_c, ρ_w) combination in the same configuration as the plots of figure 3.1.

We now compare the likelihood calculations of figure 3.1 with those arising from the consideration of just one simulated time series at a time. The densities

$$\pi(y_{sys}(\mathbf{T}_p, x^*) \mid [\mathbf{X}]_{i,\cdot} = x^*, [\mathbf{Y}]_{i,\cdot}, \rho_c, \rho_w)$$

are plotted in figure 3.2 with the correct specification of ρ_c and ρ_w along with a range of incorrect specifications, in the same configuration as figure 3.1. The calculation of these densities is still much faster than those that account for all the simulations, despite our algebraic tricks for efficiently calculating the moments of multiple time series. The great danger of inferring the location of x^* with these likelihoods is illustrated most clearly by the subfigures in 3.2 on and above the main diagonal. In these cases ρ_w is large so we expect that there is an input value at which the simulator weather signal mostly coincides with the system weather. When this signal is not found and all the simulations available are in the tail of the likelihood , the least bad simulation accumulates the majority of the likelihood, giving the impression that x^* has been identified when really it has not.



(a) $\rho_c = 0, \rho_w = 0.$

(b) $\rho_c = 0, \rho_w = 0.5.$

(c) $\rho_c = 0, \rho_w = 0.95.$

(d) $\rho_c = 0.5, \rho_w = 0.$

(e) $\rho_c = 0.5, \rho_w = 0.5.$

(f) $\rho_c = 0.5, \rho_w = 0.95.$

(g) $\rho_c = 0.95, \rho_w = 0.$

(h) $\rho_c = 0.95, \rho_w = 0.5.$

(i) $\rho_c = 0.95, \rho_w = 0.95.$

Figure 3.2: The likelihood for the system time series calculated by smoothing each simulated time series individually. The solid black interpolating lines are solely to guide the eye, since the likelihood is defined only at the input coordinates at which the simulator was run. The solid vertical line marks the location of x^*. Subfigure 3.2(h) contains the plot produced using the true (ρ_c, ρ_w) values.

In table 3.2 we present the normalised sums of likelihoods equivalent to (3.1), for the case in which we smooth one simulation at a time and restrict our attention to only the simulations in the training set. These are the values

$$\hat{\pi}(y_{sys}(\mathbf{T}_p, x^*) \mid \mathbf{Y}, \rho_c, \rho_w) = \frac{\sum_{i=1}^{13} \pi(y_{sys,p} \mid x_i = x^*, y_{sim}([\mathbf{X}]_{i,\cdot}), \rho_c, \rho_w)}{\sum_{\rho_c} \sum_{\rho_w} \sum_{i=1}^{100} \pi(y_{sys} \mid x_i = x^*, y_{sim}([\mathbf{X}]_{i,\cdot}), \rho_c, \rho_w)}. \tag{3.19}$$

	$\rho_w = 0$	$\rho_w = 0.5$	$\rho_w = 0.95$
$\rho_c = 0$	0.039	0.009	0.000
$\rho_c = 0.5$	0.109	0.020	0.000
$\rho_c = 0.95$	0.738	0.086	0.000

Table 3.2: The approximate normalised sums of likelihoods, calculated using (3.19), corresponding to the plots of figure 3.2.

As might be anticipated, since we consider only a small subset of all possible simulations, none of which show weather behaviour matching the system weather, we gather almost no evidence to support the possibility that ρ_w could be non-zero. Although incorrect, these figures serve to steer us away from the right-most posteriors of figure 3.2 that would erroneously identify x^*.

The conclusions we have made here mostly conform to common sense. Our inferences for ρ_c and ρ_w are strongly dependent on whether we use the emulator to structure the information available from the simulations. Misjudging the fidelity of the simulator to the system can lead to misidentification of x^*; this is a problem that is greatly exacerbated when we restrict our attention to a small subset of the simulator domain, namely the locations of observed simulations.

In the last calculation before leaving the example we compute what we referred to earlier, in section 2.4.1.1, as the simulator smooth. To do this, we calculate the joint expectation, variance and covariance for the system output and system climate given the simulated series conditional on the true values for (ρ_c, ρ_w) and $x_i = x^*$ for $i = 1, \ldots, 100$,

$$\mathbb{E}\left(c_{sys}(\mathbf{T}, x^*) \mid \mathbf{Y}\right) = \mathrm{Cov}\left(c_{sys}(\mathbf{T}, x^*)\,,\ \mathrm{vec}\,(\mathbf{Y})\right) \mathrm{Var}\,(\mathrm{vec}\,(\mathbf{Y}))^{-1}\,\mathrm{vec}\,(\mathbf{Y})\,, \tag{3.20}$$

$$\mathbb{E}\left(y_{sys}(\mathbf{T}, x^*) \mid \mathbf{Y}\right) = \mathrm{Cov}\left(y_{sys}(\mathbf{T}, x^*)\,,\ \mathrm{vec}\,(\mathbf{Y})\right) \mathrm{Var}\,(\mathrm{vec}\,(\mathbf{Y}))^{-1}\,\mathrm{vec}\,(\mathbf{Y})\,, \tag{3.21}$$

$$\begin{aligned}\mathrm{Var}\left(c_{sys}(\mathbf{T},x^*)\mid\mathbf{Y}\right)&=\mathrm{Var}\left(c_{sys}(\mathbf{T},x^*)\right)\\&\quad-\mathrm{Cov}\left(c_{sys}(\mathbf{T},x^*)\ ,\ \mathrm{vec}\,(\mathbf{Y})\right)\mathrm{Var}\,(\mathrm{vec}\,(\mathbf{Y}))^{-1}\,\mathrm{Cov}\left(c_{sys}(\mathbf{T},x^*)\ ,\ \mathrm{vec}\,(\mathbf{Y})\right)^T,\end{aligned}\tag{3.22}$$

$$\begin{aligned}\mathrm{Var}\left(y_{sys}(\mathbf{T},x^*)\mid\mathbf{Y}\right)&=\mathrm{Var}\left(y_{sys}(\mathbf{T},x^*)\right)\\&\quad-\mathrm{Cov}\left(y_{sys}(\mathbf{T},x^*)\ ,\ \mathrm{vec}\,(\mathbf{Y})\right)\mathrm{Var}\,(\mathrm{vec}\,(\mathbf{Y}))^{-1}\,\mathrm{Cov}\left(y_{sys}(\mathbf{T},x^*)\ ,\ \mathrm{vec}\,(\mathbf{Y})\right)^T,\end{aligned}\tag{3.23}$$

$$\begin{aligned}\mathrm{Cov}\left(c_{sys}(\mathbf{T},x^*)\ ,\ y_{sys}(\mathbf{T},x^*)\mid\mathbf{Y}\right)&=\mathrm{Cov}\left(c_{sys}(\mathbf{T},x^*)\ ,\ y_{sys}(\mathbf{T},x^*)\right)\\&\quad-\mathrm{Cov}\left(c_{sys}(\mathbf{T},x^*)\ ,\ \mathrm{vec}\,(\mathbf{Y})\right)\mathrm{Var}\,(\mathrm{vec}\,(\mathbf{Y}))^{-1}\,\mathrm{Cov}\left(y_{sys}(\mathbf{T},x^*)\ ,\ \mathrm{vec}\,(\mathbf{Y})\right)^T,\end{aligned}\tag{3.24}$$

where the quantities appearing in the expressions above are constructed with the basic quantities (3.11)-(3.14),

$$\begin{aligned}\mathrm{Var}\left(c_{sys}(\mathbf{T},x^*)\right)&=k_{cx}(x^*,x^*)\mathbf{A}_t,\\\mathrm{Var}\left(y_{sys}(\mathbf{T},x^*)\right)&=k_{cx}(x^*,x^*)\mathbf{A}_t+k_{wx}(x',x')\mathbf{B}_t,\\\mathrm{Cov}\left(c_{sys}(\mathbf{T},x^*)\ ,\ \mathrm{vec}\,(\mathbf{Y})\right)&=\rho_c\mathbf{A}_t\otimes\mathbf{a}_{x'},\\\mathrm{Cov}\left(y_{sys}(\mathbf{T},x^*)\ ,\ \mathrm{vec}\,(\mathbf{Y})\right)&=\rho_c\mathbf{A}_t\otimes\mathbf{a}_{x'}+\rho_w\mathbf{B}_t\otimes\mathbf{b}_{x'},\\\mathrm{Cov}\left(c_{sys}(\mathbf{T},x^*)\ ,\ y_{sys}(\mathbf{T},x^*)\right)&=k_{cx}(x',x')\mathbf{A}_t,\\\mathrm{Var}\,(\mathrm{vec}\,(\mathbf{Y}))&=(\mathbf{A}_t\otimes\mathbf{A}_x)+(\mathbf{B}_t\otimes\mathbf{B}_x).\end{aligned}$$

Now, with (3.20)-(3.24) we calculate expectations for $c_{sys}(\mathbf{T},x^*)$ given the observed system output values $y_{sys}(\mathbf{T},x^*)$,

$$\begin{aligned}&\mathbb{E}\left(c_{sys}(\mathbf{T},x^*)\mid\mathbf{Y}\right)+\\&\mathrm{Cov}\left(c_{sys}(\mathbf{T},x^*)\ ,\ y_{sys}(\mathbf{T},x^*)\mid\mathbf{Y}\right)\mathrm{Var}\left(y_{sys}(\mathbf{T},x^*)\mid\mathbf{Y}\right)^{-1}\left(y_{sys}(\mathbf{T},x^*)-\mathbb{E}\left(y_{sys}(\mathbf{T},x^*)\mid\mathbf{Y}\right)\right).\end{aligned}\tag{3.25}$$

The simulator smooth is then approximated as the average of the 100 conditional expectations (3.25) for which a different point in the discretised input space is identified with x^*. The weights for the average are given by (3.17). In figure 3.3 we plot the simulator smooth as well as the summands from which it is calculated, and the smooth of the system data without any information from the simulations at all. The two smooths are very similar within the range of the observed system data but significantly different beyond it. This observation reminds us that the simulator smooth is not really a tool for interpolation, because the same system data we use to calibrate the simulator renders the simulated past climates all but redundant for informing the smooth there. But, by identifying likely values for x^*, the corresponding values of $y_{sys}(\mathbf{T}_f, x^*)$ are highly informative for the future system values while the past system values alone are not.

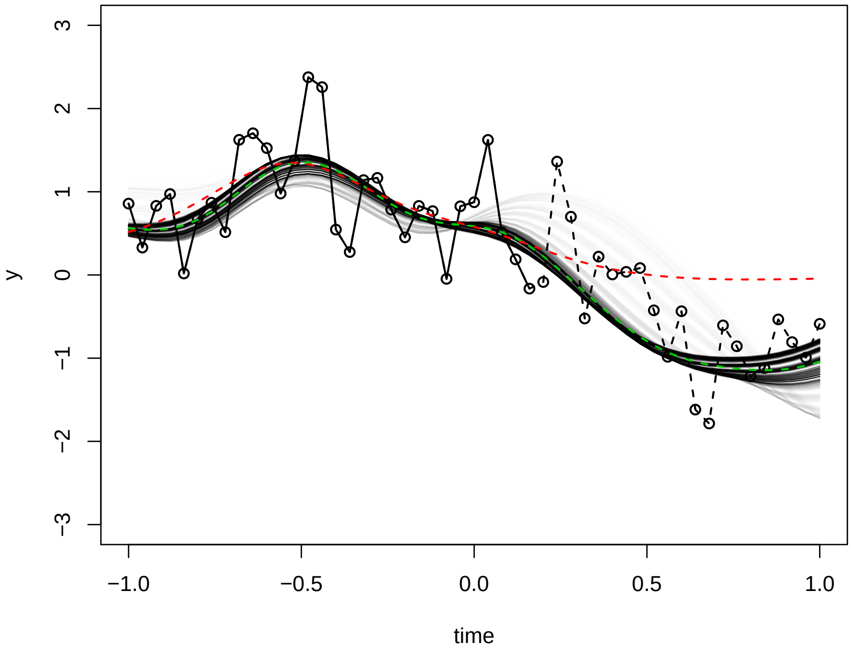


Figure 3.3: The jagged solid black line in this plot interpolates the observed system data that are used in our calibration calculations; the dashed black interpolates the system data that was held back. The red dashed line shows the expectation for the system climate given only the observed system values and no simulated data; the green dashed line shows the expectation for the system climate given the observed system values and all the simulated data. The green dashed line is computed as an average whose summands are plotted as the smooth black lines; the opacity of these lines is proportional to their weight in that average.

3.2 Chapter summary

We wish to argue that focusing attention on the smooth climate term is a good idea for three reasons:

1. Firstly, on a practical level, it allows us to form an intelligible image of the simulator's output surface, and to navigate it, without getting caught up on small local features.

2. Secondly, it represents the, usually appropriate, admission that the simulator is significantly discrepant from the system on the scale of high-frequency variation.

3. Thirdly, it allows us to rationalise the exploration of the simulator's variation across its input space rather than concentrating on its value at a single sufficient input.

The first two reasons reflect our inability to find the correct input and the simulator's inability to reproduce the observed data. But when the largest modes of variation are also the smoothest and we prioritise them to the extent that rough modes are considered noise, our algorithms are less likely to stall, obsessing over local optima in the likelihood, and the same is true of us as scientists.

These considerations are relevant to ABC methods, on which much interesting research, such as [11] and [45], is currently focused. These methods derive likelihood functions from a selection of summary statistics, and in doing so discard much of the information from the observed data. Their application is most often justified by the first argument: on the infeasibility of utilising the full likelihood. But we suspect that in most cases the second argument also plays a significant role. For example, Wood's recent work on synthetic likelihood [60], applied to the Ricker model for population dynamics, is introduced as a way to negotiate a pathologically rough likelihood surface, but when a real data set is introduced, numerical approximations of the likelihood derived from filtering techniques appear to be very low, suggesting that the model is not fit to explain single time-step changes. However, by fitting it to 'dynamically important' summary statistics Wood presumes and, to an extent, illustrates that the Ricker model may account for certain longer-term features in the data.

Although example 3.1.2 does not utilise the concept fully, we are keen to recommend structuring the simulator's mismatch to the system with the pseudo-spatial coordinate v. We find the common alternative, structuring it via one or more additive discrepancy terms, to be comparatively confusing. Specifically, it is sometimes hard to think about the orthogonality properties of an additive discrepancy term, which are determined by its location in the model. To work out the appropriate location we must ask ourselves whether the simulator or system values are informative for the discrepancy between them. If the simulator returns an unusually high temperature, for example, would we presume that it is overestimating the system value and that the discrepancy is likely to be negative, or do we put all our trust in the simulator output? Is it more appropriate to specify

$$y_{sim} = y_{sys} \oplus \epsilon,$$

or

$$y_{sys} = y_{sim} \oplus \epsilon\ ?$$

Equivalently, if we measure an unusually high temperature outside would we assume the simulator will produce a lower value, or that it will anticipate the heatwave? We suspect that in most cases we will find ourselves sympathising with both answers, which we can model by introducing an intermediate object y_R, in a similar way to our introduction of the climate:

$$y_{sim} = y_R \oplus \epsilon_{sim}, \qquad\qquad y_{sys} = y_R \oplus \epsilon_{sys}.$$

By hypothesising the existence of an additional object we are over-parameterising insofar as its value is unknowable unless we can meaningfully identify it with something, from outside our statistical analysis, of which we have prior knowledge. If it can be identified with an upgraded version of the simulator or with a familiar theoretical device, like the climate, for example, then we can see value in assuming its existence.

By specifying variances for y_R, ϵ_{sys} and ϵ_{sim} we indirectly specify the variances of the simulator and the system, and their correlation. We feel this would be an unhelpful modelling choice however, if our intuition for the system and simulator is more highly developed than that for the intermediate object y_R. In our opinion it will most often be easier to work from the other direction, and to specify the variances and correlation directly. A spatial field is the natural tool for this. By defining the discrepancy space Ω_v we also prepare the ground for the generalisation in which we analyse the outputs from a collection of different simulators, which occupy different positions in the discrepancy space.

Chapter 4

Emulation and calibration

As we head towards the application of our ideas for emulation and calibration to real data, we will see that there are several obstacles that need to be overcome. In this chapter we develop a range of methods for constructing, fitting, and calibrating with emulators that circumvent these obstacles. The methods are informed by our work with smoothing techniques, and are suited to the large data sets and high-dimensional input spaces that frequently characterise the analysis of complex physical systems.

The FAMOUS data, introduced in section 1.1.5.1 and to be analysed in chapter 5, raise two significant challenges for the grid-based approach. Firstly, many of the simulations terminated prematurely due to crashes of the distributed system on which they were run. This means that the resulting grids have a significant number of missing values. Secondly, when considering an input space of even moderately high dimension, the number of grid points on a full design quickly grow unmanageably large. We anticipate such problems being the norm rather than the exception for the majority of practical applications and so we are motivated to seek ways to escape the grid structure while retaining, or even improving, computational tractability.

We start by introducing two conjugate models for Bayesian linear regression that represent key mechanisms within our emulation procedures: the normal inverse gamma (NIG) model and the normal inverse Wishart (NIW) model. The NIG model is a well-known device that we will use for fitting what we will call the Nyström emulators or approximate basis emulators. A full description of these emulators is the subject of section 4.3. The NIW model, the natural multivariate extension of the NIG model, is less well studied and is relied upon in our other emulation scheme, which we will describe in section 4.2.

4.1 Conjugate linear models

We now zoom out from the context of computer simulators to review some results from Bayesian linear modelling that will help us to fit our emulators and calibrate with them. Results for the normal inverse gamma construction are informed predominantly by O'Hagan and Foster[35, chap. 11], while those for the normal inverse Wishart are extensions that we have derived in order to accommodate multivariate simulator outputs. In particular, we have in mind the case in which the components of the output relate to coefficients for smooth basis functions of time, but the model is also applicable to scalar outputs of one variable arising from alternative forcing scenarios, or to outputs with different physical interpretations.

4.1.1 The NIG equipped linear model

We write the linear model with a scalar output variable, y; a p-dimensional column vector input variable, x, which includes a constant intercept component; a p vector of coefficients β; and scalar error term ϵ as

$$y = x^T\beta + \epsilon.$$

We then use a bold $\mathbf{Y}$ and $\boldsymbol{\epsilon}$ to denote the concatenation of n observations and error terms, forming column vectors. The input vectors are transposed and stacked up as rows of the $n \times p$ matrix $\mathbf{X}$ so that

$$\mathbf{Y} = \mathbf{X}\beta + \boldsymbol{\epsilon}.$$

A common way to arrive at the NIG model is to consider an infinite mixture of normal models weighted according to an inverse gamma distribution,

$$\beta|\sigma^2 \sim \mathrm{N}\left(\mathbf{m}, \sigma^2\mathbf{V}\right) \qquad \epsilon|\sigma^2 \sim \mathrm{N}\left(0, \sigma^2\mathbf{D}\right) \qquad \sigma^2 \sim \text{Inv-Gamma}\left(a, d\right).$$

It follows from this specification that the the conditional distribution for σ^2 given β is also inverse gamma while the marginal for β follows a multivariate t-distribution, the parameterisation of which is defined in appendix D.1.1 along with those of the other distributions referenced in this work,

$$\beta \sim \mathrm{t}_d\left(\mathbf{m}, a\mathbf{V}\right), \qquad \sigma^2|\beta \sim \text{Inv-Gamma}\left(a + (\beta - \mathbf{m})^T\mathbf{V}^{-1}(\beta - \mathbf{m}), d + p\right).$$

Given training data $\{\mathbf{X}, \mathbf{Y}\}$, the adjusted NIG parameters can be shown to be:

$$\mathbf{V}^* = \left(\mathbf{V}^{-1} + \mathbf{X}^T\mathbf{D}^{-1}\mathbf{X}\right)^{-1}, \tag{4.1}$$

$$\mathbf{m}^* = \mathbf{V}^*\left(\mathbf{V}^{-1}\mathbf{m} + \mathbf{X}^T\mathbf{D}^{-1}\mathbf{Y}\right), \tag{4.2}$$

$$a^* = a + (\mathbf{Y} - \mathbf{X}\mathbf{m})^T(\mathbf{D} + \mathbf{X}\mathbf{V}\mathbf{X}^T)^{-1}(\mathbf{Y} - \mathbf{X}\mathbf{m}), \tag{4.3}$$

$$d^* = d + n, \tag{4.4}$$

and with some fiddly but basic algebra, alternative expressions for a^* can be derived:

$$a^* = a + \mathbf{m}^T\mathbf{V}^{-1}\mathbf{m} + \mathbf{Y}^T\mathbf{D}^{-1}\mathbf{Y} - \mathbf{m}^{*T}(\mathbf{V}^*)^{-1}\mathbf{m}^*, \tag{4.5}$$

$$a^* = a + (\mathbf{Y} - \mathbf{X}\mathbf{m}^*)^T(\mathbf{D} + \mathbf{X}\mathbf{V}\mathbf{X}^T)(\mathbf{Y} - \mathbf{X}\mathbf{m}^*). \tag{4.6}$$

Expression (4.6), in particular, proves to be useful for establishing a relationship with the GCV criterion, commonly used in the smoothing literature, further on in this subsection.

We should note that the expression for the posterior mean $\mathbf{m}^*$ involves neither a nor d, meaning that it is the same mean we would calculate with a model for which σ^2 is assumed known. The marginal posterior for a single new output $y(x')$, given x', is a t-distribution $t_{d^*}(x'^T\mathbf{m}^*, a^*(1 + x'^T\mathbf{V}^*x'))$. More generally, the joint density for a vector of outputs is given by

$$\pi(\mathbf{Y}|\mathbf{X}) = \frac{a^{d/2}\Gamma((d+n)/2)}{|\mathbf{D} + \mathbf{X}\mathbf{V}\mathbf{X}^T|^{1/2}\pi^{n/2}\Gamma(d/2)}\left(a + (\mathbf{Y} - \mathbf{X}\mathbf{m})^T(\mathbf{D} + \mathbf{X}\mathbf{V}\mathbf{X}^T)^{-1}(\mathbf{Y} - \mathbf{X}\mathbf{m})\right)^{-(d+n)/2}, \tag{4.7}$$

$$= \frac{1}{\pi^{n/2}|\mathbf{D}|^{1/2}}\frac{\Gamma(d^*/2)}{\Gamma(d/2)}\frac{|\mathbf{V}^*|^{1/2}}{|\mathbf{V}|^{1/2}}\frac{(a^*)^{-d^*/2}}{a^{-d/2}}. \tag{4.8}$$

An alternative formulation of a^* and $\mathbf{m}^*$ shows their relation to the classical estimates for σ^2 and β, which we denote with ˇ accents:

$$\mathbf{m}^* = (\mathbf{I} - \mathbf{A})\mathbf{m} + \mathbf{A}\check{\beta},$$

$$a^* = a + (n - p)\check{\sigma}^2 + (\mathbf{m} - \check{\beta})^T\left(\mathbf{V} + (\mathbf{X}^T\mathbf{D}^{-1}\mathbf{X})^{-1}\right)^{-1}(\mathbf{m} - \check{\beta}),$$

where

$$
\begin{aligned}
(n-p)\check{\sigma}^2 &= (\mathbf{Y}-\mathbf{X}\check{\beta})^T\mathbf{D}^{-1}(\mathbf{Y}-\mathbf{X}\check{\beta}),\\
\check{\beta} &= (\mathbf{X}^T\mathbf{D}^{-1}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{D}^{-1}\mathbf{Y},\\
\mathbf{A} &= (\mathbf{V}^{-1}+\mathbf{X}^T\mathbf{D}^{-1}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{D}^{-1}\mathbf{X}.
\end{aligned}
$$

Additionally, the posterior estimates implied by the NIG model for the expectation and variance for β, given a set of training data, also coincide with the Bayes Linear estimates for them in [17, p. 266]. This result allows us to formally divorce the estimates from the normal and inverse gamma distributions that are used to construct them here, and so ought to lend the estimates conceptual and practical robustness to data that are not well described by these distributions.

O'Hagan and Forster note that setting $a = d = 0$ and $\mathbf{V}^{-1} = \mathbf{0}$ leads to the Jeffreys prior $\pi(\beta, \sigma^2) \propto \sigma^{-(p+2)}$. It also follows from the model, that the prior for the scalar quantity (4.9) is given by an F-distribution,

$$
\begin{aligned}
&\frac{d}{na}(\mathbf{Y}-\mathbf{Xm})^T(\mathbf{D}+\mathbf{XVX}^T)^{-1}(\mathbf{Y}-\mathbf{Xm}) && (4.9)\\
&\qquad =\frac{d}{n}\frac{(\mathbf{Y}-\mathbf{Xm})^T(\sigma^2(\mathbf{D}+\mathbf{XVX}^T))^{-1}(\mathbf{Y}-\mathbf{Xm})}{a\sigma^{-2}}, && (4.10)\\
&\qquad \sim \frac{\chi^2_n/n}{\chi^2_d/d}, && (4.11)\\
&\qquad \sim \mathrm{F}(n,d). && (4.12)
\end{aligned}
$$

This result is useful for informing diagnostics as we will discuss later on in section 4.4.5.

4.1.1.1 The NIG model and smoothing parameter selection

We take a brief detour now to establish a point of contact between the type of quantities we are constructing with the NIG model, and those that feature in the mainstream smoothing literature. The parameter σ^2 in the NIG linear model serves to scale the variance of β and ϵ simultaneously. However, we are frequently interested in the size of their variances relative to each other. This is the case when we want to treat the smooth climate component as arising from the regressors and the rough weather part as arising from the error term. To accommodate this we can introduce a further parameter λ such that,

$$\mathbf{V} = \lambda^{-1}\mathbf{V}'. \tag{4.13}$$

The matrix $\mathbf{V}'$ here is a fixed variance matrix and the scalar λ plays the role of a roughness penalty, a dimensionless variable that we can adjust in order to scale $\mathbf{V}'$ up or down. Conditional on λ we have the NIG model once again, meaning that we can numerically maximise the likelihood (4.8) with respect to λ, possibly constrained by a prior, to produce an estimate for λ. Since λ possesses no obvious conjugacy relations, there is no advantage to equipping it with a particular prior.

It is interesting to note the relationship between the marginal likelihood for $\mathbf{Y}$, as a function of λ, and the generalised cross-validation criterion, whose minimisation is commonly used as a method for choosing a smoothness parameter for smoothing spline problems:

$$GCV(\lambda) = n\frac{\text{SSE}}{(\text{Tr}\,(\mathbf{I} - \mathbf{S}_\lambda))^2}. \tag{4.14}$$

The matrix $\mathbf{S}_\lambda$ here is a smoother matrix and SSE is a sum of squared errors. Denoting the NIG posterior parameters that are produced for a specific fixed value of λ using a subscript, we can translate these objects into the notation of the NIG model. We equate the classical fitted smooth, $\hat{\mathbf{Y}}_\lambda$, with the posterior mean (4.2) arising from a zero prior mean,

$$\hat{\mathbf{Y}}_\lambda = \mathbf{X}\mathbf{m}^*_\lambda = \mathbf{X}\mathbf{V}^*_\lambda(\mathbf{V}^{-1}\mathbf{0} + \mathbf{X}^T\mathbf{D}^{-1}\mathbf{Y}) = \mathbf{S}_\lambda\mathbf{Y}, \tag{4.15}$$

so that the implied smoothing matrix, which is essentially defined by its role in (4.15), is

$$\mathbf{S}_\lambda = \mathbf{X}\mathbf{V}^*_\lambda\mathbf{X}^T\mathbf{D}^{-1}.$$

The sum of squares term in (4.14) is the obvious quantity

$$\text{SSE} = (\mathbf{Y} - \hat{\mathbf{Y}}_\lambda)^T(\mathbf{Y} - \hat{\mathbf{Y}}_\lambda),$$

so that we can write,

$$GCV(\lambda) = \frac{n}{\left(\text{Tr}\left(\mathbf{I} - \mathbf{X}\mathbf{V}^*_\lambda\mathbf{X}^T\mathbf{D}^{-1}\right)\right)^2}(\mathbf{Y} - \mathbf{X}\mathbf{m}^*_\lambda)(\mathbf{Y} - \mathbf{X}\mathbf{m}^*_\lambda). \tag{4.16}$$

Ramsay and Silverman point out that it is instructive to view this quantity as a product:

$$GCV(\lambda) = \frac{n}{\text{Tr}\,(\mathbf{I} - \mathbf{S}_\lambda)} \times \frac{(\mathbf{Y} - \hat{\mathbf{Y}}_\lambda)^T(\mathbf{Y} - \hat{\mathbf{Y}}_\lambda)}{\text{Tr}\,(\mathbf{I} - \mathbf{S}_\lambda)}, \tag{4.17}$$

where the right-hand factor is reminiscent of the unbiased estimate of the error variance from a classical linear regression problem, and the left-hand factor is seen as a multiplicative discount that rewards smoother smooths.

When we take the NIG posterior resulting from the prior that specifies $\mathbf{m} = \mathbf{0}$ and $a = d = 0$, and raise it to the power $-2/n$, we produce an expression comparable to (4.17),

$$\pi(\mathbf{Y}|\mathbf{X})^{-2/n} \propto |\mathbf{I} - \mathbf{X}\mathbf{V}_\lambda^* \mathbf{X}^T \mathbf{D}^{-1}|^{-1/n}\ (\mathbf{Y} - \mathbf{X}\mathbf{m}_\lambda^*)^T (\mathbf{I} - \mathbf{X}\mathbf{V}_\lambda^* \mathbf{X}^T \mathbf{D}^{-1})^{-1} \mathbf{D} (\mathbf{Y} - \mathbf{X}\mathbf{m}_\lambda^*), \quad (4.18)$$

$$= |\mathbf{I} - \mathbf{S}_\lambda|^{-1/n} \times (\mathbf{Y} - \hat{\mathbf{Y}}_\lambda)^T (\mathbf{I} - \mathbf{S}_\lambda)^{-1} \mathbf{D} (\mathbf{Y} - \hat{\mathbf{Y}}_\lambda), \quad (4.19)$$

which we also view as a product. As mentioned in section 2.1, it is necessary to be more careful when one or more linear combinations of the coefficients are unconstrained by the penalty, which results in zero eigenvalues for $\mathbf{V}^{-1}$. In this case, we employ the pseudo-determinant and the Moore-Penrose generalised inverse to redefine (4.19) as,

$$\pi(\mathbf{Y}|\mathbf{X})^{-2/n} \propto |\mathbf{I} - \mathbf{S}_\lambda|_+^{-1/n} \times (\mathbf{Y} - \hat{\mathbf{Y}}_\lambda)^T (\mathbf{I} - \mathbf{S}_\lambda)^{\dagger} \mathbf{D} (\mathbf{Y} - \hat{\mathbf{Y}}_\lambda).$$

The comparison of (4.14) and (4.19) is of interest because the GCV criterion has been criticised by authors such as Gu[18] for producing curves that are judged to be under-smoothed. One of Gu's recommendations is to multiply $\mathbf{S}_\lambda$ in (4.17) by a scalar slightly larger than one to compensate for this. We find that the criterion derived from the NIG likelihood encourages more smoothing and results in a more clearly defined minimum than the GCV. Although we currently cannot prove these properties always hold, and our investigations of the criteria rely on numerical explorations, we can begin to see the reason for the NIG likelihood's greater preference for smoothness by noting that the left-hand terms of (4.14) and (4.19), which we view as punishments for a smooth's wiggliness or over-fitting, satisfy the inequality

$$|\mathbf{I} - \mathbf{S}_\lambda|^{-1/n} \geq \frac{n}{\mathrm{Tr}\,(\mathbf{I} - \mathbf{S}_\lambda)}$$

due to theorem B.0.21. We argue that the likelihood-based criterion is better than the GCV criterion for choosing a smooth since it leads us to the type of inference we would like to make, specifically a smoother curve, in a principled way that feels conceptually cleaner than an ad hoc modification to the GCV formula. In example 4.1.1 we will also see the extent to which the optimising value of the likelihood-based criterion is better identified than that for the GCV.

Example 4.1.1. [Roughness inference for the FAMOUS data and the NIG model.] We now return to the FAMOUS time series we looked at in example 2.1.1 when we introduced penalty-based smoothing. We equate the matrix of basis values $\boldsymbol{\phi}$ from that example with the matrix $\mathbf{X}$ from the description of the NIG model. As we did before, we set $\mathbf{D} = \mathbf{I}$ to be the identity matrix and define the unscaled precision, or inverse variance $(\mathbf{V}')^{-1} = \mathbf{P}$, using either first or second squared derivative penalties. We then approach the data from the spline/penalty smoothing perspective and from the perspective of the NIG Bayesian linear model with a prior such that $\mathbf{m} = \mathbf{0}$ and $a = d = 0$.

The penalty approach yields a minimal loss solution for the coefficients,

$$\hat{\beta}_\lambda = \left[\lambda \mathbf{P} + \mathbf{X}^T\mathbf{X}\right]^{-1} \mathbf{X}^T\mathbf{Y} = \mathbf{V}^*\mathbf{X}\mathbf{Y}^T, \tag{4.20}$$

which coincides with the NIG posterior mean,

$$\mathbb{E}\,(\beta \mid \mathbf{Y}, \lambda) = \mathbf{m}^*_\lambda = \mathbf{V}^*\mathbf{X}^T\mathbf{Y}. \tag{4.21}$$

But the NIG model also provides us with a variance for the coefficients,

$$\mathrm{Var}\,(\beta \mid \mathbf{Y}, \lambda) = \frac{a^*}{n-2}\mathbf{V}^*. \tag{4.22}$$

Despite the coincidence of (4.20) and (4.21), the smooths we calculate from the different perspectives are different because we condition them on the minimising values of different criteria:

$$\log(GCV(\lambda)) = \log(n) - 2\log(\mathrm{Tr}\,(\mathbf{I} - \mathbf{S}_\lambda)) + \log((\mathbf{Y} - \mathbf{X}\hat{\beta}_\lambda)^T(\mathbf{Y} - \mathbf{X}\hat{\beta}_\lambda)), \tag{4.23}$$

for the smoothing spline approach and

$$-\frac{2}{n}\log(\pi(\mathbf{Y} \mid \mathbf{X}, \lambda)) = -\frac{1}{n}\log(|\mathbf{I} - \mathbf{X}\mathbf{V}^*_\lambda\mathbf{X}^T|_+) \tag{4.24}$$

$$+ \log((\mathbf{Y} - \mathbf{X}\mathbf{m}^*_\lambda)(\mathbf{I} - \mathbf{X}\mathbf{V}^*_\lambda\mathbf{X}^T)^\dagger(\mathbf{Y} - \mathbf{X}\mathbf{m}^*_\lambda)), \tag{4.25}$$

for the NIG approach.

In figure 4.1 we present plots of the GCV and log-likelihood criteria, (4.23) and (4.25), as $\log(\lambda)$ is varied along the x-axes. For both first and second derivative penalties, the log-likelihood criterion produces a more definitive minimum at a larger value of the penalty multiplier λ.

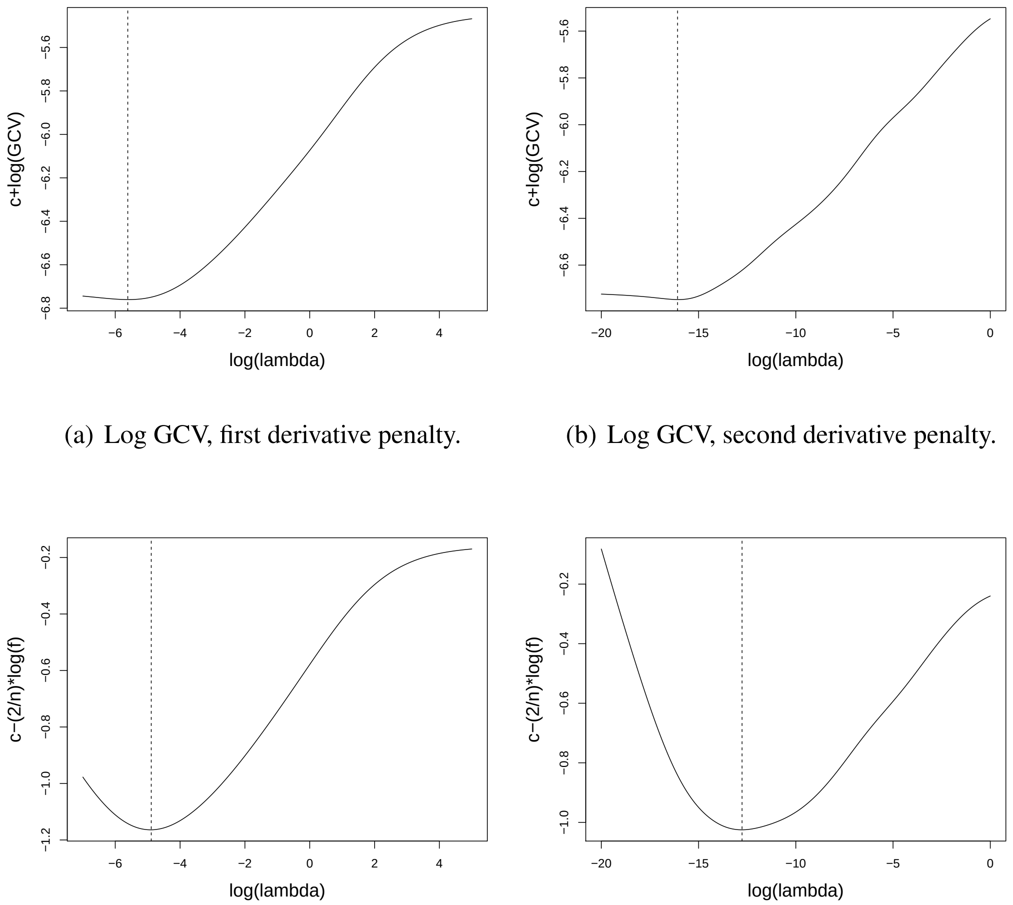
In figure 4.2 we look at the smooths resulting from the optimal values for λ as identified by the different criteria. We see that when we penalise the first derivative, the smooths are almost indistinguishable until we reach the end of the range of observations. The difference between them is more noticeable when we penalise the second derivative. In subfigure 4.2(b) it is clear that the smooth from the NIG model is smoother than that minimising the GCV. This is particularly evident from the way the latter is bent steeply downwards at the right edge of the data range; extrapolating the smooth like this is unlikely to be considered appropriate.

We partially illustrate the NIG posterior variance (4.22) with grey shaded regions, and note that the pointwise credible intervals that the regions demarcate are very conservative with regard to extrapolation. This feature is due to the climate trend's lower derivatives being completely un-penalised. For the climate in figure 4.2(b), for example, our prior variance specification says that the curvature of the climate trend is small, precisely how small being determined by λ, but that its gradient and value can be of any size. Consequently, the credible interval for the climate beyond the range of observations is determined by the degree to which we allow the climate to bend at the end of the range of observations. This sort of variance specification can be seen in opposition to that which we remarked upon in example 2.4.2. In that example we used a Matérn function, which implicitly penalized all derivatives, to derive the covariances of the climate trend, and saw that the resulting smooths all decayed back to the global mean.

4.1.2 The NIW equipped linear model

The results presented here for the normal inverse Wishart linear model are our own extrapolations of those for the normal inverse gamma case. In the multivariate setting, $\mathbf{Y}$ becomes an $n \times q$ matrix of observations consisting of n q-dimensional output vectors stacked up row by row; similarly, $\mathbf{E}$ is a matrix, of the same size as $\mathbf{Y}$, made by stacking q-dimensional error terms. As before, the input variable x is a p vector and $\mathbf{X}$ is an $n \times p$ matrix of different input vectors stacked up. The model coefficients are now kept in the $p \times q$ matrix $\boldsymbol{\beta}$. Combining these objects we write the model as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{E}. \tag{4.26}$$



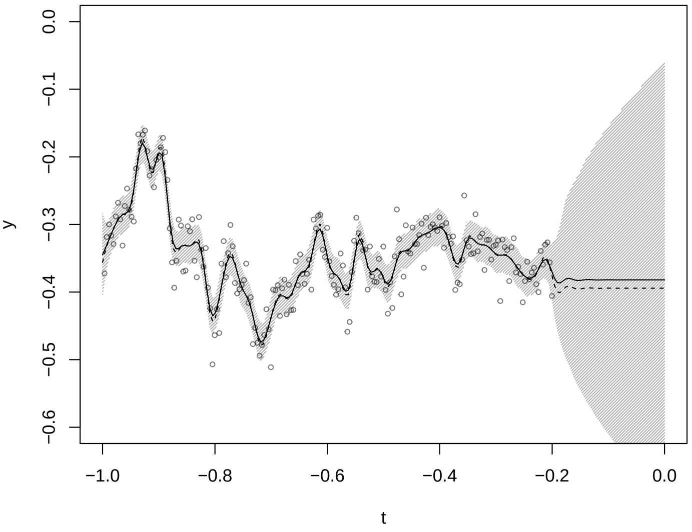
(a) Log GCV, first derivative penalty.

(b) Log GCV, second derivative penalty.

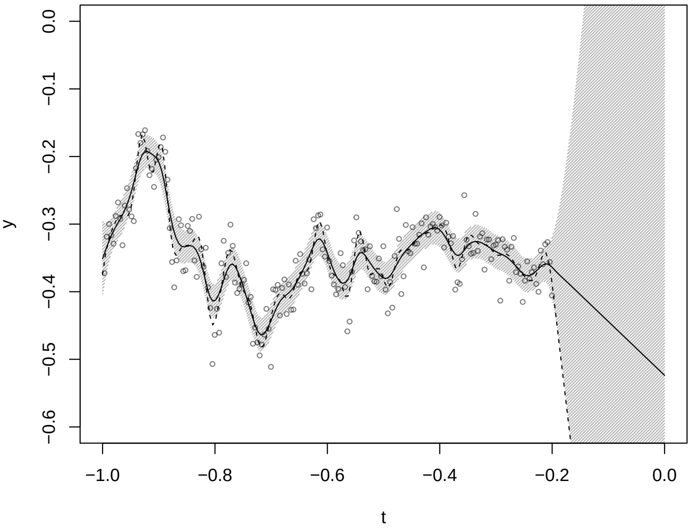
(c) Transformed log-likelihood, first derivative penalty.

(d) Transformed log-likelihood, second derivative penalty.

Figure 4.1: Plots of the log GCV and transformed log likelihood from the NIG model up to an additive constant. The dashed vertical lines mark the locations of the curves' minimal values.



(a) First derivative penalty.



(b) Second derivative penalty.

Figure 4.2: The points here show a simulated AMOC time series from FAMOUS. The solid black lines show the adjusted means for c from the NIG model with λ set to minimise the negative log-likelihood (4.19). The dashed lines show the penalty-derived smooths given values of λ that minimise the GCV criterion. The shading marks out a region two standard deviations from the mean for c according to the NIG model.

We can imagine, for instance, that the q outputs correspond to simulator outputs for different times and that the vector x contains the simulator's input variables along with an intercept term.

The implication of the model is that each of the q elements of a single output vector is produced by a univariate linear model whose coefficients are stored in the qth column of $\boldsymbol{\beta}$. We introduce covariances between the linear models via covariances between the coefficients as defined by the $q \times q$ matrix $\mathbf{H}$, which we consider random. This variable is analogous to the variable σ^2 from the NIG prior. The $p \times p$ matrix $\mathbf{V}$, which encodes the covariances between coefficients corresponding to different inputs, is considered known. The resulting covariance matrix for $\boldsymbol{\beta}$ can be written as

$$\mathrm{Var}\,(\mathrm{vec}\,(\boldsymbol{\beta})\,|\mathbf{H}) = \mathbf{H} \otimes \mathbf{V},$$

where, again, we employ the vec $(\cdot)$ notation that reads a matrix into a vector; or, considering elements individually, we write it as

$$\mathrm{Cov}\left([\boldsymbol{\beta}]_{i,j}\ ,\ [\boldsymbol{\beta}]_{k,l}|\mathbf{H}\right) = [\mathbf{H}]_{j,l}[\mathbf{V}]_{i,k}.$$

In order to achieve conjugacy we will also need

$$\mathrm{Var}\,(\mathrm{vec}\,(\mathbf{E})\,|\mathbf{H}) = \mathbf{H} \otimes \mathbf{D},$$
$$\mathrm{Cov}\left([\mathbf{E}]_{i,j}\ ,\ [\mathbf{E}]_{k,l}|\mathbf{H}\right) = [\mathbf{H}]_{j,l}[\mathbf{D}]_{i,k},$$

meaning that the error terms of the q sub models are similarly correlated. The result is that the variance of $\mathbf{Y}$ conditional on $\mathbf{H}$ can be factorised as a Kronecker product:

$$\mathrm{Var}\,(\mathrm{vec}\,(\mathbf{Y})) = \mathbf{H} \otimes (\mathbf{X}^T\mathbf{V}\mathbf{X} + \mathbf{D}). \tag{4.27}$$

Theorem B.0.4 and an assumption of normally distributed error terms allows us to contract sums of squares expressions into traces of products of matrices, with the effect that we can write the likelihood for $(\boldsymbol{\beta}, \mathbf{H})$ given $\mathbf{Y}$ as

$$\pi(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\beta}, \mathbf{H}) = (2\pi)^{-(nq)/2}\,|\mathbf{H} \otimes \mathbf{D}|\exp\left[-\frac{1}{2}\mathrm{vec}\,(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T\,(\mathbf{H} \otimes \mathbf{D})^{-1}\mathrm{vec}\,(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})\right], \tag{4.28}$$

$$= (2\pi)^{-(nq)/2}\,|\mathbf{H}|^{-n/2}|\mathbf{D}|^{-q/2}\exp\left[-\frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}\,(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T\,\mathbf{D}^{-1}\,(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})\right)\right]. \tag{4.29}$$

Now, the normal inverse Wishart distribution for $(\boldsymbol{\beta}, \mathbf{H})$ with fixed parameters $\mathbf{M}$, $\mathbf{V}$, $\boldsymbol{\Psi}$ and ν is given by

$$\pi(\boldsymbol{\beta}, \mathbf{H}) = k_{\pi(\boldsymbol{\beta},\mathbf{H})}|\mathbf{H}|^{-(\nu+p+q+1)/2} \exp\left[-\frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}\left((\boldsymbol{\beta}-\mathbf{M})^T\,\mathbf{V}^{-1}\,(\boldsymbol{\beta}-\mathbf{M}) + \boldsymbol{\Psi}\right)\right)\right]. \tag{4.30}$$

This prior implies that the marginal distribution for $\mathbf{H}$ is an inverse Wishart distribution. It also implies that the elements of $\boldsymbol{\beta}$ given $\mathbf{H}$ are distributed according to a multivariate normal so that the NIW prior can be derived as a mixture of multivariate normal distributions. The normalising constant in (4.30) can be found by integrating out $\boldsymbol{\beta}$ and comparing it to the inverse Wishart density function, a calculation presented in appendix C.2.1, to reveal:

$$k_{\pi(\boldsymbol{\beta},\mathbf{H})} = \frac{|\boldsymbol{\Psi}|^{\nu/2}}{(2\pi)^{pq/2}|\mathbf{V}|^{q/2}\Gamma_q(\nu/2)2^{\nu q/2}}.$$

We also note that the Jeffreys prior for the NIW distribution is given by

$$\pi(\boldsymbol{\beta}, \mathbf{H}) \propto |\mathbf{H}|^{-(p+q+1)/2},$$

which results from the limit of taking $\mathbf{V}^{-1}$ towards to the zero matrix, along with ν and $\boldsymbol{\Psi}$.

Theorem 4.1.2 (The conjugate parameter updates for the NIW model)**.** *Given that* $(\boldsymbol{\beta}, \boldsymbol{H})$ *is distributed as a NIW variable with parameters* $\{\boldsymbol{M}, \boldsymbol{V}, \boldsymbol{\Psi}, \nu\}$ *and that the conditional distribution of* $\boldsymbol{Y}$ *given* $\{\boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{H}\}$ *is the multivariate normal whose density is given by* (4.29)*, the posterior for* $(\boldsymbol{\beta}, \boldsymbol{H})$ *given* $\{\boldsymbol{X}, \boldsymbol{Y}\}$ *is also NIW with parameters*

$$\boldsymbol{V}^* = \left(\boldsymbol{V}^{-1} + \boldsymbol{X}^T\boldsymbol{D}^{-1}\boldsymbol{X}\right)^{-1}, \tag{4.31}$$

$$\boldsymbol{M}^* = \boldsymbol{V}^*\left(\boldsymbol{V}^{-1}\boldsymbol{M} + \boldsymbol{X}^T\boldsymbol{D}^{-1}\boldsymbol{Y}\right), \tag{4.32}$$

$$\boldsymbol{\Psi}^* = \boldsymbol{\Psi} + (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{M})^T\left(\boldsymbol{D} + \boldsymbol{X}\boldsymbol{V}\boldsymbol{X}^T\right)^{-1}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{M}), \tag{4.33}$$

$$\nu^* = \nu + n. \tag{4.34}$$

Corollary 4.1.3 (The marginal density for $\mathbf{Y}$)**.** *The marginal probability density function for* $\boldsymbol{Y}$ *given* $\boldsymbol{X}$*, given the assumptions described in theorem 4.1.2, is*

$$\pi(\boldsymbol{Y} \mid \boldsymbol{X}) = \pi^{-nq/2}|\boldsymbol{D}|^{-q/2}\frac{\Gamma_q(\nu^*/2)}{\Gamma_q(\nu/2)}\frac{|\boldsymbol{V}^*|^{q/2}}{|\boldsymbol{V}|^{q/2}}\frac{|\boldsymbol{\Psi}|^{\nu/2}}{|\boldsymbol{\Psi}^*|^{\nu^*/2}}. \tag{4.35}$$

Proof. The proofs for results 4.1.2 and 4.1.3 are included as appendices in C.1 and C.2.2 respectively. □

Expression (4.35) is the multivariate version of (4.8); it is the marginal likelihood we would use to re-weight a prior for a single x^* given $y(x^*)$. When we look at it as a function of x^*, which is a single column vector of p input quantities, the only terms we need to consider are the two determinants

$$\pi(y(x^*) \mid x^*) \propto |\mathbf{V}^*|^{q/2}|\mathbf{\Psi}^*|^{-(\nu+1)/2}, \tag{4.36}$$

since these are the only ones that contain x^*. Expanding (4.36), and remembering that $y(x^*)$ is a q-dimensional column vector and that $\mathbf{D}$ is a square matrix of size $N_x \times N_x$, so a scalar, leads to

$$\begin{aligned}\pi(y(x^*) \mid x^*) \propto & |\mathbf{V}^{-1} + x^*\mathbf{D}^{-1}x^{*T}|^{-q/2} \\ & \times |\mathbf{\Psi} + \left(y(x^*)^T - x^{*T}\mathbf{M}\right)^T \left(\mathbf{D} + x^{*T}\mathbf{V}x^*\right)^{-1} \left(y(x^*)^T - x^{*T}\mathbf{M}\right)|^{-(\nu+n)/2},\end{aligned}$$

which we can re-express in terms of quadratic forms rather than determinants using theorem B.0.7,

$$\pi(y(x^*) \mid x^*) \propto (\mathbf{D} + x^{*T}\mathbf{V}x^*)^{-q/2} \left(1 + \frac{\left(y(x^*)^T - x^{*T}\mathbf{M}\right)\mathbf{\Psi}^{-1}\left(y(x^*)^T - x^{*T}\mathbf{M}\right)^T}{\mathbf{D} + x^{*T}\mathbf{V}x^*}\right)^{-(\nu+1)/2}. \tag{4.37}$$

Function (4.37) is the same as that arrived at by Plessis and van der Merwe [40] and Brown [6] in their work on the Bayesian calibration of multivariate linear models. It is an interesting function, resembling a multivariate t-distribution in x^* but with the inclusion of the factor

$$\mathbf{D} + x^{*T}\mathbf{V}x^*,$$

which serves to flatten out the likelihood when x^* grows large in directions where the variance for the coefficients, $\mathbf{V}$, lends it leverage. As long as $\mathbf{V}$ is positive definite, when x^* gets very large in any direction, the right-hand factor of (4.37) tends to a constant, leaving the tail behaviour to the left-hand term. This also resembles a multivariate t-distribution, but centred on the origin. By referring to the convergence properties of the multivariate t-density, we see that the integral of our likelihood will only converge when $q > p$. As a consequence, when we have more output quantities to calibrate against than input quantities to calibrate, the likelihood leads to a proper posterior for a vague, uniform prior on x^*. Note that this is not the case otherwise, and that, in particular, the posterior resulting from the NIG likelihood, (4.35), and an improper uniform prior for x^* diverges. Our studies of function (4.37) have not produced further results of interest, however, so we continue now without discussing the properties of such posterior densities.

The conjugacy of the prior allows us to update the NIW model extremely quickly. Equally quick are downdates, derived from the inversion of equations (4.31)-(4.34), and formulated explicitly in appendix C.3. A downdate refers to the adjustment of the model's parameter distribution upon the removal, or un-learning, of training data. The concept is useful for the calculation of leave-one-out diagnostics when we come to checking the model's fit, as we do in section 5.1.5.

The model is appropriate for a reasonably small selection of outputs, to keep down the cost of frequent matrix inversions, which are all thought to depend on the inputs to a similar degree and in different ways. We make this recommendation on the basis that the nature of the variation of each linear combination of the outputs across the input space is a priori identical after scaling. The model would therefore be inappropriate if certain linear combinations of the output corresponded to weather-like terms that are expected to vary rapidly over the input space, and others to climate-like trends that are expected to vary more gradually over the input space.

4.2 The NIW emulator

Having introduced the NIW linear model, in this section we describe how we can employ it for the purposes of emulation of time series data. With the resulting NIW emulator we treat coefficients for basis functions in time, rather than the simulator output values at specific times, as the emulator's response variable. Since the coefficients are not observable, we rely on a Gibbs sampling procedure to simulate coefficient values that can be conditioned on. By employing fewer basis functions than simulator observations we reduce the size of the matrices we would otherwise need to manipulate, and by repeatedly simulating basis coefficients conditional on simulator outputs we effectively integrate out the missing data from crashed simulations.

The inverse Wishart component of the NIW model provides a formal mechanism for us to learn about the covariance properties of the climate signal via the covariances between the basis coefficients. Accordingly, the Gibbs sampler also includes a sampling step from an inverse Wishart distribution for the variance matrix $\mathbf{H}$.

The precise role of the NIW model in our emulator is most easily illustrated by describing the imagined generative mechanism for a set of N_x simulator outputs. We suppose that variance matrix $\mathbf{H}$ is a member of the population described by the inverse Wishart distribution

$$\mathbf{H} \sim \mathrm{IW}_{\nu}\left(\boldsymbol{\Psi}\right), \tag{4.38}$$

and that given $\mathbf{H}$, the elements of the $N_x \times q$ matrix of basis coefficients, $\boldsymbol{\beta}$, are drawn from a multivariate normal distribution according to

$$\mathrm{vec}\left(\boldsymbol{\beta}\right) \mid \mathbf{H} \sim \mathrm{N}\left(\mathbb{E}\left(\mathrm{vec}\left(\boldsymbol{\beta}\right)\right), \mathbf{H} \otimes \mathbf{K}\right), \tag{4.39}$$

where the fixed matrix $\mathbf{K}$ encodes correlations of coefficients between series and $\mathbf{H}$ encodes correlations within them. The basis coefficients are then multiplied by a $q \times N_t$ matrix of basis function values to produce an $N_x \times N_t$ matrix of values for the climate trend,

$$\mathbf{C} = \boldsymbol{\beta}\boldsymbol{\phi}^T, \tag{4.40}$$

where the ith row of $\mathbf{C}$ contains the climate trend underlying the ith simulation,

$$[\mathbf{C}]_{i,j} = c([\mathbf{T}]_j, [\mathbf{X}]_{i,\cdot}). \tag{4.41}$$

To the climate trends, independent multivariate normal vectors of weather values are added to produce the $N_x \times N_t$ matrix of observable quantities $\mathbf{Y}$,

$$\mathrm{vec}\left(\mathbf{Y}\right) \mid \mathbf{C} \sim \mathrm{N}\left(\mathrm{vec}\left(\mathbf{C}\right), \mathbf{K}_w \otimes \mathbf{I}\right). \tag{4.42}$$

In the final step of the generative mechanism we imagine that the full grid structure of observations is corrupted when a set of elements of $\mathbf{Y}$ is deleted. To keep track of which values still remain, it is useful to introduce the sets I_i for $i = 1, \ldots, N_x$, which contain the column numbers of un-deleted output values for each simulation, or, equivalently, for each row of $\mathbf{Y}$.

In the following two subsections we expand on the process of fitting the NIW emulator to a ragged array of data and calibrating with it, while in the complementary examples 4.2.1 and 4.2.2, we walk through these processes with synthetic data sets in order to further demonstrate and clarify features of the calculations involved.

4.2.1 Fitting the NIW emulator

Algorithm 1 serves to describe concisely our Gibbs fitting procedure. It consists of $N_x + 1$ update operations: N_x for the individual coefficient vectors and one for the matrix $\mathbf{H}$.

In describing these operations we use the subscript notation whereby $[\boldsymbol{\beta}]_{i,\cdot}$ denotes the ith row of the matrix $\boldsymbol{\beta}$ and $[\boldsymbol{\beta}]_{-i,\cdot}$ denotes the version of $\boldsymbol{\beta}$ with the ith row removed. We also mix positive and negative subscripts so that $[\mathbf{K}]_{i,-i}$ is the vector formed by taking the ith row of $\mathbf{K}$ and deleting its ith column.

In lines 4 and 5 of algorithm 1 we adjust the moments of the basis coefficients for a particular simulated climate given the current estimates for all the others. Lines 6 and 7 serve to customise the full basis matrix and weather variance matrix, which are the same as those in expressions (4.40) and (4.42) respectively, according to the particular values of simulation j that are available. The customised matrices are then used in 8 and 9 to induce moments for the normal distribution of simulator output values. In lines 10 and 11 we adjust the coefficient moments again, this time by the simulator output values.

In lines 12 and 14 we simulate coefficient vectors and the matrix $\mathbf{H}$, and use them to redefine the state of the system, overwriting their previous values. Finally, in lines 15 and 16 we update a running mean of the sampler's coefficient values from the post-burn-in iterations. The running mean, which encodes our estimate for the joint smooths, is the algorithm's output.

Algorithm 1 NIW Gibbs sampler

```
Initiate $\boldsymbol{\beta} \leftarrow \mathbb{E}(\boldsymbol{\beta})$, $\mathbf{H} \leftarrow \mathbb{E}(\mathbf{H})$
for $i = 1, \ldots, N_{it}$ do
    for $j = 1, \ldots, N_x$ do
        $\mu_\beta \leftarrow \mathbb{E}\left([\boldsymbol{\beta}]_{j,\cdot}\right) + [\mathbf{K}]_{j,-j}[\mathbf{K}]^{-1}_{-j,-j}([\boldsymbol{\beta}]_{-j,\cdot} - \mathbb{E}\left([\boldsymbol{\beta}]_{-j,\cdot}\right))$
        $\Sigma_\beta \leftarrow ([\mathbf{K}]_{j,j} - [\mathbf{K}]_{j,-j}[\mathbf{K}]^{-1}_{-j,-j}[\mathbf{K}]_{-j,j}) \times \mathbf{H}$
        $\tilde{\boldsymbol{\phi}} \leftarrow [\boldsymbol{\phi}]_{I(j),\cdot}$
        $\tilde{\mathbf{K}}_w \leftarrow [\mathbf{K}_w]_{I(j),I(j)}$
        $\mu_y \leftarrow \tilde{\boldsymbol{\phi}}\mu_\beta$
        $\Sigma_y \leftarrow \tilde{\boldsymbol{\phi}}\Sigma_\beta\tilde{\boldsymbol{\phi}}^T + \tilde{\mathbf{K}}_w$
        $\mu_\beta \leftarrow \mu_\beta + \Sigma_\beta\tilde{\boldsymbol{\phi}}^T\Sigma_y^{-1}(y - \mu_y)$
        $\Sigma_\beta \leftarrow \Sigma_\beta - \Sigma_\beta\tilde{\boldsymbol{\phi}}^T\Sigma_y^{-1}\tilde{\boldsymbol{\phi}}\Sigma_\beta$
        Simulate $[\boldsymbol{\beta}]_{j,\cdot} \sim \mathrm{N}\left(\mu_\beta, \Sigma_\beta\right)$
    end for
    Simulate $\mathbf{H} \sim \mathrm{IW}_{\nu+N_x}\left(\boldsymbol{\Psi} + (\boldsymbol{\beta} - \mathbb{E}(\boldsymbol{\beta}))^T\mathbf{K}^{-1}(\boldsymbol{\beta} - \mathbb{E}(\boldsymbol{\beta}))\right)$
    if $i > N_{burn}$ then
        $\bar{\boldsymbol{\beta}} \leftarrow ((i - N_{burn})\bar{\boldsymbol{\beta}} + \boldsymbol{\beta})/(i - N_{burn} + 1)$
    end if
end for
return $\bar{\boldsymbol{\beta}}$.
```

Example 4.2.1 (A synthetic example of fitting the NIW emulator)**.** In this lightweight example we adopt the model for emulating a simulator's output described in lines (4.38)-(4.42). Our input variable is one-dimensional, predominantly in order to produce easily interpretable plots, and the $N_x = 23$ simulation input coordinates are spaced equally along the interval $[-1, 1]$. The fixed matrix $\mathbf{K}$ is derived from a Matérn autocovariance function,

$$[\mathbf{K}]_{i,j} = k_{Mat}(|x_i - x_j|, u = 0.3, \nu = 2). \tag{4.43}$$

To create the climate basis we select $p = 27$ equally spaced points over the time interval $[-1, 1]$. These points mark the centres of Matérn autocovariance functions that we use as basis functions, the span of the basis functions then defines the space of possible climate trends.

We derive a prior mean, $\mathbb{E}(\mathbf{H})$, for the variance matrix $\mathbf{H}$ by inverting the variance matrix, defined by the same Matérn autocovariance function (4.43), for the basis centre points. The reasoning behind this choice will become clearer in section 4.4 where we expand on our understanding of basis function approximations to autocovariances. We then generate a true value for $\mathbf{H}$ by sampling from an inverse Wishart distribution with mean $\mathbb{E}(\mathbf{H})$ and $\nu = 60$ degrees of freedom. Conditional on $\mathbf{H}$, the coefficients are multivariate normal with moments

$$\mathbb{E}(\boldsymbol{\beta}) = \mathbf{0}, \qquad \mathrm{Cov}\left([\boldsymbol{\beta}]_{i,\cdot}\ ,\ [\boldsymbol{\beta}]_{j,\cdot} \mid \mathbf{H}\right) = [\mathbf{K}]_{i,j}\mathbf{H}.$$

We then specify the variance of the multivariate normal weather signals using another Matérn autocovariance function so that

$$\mathbb{E}\left([\mathbf{W}]_{i,j}\right) = \mathbb{E}\left(w(t_j, [\mathbf{X}]_{i,\cdot})\right) = \mathbf{0}, \qquad \mathrm{Cov}\left([\mathbf{W}]_{i,j}\ ,\ [\mathbf{W}]_{k,l}\right) = [\mathbf{I}]_{i,k}[\mathbf{K}_w]_{j,l}, \tag{4.44}$$

$$[\mathbf{K}_w]_{j,l} = 0.4^2 k_{Mat}(|t_j - t_l|, 0.05, 1). \tag{4.45}$$

The complete grid of simulator outputs, which is not observed, consists of $N_x = 23$ simulations of $N_t = 100$ equally spaced output values. This grid is corrupted by deleting elements in the ith row of $\mathbf{Y}$ after a particular simulation termination time. For this example, we generated the termination times by multiplying samples from the binomial distribution $\mathrm{B.nom}\,(n = 5, p = 0.7)$ by twenty. The burn-in period for the algorithm, which we choose by eye, consists of $N_{burn} = 1000$ sweeps through all of the $N_x + 1$ parameter updates.

In figure 4.3 we plot the synthetic training data as well as the unobserved climate trends generated by the model. In figure 4.4(a) we show the expectations of the climate trends inferred from considering only one simulation at a time; we will refer to these as the individual smooths. For the calculation of these smooths we have also treated $\mathbf{H}$ as fixed at its prior mean. This is because the conjugate mechanism for learning about $\mathbf{H}$ relies on our ability to condition on observed coefficient values. Without such observations or the Gibbs sampler's simulations of them, learning about $\mathbf{H}$ requires a numerical integration procedure that is unacceptably costly given that the individual smooths are intended to be a quick and crude alternative to the joint smooths.

Figure 4.4(b) shows the climate trends that result from the average of the basis coefficient values over the post burn-in Gibbs sampler iterations. These constitute our estimates for the joint smooths given the ragged array of observations.

Even without quantitative assessment, it is clear from the plots that the joint smooths provide a better approximation of the climate trends. Firstly, the structured relationship between the trends is clearly visible in the way the curves in figure 4.4(b) are more closely aligned with each other that those in figure 4.4(a). This sharing of information between the simulations is most noticeable towards the end of the time domain when most of the simulations have crashed. This is evident in the way most of the curves in figure 4.4(a) decay smoothly back to zero, while those of figure 4.4(b), using information from the simulations that did not crash, maintain flatter trajectories above zero.

In regard to the degree and type of smoothing, the individual smooths appear to be over-smoothed in the sense that changes in direction of the climate expectation tend to be sharper in figure 4.4(b). The appearance of sharper turns requires that the preference for smoothness described by $\mathbb{E}(\mathbf{H})$ is, to an extent, counteracted. This is possible with the Gibbs emulator: firstly, because information from all the simulations is combined to resist the prior for smoothness; and secondly, because the definition of smoothness, as described by $\mathbf{H}$, may be modified by the inverse Wishart component of the model.

Unfortunately, even on the scale of our toy example, the Gibbs sampler is not particularly quick in human time, and we suspect its application to significantly larger data sets would lead to unsatisfactory waiting times. We identify three reasons for the algorithm's slow progress. Firstly, mixing is slowed by the strong correlations between climate series. The usual response to this problem would be to alter the sampler to update multiple coefficient vectors at once; to do so however would eventually lead to greater computational costs from handling and simulating larger arrays of numbers. Secondly, the sampler spends a considerable amount of time recomputing quantities that are almost the same from one iteration to the next. With this comment we refer to the sampler's cyclic update schedule, which visits coefficients mostly pinned down by long series of observations just as frequently as those with very few observations, whose posterior variance is large. Thirdly, our looped coding in R is not particularly efficient. This is an implementation issue rather than a methodological one, but it is still a consideration for our work.

Of these three issues, we find the second particularly interesting. For not only do we anticipate having to deal with coefficient vectors with differing posterior variance,

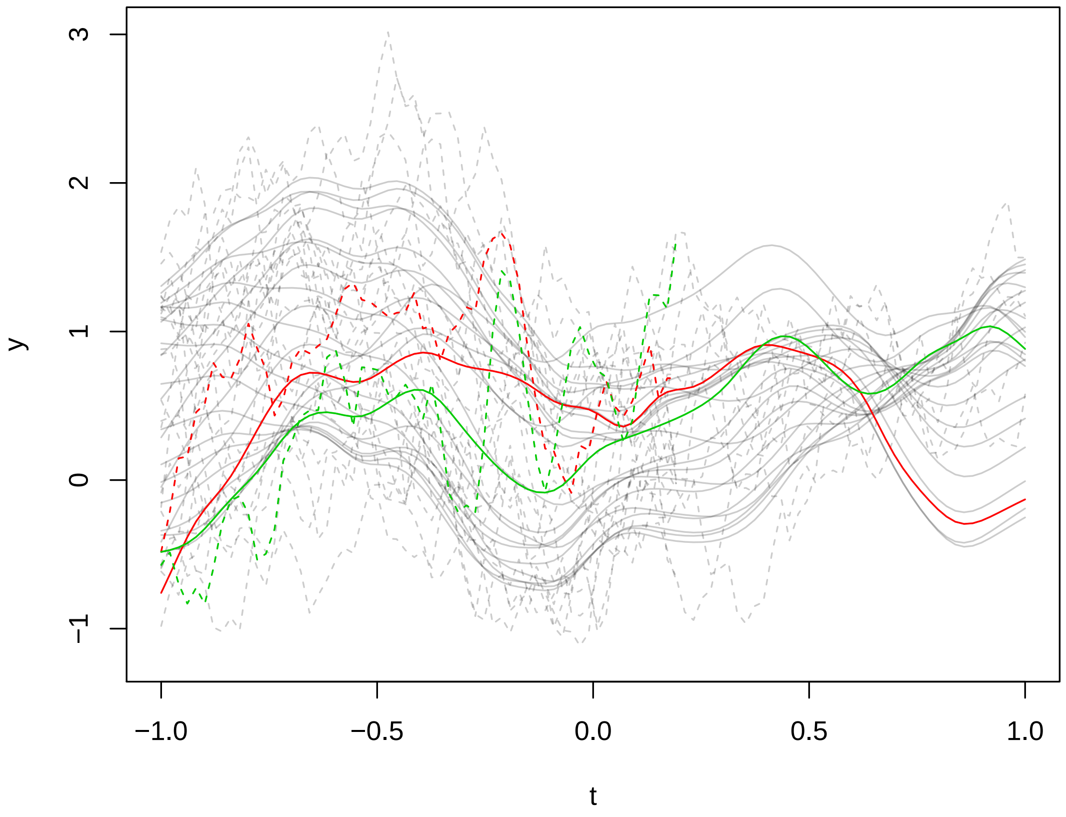
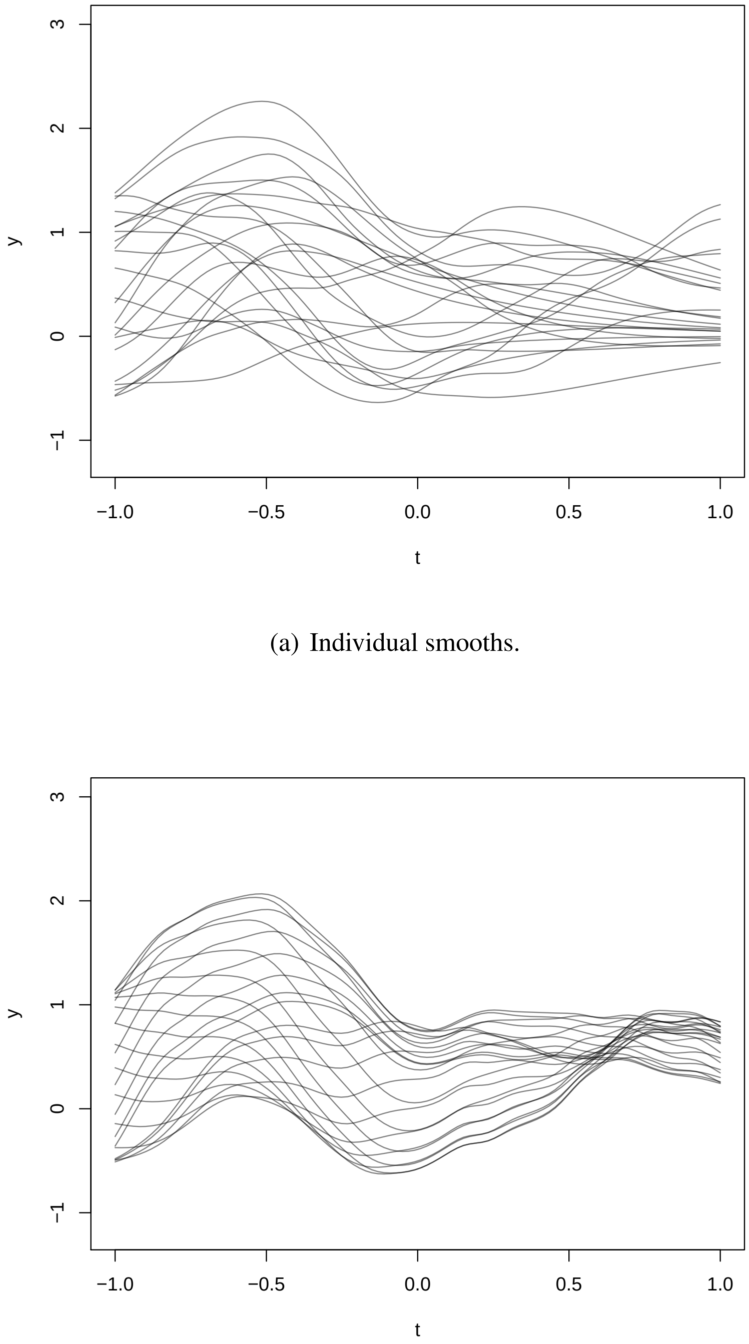


Figure 4.3: Interpolations of the synthetic data used to test the Gibbs algorithm are plotted here with dashed lines; the unobserved climate trends are plotted with solid lines. Two pairs of observed output and unobserved climate trend have been picked out arbitrarily and coloured red and green to better illustrate the nature and size of the weather component.



(a) Individual smooths.

(b) Smooths from the Gibbs sampler.

Figure 4.4: Subfigure 4.4(a) shows the individual smooths of example 4.2.1's synthetic data, these are the expectations for each $c(x_i, t)$ given only $y(x_i, t_j)$ for $j = I_i$. Subfigure 4.4(b) shows the climate trends calculated by averaging over the post burn-in iterations of the Gibbs sampler that targets the joint smooth.

we also look ahead to the calibration stage in which coefficient vectors corresponding to simulations with inputs close to x^* will have greater influence on our calibration inferences than those that are far away. In an effort to recognise the role of the variable variance, we can choose to modify the Gibbs sampler, randomising the order of the updates and setting visitation probabilities, which we collect together and denote α, according to

$$[\alpha]_i \propto \mathrm{Var}\left([\boldsymbol{\beta}]_{i,\cdot} \mid [\boldsymbol{\beta}]_{-i,\cdot}, \mathbf{H}, y([\mathbf{T}]_{I(i)}, [\mathbf{X}]_{i,\cdot})\right), \tag{4.46}$$

or to recognise the differing importance of each simulation to the calibration procedure we may choose to set the visitation probabilities according to an estimate of the posterior for x^*, which we write as

$$[\alpha]_i \underset{\sim}{\propto} \pi_{x^*}([\mathbf{X}]_{i,\cdot}). \tag{4.47}$$

We may also choose to invest in the calculation of weights that are less ad hoc, with theoretical optimality properties in terms of either the sampler's convergence speed or the asymptotic variance of the estimates it produces. Interesting work informing these calculations has been produced by Levine[28]. However, in preliminary experiments following on from example 4.2.2 we detect almost no advantage to using the modified visitation probabilities (4.46) and (4.47). We suspect that this fact may be related to the very small number of simulations here. When there are a large number of simulations and the vast majority are mostly left out of the algorithm's visitation schedule this finding may be reversed, but for now we refrain from engaging in a thorough investigation of visitation probabilities.

4.2.2 Calibration with the NIW emulator

Our strategy for inferring the value of an input, x^*, given an observation of the corresponding output, $y^* = y(x^*)$, involves discretising the input space to a finite set of N_{x^*} points whose coordinates we store in the $N_{x^*} \times p$ matrix $\mathbf{P}$. The values of the climate basis coefficients at these coordinates, conditional on $\mathbf{H}$ and the coefficients of the simulated climates, are normally distributed, and we create the $N_{x^*} \times q$ matrix, $\boldsymbol{\beta}'$, to store their values. This means that the likelihood, which we would use to adjust a prior over the set of possible x^* values given $\boldsymbol{\beta}$, is calculated using a multivariate normal density.

Since the climate basis coefficients for the simulated series are unknown, the likelihood we calculate in practice is formed by taking an average over the Gibbs sampler iterations. So our calibration calculation requires that we insert algorithm 2 as a subroutine immediately after line 15 of algorithm 1, and alter the final line to return $\bar{l}$ along with $\bar{\beta}$.

To describe algorithm 2 we also need to introduce some additional notation. Firstly, we need to define matrices **G** and **J** whose elements encode variances and covariances for basis coefficients at different points in the input space, in the same way the matrix **K** does,

$$[\mathbf{G}]_{i,i}\mathbf{H} = \mathrm{Var}\left([\boldsymbol{\beta}']_{i,\cdot} \mid \mathbf{H}\right), \qquad [\mathbf{J}]_{i,j}\mathbf{H} = \mathrm{Cov}\Big([\boldsymbol{\beta}']_{i,\cdot}\,,\ [\boldsymbol{\beta}]_{j,\cdot} \mid \mathbf{H}\Big).$$

Secondly, we define

$$\pi_{MVN}(y; \mu, \Sigma)$$

to be the value of the multivariate normal density function, with mean μ and variance Σ, at y.

Algorithm 2 NIW Gibbs sampler calibration subroutine

```
for $k = 1, \ldots, N_{x^*}$ do
    $\mu_{\beta^*} \leftarrow \mathbb{E}(\boldsymbol{\beta}) + [\mathbf{J}]_{k,\cdot}\mathbf{K}^{-1}(\boldsymbol{\beta} - \mathbb{E}(\boldsymbol{\beta}))$
    $\Sigma_{\beta^*} \leftarrow ([\mathbf{G}]_{k,k} - [\mathbf{J}]_{k,\cdot}\mathbf{K}^{-1}[\mathbf{J}]_{k,\cdot}^{T}) \times \mathbf{H}$
    $\mu_{y^*} \leftarrow \boldsymbol{\phi}\mu_{\beta^*}$
    $\Sigma_{y^*} \leftarrow \boldsymbol{\phi}\Sigma_{\beta^*}\boldsymbol{\phi}^{T} + \mathbf{K}_{w}$
    $[l]_k \leftarrow \pi_{MVN}(y^*; \mu_{y^*}, \Sigma_{y^*})$
end for
$\bar{l} \leftarrow ((i - N_{burn})\bar{l} + l)/(i - N_{burn} + 1)$
```

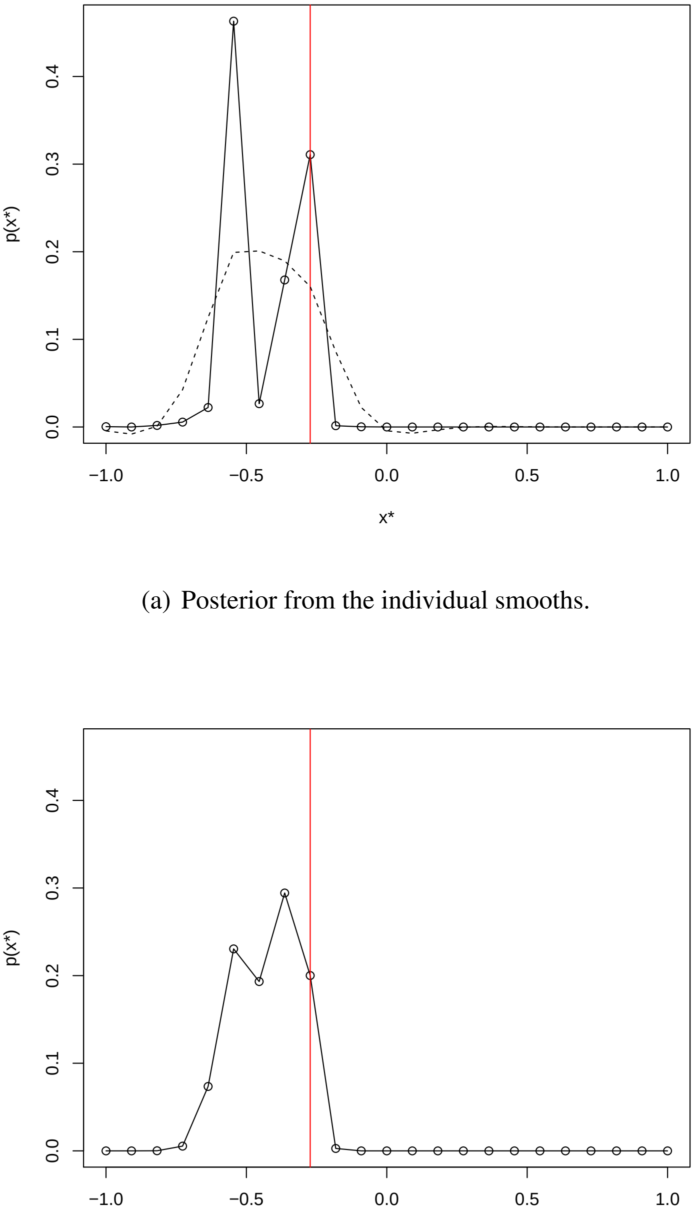
Example 4.2.2 (A synthetic example of calibrating with the NIW emulator)**.** In the following experiment, which continues directly from example 4.2.1, we choose to specify the matrix, **P**, of candidate x^* coordinates to coincide with the matrix, **X**, of simulation coordinates. This is partly for convenience, since it simplifies the calculations of algorithm 2, and partly so that the posterior inferences from the Gibbs procedure may be compared more easily to those in which the simulations are considered individually.

So we choose one of the N_x inputs to be the true system input parameter. To its corresponding true coefficient vector, β^*, we add a new simulated weather term that is independent of all other quantities, with the implication that the system and simulator climate functions are the same whereas their weather functions are independent. The result is a new signal, y^*, that we treat as a time series of system observations. Note that we are not incorporating a simulator discrepancy term for the climate signal here in order not to distract from the basic mechanisms of the calibration calculation.

Figure 4.5(b) shows the curve that interpolates the approximate posterior values for x^*, resulting from the specification of equal prior probabilities on the N_x input coordinates and likelihood calculated with the Gibbs sampler algorithm. Figure 4.5(a) meanwhile shows the equivalent posterior for the case in which a likelihood for y^* is calculated simply by considering simulations individually and not allowing the matrix $\mathbf{H}$ to vary from its prior mean.

To be explicit, these individual likelihoods are computed by setting the prior variance matrix for the basis coefficients to the prior mean value of the inverse Wishart distribution used in the Gibbs calculation. Conditional on this matrix, the moments for the coefficients of an individual simulated climate are adjusted only by the corresponding simulator outputs. The mean and variance for the coefficients then define a multivariate normal density, in the same manner as in lines 4-6 of algorithm 2, which is used as a likelihood for the location of x^*.

As with example 3.1.2, our conclusions regarding the adequacy of the calibration resulting from the individual and joint smoothing procedures are mixed. The likelihood calculations in which the series are considered individually can be performed extremely quickly; they are effectively instant in comparison to the joint calculation. In this example, however, there is clearly structure in the posterior that is degraded when we consider the smooths individually. The sharp valley in figure 4.5(a), for example, arguably represents an inappropriate inference that could steer our attention from an input worthy of further investigation. The broader characteristics, specifically, the posterior mean and variance are well approximated by the likelihood based on the individual smooths however. This notion is demonstrated when we smooth the likelihood arising from the individual smooths to produce a curve not dissimilar to the likelihood calculated using the joint smooth.



(a) Posterior from the individual smooths.

(b) Posterior from the Gibbs sampler.

Figure 4.5: In subfigure 4.5(a) we plot the posterior for x^* arising from equal prior probability masses on the simulation inputs and the likelihoods of the system time series given the individual smooths of the simulator data. Also plotted here, with a dashed line, is a smooth of this posterior. In subfigure 4.5(b) we plot the posterior arising from the same prior and the likelihood calculated using the Gibbs sampler.

4.2.3 Concluding notes on the NIW emulator

The NIW emulator represents an intermediate step in our development of a suitable emulator for simulated and system time series. It is not a model we will advance further in this thesis, mainly because of the Gibbs sampler's computational demands, but it does serve to highlight several important concepts that influence the rest of our work. It shows us how basis functions help us to deal with large quantities of missing data as well as large quantities of observed data, and how parameterising a curve by its basis function coefficients can lead to a convenient method for inferring covariance parameters. It shows us how we can use a Gibbs sampling argument to formalise the process of returning to time series and re-smoothing them with different covariance parameters. And it shows us that we might be able to prioritise these return visits in the light of system data.

4.3 Nyström-basis emulators

In this section we look at a significantly different approach to modelling the simulator and the system. With these models we can also deal with ragged arrays of training data, but the primary motivation for the work here is the computational cost usually associated with modelling in high dimensions. With the models that we will introduce, we leave the grid structure characterising examples 2.3.2 and 3.1.2 behind. In doing so we can save on computation by drastically cutting down on the size of the arrays we need to handle. We find, however, that a consequence of leaving the grid structure is that we will have to abandon the inverse Wishart mechanism for learning about variances and the algebraic shortcuts that rely on the factorisation of certain variance matrices.

We start in section 4.3.1 by introducing the idea of an optimal, and therefore minimal, basis for a random field. The optimality of the basis arises from the way it focuses on both a specific part of the input space, as defined by a prior for x^*, and on a specific part of a space of functions, as defined by a covariance function for the simulator output. The name 'Nyström-basis emulator' is inherited from the 'Nyström method' for approximating the solution to the eigenfunction problem to which the optimality condition relates. We move on to describe the conventional method for the estimation of such a basis, and then further, to describe a novel iterative estimation strategy that gathers the basis's degrees of freedom to the region of the input space defined by an estimate of the posterior for x^*. This focusing technique is inspired by the weighted re-visiting strategy we saw in section 4.2.1. Finally, in section 4.4, we develop a particularly parsimonious linear model tailored to the input locations of the available simulations and the covariance function of the simulator output.

We will find, as we did in section 3, that by explicitly including the time variable t or, more precisely, by distinguishing it from the input parameter x we introduce notational clutter that obscures the mathematical structure of the expressions justifying and explaining our basis approximations. For this reason, in the following sections, we will often think about simulators and systems with a scalar output, and single vector input formed by concatenating the input variables with the output index, time. This combined input vector is denoted

$$\xi^T = (t, x^T),$$

while the matrix formed by stacking such vectors as rows is denoted Ξ. We will, however, sometimes need to unfold ξ back into (t, x) to make certain expressions required in chapter 5 more explicit.

4.3.1 Approximately optimal basis functions

Despite initial enthusiasm for the FDA approach, we have found it very difficult to anticipate the consequences of a choice of basis, and a penalty or variance specification for each component, for the covariance function to which they lead. This is unsatisfactory because our intuition for the covariance function is, more often than not, more highly developed than that for its components. It is, for example, easier to specify confidently the mean, variance and approximate correlation decay length for a field than the expected size of the linear, quadratic and cubic components in its expansion. It seems almost perverse to create another inverse problem whereby we attempt to construct pseudo-mechanistic rules, in the form of roughness penalties, that result in the smoothness properties we want to encourage as part of our prior specifications. So when data are not plentiful, and our priors are important to our posterior inferences, we reject the bottom-up approach to constructing the emulator from a set of basis functions unless there is a significant theoretical underpinning to a particular penalised or conserved quantity. Still, the basis expansion of a field is vital for taking control of the size of the inference calculations. Our preferred strategy is a top-down approach whereby approximate basis functions are constructed from precisely specified covariance functions.

Whether our strategy is top-down or bottom-up, the construction of a basis for a space of functions of many variables has the potential to lead to an overwhelmingly large number of terms. The problem tends to arise from the tensoring of low-dimensional objects, either grids or basis functions for example, to create higher-dimensional ones. When we do this, the resulting objects are equivalent to rectangular grids. As the dimension of the grid increases, the objects near the corners tend to become less important while the proportion of grid points near the corners grows. Understanding of this rather vague assertion can be furnished by examining the following examples.

4.3.2 Corners of high-dimensional cubes

4.3.2.1 Corners of probability distributions

In this first example, which is particularly relevant when thinking about the placement of nodes for the approximation of integrals, we look at the distribution of mass described by a probability density function as it is generalized to increasingly high dimensional settings.

Consider a hypercube, denoted C, symmetric about each axis, and containing 0.95 of the mass of a unit multivariate normal distribution. We then introduce the sphere, $\mathcal{S}$, bounded by the hypercube, pressed up against its faces. We can calculate the volume, V, and probability mass, M, inside the cube and the sphere as

$$\begin{aligned} V(C) &= (2r)^D, & M(C) &= 0.95, \\ V(\mathcal{S}) &= \frac{(\sqrt{\pi}r)^D}{\Gamma(D/2+1)}, & M(\mathcal{S}) &= P(\chi^2_D < r^2), \end{aligned}$$

where $P(\chi^2_D < r^2)$ refers to the chi-squared cumulative distribution function with D degrees of freedom and r is calculated using the inverse cumulative distribution function of a unit normal variable,

$$r = \Phi^{-1}(0.975).$$

Plots 4.6(a) and 4.6(b) show the quantities

$$\frac{M(\mathcal{S})}{M(C)} \tag{4.48}$$

and

$$\frac{V(\mathcal{S})}{V(C)}, \tag{4.49}$$

which describe the probability mass and the volume within the hypersphere as a fraction of that contained within the hypercube. It is clear here that the proportion of the volume within the sphere, away from the corners, decreases rapidly as the dimension increases, much faster than the rate at which its relative mass decreases. The result is that in high dimensions the corners of the cube take up more space and constitute more of the mass, but the relative density in the corners decreases. It is in this sense that the corners become more of a burden and less important. The notion is illustrated in figure 4.6(c), which plots the log ratio of the average density within the sphere and within the cube as measured by

$$\log\left[\frac{M(\mathcal{S})/V(\mathcal{S})}{M(C)/V(C)}\right]. \tag{4.50}$$

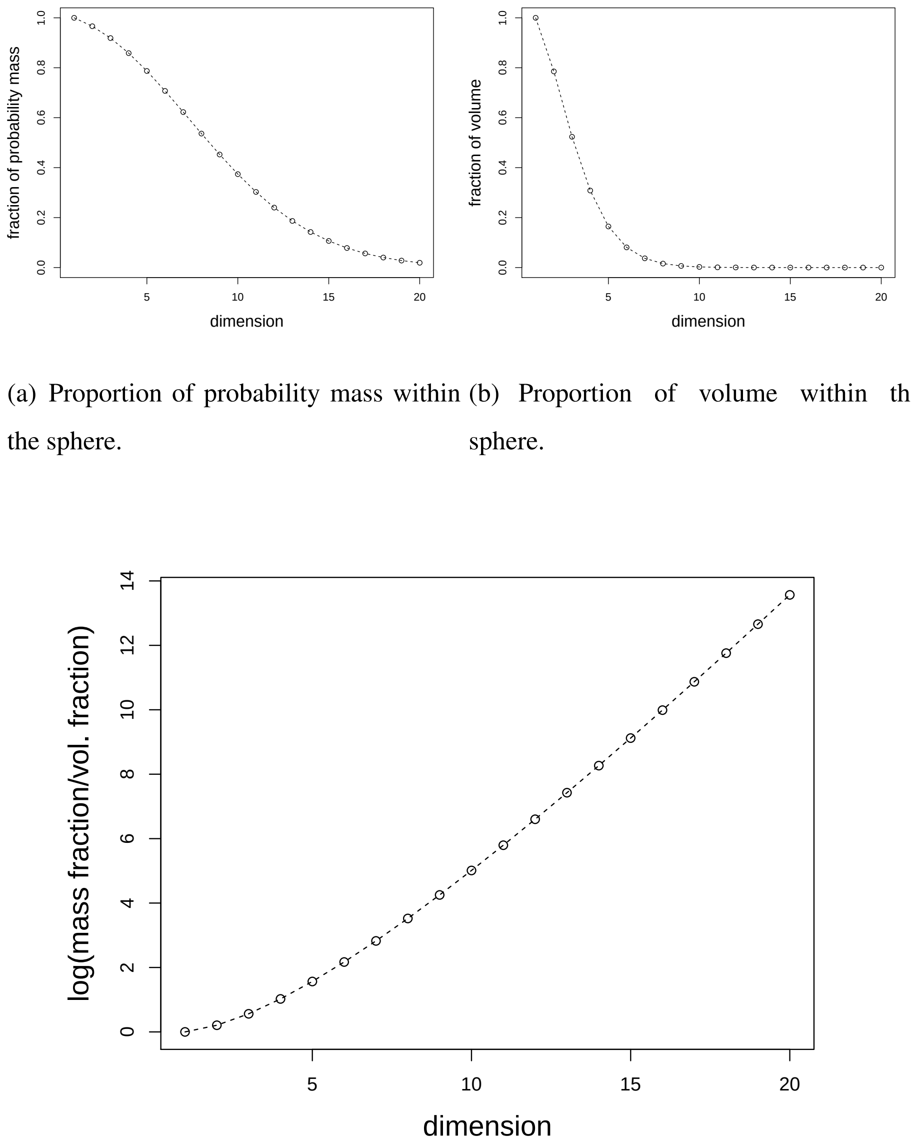
4.3.2.2 Corners of arrays of basis functions

In this example we demonstrate how the same high-dimensional phenomenon described in section 4.3.2.1 is relevant to discrete sets, namely sets of basis functions, as well as continuous sets of real numbers. Here the analogue of the probability density that gives higher weight to points near the centre of the distribution is a covariance specification that attributes more variance to smoother basis functions. The analogue of a region's volume is the cardinality of a set of basis functions and the analogue of its mass is the variance attributable to that set.

Let us consider variable selection for a multiple linear regression problem using the model

$$y = \beta^T \phi(\xi) \oplus e = \sum_{i\in\Omega} [\beta]_i [\phi(\xi)]_i \oplus e. \tag{4.51}$$

We define $\xi = (\xi_1, \xi_2, \ldots, \xi_D)^T \in \Re^D$ to be a D-dimensional input quantity, whose components are each independently identically distributed according to the density, over one dimension, $\pi(\cdot)$.



(a) Proportion of probability mass within the sphere.

(b) Proportion of volume within the sphere.

(c) The logged fraction of average densities for the sphere and hypercube.

Figure 4.6: In subfigures 4.6(a), 4.6(b) and 4.6(c) we present plots of the quantities (4.48), (4.49) and (4.50) respectively as the dimension of the input domain increases.

To construct the multivariate regressors $\phi(\xi)$, we start by considering an ordered set of one-dimensional regressor functions, for which a natural candidate is the set of the first $N+1$ polynomials $\{P_0, P_1, \ldots, P_N\}$ that are orthonormal with respect to $\pi(\cdot)$. This property means that for a scalar input, t,

$$\int_{\Re} P_j(t) P_k(t) \pi(t)\,\mathrm{d}t = \delta_{j,k},$$

so that for the univariate regression problem in which we have zero expectations for all the regression coefficients, the variation in the output quantity, y, attributable to regressor j is equal to the variance of the jth coefficient and is uncorrelated to that attributable to any other regressor.

The independence of the components of the input quantity ξ means that these properties are inherited by the multiple regression model that employs as regressors products of the univariate regressor functions. The expectations for regression coefficients and the summands of (4.51) are then given by

$$\mathbb{E}\left([\beta]_i\right) = 0 \qquad \text{for } i \in \Omega, \tag{4.52}$$

$$[\phi(\xi)]_i = P_{i_1}(\xi_1) P_{i_2}(\xi_2) \ldots P_{i_D}(\xi_D), \tag{4.53}$$

$$\mathrm{Var}\left([\beta]_i [\phi(\xi)]_i\right) = \mathrm{Var}\left([\beta]_i\right), \tag{4.54}$$

$$\mathrm{Cov}\Big([\beta]_i [\phi(\xi)]_i\ ,\ [\beta]_j [\phi(\xi)]_j\Big) = 0 \qquad \text{for } i \neq j, \tag{4.55}$$

where the expectations implicit to the variance expressions are over both the values of the regressor coefficients and the input quantity, and where the un-subscripted indices i and j label coefficients, multivariate regressors and their one-dimensional components via the ordered vectors of the sub-indices $(i_1, \ldots, i_D)$ and $(j_1, \ldots, j_D)$.

The full tensored set of N regressors from each of the D dimensions, which we call C, for cube, may be written as

$$C = \{P_{i_1}(\xi_1) P_{i_2}(\xi_2) \ldots P_{i_D}(\xi_D) \mid i_1, i_2, \ldots, i_D \geq 0,\ \max(i_1, i_2, \ldots, i_D) \leq N\}.$$

The set $\mathcal{S}$, for simplex, includes interactions only up to a combined order of N and can be written as

$$\mathcal{S} = \{P_{i_1}(\xi_1) P_{i_2}(\xi_2) \ldots P_{i_D}(\xi_D) \mid i_1, i_2, \ldots, i_D \geq 0,\ i_1 + i_2 + \ldots + i_D \leq N\}.$$

Now suppose that the variance specification for the regressor coefficients and the error term is,

$$\mathrm{Var}\,(\beta_i) = \rho^{\sum_{d=1}^{D} i_d}, \qquad \mathrm{Var}\,(e) = \sigma^2,$$

where ρ is a constant strictly between zero and one. The specification means that higher order polynomial trends, which we may understand as being rougher, contribute less to the function y. It follows that the set sizes, denoted $V(\cdot)$ for volume, and the sums of variances for y attributable to the included regressors, denoted $M(\cdot)$ for mass, are given by

$$V(C) = (1+N)^D, \qquad M(C) = \left(\frac{1-\rho^N}{1-\rho}\right)^D, \tag{4.56}$$

$$V(\mathcal{S}) = \binom{N+D}{D}, \qquad M(\mathcal{S}) = \sum_{k=0}^{N} \binom{k+D-1}{D-1}\rho^k. \tag{4.57}$$

Expressions (4.56), for the full set of regressors, are derived simply from raising the grid length and the formula for a geometric series to the power of D. Expressions (4.57) are derived by noticing that there are $\binom{k+D-1}{D-1}$ combinations of D indices that sum to k, and from the identity

$$\sum_{k=0}^{N} \binom{k+D-1}{D-1} = \binom{N+D}{D}.$$

For the sake of example we set $\rho = 0.5$, $N = 6$ and show, in figure 4.7(a), the values of

$$\frac{M(\mathcal{S})}{M(C)}, \tag{4.58}$$

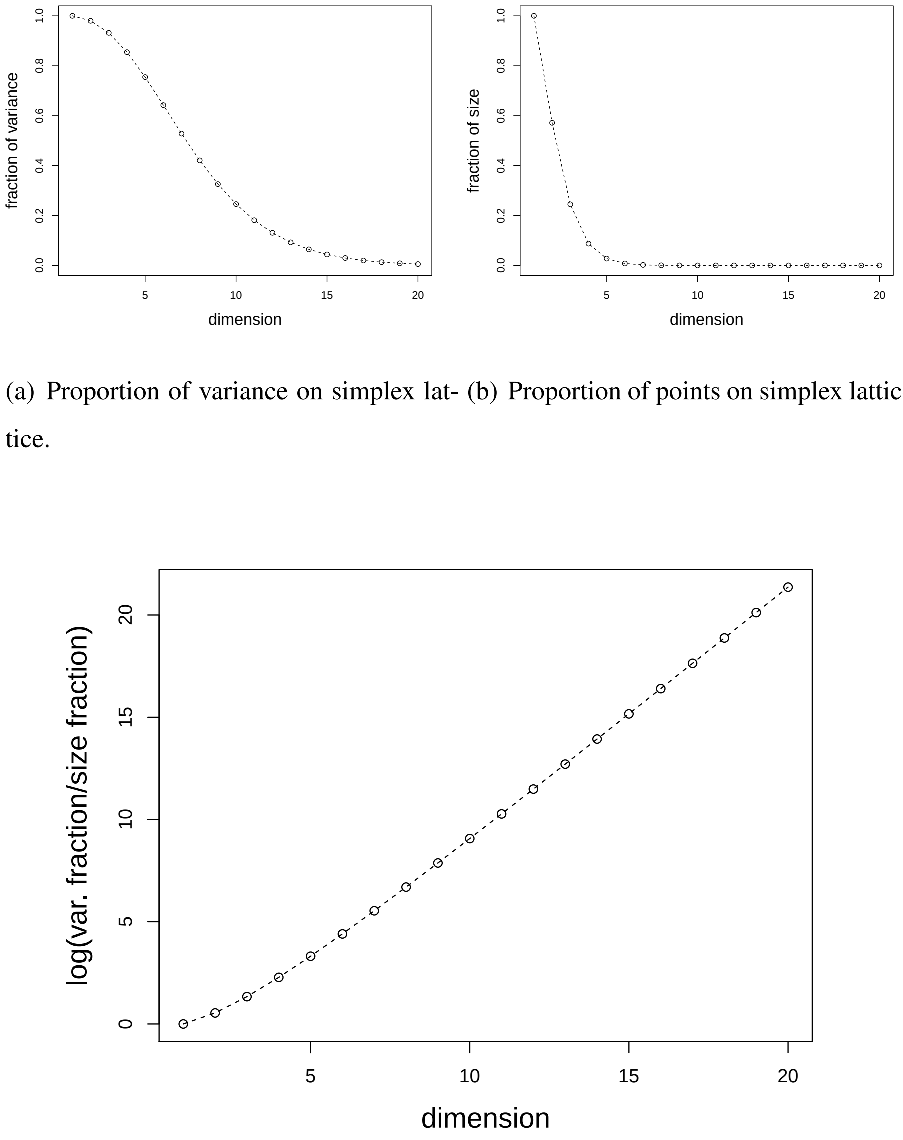
as the dimension of the objects increases, and in figure 4.7(b) we show

$$\frac{V(\mathcal{S})}{V(C)}, \tag{4.59}$$

the cardinality of $\mathcal{S}$ over that of C. Then, in figure 4.7(c) we illustrate the ratio of these two quantities by plotting

$$\log\left[\frac{M(\mathcal{S})/V(\mathcal{S})}{M(C)/V(C)}\right]. \tag{4.60}$$

The point that we make here is that by retaining only lower-order interaction terms we discard most of the full set's regressor functions while, because of the variance specification, preserving a disproportionately large amount of the field's variance. Figure 4.7(c)



(a) Proportion of variance on simplex lattice.

(b) Proportion of points on simplex lattice.

(c) The logged fraction of variance and set size for the simplex lattice.

Figure 4.7: In subfigures 4.7(a), 4.7(b) and 4.7(c) we present plots of the quantities (4.58), (4.59) and (4.60), respectively, as the dimension of the input domain increases.

demonstrates how this point becomes more important as the dimension of the regression increases, by plotting quantity (4.60), which is interpretable as a logged fraction of variance densities for the two sets.

Shaving off the corners of fully tensored objects thus has the potential to save a large number of degrees of freedom, and a large amount of computational demand. It is the secret to the success of sparse grid methods for numerical integration and their ability to mitigate the curse of dimensionality. A comprehensive introduction to sparse grids is given in [8] while a paper with more of an emphasis on their application to integrating over likelihoods is available in [20].

For simulators like FAMOUS, producing an ensemble of around one thousand simulations requires several weeks, and this is after a potentially longer wait for the computing resources to become available. Because of this computational cost, and due to simulations crashing, we are unlikely to be able to produce full grids of simulator data. Similarly, exploratory emulation with a basis of more than one thousand members, which results from tensoring four-member univariate bases from five dimensions for example, is likely to render emulation calculations slow. The emulator still ought to be orders of magnitude faster than the simulator, but it may become frustratingly time consuming as we refit it thousands of times under different covariance specifications, and without full grids of simulated data we cannot speed the calculations up with the methods we derived in section 2.3. We therefore commit our further research to emulation techniques that avoid dependencies on tensored designs and tensored bases.

Both examples of grids in this subsection are relevant to us because our interest lies in a function that is localised in a space of functions, because the climate is believed to be smooth, and then we are interested only in its values in certain regions of the input space, because often much of a simulator's input space leads to implausible or non-physical outputs.

4.3.2.3 An optimal basis

We continue our progress towards the proposal of a basis that recognises the relative redundancies in fully tensored objects by stating and proving an optimality result. The result is phrased as a theorem and a proof, which defers several mathematical details whose complete exposition would necessitate the introduction of a significant amount of additional theory. The theorem is essentially a version of the Karhunen-Loève theorem, whose implications reach from the classical work of Karl Pearson [37] on principal component analysis, to the contemporary work of Xiu[61] on polynomial chaos (PC). This latter field is especially interesting to us since it has been driven mostly by the desire of the applied mathematics community to propagate input uncertainty through simulators for physical systems. In this context a PC expansion is used to provide a basis for an approximation to a simulator's output given a particular distribution for the input variable. A more detailed discussion of the relationship between PC and methods more familiar to statisticians is provided in [34], but we note that while the orthogonal polynomial bases of the PC method are employed with stability, convenience and efficiency in mind, they are not derived from optimality arguments and covariance specifications for the simulator output, and they do not scale naturally to high-dimensional fields unless we are careful to select only certain tensored polynomials as discussed in section 4.3.2.2.

Before launching into the theorem we ought to provide a little mathematical background to the objects and deferred details the theorem involves. Firstly, we need to define what we mean by an integral with respect to a real function with domain Ω: we adopt the squared-bracket notation,

$$\int_H \mathcal{L}(f)\,\mathrm{D}\,[f] = \int_{\Re} \cdots \int_{\Re} \mathcal{L}(f) \prod_{\xi\in\Omega} df(\xi),$$

to denote the integral of a functional $\mathcal{L}$ with respect to function f over a set of functions labelled H. We thus think of the functional integral as an integration over an infinite-dimensional vector space. Secondly, we need to talk about the set of functions we integrate and optimise over in the theorem: we rely on the theory for reproducing kernel Hilbert spaces, specifically the Riesz representation theorem, to associate an autocovariance function, $k(\cdot,\cdot)$, with a unique space of functions H. This space contains functions f that are smooth in the sense that

$$f(\xi') = \int_\Omega k(\xi',\xi'')f(\xi'')\,\mathrm{d}\xi'' \qquad\qquad \forall f \in H,$$

with the implication that the functions' values coincide with their local averages as defined by the kernel. For a more formal introduction to reproducing kernel Hilbert spaces we recommend consulting [58].

Let us imagine that we will be given the values of climate function $c(\xi, \hat{v})$ at all ξ and a single $\hat{v}$, and that we will then be asked to make a prediction for $c(\xi^*, \hat{v})$, given ξ^*, using a finite number of basis functions and an error term. We now consider the problem of choosing the basis functions and the error variance before either of these events, and respond to it with theorem 4.3.1.

We find that our understanding of the imagined problem is facilitated by considering the metaphorical situation in which we will be shown a complete image of an object and asked to sketch a copy of it. The image will then be withdrawn and we will be required to describe the object by referring to our sketch. The prior for the function input and the covariance function for its output correspond to clues for the type of object we will be shown: a machine, an animal or a landscape, for example. The choice of basis corresponds to the selection of tools we can use for the sketch such as a pencil, paintbrush, or ruler. The basis coefficient moments relate to how often, and in which combinations, we will need to use our tools. And the regression model's independent error terms relate to the loss of accuracy we anticipate that our sketch will result in. This problem is one step removed from the emulation problem because in practice we will not be given the simulator climate function, corresponding to the complete image, to learn from; we will only be shown distorted glimpses of it via observations of the output, which has been contaminated with the weather signal.

Theorem 4.3.1. *The first N eigenfunctions of the operator T,*

$$T[f](\xi') = \int k_{c\xi}(\xi', \xi'')\pi_{\xi^*}(\xi'')f(\xi'')\,\mathrm{d}\xi'', \tag{4.61}$$

represent an optimal finite basis for linear regression, in the sense of minimising over choices of basis whose members are in H, the expected squared loss,

$$\mathcal{L} = \mathbb{E}\left((c(\xi) - \beta^T \phi(\xi))^2\right), \tag{4.62}$$

where $\phi(\xi)$ is a column vector of basis function values at ξ; π_{ξ^*} is the prior distribution for the input value, ξ^*, at which we must make a prediction; $k_{c\xi}(\cdot,\cdot)$ is the kernel that defines our covariance specification for the zero mean field $c(\cdot)$; β is a vector of coefficients independent of ξ^* and $c(\xi^*)$; and the expectation in (4.62) *is taken over both a finite vector space, Ω, for possible values of ξ, and the reproducing kernel Hilbert space H defined by the kernel $k_{c\xi}$, for possible values of $c(\cdot)$.*

Proof. By differentiating inside the expectation and solving for zero we can derive the regression coefficients, for known regressors and a specific instantiation of c, minimizing the expected loss. These are:

$$\hat{\beta} = \mathbb{E}\left(\phi(\xi)\phi(\xi)^T\right)^{-1}\mathbb{E}\left(\phi(\xi)c(\xi)\right) = \left(\int \phi(\xi)\phi(\xi)^T\pi_{\xi^*}(\xi)\,\mathrm{d}\xi\right)^{-1}\left(\int \phi(\xi)c(\xi)\pi_{\xi^*}(\xi)\,\mathrm{d}\xi\right), \tag{4.63}$$

which are analogues of the classical least squares estimates with sums over observed values replaced by expectations, which we write as integrals weighted by the prior distribution. Given these coefficients, the loss to be minimised is

$$\mathcal{L} = \int_H\int_\Omega \left(c(\xi)^2 - 2c(\xi)\hat{\beta}^T\phi(\xi) + \hat{\beta}^T\phi(\xi)\phi(\xi)^T\hat{\beta}\right)\pi_{\xi^*}(\xi)\,\mathrm{d}\xi\;[Dc], \tag{4.64}$$

$$= \int_H\int_\Omega c(\xi)^2\pi_{\xi^*}(\xi)\,\mathrm{d}\xi\;\mathrm{D}\left[c\right] - \int_H \hat{\beta}^T\left(\int_\Omega \phi(\xi)\phi(\xi)^T\pi_{\xi^*}(\xi)\,\mathrm{d}\xi\right)\hat{\beta}\;\mathrm{D}\left[c\right]. \tag{4.65}$$

The first term does not depend on the regressors, so we concentrate on the second term and seek to maximise

$$\mathcal{J} = \int_H \hat{\beta}^T\left(\int_\Omega \phi(\xi)\phi(\xi)^T\pi_{\xi^*}(\xi)\,\mathrm{d}\xi\right)\hat{\beta}\,\mathrm{D}\left[c\right],$$

which represents an expression for the variation in c attributable to the regressors. At this point we introduce the eigenfunctions u_i satisfying

$$\int_\Omega k_{c\xi}(\xi,\xi')\pi_{\xi^*}(\xi')u_i(\xi')\,\mathrm{d}\xi' = \lambda_i u_i(\xi), \tag{4.66}$$

$$\int_\Omega u_i(\xi')u_j(\xi')\pi_{\xi^*}(\xi')\,\mathrm{d}\xi' = \delta_{i,j}, \tag{4.67}$$

$$\lambda_0 \geq \lambda_1 \geq \ldots \geq 0. \tag{4.68}$$

The non-negativity and summability of the eigenvalues, which we take as given, follows via Mercer's theorem, as long as we restrict $k_{c\xi}$ to be a continuous symmetric non-negative definite kernel, and π_{ξ^*} to be a distribution with moments of all orders. The continuity requirement prevents us specifying c as a white-noise process for example. We also claim that any basis function in H can be described by a linear combination of the eigenfunctions, and we do so while imposing an orthonormality condition such that

$$[\phi(\xi)]_i = \sum_{k=0}^{\infty} d_{ik}u_k(\xi), \tag{4.69}$$

$$\sum_{k=0}^{\infty} d_{ik}d_{jk} = \delta_{i,j}. \tag{4.70}$$

As a consequence,

$$\left(\int_\Omega \phi(\xi)\phi(\xi)^T \pi_{\xi^*}(\xi)\,\mathrm{d}\xi\right) = \mathbf{I},$$

which serves to simplify our calculations by reducing $\mathcal{J}$ and $\hat{\beta}$ to

$$\mathcal{J} = \int_H \hat{\beta}^T\hat{\beta}\,\mathrm{D}\,[c]$$

and

$$\hat{\beta} = \int_\Omega \phi(\xi)c(\xi)\pi_{\xi^*}(\xi)\,\mathrm{d}\xi,$$

So, trading the matrix notation for explicit sums, we can write

$$\begin{aligned}\mathcal{J} &= \int_H \hat{\beta}^T\hat{\beta}\,\mathrm{D}\,[c]\,,\\ &= \int_H \sum_{i=0}^{N}\left(\int_\Omega c(\xi)\sum_{k=0}^{\infty} d_{ik}u_k(\xi)\pi_{\xi^*}(\xi)\,\mathrm{d}\xi\right)\left(\int_\Omega c(\xi')\sum_{l=0}^{\infty} d_{il}u_l(\xi')\pi_{\xi^*}(\xi')\,\mathrm{d}\xi'\right)\mathrm{D}\,[c]\,.\end{aligned}$$

Taking the functional integral inside the sums, we see that it can be replaced by the kernel function,

$$\begin{aligned}\mathcal{J} &= \int_\Omega\int_\Omega \sum_{i=0}^{N}\sum_{k=0}^{\infty}\left(\int_H c(\xi)c(\xi')\,\mathrm{D}\,[c]\right) d_{ik}u_k(\xi)\pi_{\xi^*}(\xi)\,\mathrm{d}\xi \sum_{l=0}^{\infty} d_{il}u_l(\xi')\pi_{\xi^*}(\xi')\,\mathrm{d}\xi',\\ &= \int_\Omega\int_\Omega \sum_{i=0}^{N}\sum_{k=0}^{\infty} k(\xi,\xi')d_{ik}u_k(\xi)\pi_{\xi^*}(\xi)\,\mathrm{d}\xi \sum_{l=0}^{\infty} d_{il}u_l(\xi')\pi_{\xi^*}(\xi')\,\mathrm{d}\xi'.\end{aligned}$$

We then use the eigenfunction property 4.66, which removes explicit reference to the kernel,

$$\begin{aligned}\mathcal{J} &= \int_\Omega \sum_{i=0}^{N}\sum_{k=0}^{\infty}\sum_{l=0}^{\infty} d_{ik}d_{il}\left(\int_\Omega k_{c\xi}(\xi,\xi')u_k(\xi)\pi_{\xi^*}(\xi)\,\mathrm{d}\xi\right)u_l(\xi')\pi_{\xi^*}(\xi')\,\mathrm{d}\xi',\\ &= \int_\Omega \sum_{i=0}^{N}\sum_{k=0}^{\infty}\sum_{l=0}^{\infty} \lambda_k d_{ik}d_{il}u_k(\xi')u_l(\xi')\pi_{\xi^*}(\xi')\,\mathrm{d}\xi',\end{aligned}$$

and orthonormality property 4.67, which removes explicit reference to the eigenfunctions,

$$\begin{aligned}\mathcal{J} &= \sum_{i=0}^{N}\sum_{k=0}^{\infty}\sum_{l=0}^{\infty} \lambda_k d_{ik}d_{il}\left(\int_\Omega u_k(\xi')u_l(\xi')\pi_{\xi^*}(\xi')\,\mathrm{d}\xi'\right),\\ &= \sum_{i=0}^{N}\sum_{k=0}^{\infty}\sum_{l=0}^{\infty} \lambda_k d_{ik}d_{il}\delta_{k,l}.\end{aligned}$$

The Kronecker delta serves to collapse one of the sums to leave equation (4.71), which is a weighted sum of eigenvalues. The weights must sum to one due to (4.70), from which we deduce that $\mathcal{J}$ is maximised when the maximum possible weight, consistent with the orthogonality condition, is allocated to the largest eigenvalues,

$$\mathcal{J} = \sum_{i=0}^{N} \sum_{k=0}^{\infty} \lambda_k d_{ik} d_{ik}, \tag{4.71}$$

$$\leq \sum_{k=0}^{N} \lambda_k. \tag{4.72}$$

The maximal weighting, achieving upper bound (4.72), requires that $d_{ik} = \delta_{i,k}$. As a consequence, all the weight in the eigenfunction expansions of the basis functions, (4.69), is concentrated on individual eigenfunctions. □

Corollary 4.3.2. *In the context of theorem 4.3.1, the minimum expected squared loss achievable with a basis of N elements is*

$$\hat{\mathcal{L}} = \sigma_c^2 - \sum_{j=1}^{N} \lambda_j, \tag{4.73}$$

where $\sigma_c^2 = k(\xi, \xi)$.

The corollary follows from putting the optimal value for $\int_H \hat{\beta}^T \hat{\beta} \, \mathrm{D}[y]$ back into (4.65).

4.3.2.4 The approximate eigenfunction basis

Zhu [63] gives an analytic expression for the eigenfunctions of the squared exponential kernel and a Gaussian prior probability density, but in general such results seem to be very rare. Fortunately, numerical approximation of the eigenfunctions, known as Nyström's method, is a viable alternative. Nyström's method involves approximating an integral with respect to a continuous probability measure by a sum over finitely many points, which we refer to as nodes. We can use it to approximate the solution to the eigenfunction problem:

$$T[u](\xi) = \lambda u(\xi) = \int_\Omega k(\xi, n)\, \pi_{\xi^*}(n)\, u(n)\, \mathrm{d}n \approx \sum_{j=1}^{N} k(\xi, [\mathbf{N}]_{j,\cdot})\, w_j\, u([\mathbf{N}]_{j,\cdot}). \tag{4.74}$$

Notice that we have altered the notation for the integrated variable from ξ in (4.61) to n in (4.74). This is to make way for notation to describe basis approximation nodes. We store the N p-dimensional node coordinates $\mathbf{N}_i$, $i = 1, \ldots, N$ as the rows of the $N \times p$

matrix $\mathbf{N}$. By evaluating (4.74) at the nodes we produce an approximate finite-dimensional eigenvector problem,

$$\lambda u([\mathbf{N}]_{i,\cdot}) \approx \sum_{j=1}^{N} k([\mathbf{N}]_{i,\cdot}, [\mathbf{N}]_{j,\cdot})\ w_j\ u([\mathbf{N}]_{j,\cdot}),$$

which we associate with the exact finite-dimensional (right) eigenvector problem,

$$\mathbf{U\Lambda} = \mathbf{KWU}, \qquad\qquad \mathbf{U}^T\mathbf{WU} = \mathbf{I}, \tag{4.75}$$

where $[\mathbf{K}]_{i,j} = k([\mathbf{N}]_{i,\cdot}, [\mathbf{N}]_{j,\cdot})$ are elements of the variance matrix for the climate values at the basis nodes; $[\mathbf{W}]_{i,j} = \delta_{i,j}w_j$ are the elements of a diagonal matrix containing the integration weights; $\mathbf{U}$ is the matrix whose jth column contains the jth eigenvector of $\mathbf{KW}$; and $\mathbf{\Lambda}$ is the diagonal matrix containing the eigenvalues of $\mathbf{KW}$.

Given $\mathbf{K}$ and $\mathbf{W}$, we solve 4.75 numerically to produce $\mathbf{U}$ whose i, jth element we equate with the value of the jth eigenfunction of T at node i, and $\mathbf{\Lambda}$ whose jth diagonal element we equate with the jth eigenvalue of T. Feeding the normalised right-eigenvector solutions to (4.75) back into (4.74) we see that our eigenfunction approximations are linear combinations of covariance kernels centred on the nodes,

$$\phi_m(\xi) \approx [\mathbf{\Lambda}]^{-1}_{m,m} \sum_{j=1}^{N} k(\xi, [\mathbf{N}]_{j,\cdot})\ [\mathbf{W}]_{j,j}\ [\mathbf{U}]_{j,m},$$

and that our approximate eigenvalues are the eigenvalues of $\mathbf{KW}$.

With the Nyström method we are free to specify any valid covariance function; there is no longer an advantage to specifying one that is factorisable, as there was in examples 2.3.2 and 3.1.2. We may, for example, employ radial isotropic kernels which allow for correlation lengths that are not aligned to the input axes.

In regard to computation, we note that it is in fact preferable to work with the symmetrised version of the eigen-decomposition as it can be computed more quickly. The symmetrised problem has the same eigenvalues as the original, and eigenvectors for each are easily derivable from the other via the multiplication of a diagonal matrix. These relationships are clear upon noticing that the defining equation for the eigenvectors of the symmetrised problem,

$$\mathbf{W}^{1/2}\mathbf{K}\mathbf{W}^{1/2}\tilde{\mathbf{U}} = \tilde{\mathbf{U}}\tilde{\mathbf{\Lambda}}, \tag{4.76}$$

can be rearranged to reveal the equation for the asymmetric version,

$$\mathbf{KW}(\mathbf{W}^{-1/2}\tilde{\mathbf{U}}) = (\mathbf{W}^{-1/2}\tilde{\mathbf{U}})\tilde{\mathbf{\Lambda}},$$
$$\mathbf{KWU} = \mathbf{U\Lambda},$$

so that,

$$\mathbf{U} = \mathbf{W}^{-1/2}\tilde{\mathbf{U}}, \qquad \text{and} \qquad \mathbf{\Lambda} = \tilde{\mathbf{\Lambda}}.$$

As for computing the Nyström approximation in practice, a sparse Gaussian quadrature rule whose weighting function corresponds to the prior for ξ^* would appear to be a natural choice for the approximation in (4.74). However, the rule's weight matrix may lead to negative eigenvalues, corresponding to negative variances for basis coefficients, to which an appropriate response is not obvious. Sobol sequences or other quasi-random space-filling designs represent valuable alternatives, and can be modified to reflect a range of priors if we transform sequences based on the unit uniform distribution with the quantile function of those priors.

4.3.2.5 Applying the approximate eigenfunction basis

Firstly, we need to confirm exactly what our finite basis approximate model is targeting. For the 'full' or 'target' model, we define the physical quantity of interest to be the sum of independent, weakly stationary climate and weather components,

$$y(\xi, v) = c(\xi, v) \oplus w(\xi, v), \tag{4.77}$$

and assume that the prior mean values have been removed so that,

$$\mathbb{E}\left(c(\xi, v)\right) = 0, \qquad \mathbb{E}\left(w(\xi, v)\right) = 0. \tag{4.78}$$

We then specify covariances between the component values using autocovariance functions that are factorisable as follows:

$$\mathrm{Cov}\left(c(\xi', v')\,,\; c(\xi'', v'')\right) = k_{c\xi}(\xi', \xi')k_{cv}(v', v''), \tag{4.79}$$
$$\mathrm{Cov}\left(w(t', x', v')\,,\; w(t'', x'', v'')\right) = k_{wt}(t', t'')k_{wx}(x', x'')k_{wv}(v', v''). \tag{4.80}$$

Our intention here is not to enforce factorisability on the autocovariance function for the climate over the ξ space in order to produce a more flexible model, one whose covariance parameters can be more comprehensively tuned according to the time and input dependencies that we observe in the simulated data. The factorisability of the contribution to (4.79) from separations in the discrepancy space is imposed in anticipation of the fact that, for the time being, we do not plan on observing many different simulators spread over the discrepancy space. Therefore, the added flexibility afforded by discarding this particular factorisability property will not be utilised and would only serve to complicate the model. The full factorisability of the weather's autocovariance is also specified to reduce the model's complexity, but also by the experience, discussed in section 5.1.1, that the weather terms of climate simulators are particularly sensitive to the input parameters. So we specify the factorisability primarily in anticipation of setting the input and discrepancy contributions to the Kronecker delta function:

$$k_{wx}(x', x'') = \delta_{x',x''}, \qquad\qquad k_{wv}(v', v'') = \delta_{v',v''}. \tag{4.81}$$

Now, we cite theorem 4.3.1 to motivate the adoption of a basis for describing the climate over the ξ space. We define $\phi(\xi)$ to be an N-dimensional column vector containing the values of the first N eigenfunctions of T, where

$$T[u](\xi') = \int_{\Omega_\xi} k_{c\xi}(\xi', \xi'')\, \pi_{\xi^*}(\xi'')\, u(\xi'')\, \mathrm{d}\xi''$$

and

$$\pi_{\xi^*}(\xi'') = \pi_t(t'')\pi_{x^*}(x''). \tag{4.82}$$

The functions $\pi_{x^*}(\cdot)$ and $\pi_t(\cdot)$ in (4.82) refer to the prior distribution for the true system input parameters, and to a distribution that serves to identify the period during which we are most interested in emulating the simulator and system output.

From here we define a linear regression model whose true coefficient values, described by (4.63), would minimise the expected squared prediction error, in the sense given by (4.62), given complete knowledge of a particular climate function $c(\xi, v)$ over a subspace formed by holding v fixed. We write the approximation as

$$y(\xi, v) \approx \breve{c}(\xi, v) \oplus w(\xi, v), \tag{4.83}$$

$$\breve{c}(\xi, v) = \beta(v)^T \phi(\xi) \oplus e_c(\xi, v). \tag{4.84}$$

We specify that basis coefficients relating to different basis functions are uncorrelated and have variances equal to the eigenvalues of the operator T. The coefficients relating to the same basis function, but for different subspaces corresponding to different discrepancy space coordinates, have correlations determined by k_{cv}, so that

$$\mathbb{E}\left([\beta(v)]_i\right) = 0, \qquad \mathrm{Cov}\left([\beta(v')]_i\ ,\ [\beta(v'')]_j\right) = \lambda_i \delta_{i,j} k_{cv}(v', v'').$$

Meanwhile, every climate error term, denoted e_c in (4.84), is modelled as independent to all other quantities with variance corresponding to the residual expected loss (4.73). With the introduction of the climate error term we are effectively replacing variation not captured by the eigen-basis with white noise, spreading the discarded eigenvalues or variances evenly over the spectral domain:

$$\mathbb{E}\left(e_c(\xi, v)\right) = 0, \qquad \mathrm{Cov}\left(e_c(\xi', v')\ ,\ e_c(\xi'', v'')\right) = \left(\sigma_c^2 - \sum_{j=1}^{N} \lambda_j\right) \delta_{\xi',\xi''} \delta_{v',v''}.$$

Finally, the weather terms are modelled faithfully to the full model:

$$\mathbb{E}\left(w(t, x, v)\right) = 0, \qquad \mathrm{Cov}\left(w(t', x', v')\ ,\ w(t'', x'', v'')\right) = k_{wt}(t', t'') \delta_{x',x''} \delta_{v',v''}.$$

We can then use the Bayes linear formulae to adjust our expectation for the coefficients given observations of y. We can also model uncertainty for the variance of y by introducing a single multiplicative constant to both $k_{c\xi}$ and k_{wt}, which is inherited by the linear model's coefficient and the residual variances, and employing the NIG machinery.

Now, since we only have access to numerically approximated eigenfunctions, our basis is already naturally limited to the size of the matrix $\mathbf{K}$ in (4.75), which is determined by the number of nodes employed in the approximation. Further truncation will lead to a decrease in computational demand when it comes to fitting the linear model, since on each adjustment of the basis coefficients we are required to invert the coefficients' variance matrix. The other effect of the truncation is to smooth out the likelihood for observations, which can help to stabilise our calibration calculations as we saw in example 3.1.2.

We see the eigenfunction approximation as generating a set of regressors, whose orthogonality reduces the regression calculations' redundancy and potential for instability, and the eigenvalue approximations as providing a variable ranking and selection criterion.

In this way the eigen-basis approximation fulfils a role similar to a variable selection procedure, but is based on the prior covariance specification for y and ξ^* rather than the likelihood of observed data.

It is also useful to look at the linear model that retains all the approximate eigenfunction regressors but linearly transforms them back to the regressors that are the autocovariance functions centred on each node. This is the case because it turns out to be a more natural parameterisation for understanding the related approximation in section 4.4. Under the reparameterisation, the model is described using the coefficient vector $\tilde{\beta}$, where

$$y(\xi, v) \approx \beta(v)^T \mathbf{\Lambda}^{-1}\mathbf{U}^T\mathbf{W}k_{c\xi}(\mathbf{N}, \xi) \oplus e_c(\xi, v) \oplus w(\xi, v), \tag{4.85}$$

$$y(\xi, v) \approx \tilde{\beta}(v)^T k_{c\xi}(\mathbf{N}, \xi) \oplus e_c(\xi, v) \oplus w(\xi, v). \tag{4.86}$$

The variance matrix for the $\tilde{\beta}$ is then given by,

$$\begin{aligned}\mathrm{Var}\left(\tilde{\beta}(v)\right) &= \mathbf{W}\mathbf{U}\mathbf{\Lambda}^{-1}\mathrm{Var}\left(\beta(v)\right)\mathbf{\Lambda}^{-1}\mathbf{U}^T\mathbf{W},\\ &= \mathbf{W}^{1/2}\tilde{\mathbf{U}}\mathbf{\Lambda}^{-1}\tilde{\mathbf{U}}^T\mathbf{W}^{1/2},\\ &= \mathbf{W}^{1/2}\mathbf{W}^{-1/2}\mathbf{K}^{-1}\mathbf{W}^{-1/2}\mathbf{W}^{1/2},\\ &= \mathbf{K}^{-1}.\end{aligned}$$

A particularly nice property of formulation (4.86) is that it draws attention to the node locations, and implicitly to the distance of every other point from them. At the node locations the first two moments of the basis approximation match those of the full model, which can be seen by writing

$$\mathrm{Var}\left(\tilde{\beta}(v)k_{c\xi}(\mathbf{N}, \mathbf{N})\right) = \mathbf{K}\mathbf{K}^{-1}\mathbf{K} = \mathbf{K},$$

and further away from the node locations the approximation become less good. In section 4.4 we will construct an approximation, similar to the Nyström-basis model, while holding the idea that we are expanding around the node locations as an integral part of our reasoning, rather than as a peripheral or coincidental feature.

4.3.2.6 The importance-weighted eigenfunction approximation

We are not limited to standard quadrature designs or random sequences however; appropriately weighted, we can choose from a much wider class of node designs for the sum in (4.74). This is where a great strength of the method is revealed. We can, for example, create a basis from an MCMC sample. In this way we can craft bases to strangely shaped regions of interest. In particular, we can tailor the basis to an approximation of the posterior for ξ^* given simulated and system data. We will call the tailored basis an importance basis after the importance sampling procedure, which we now review.

Importance sampling is a technique for generating an approximate sample from a target distribution using a weighted sub-sample of another sample drawn from an approximating distribution. It is especially useful when it is easy to sample from the approximating distribution but difficult, or impossible, to sample from the target distribution. The second reason the technique is useful, and the reason that helps us now, is that the approximating distribution may be positioned not to match the target distribution but to produce samples at parameter values that are important to the posterior quantities we seek to estimate. Importance sampling is known as a variance reduction technique because a Monte Carlo estimate that samples important parameter values only rarely will tend to exhibit jumps upon those rare occasions. As an example, in [12], Gamerman advocates the use of fat-tailed importance distributions because estimates of moments from random samples are particularly sensitive to the extreme values that are sampled only rarely under light-tailed distributions.

An importance sample from a target distribution with density π is produced by drawing a random or pseudorandom sample of nodes $\mathbf{N}_j$ from an importance density and computing normalised importance weights $\hat{w}_j$,

$$n_j \sim \pi_{Imp}, \tag{4.87}$$

$$w_j \propto \frac{\pi(n_j)}{\pi_{Imp}(n_j)}, \tag{4.88}$$

$$\hat{w}_j = \frac{w_j}{\sum_j w_j}, \tag{4.89}$$

where the normalising constants of neither density are required since we normalise them away with line (4.89). The nodes and weights can then be used to approximate the target distribution by a multinomial distribution or to approximate integrals over it according to

$$\pi(x) \approx \text{M.nom}\left([\mathbf{N}]_j; \hat{w}_j\right), \tag{4.90}$$

$$\int h(x)\pi(x)\,\mathrm{d}x \approx \sum_j \hat{w}_j h([\mathbf{N}]_{j,\cdot}). \tag{4.91}$$

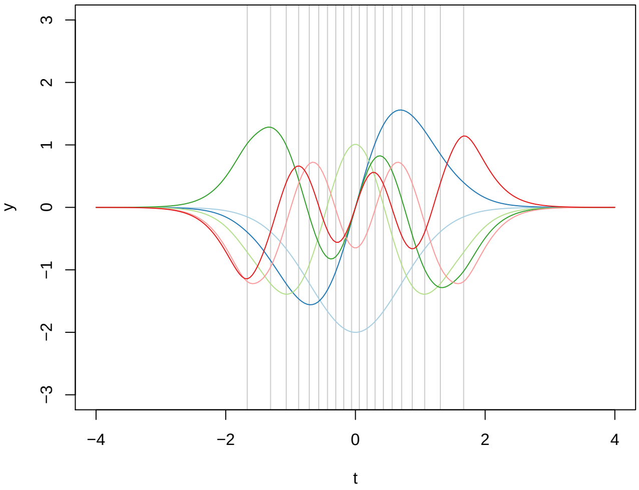
In our case we can use an estimate for the posterior distribution for ξ^* as an importance distribution to simulate nodes for approximation (4.74). We would then calculate the weights in that expression by dividing the prior at the node locations by the importance density there, and normalising them.

As an example of the effects achievable with the importance weighting technique, figure 4.8(a) shows a plot of the first six of twenty basis functions arising from a unit normal input prior π; a Matérn function for the autocovariance k; and unweighted nodes at twenty equally spaced quantiles of π. The functions are scaled relative to each other by the square roots of their eigenvalues. This plot is compared to figure 4.8(b), in which we present a plot of the first six basis function computed using an importance distribution that differs from the prior. Specifically, we place the nodes at twenty equally spaced quantiles of a unit normal centred on minus one. The variance of the resulting field grows towards the centre of the plot's t axis, the approximate location of the bulk of $\mathrm{N}(0, 1)$'s mass, but the potential for high-frequency variation is concentrated towards the left since this is where the nodes are located. This basis would be appropriate for modelling values of a function whose inputs are a priori believed to be distributed according to $\mathrm{N}(0, 1)$ but whose high frequency behaviour we know, or suspect, to be more important in the region of $\mathrm{N}(-1, 1)$'s greatest density.

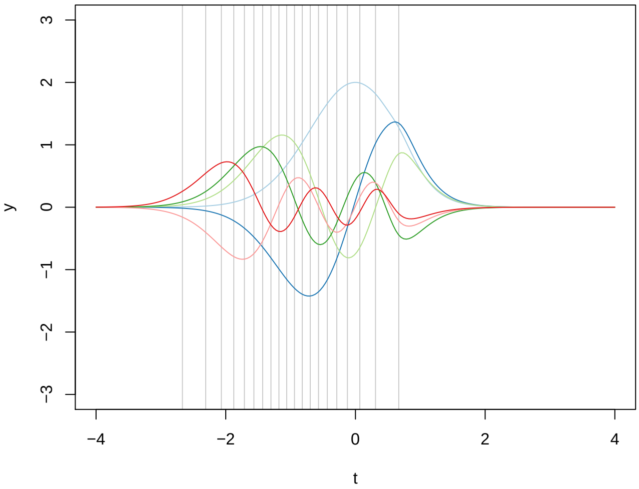
4.4 The Cholesky-basis emulator

As in section 4.3.2.5, our aim is still to derive a linear model that is suitable for learning about an unknown smooth climate function whose second moments are described by a known autocovariance function.

The theoretical motivation behind the eigenfunction basis is the idea that we can model a maximal amount of variation with a finite number of basis functions and so minimise the cost of our calculations in terms of this number. Although we can truncate the approximate eigenfunction basis, each member still relies on all of a large number of nodes, which each contribute to the cost of a basis function evaluation. Empirically we find that we can thin out virtually redundant nodes in a convenient fashion by Cholesky decomposing the variance matrix for the climate values at the node locations. This leads our



(a) Unweighted Nyström -basis.



(b) Weighted Nyström -basis.

Figure 4.8: Weighted and unweighted Nyström-bases. For the unweighted basis the nodes are positioned at quantiles of the unit normal distribution and the weights for the Nyström approximation are all equal. For the weighted basis the nodes are positioned at quantiles of a normal importance density with unit variance and mean -1; the weights for this Nyström approximation are proportional to the mean zero unit normal density over the importance density.

investigation towards to an approximation whose modification of the full model structure is understood in terms of retained nodes rather than retained eigenfunctions. We refer to a basis produced using a node thinning procedure and no eigenfunction truncation as a Cholesky-basis.

With pivoting enabled, the Cholesky decomposition of a variance matrix $\mathbf{K}$ sorts the rows and columns, which correspond to individual nodes, according to the variance of the field at the node locations given observations at all preceding nodes. And from the main diagonal of the Cholesky factor we can read off the square roots of these variances.

To better understand the relationship between the basis node locations and the Cholesky pivot we need to take a closer look at what the Cholesky decomposition algorithm actually does when we summon it from a library of numerical algebra functions. A pseudo-code breakdown of the standard Cholesky algorithm is presented in appendix B.0.22. The breakdown allows us to see that the Cholesky pivoting subroutine can be understood as effectively guiding a stepwise search that, at each iteration, selects the variable with greatest variance given the values of all previously selected variables. We call this greatest variance the maximal variance, and its corresponding variable the maximal variable; the maximal variance can be understood as approximating the variance for all the unselected variables that is resolved upon linear adjustment from the maximal variable. It is in this sense that the Cholesky thinning procedure for the node locations selects locations that are representative for the discarded candidate locations.

It can also be shown that the first N components of the Cholesky pivot, the ordered vector of indices for the selected variables, identify the $N \times N$ submatrix of $\mathbf{K}$ with the greatest determinant. Another feature worth noting is that the computational costs of standard implementations of both the Cholesky decomposition and the symmetric eigendecomposition increase with the cube of the number of rows of $\mathbf{K}$, but that the Cholesky decomposition is around ten times faster.

We now ask ourselves whether we can write an algorithm that is more specifically relevant to the calibration problem; one that identifies a suitable set of nodes for approximations like (4.74). The answer lies in altering the criterion that is used to pivot or re-order the rows and columns. Maximisation of our proposed criterion may be phrased as identifying the climate observations that are best placed to channel information from the simulated series to the system series. The best climate variables then define basis functions for our emulator in the form of autocovariance functions centred on the variables, as made explicit in the alternate parameterisation of the eigen-basis (4.86). To quantify a climate variable's informativeness for the system climate we consider a statistic, a function of ξ, that takes the form of an integral of Bayes linear resolved variances,

$$Q(\xi, v) = \int \pi_{\xi^*}(\xi')\mathrm{Cov}\left(c(\xi', v^*)\,,\; c(\xi, v)\right)^2 \mathrm{Var}\left(c(\xi, v)\right)^{-1} \,\mathrm{d}\xi', \tag{4.92}$$

the function π_{ξ^*} being the prior for ξ^*, as in (4.82). Our maximisation of the criterion begins with the assumption that the available simulations mark the best places for potential basis nodes, we use them to define a set of candidate basis nodes and store them in the matrix $\mathbf{\Xi}$. The domain of the criterion thus shrinks to a finite set of points whose values we store in a vector $\mathbf{Q}$,

$$[\mathbf{Q}]_i = Q([\mathbf{\Xi}]_{i,\cdot}, \hat{v}) = \int \pi(\xi^*)\mathrm{Cov}\left(c(\xi^*, v^*)\,,\; c([\mathbf{\Xi}]_{i,\cdot}, \hat{v})\right)^2 \mathrm{Var}\left(c([\mathbf{\Xi}]_{i,\cdot}, \hat{v})\right)^{-1} \,\mathrm{d}\xi^*. \tag{4.93}$$

We then need to approximate integral (4.93) by a finite sum, which can be thought of as a Monte Carlo estimate or quadrature approximation whose nodes are referred to as integration nodes and are stored in the matrix $\mathbf{P}$:

$$[\mathbf{Q}]_i \approx \sum_j w_j \mathrm{Cov}\left(c([\mathbf{P}]_{j,\cdot}, v^*)\,,\; c([\mathbf{\Xi}]_{i,\cdot}, \hat{v})\right)^2 \mathrm{Var}\left(c([\mathbf{\Xi}]_{i,\cdot}, \hat{v})\right)^{-1}. \tag{4.94}$$

We find it helpful to think of the nodes of the integration grid as atoms of a discrete distribution for ξ^*.

The new algorithm, which we describe in Algorithm 3, behaves just like the algorithm for Cholesky decomposition, with the exception that it takes two matrix arguments,

$$\mathbf{C} = \mathrm{Cov}\left(c(\mathbf{P}, v^*)\,,\; c(\mathbf{\Xi}, \hat{v})\right) \qquad \text{and} \qquad \mathbf{K} = \mathrm{Var}\left(c(\mathbf{\Xi}, \hat{v})\right),$$

and one vector argument, w, consisting of the integration weights. The algorithm includes a stopping rule, which in the standard Cholesky algorithm alerts us to numerical indefiniteness of the matrix to be decomposed. Here, we use the stopping rule to tell us when further inducing variables are unable to produce a Q value, an expected variance resolution for $c(\xi^*, v^*)$, greater than a certain threshold value. If the stopping rule is not activated, the algorithm returns: $\mathbf{R}$, an upper-right triangular square root of $\mathbf{K}$; ϱ, a pivot vector which orders the inducing variables according to their potential to resolve variance for $c(\xi^*, v^*)$; and ϵ, a vector of values giving the maximum criterion value from each loop of the algorithm.

The algorithm serves as a thinning procedure when we submit to it a set of candidate nodes in the matrix $\boldsymbol{\Xi}$, and we choose to retain the N nodes indexed by the pivot vector up to the point at which the stopping rule is activated. Additionally, the principal $N \times N$ submatrix of $\mathbf{R}$ provides the Cholesky factor for the variance matrix at these nodes.

Algorithm 3 Modified Cholesky decomposition with pivoting

```
Input $\mathbf{C}$, $\mathbf{K}$, $w$
Initialise $\mathbf{R} \leftarrow \mathbf{0}$, $\varrho \leftarrow 1 : n$, $\epsilon \leftarrow \mathbf{0}$
for $j = 1, \ldots, n$ do
    procedure PIVOTING SUBROUTINE
        if $j > 1$ then
            $[\mathbf{Q}]_{j-1} \leftarrow 0$                          ▷ Skip previously selected variables.
        end if
        for $i = j, \ldots, n$ do
            $[\mathbf{Q}]_i \leftarrow w^T[\mathbf{C}]^2_{\cdot,i}/[\mathbf{K}]_{i,i}$   ▷ Evaluate pivoting criterion.
        end for
        if $\max_i [[\mathbf{Q}]_i] < \epsilon^*$ then
            Escape For-loop and terminate algorithm
        end if
        $\epsilon_j \leftarrow \max_i [[\mathbf{Q}]_i]$                ▷ Identify maximal criterion value.
        $q \leftarrow \arg\max_i \; [[\mathbf{Q}]_i]$              ▷ Identify maximal variable.
        $[\mathbf{K}]_{\cdot,j} \leftrightarrows [\mathbf{K}]_{\cdot,q}$                   ▷ Re-order variables.
        $[\mathbf{K}]_{j,\cdot} \leftrightarrows [\mathbf{K}]_{q,\cdot}$
        $[\mathbf{C}]_{\cdot,j} \leftrightarrows [\mathbf{C}]_{\cdot,q}$
        $[\mathbf{R}]_{\cdot,j} \leftrightarrows [\mathbf{R}]_{\cdot,q}$
        $\varrho_j \leftrightarrows \varrho_q$
    end procedure
    $[\mathbf{R}]_{j,j} \leftarrow \sqrt{[\mathbf{K}]_{j,j}}$                 ▷ Update Cholesky factor.
    $[\mathbf{R}]_{j,(j+1):n} \leftarrow [\mathbf{K}]_{j,(j+1):n}[\mathbf{R}]^{-1}_{j,j}$
    $[\mathbf{C}]_{\cdot,(j+1):n} \leftarrow [\mathbf{C}]_{\cdot,(j+1):n} - [\mathbf{C}]_{\cdot,j}[\mathbf{K}]_{j,(j+1):n}[\mathbf{K}]^{-1}_{j,j}$
    $[\mathbf{K}]_{(j+1):n,(j+1):n} \leftarrow [\mathbf{K}]_{(j+1):n,(j+1):n} - [\mathbf{K}]_{(j+1):n,j}[\mathbf{K}]_{j,(j+1):n}[\mathbf{K}]^{-1}_{j,j}$.
end for
return $\mathbf{R}$, $\varrho$, $\epsilon$.
```

We find contemporary research from the machine learning community mirroring our ideas here in the work of Quiñonero-Candela and Rasmussen. In [41], they describe a range of methods for the approximation of Gaussian processes based on the premise that conditional on a certain set of field values, which they call *inducing variables*, all or most of the remaining values are mutually independent. The coefficients of the basis functions that we use to define our linear model emulator play the same role as inducing variables, so we can see the Cholesky node-selection algorithm as also being a method for selecting inducing climate variables.

Approximate analogues to our modified Cholesky algorithm have also been proposed in the machine learning literature, where such forward-selection strategies are referred to as greedy algorithms and models constructed with inducing variables may also be called vector machines. Lawrence et al.[27], for example, describe the sequential construction of a Cholesky factor using a greedy algorithm that pivots candidate variables according to an entropy score. As far as we are aware, however, there is no peer-reviewed work in which vector machine-type Bayesian models are constructed in anticipation of a calibration problem of this sort.

The inducing variable methods encourage a change of perspective from basis-oriented modelling strategies; the inducing variables themselves represent an appealing way of re-interpreting the finite basis approximation to the full rank random field in terms of landmark or reference climate values, which we feel are, in many ways, easier to relate to than basis function coefficients. As such we frequently switch between the basis function and inducing variable interpretations as and when it is convenient.

We use the Cholesky-basis to construct another linear regression model for approximating values of the field described by the full model of (4.77)-(4.81). This time the column vector of basis functions, $\phi(\xi)$, is composed of climate covariance functions centred on the basis node locations. For a single, fixed v, the covariances between the basis coefficients are recognisable from the alternate parameterisation of the eigen-basis linear model (4.86); they are derived from the matrix $\mathbf{K} = k_{c\xi}(\mathbf{N}, \mathbf{N})$, which describes the prior variance matrix for the inducing climate variables in the full model. For different discrepancy coordinates we shrink the correlations using k_{cv} just like we did with the eigen-basis model:

$$y(\xi, v) \approx \beta(v)^T \phi(\xi) \oplus e_c(\xi, v) \oplus w(\xi, v), \qquad [\phi(\xi)]_i = k_{c\xi}([\mathbf{N}]_{i,\cdot}, \xi), \tag{4.95}$$

$$\mathbb{E}\left([\beta(v)]_i\right) = 0, \qquad \mathrm{Cov}\Big([\beta(v')]_i\ ,\ [\beta(v'')]_j\Big) = [\mathbf{K}^{-1}]_{i,j} k_{cv}(v', v''). \tag{4.96}$$

A major difference of the Cholesky-basis model, inspired by the inducing variable interpretation of the approximation, is that we specify the variance of the climate error term e_c, not as homogeneous over its domain, but as the residual variance of the full field given the values of the selected inducing variables:

$$\mathbb{E}\left(e_c(\xi, v)\right) = 0, \qquad \text{Var}\left(e_c(\xi, v)\right) = k_{c\xi}(\xi, \xi) - k_{c\xi}(\xi, \mathbf{N})k_{c\xi}(\mathbf{N}, \mathbf{N})^{-1}k_{c\xi}(\mathbf{N}, \xi), . \tag{4.97}$$

Finally, the weather term of the Cholesky-basis model is also specified as it is in the full model:

$$\mathbb{E}\left(w(t, x, v)\right) = 0, \qquad \text{Cov}\left(w(t', x', v')\,,\; w(t'', x'', v'')\right) = k_{wt}(t', t'')\delta_{x',x''}\delta_{v',v''}.$$

Note that the variance of the linear model's error term, $e_c(\xi)$, decreases to zero at the basis node locations. This feature allows us to explicitly associate the model's regression coefficients with values of the approximate climate at these locations,

$$c(\xi) \approx \beta^T \phi(\xi) \oplus e_c(\xi), \tag{4.98}$$

$$c(\mathbf{N}) \approx \mathbf{K}\beta. \tag{4.99}$$

Given expression (4.99), we can see that specifying the coefficients' variance as $\mathbf{K}^{-1}$ means that the climate covariances between the node locations induced by the Cholesky-basis linear model matches those induced by the autocovariance function of the full model.

As with the eigen-basis model, the computational savings introduced by this linear model come from modelling certain climate residual terms, e_c, as uncorrelated to each other. The loss of structure the linear model approximation brings about ought to be small if the basis functions account for most of y's variation, so that the residuals are small. To be explicit, the savings arise from not computing the full residual variances and from not inverting them. The inversion of the approximate variance for values of c now requires one inversion for the variance matrix for the node locations as well as a series of others on the scale of the sets of $e_c(\xi)$ whose correlations are preserved.

In the extreme case, we can model all the residuals as mutually independent,

$$\text{Cov}\left(e_c(\xi', v')\,,\; e_c(\xi'', v'')\right) = 0. \tag{4.100}$$

Quiñonero-Candela refers to this type of specification as a FITC (Fully Independent Training Conditional) approximation to the full covariance specification. The approximation is said to possess a global Markov property, meaning that all pairs of subsets of variables are conditionally independent given the inducing variables. Equivalently, the inducing variables can be said to separate the field in the sense of (3.2). Alternatively, we can treat particular sets of residuals as independent but preserve the correlations within them. This is referred to by Quiñonero-Candela as the PITC (Partially Independent Training Conditional) approximation; it represents a more sophisticated approximation strategy than simply modelling all the residuals as independent. In our application of the Cholesky-basis approximation to real time series data in chapter 5 and to synthetic time series data in section 4.4.1, for example, we choose to retain the correlations within each series but ignore the correlations between series. In this case, the climate residuals have the following covariance structure,

$$\begin{aligned}\mathrm{Cov}\left(e_c(t',x',v')\,,\ e_c(t'',x',v')\right) =&\mathrm{Cov}_{c(\mathbf{N},v')}\left(c(t',x',v')\,,\ c(t'',x',v')\right),\\ =&k_{c\xi}((t',x'),(t'',x'))\\ &-k_{c\xi}((t',x'),\mathbf{N})^T k_{c\xi}(\mathbf{N},\mathbf{N})^{-1}k_{c\xi}(\mathbf{N},(t'',x')),\end{aligned}$$

so that the variance of values of the approximate climate term over time, at a single input and discrepancy coordinate is given by

$$\begin{aligned}&\mathrm{Cov}\left(\check{c}(t',x',v')\,,\ \check{c}(t'',x',v')\right) \qquad (4.101)\\ &\quad= k_{c\xi}((t',x'),\mathbf{N})\mathrm{Var}\left(\beta(v')\right)k_{c\xi}(\mathbf{N},(t'',x'))\\ &\qquad\qquad+k_{c\xi}((t',x'),(t'',x'))-k_{c\xi}((t',x'),\mathbf{N})k_{c\xi}(\mathbf{N},\mathbf{N})^{-1}k_{c\xi}(\mathbf{N},(t'',x')),\\ &\quad= k_{c\xi}((t',x'),\mathbf{N})(\mathrm{Var}\left(\beta(v')\right)-k_{c\xi}(\mathbf{N},\mathbf{N})^{-1})k_{c\xi}(\mathbf{N},(t'',x'))\\ &\qquad\qquad+k_{c\xi}((t',x'),(t'',x')). \qquad (4.102)\end{aligned}$$

Notice that before we have assimilated any data $\mathrm{Var}\left(\beta(v')\right) = k_{c\xi}(\mathbf{N},\mathbf{N})^{-1}$, and the approximation's covariance between time points for an individual simulation is faithful to the full model. As $\mathrm{Var}\left(\beta(v)\right)$ is reduced to zero via the adjustment from data, the variance of the climate trend approaches that which would have resulted from direct observation of the climate trend at the basis node locations. Crucially, the covariances within the series are maintained. This is not the case for the covariances between simulations, which we insist are induced solely through the inducing variables, with the effect that

$$\mathrm{Cov}\left(e_c(t',x',v')\,,\ e_c(t'',x'',v'')\right)=\delta_{x',x''}\delta_{v',v''}\mathrm{Cov}_{c(\mathbf{N},v')}\left(c(t',x',v')\,,\ c(t'',x'',v'')\right), \qquad (4.103)$$

where we have used $\mathrm{Cov}_{c(\mathbf{N},v')}\left(c(t',x',v')\,,\ c(t'',x'',v'')\right)$ to denote the adjusted or residual covariance between $c(t',x',v')$ and $c(t'',x'',v'')$ given $c(\mathbf{N},v')$ according to the full model.

We can view the Cholesky-basis approximation from a number of angles, corresponding to opportunities for different approximations that have similar or equivalent consequences. From one angle we can see the approximation as enforcing sparsity on the matrix describing the residual variance of climate values conditional on the inducing variables. In this way the approximation can be called a sparse approximation. Alternatively, we can look at the approximation as an approximate factorisation of a joint probability distribution, along the lines of:

$$\pi(c(\mathbf{N},v),c(\Xi,v)) = \pi(c(\mathbf{N},v))\pi(c(\Xi,v)|c(\mathbf{N},v)) \approx \pi(c(\mathbf{N},v))\prod_i \pi(c([\Xi]_{i,\cdot},v)|c(\mathbf{N},v)).$$

This is the way Tresp[55] arrives at a related approximation method, which he calls the Bayesian Committee Machine.

Because we ignore the residual correlations, the approximate field effectively becomes rougher further away from the nodes. Typically we would view this is as a degradation of the full model, but we note that we could also embrace it as a way to introduce non-stationarity into the model. The field described by the approximation is smoother in regions of the space filled more densely with nodes. This is not a feature we will have time to develop however.

4.4.1 An extended synthetic example: analysing the effects of the basis approximations

We now re-examine examples 2.3.2 and 3.1.2 in order to investigate the accuracy of the eigen- and Cholesky-basis approximations of a model whose moments are specified by autocovariance functions. In those first examples, the output quantity y was described as a sum of climate and weather terms,

$$y(t,x,v) = c(t,x,v) \oplus w(t,x,v), \tag{4.104}$$

for which we defined autocovariance functions as products of Matérn autocovariances in each direction,

$$\mathrm{Cov}\left(c(t',x',v')\,,\;c(t'',x'',v'')\right) = k_{ct}(t',t'')k_{cx}(x',x'')k_{cv}(v',v''), \tag{4.105}$$

$$\mathrm{Cov}\left(w(t',x',v'')\,,\;w(t'',x'',v'')\right) = k_{wt}(t',t'')k_{wx}(x',x'')k_{wv}(v',v''). \tag{4.106}$$

We then simulated a grid of simulator data consisting of $N_x = 13$ simulations of time series of length $N_t = 51$. In this extended example we modify (4.106) slightly by setting the autocovariance factors for the weather term relating to the input and discrepancy spaces from Matérn functions to Kronecker delta functions. The effect of this is to render the weather signals from different simulations or system instantiations independent.

Our approximations to the full model, whose distinguishing features we recapitulate in the following list, all describe the climate as a sum of basis functions plus a climate 'error' term e_c:

$$y(t,x) \approx \breve{c}(t,x) \oplus w(t,x) = \phi(t,x)\beta \oplus e_c \oplus w(t,x).$$

1. The eigen-basis approximation, with a uniform prior for x^* and a uniform distribution over the time variable, uses approximate eigenfunctions of the operator T,

   $$T[f](t',x') = \int_{-1}^{1}\int_{-1}^{1} k_{ct}(t',t'')k_{cx}(x',x'')f(t'',x'')\,\mathrm{d}t''\,\mathrm{d}x'',$$

   to define its basis functions. It compensates for climate variation not captured in the basis by adding a homogeneous white noise-type climate error term.

   To implement the current example's eigen-basis approximation we generate a Sobol sample of 663 points over $[-1,1]^2$, which constitute the equally weighted nodes for the Nyström approximation of the eigenfunctions and eigenvalues. The first N of these are retained to define the basis and the linear model as explained in section 4.3.2.5.

2. The Cholesky-basis approximations use climate autocovariance functions centred on specific node locations as their basis functions. Specifically, we generate the Cholesky-basis by Cholesky decomposing the variance matrix for $c(\xi,\hat{v})$ at the $N_t \times N_x (= 51 \times 13 = 663)$ grid of simulation coordinates with algorithm 3 and by retaining the pivot's first N entries. The climate autocovariance functions centred

on the locations indexed by the retained pivot entries constitute the Cholesky-basis functions. We then consider the two variants of the Cholesky-basis model which treat the climate error terms in different ways:

(a) The FITC Cholesky-basis approximation, like the eigen-basis approximation, compensates for variation that is not captured by the basis with independent climate error terms. The variance of these terms is not homogeneous however, but varies according to the residual climate variation that is not resolved upon the full adjustment by the inducing climate variables.

(b) The PITC Cholesky-basis approximation models the error terms even more closely by allocating them the same heterogeneous variance specification as the FITC approximation while also incorporating covariance between sets of them corresponding to the individual time series.

Our examination of the approximations, and their relation to the full model, consists of three parts: firstly, in section 4.4.1.1, we examine the moments for the simulator climate and simulator output values induced by the full and approximate models before any simulations have been observed; secondly, in section 4.4.1.2, we look at the moments for the system climate and system output induced by the full and approximate models following adjustment by the simulator data; and thirdly, in section 4.4.1.3, we study the likelihood for x^* given an observation of a time series of system values.

4.4.1.1 The prior moments for the simulator climate and weather

We compare the moments arising from the basis approximations to those of the full model with a statistic normally used to measure the Kullback-Leibler divergence (KLD) between two normal distributions. When the approximating and target normal distributions, denoted $\hat{\mathcal{N}}$ and $\mathcal{N}^*$, are parameterised with mean and variance $(\hat{\mu}, \hat{\Sigma})$ and (μ^*, Σ^*) respectively, then the KLD of $\hat{\mathcal{N}}$ from $\mathcal{N}^*$ is written as:

$$D_{KL}(\mathcal{N}^* \| \hat{\mathcal{N}}) = \frac{1}{2}\left(\mathrm{Tr}\left(\hat{\Sigma}^{-1}\Sigma^*\right) + (\hat{\mu} - \mu^*)^T \hat{\Sigma}^{-1}(\hat{\mu} - \mu^*) - k - \log\left(\frac{|\Sigma^*|}{|\hat{\Sigma}|}\right)\right). \tag{4.107}$$

The KLD statistic may be interpreted as an expected benefit or advantage to employing a notionally true distribution rather than an approximation of it, or as a measure of information loss, but we use it now primarily as a convenient way to measure the difference between the first two moments induced by our approximating and full models.

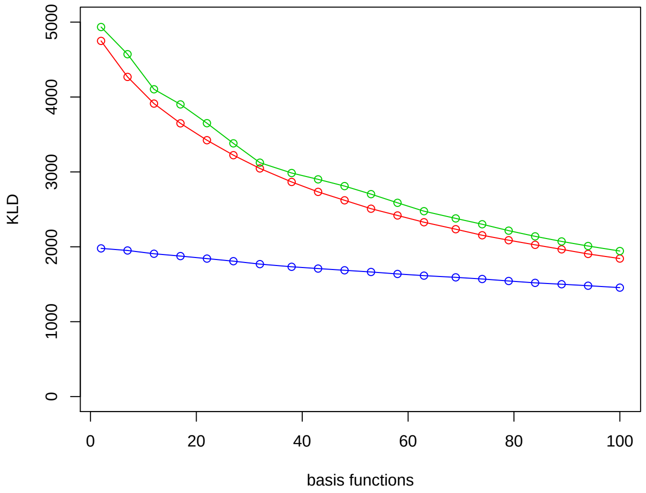
In figure 4.9(a) we plot the KLD statistics produced by comparing the prior means and full covariances for the simulator climate and simulator output quantities on the $N_t \times N_x$ grid of simulation coordinates. The plots show that the eigen-basis emulator is less divergent from the full model than the FITC variant of the Cholesky-basis emulator. The implication here is that the heterogeneous error variance in the latter model is less important than the optimal choice of basis functions in the former. The KLD statistics for the PITC variant of the Cholesky-basis emulator show that retaining some of the correlations between the climate error terms outweighs both of these features, and renders the PITC approximation a significantly better approximation to the full model.

Figure 4.9(b) shows how the differences between the approximations are diminished when we view them in the context of contributing to the specification of the outputs' moments. This occurs because the variance for the independent weather terms, which we add to the approximate climate variances, serves to dwarf the differences between the climate approximations for all but very crude approximations.

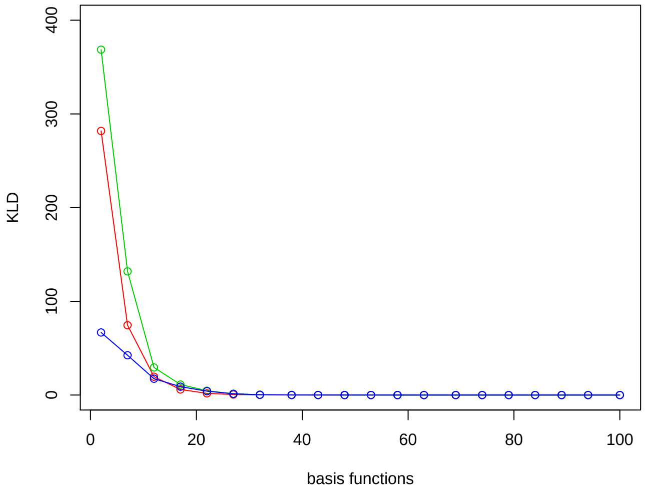
4.4.1.2 The adjusted moments for the climate and output

For the second round of tests we look at the same KLD statistics for the approximating and full models as in section 4.4.1.1, but now the moments we compare with (4.107) are those arising from the models whose coefficients have been adjusted by the simulator data.

Figure 4.10 appears to convey the same message as figure 4.9. It is difficult to interpret the significance of actual values of the KLD statistics, however, so in figure 4.12 we plot the interpolating surfaces of the differences between the full emulator's adjusted expectation and those of the approximate basis emulators. Context for the shape and size of these surfaces is provided by figure 4.11, which shows the adjusted expectation for the climate surface as calculated with the full model. We can see that with only 27 basis functions, all the approximate models lead to mean surfaces that are almost indistinguishable from the full model.

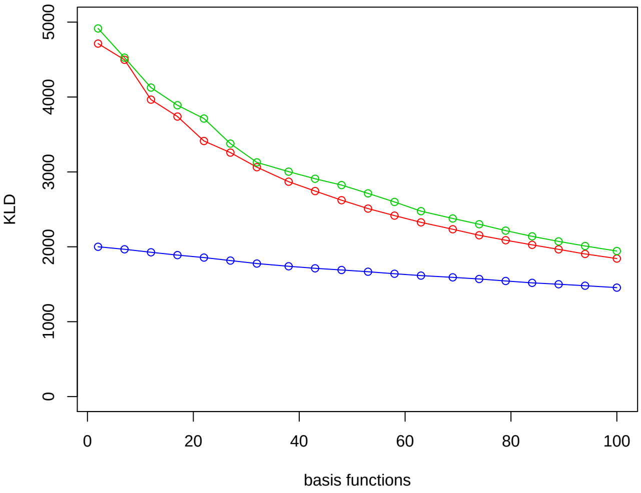


(a) KLD statistics for the simulator climate prior to adjustment.

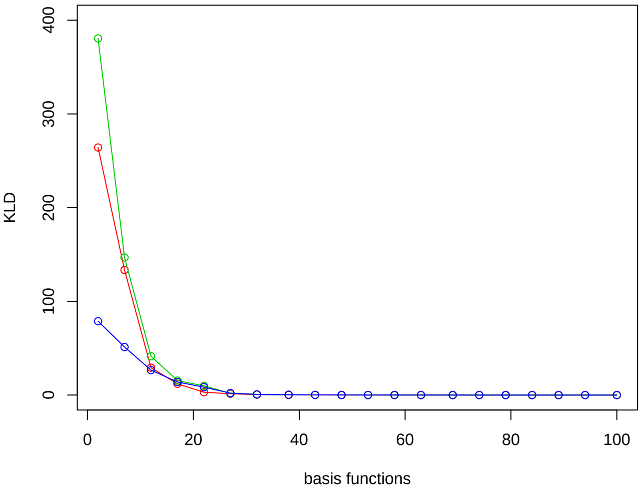


(b) KLD statistics for the simulator output prior to adjustment.

Figure 4.9: KLD statistics quantifying the divergence of the approximate models, with varying basis sizes, from the full model. The eigen-basis model is plotted in red, the Cholesky FITC model in green, and the Cholesky PITC model in blue.



(a) KLD statistics for the system climate after adjustment.



(b) KLD statistics for the system output after adjustment.

Figure 4.10: KLD statistics quantifying the divergence of the adjusted approximate models, with varying basis sizes, from the adjusted full model. The eigen-basis model is plotted in red, the Cholesky FITC model in green, and the Cholesky PITC model in blue.

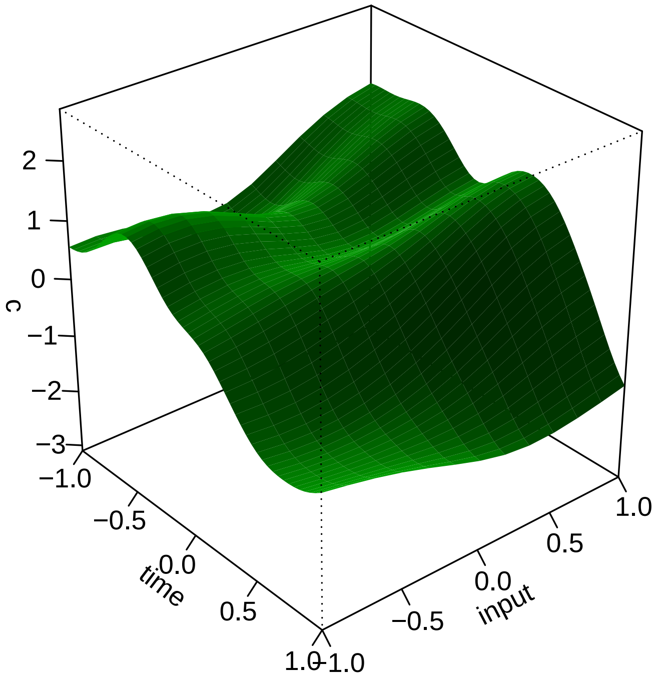
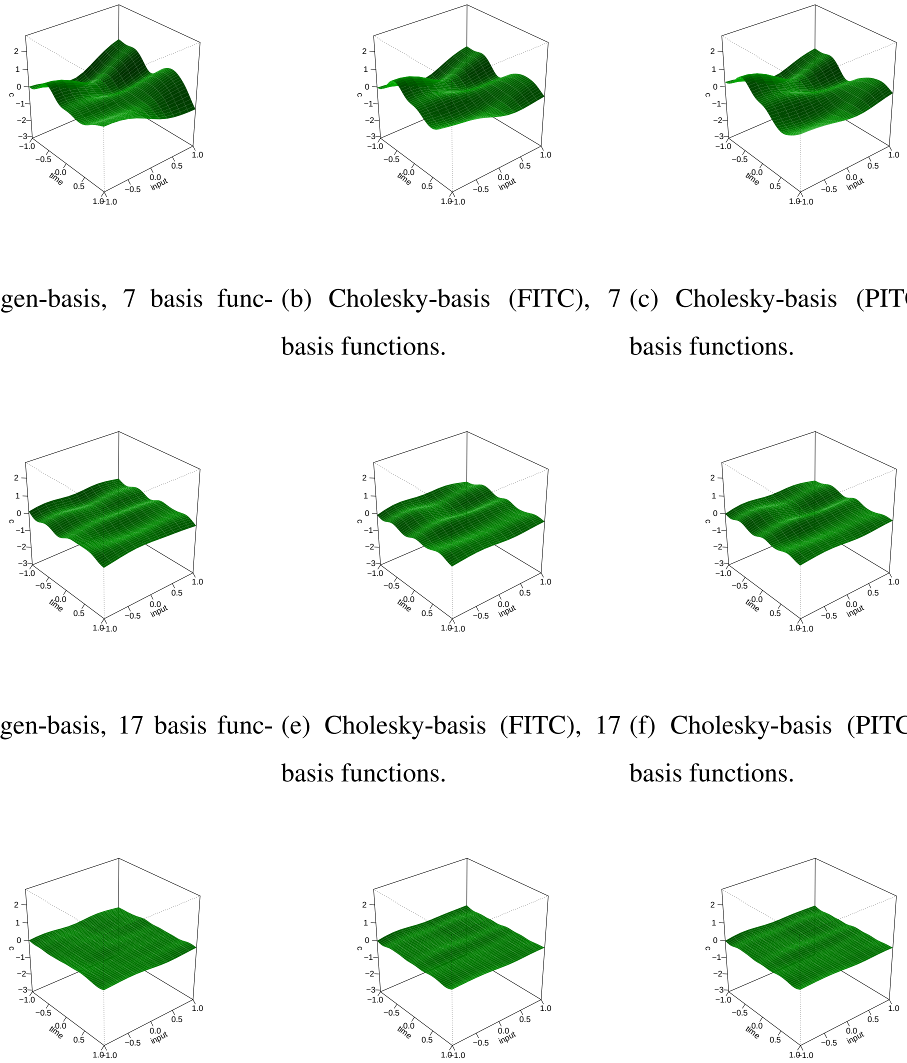


Figure 4.11: The interpolated adjusted expected values for the climate trend as calculated using the full emulator.



(a) Eigen-basis, 7 basis functions. (b) Cholesky-basis (FITC), 7 basis functions. (c) Cholesky-basis (PITC), 7 basis functions.

(d) Eigen-basis, 17 basis functions. (e) Cholesky-basis (FITC), 17 basis functions. (f) Cholesky-basis (PITC), 17 basis functions.

(g) Eigen-basis, 27 basis functions. (h) Cholesky-basis (FITC), 27 basis functions. (i) Cholesky-basis (PITC), 27 basis functions.

Figure 4.12: Plots of the difference between the surface of adjusted expected values for the climate as calculated with the full model and with the basis approximations.

4.4.1.3 The likelihood for the system input

Finally, we consider the approximations in terms of the calibration results to which they lead. To do so we first discretise the input space to a set of 100 equally spaced points. We then simulate a time series of system output values using multivariate normal distribution with moments informed by the full model, and a value for x^* chosen from amongst the points of the discretised space.

Our approach is to look at the normalised likelihoods given the simulated system output at the 100 candidate input points as if they constituted a posterior probability distribution for x^*. We then compute a version of the KLD statistic for discrete distributions: specifically, the KLD of an approximating discrete probability distribution Q from a target distribution P is given by

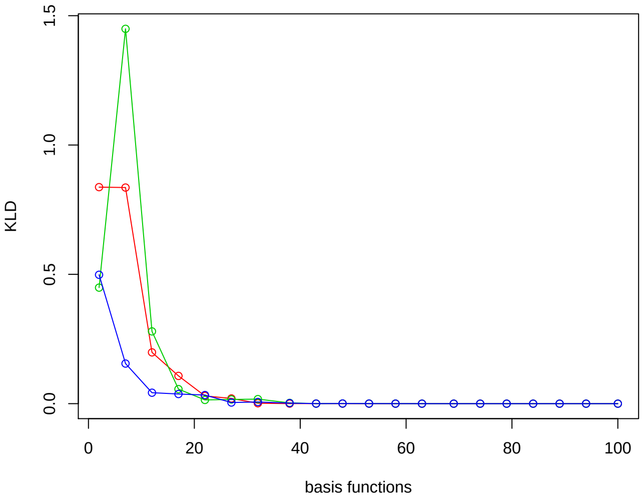
$$D_{KL}(P||Q) = \sum_i P(x_i) \log\left(\frac{P(x_i)}{Q(x_i)}\right).$$

Figure 4.13 shows KLD statistics suggesting that the eigen-basis emulator may be slightly better than the Cholesky FITC emulator in approximating the full model, but that the Cholesky PITC is superior to both. It also shows that the differences between the approximations become very small as their bases reach around 30 members.

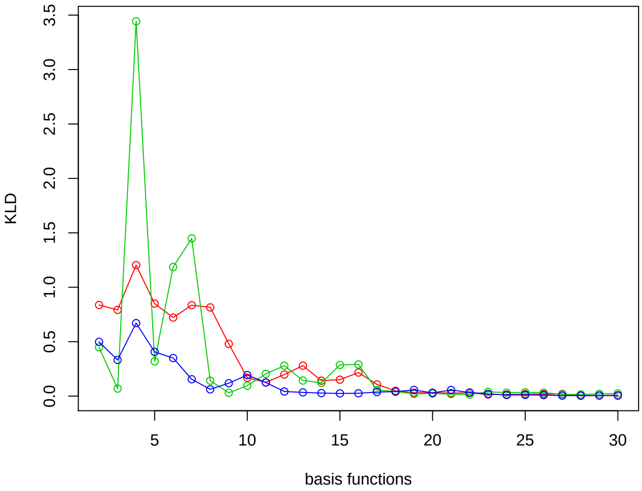
But again, despite the reassurance that the approximations really do get better as the number of basis functions increases, the KLD statistics are not very enlightening for the quality of the approximations. So in figure 4.14 we plot the normalised likelihoods arising from the approximations with bases of increasing size alongside that from the full emulator. We see that the shapes of the approximate likelihoods quickly converge on that of the true likelihood. It also appears that, in this particular instantiation of the synthetic example, the eigen-basis approximation better identifies x^* than the full model. It ought to be stressed that this is an anomaly and that, by definition, the full model, which was used to simulate the data, is expected to attribute a higher log-likelihood to x^*.

4.4.2 Making inferences for x^*

To make inferences for x^* is the aim of what we have, until now, referred to rather informally as our calibration procedure. In this section, as well as the next, we explain

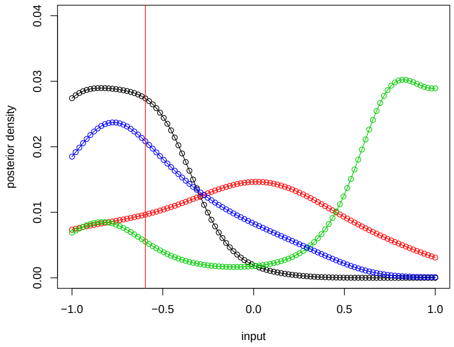


(a) 2-100 basis functions.

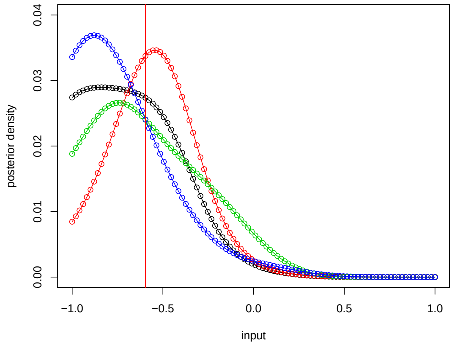


(b) 2-30 basis functions.

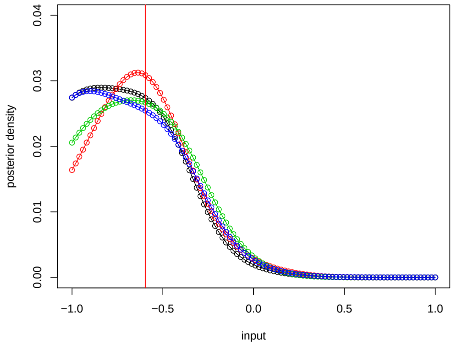
Figure 4.13: Plots showing the Kullback-Leibler divergences of the posterior distributions for x^*, calculated using the approximate models, from the posterior calculated using the full model. As in the other plots in this section, the statistics corresponding to the eigenbasis model are plotted in red, the Cholesky FITC model in green, and the Cholesky PITC model in blue. Subfigure 4.13(b) provides a higher resolution examination of the statistics for the smaller bases.



(a) 7 basis functions.



(b) 17 basis functions.



(c) 27 basis functions.

Figure 4.14: Normalised likelihoods, interpreted as posterior distributions, for x^*. The black line interpolates the probabilities arising from the full model, the red line interpolates those from the eigen-basis model, the green line interpolates those from the Cholesky FITC model, and the blue line interpolates those from the Cholesky PITC model. The red vertical lines mark the location of the true values of x^*.

precisely the form of the inferences we intend to make and the calculations that they involve.

Focusing on the PITC Cholesky-basis model, and using the shorthand notation

$$\rho^* = k_{cv}(v^*, \hat{v}),$$

to denote the correlation brought about by separations in the discrepancy space, we can write the expectation for the system climate trend in terms of the adjusted basis coefficients for the simulator climate:

$$\mathbb{E}\left(\breve{c}(\xi', v^*)\right) = \rho^* \phi(\xi')^T \mathbb{E}\left(\beta(\hat{v})\right),$$

where, notationally, $\hat{v}$ is not interpreted as much as an argument of the expected system climate as an index for the simulator climate coefficients.

Following the approximation's climate error variance specification, as described in (4.103), the covariance between climate values within a series is given by,

$$\begin{aligned} \mathrm{Cov}\left(\breve{c}(t', x', v^*)\,,\ \breve{c}(t'', x', v^*)\right) =& \rho^{*2} \phi(t', x')^T (\mathrm{Var}\left(\beta(\hat{v})\right) - k_{c\xi}(\mathbf{N}, \mathbf{N})^{-1}) \phi(t'', x') \\ &+ k_{c\xi}((t, x'), (t', x')) k_{cv}(v^*, v^*), \end{aligned}$$

while the covariance for climate values belonging to different series is,

$$\mathrm{Cov}\left(\breve{c}(t', x', v')\,,\ \breve{c}(t'', x'', v'')\right) = \rho^{*2} \phi(t', x')^T \mathrm{Var}\left(\beta(\hat{v})\right) \phi(t'', x'').$$

These quantities, with the addition of variance terms attributable to the weather, induce the following expectations and variances for the system output,

$$\mathbb{E}\left(y(t, x^*, v^*)\right) \approx \mathbb{E}\left(\breve{c}(t, x^*, v^*)\right), \tag{4.108}$$

$$\begin{aligned} \mathrm{Cov}\left(y(t, x^*, v^*)\,,\ y(t', x^*, v^*)\right) \approx& \mathrm{Cov}\left(\breve{c}(t, x^*, v^*)\,,\ \breve{c}(t', x^*, v^*)\right) \\ &+ \mathrm{Cov}\left(w(t, x^*, v^*)\,,\ w(t', x^*, v^*)\right). \end{aligned} \tag{4.109}$$

Now, as we begin to consider tackling real data, rather than synthetic data simulated in a known and controlled way, we face the important decision of whether or not to associate statistics (4.108) and (4.109) with a probability distribution, which would allow us to construct a likelihood for the system inputs given a set of system outputs. This likelihood may then inform simulator diagnostics and contribute to a posterior distribution for x^*, given that we are also prepared to assign x^* with a prior distribution.

One property of a potential posterior distribution for x^*, written $\pi_{x^*|\mathbf{Y}}$ here, that we find particularly interesting is the $\alpha \times 100\%$ highest posterior density credible set (HPDCS), which we will assume always exists for the type of posteriors we will construct. An HPDCS is an $\alpha \times 100\%$ posterior credible set for which there exists an h such that,

$$\int \pi_{x^*|\mathbf{Y}}(x)\, \mathbf{1}(x \in \Omega_{HPDCS})\, \mathrm{d}x = \alpha, \qquad \text{and} \qquad \Omega_{HPDS} = \{x \mid \pi_{x^*|\mathbf{Y}}(x) > h\},$$

where $\mathbf{1}(\cdot)$ is the indicator function taking value one when its argument is a true proposition and zero otherwise. The significance of the HPDCS lies in the fact that it is often understood as being the smallest of all $\alpha \times 100\%$ posterior credible sets when we quantify size with,

$$|\Omega| = \int \mathbf{1}(x \in \Omega)\, \mathrm{d}x, \tag{4.110}$$

which can be seen as generalising the length of a one-dimensional credible interval. We imagine locating the HPDCS by lowering a horizontal plane over the posterior surface and stopping once α of the posterior mass is accounted for by the parameter values at which the density rises above the plane.

The HPDCS is a natural description of the posterior when our prior for x^* is uniform, but when it is not, the size of the region as described by (4.110) is questionable. We argue that it is more natural to concentrate on the $\alpha \times 100\%$ credible set, which is smallest with respect to the prior density for x^*, because we are particularly interested in the proportion of the inputs that we had once thought plausible that are discredited by the data. We define the size of a set Ω with respect to a distribution π to be,

$$|\Omega|_\pi = \int \pi(x)\, \mathbf{1}(x \in \Omega)\, \mathrm{d}x. \tag{4.111}$$

The credible set that is smallest with respect to the prior for x^* is the highest likelihood credible set (HLCS); a justification for this claim is presented in appendix D.2.1.

In practice we consider two methods for approximating (4.111) and other integrals over the input space. The first, and the simpler of the two requires that we generate a large pseudo-random or quasi-random sample from the prior. We evaluate the likelihood for each sample member and normalise the likelihoods so that they sum to one. We then sort the normalised likelihoods into decreasing order and select the first few such that their sum exceeds α. The proportion of samples corresponding to the selected likelihoods is our estimate of the HLCS's size:

$$|\Omega_{HLCS}|_{\pi_{x^*}} \approx \frac{1}{N} \sum_{j=1}^{N} \mathbf{1}(x \in \Omega_{HLCS}),$$

where the sum is over the N members of the prior sample. The prior density does not feature explicitly in these calculations since it exerts its influence via the distribution of the prior sample members.

The second, more sophisticated, method utilises the sparse grid methodology we mentioned in our discussion of the redundancy in full-grid designs in high dimensions and in section 2.4.1 as part of our discussion on integrating over covariance parameters. The sparse grid integration schemes provide us with arrays of integration node locations and integration weights for making approximations of the form,

$$\int \pi_{imp}(x) f(x)\,\mathrm{d}x \approx \sum_{j=1}^{N} w_j f(x_j). \tag{4.112}$$

The function π_{imp} in (4.112) plays the role of an importance density and characterises the integration scheme. To approximate an integral that does not explicitly include the importance density as a factor, we simply include its reciprocal in the function that is evaluated at the integration nodes,

$$\int f(x)\,\mathrm{d}x = \int \pi_{imp}(x) \frac{f(x)}{\pi_{imp}(x)}\,\mathrm{d}x \approx \sum_{j=1}^{N} w_j \frac{f(x_j)}{\pi_{imp}(x_j)}.$$

Thus, we approximate the normalising constant c for the posterior as

$$c = \int \pi_{y^*}(y^* \mid x) \pi_{\xi^*}(x)\,\mathrm{d}x \approx \sum_{j=1}^{N} w_j \frac{\pi_{y^*}(y^* \mid x_j) \pi_{\xi^*}(x_j)}{\pi_{imp}(x_j)},$$

and we think of the values r_j, in the context of approximating integrals, as playing a role analogous to posterior probability masses,

$$r_j = \frac{w_j \pi_{y^*}(y^* \mid x_j) \pi_{\xi^*}(x_j)}{c \pi_{imp}(x_j)}.$$

There are two features that recommend the use of Gaussian quadrature schemes: one is that by aiming the importance density at the posterior for x^* we focus our emulator evaluations on inputs in the parts of the input space that are most interesting; the other is that by choosing an importance density that is almost proportional to the posterior, the approximation can be shown to be almost exact.

As with the random prior sample, to approximate $|\Omega|_{\pi_{x^*}}$ here we sort the integration nodes into decreasing order with respect to their likelihoods and select the first few such the r_j values sum to more than α. The size of the HLCS is then a sum over this set of selected nodes,

$$|\Omega_{HLCS}|_{\pi_{x^*}} \approx \sum_{j=1}^{N} w_j \frac{\pi_{x^*}(x_j)}{\pi_{imp}(x_j)} \mathbf{1}(x \in \Omega_{HLCS}).$$

Next, we look at a way to summarise more information from the posterior for x^* in a natural fashion. The summary we adopt is motivated by the affinity of Gaussian quadrature with the estimation of moments. Specifically, we approximate the first few moments of the posterior with the quadrature scheme and, to make sense of them, associate them with an Gram-Charlier A (GCA) series, which we interpret as a sketch of the posterior density. The series is not guaranteed to constitute a valid probability density function because it can take negative values, but we can still use it to produce approximate plots, approximate integrals and approximate MCMC algorithms so long as we perturb the negative densities to a small positive value. The particular form of GCA series we concentrate on acts like a functional expansion of the posterior density about a Gaussian with the posterior's first two moments.

For example, the GCA series approximation for a one-dimensional posterior π with mean and variance μ and σ^2 is

$$\pi_{x^*|\mathbf{Y}}(x) \approx \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)\left(1 + \sum_{k=3}^{N} [\vartheta]_k H_k\left(\frac{x-\mu}{\sigma}\right)\right).$$

The H_k functions here are the normalised probabilists' Hermite polynomials. Calculating the series' parameters requires that we first estimate the posterior mean and variance, then use these to estimate the coefficients ϑ. The reason for introducing the Hermite polynomials is that their orthonormality properties imply,

$$\int \pi(x \mid y) H_k\left(\frac{x-\mu}{\sigma}\right) \mathrm{d}x \approx [\vartheta]_k,$$

so that the series parameters are particularly straightforward to compute:

$$\mu \approx \sum_{j=1}^{N} r_j x_j, \tag{4.113}$$

$$\sigma^2 \approx \sum_{j=1}^{N} r_j (x_j - \mu)^2, \tag{4.114}$$

$$[\vartheta]_k \approx \sum_{j=1}^{N} r_j H_k\left(\frac{x_j - \mu}{\sigma}\right). \tag{4.115}$$

The GCA series approximation arising from estimates (4.113)-(4.115) ought to paint an accurate picture of $\pi(x \mid y)$ when it is almost normal, and small values of β produce only minor corrections for its shape.

In D dimensions we parameterise the GCA series approximation of a posterior distribution with mean vector μ and variance matrix Σ in terms of its principal component scores,

$$\pi(x \mid y) \approx (2\pi)^{-D/2}|\Sigma|^{-1/2}\left(\prod_{d=1}^{D} \exp\left(-\frac{[u]_d^2}{2}\right)\right)\left(1 + \sum_{k=3}^{N} \sum_{\sum i_d = k} [\vartheta]_{(i_1,\ldots,i_D)} \prod_{d=1}^{D} H_{i_d}([u]_d)\right),$$
$$u = \mathbf{\Lambda}^{-1/2}\mathbf{U}^T(x - \mu),$$

where $\mathbf{\Lambda}$ and $\mathbf{U}$ are the $D \times D$ matrices of eigenvalues and eigenvectors of Σ. This normalised parameterisation is also the key to iteratively improving the accuracy of the quadrature estimates more generally, because once we have (potentially crude) estimates for the mean and variance of π we may recompute the set of quadrature node locations by transforming the grid whose importance density is a unit multivariate normal density from the space of the principal component scores, u, back up into the input space of x.

Before we move on we reiterate that references to the ‘integration nodes’ in this section, and those that follow, refer to the arrays described above: either the random sample from the prior or the sparse grid. They are distinct from the basis nodes, which are the locations of the inducing variables and the centres of the Cholesky-basis functions. Additionally, the integration node with the greatest posterior density value will be referred to as the MAP estimate for x^*, and denoted x^*_{MAP}.

4.4.3 Calibrating the covariance parameters

In the examples to come, we will choose a correlation function for the climate of the form

$$k_{c\xi}(\xi, \xi') = \exp\left[-(\xi - \xi')^T \mathbf{L}^{-1} \mathbf{L}^{-T} (\xi - \xi')\right],$$

because it is extremely quick to compute and allows us to describe correlation lengths, which accommodate ridges in the climate surface that are not aligned to the input axes. The matrix $\mathbf{L}$ here is an upper triangular matrix which fulfils a role analogous to the parameter l_{SE} in (2.41) for the squared exponential autocovariance function in one dimension. This parameterisation of the correlation lengths ensures that the product $\mathbf{L}^{-1}\mathbf{L}^{-T}$ is PSD, and that consequently the covariance matrices derived from the the correlation function are PSD too. We may also choose to modify the parameterisation slightly by insisting that the diagonal elements of $\mathbf{L}$ are strictly positive, with the effect that the parameterisation becomes a bijection onto the set of PSD matrices.

The pursuit of inferences for covariance parameters has motivated a great deal of interesting research, and the construction of clever algorithms. We can, like Browne [7] for example, construct an RWM algorithm that jumps from one proposal to another using a Wishart distribution whose mean is the current state of the chain, thus staying within the set of PSD matrices. We can try to reparameterise the correlation length matrix like Pinheiro and Bates[39] in an attempt to find opportunities to exploit our intuition or prior knowledge for the field, if we have any. We can also try to steer our algorithm around the set of PSD matrices by computing the gradient of the likelihood with respect to the correlation length matrix, as recommended by MacKay[30]. We find the random walk is aimless and slow, and that the reparameterisation and the differentiation calculations can be fiddly and prone to human error. The mathematical and computational efforts required to overcome these difficulties, by themselves, would not represent sufficient reasons to abandon an investment in sophisticated algorithms, but they are compounded by the suspicion that formal searches for $\mathbf{L}$ are likely to represent misplaced effort since the system is not a Gaussian process with the specified autocovariance function. Furthermore, we will only be able to use an approximation to the full Gaussian process anyway. Our priority is to move away from potentially inappropriate initial estimates without introducing excessive complication or opportunities for errors in our calculations. So our preferred way forward is to strive to make likelihood evaluations for the emulator as fast as possible and to use a robust, brute-force Nelder-Mead optimiser on a posterior arising from an informative prior that protects against over-fitting.

4.4.4 Emulator validation

As our first diagnostic, we ought to assess the adequacy of the numerical approximation to the integral that defines the pivoting criterion in the modified Cholesky algorithm 3. To do this, we run the algorithm on the variance matrix for the climate values at the integration nodes. We should see the elements of pivoting criterion vector drop to small values well before the last of the integration nodes is selected. If this occurs there is some redundancy in the integration grid; if not, we have a sign that the integration nodes are positioned too sparsely and have left parts of the input space unaccounted for.

Next, we suggest inspecting leave-one-out diagnostics for simulations with reference to their marginal input coordinates and to their proximity to the basis nodes. The basis nodes are the conduits between which simulations can share information with each other. This will be complementary information if the emulator is well-suited to the data in question, or contradictory information if the emulator is significantly misspecified. Thus simulations close to many nodes ought be most useful for diagnosing a poorly fitting emulator and we suggest they are prioritised in checking procedures. We measure the proximity of a set of input values to the nodes with the trace of the variance resolved upon adjustment by the inducing variables:

$$\mathrm{Tr}\left(k(\xi, \mathbf{N})\mathrm{Var}\left(\beta\right) k(\mathbf{N}, \xi)\right). \tag{4.116}$$

The scalar quantity (4.9) from section 4.1.1, measured against the f-distribution that describes its prior, is a natural test statistic for the leave-one-out diagnostics.

4.4.5 Simulator validation

Although the fit of the emulator is important to us, the question of primary importance to the scientists and stakeholders involved is whether they ought to pay attention to the simulator, and whether it is sufficiently authoritative to inform predictions and policies.

It may be that we are particularly reluctant to use the language of probability when making explicit statements about x^* to these stakeholders. We may feel uncomfortable talking about the HLCS of section 4.4.2, for example, when our prior does not represent beliefs we would wish to be held to. The approach used in the history matching exercises [9] and [56], by members of the department at Durham University, is characterised by caution. Their methods involve the use of thresholds to partition the input space into the ‘ruled-out set’ (ROS) and the ‘not ruled-out yet set’ (NROYS). We label these sets Ω_{ROS} and Ω_{NROYS} respectively. The thresholds used tend to be based on probabilistic bounds provided by results such as Gauss’ inequality and Chebyshev’s inequality, described in appendix D.2.2 and D.2.3, which ought to be more forgiving to discrepancies than likelihoods based on the normal distribution.

The history matching strategy is emphatically not a fully Bayesian analysis, as it does not result in probabilistic statements for the location of x^*. The set of points not ruled out could be empty. We would then reject the idea that there is a best input and a best simulation without entertaining an alternative hypothesis. Another feature of the threshold strategy is that it focuses our attention on a point-wise plausibility measure, not on the plausibility of x^* being in a particular region.

The proportion of the set of a priori plausible inputs within the NROYS is a quantity with which we may communicate the degree to which a calibration exercise has identified plausible input values. For the approximation of the size of the NROYS we consider the same two methods described in section 4.4.2. With the first, we simply generate a large sample from the prior, count the number of sample members that pass the plausibility test and divide by the sample size.

The second method requires that we evaluate the discrepancy statistic at the integration nodes of a quadrature scheme and perform the following sum,

$$|\Omega_{NROYS}|_{\pi_{x^*}} \approx \sum_j w_j \frac{\mathbf{1}(x_j \in \Omega_{NROYS})\pi_{x^*}(x_j)}{\pi_{imp}(x_j)}. \tag{4.117}$$

We note that quantification of the NROYS in this way, measures its size with respect to the prior distribution for x^*. Being able to compare the history matching statistic (4.117) to the likelihood-based calculations of section 4.4.2 is part of our motivation for preferring the HLCS to the HPDCS.

Typically the test criterion takes the form of a normalised squared discrepancy along the lines of

$$d_{ROS}(x) = (y^* - \mathbb{E}_{\mathbf{Y}}(y(x)))^T \mathrm{Var}_{\mathbf{Y}}(y(x))^{-1} (y^* - \mathbb{E}_{\mathbf{Y}}(y(x))), \tag{4.118}$$

and the threshold is set with reference to an approximating distribution. For example, if we are willing to liken the imagined variability of (4.118) to a chi-squared distribution, we may choose to employ a three-sigma type rule and deem criterion values over $q + 3\sqrt{2q}$ to signify implausibility, where q is the dimension of the vector $y(x^*)$. Because plausibility is not conserved in the way probability mass is, the NROYS is indicative simultaneously of the degree of uncertainty reduction for x^* given the model and for the validity of the model. The latter information is normalised away in the calculation of the posterior credible sets.

4.5 Chapter summary

In this chapter we have reviewed the NIG model and NIW models, which we can use to learn about the smoothness of the climate as well as its values. Furthermore, we have incorporated it into a Gibbs sampler algorithm that serves to reduce the dimension of the regression, and accommodate missing data. Still, the method is too demanding to be satisfactory and so prompts a move to more radical approximations.

Our next model, the importance-weighted eigen-basis emulator, represents a highly efficient and adaptive emulation tool. It addresses the problem of regressor selection in a way that utilises our intuition for the structure of the simulator output and the location of the most relevant parts of the input space. Computationally, the linear model arising from the eigen-basis approximation allows us to quickly assimilate data sets larger than those we could handle with the full covariance function. The optimality result, theorem 4.3.1, helps us to understand the importance of the eigen-basis, whose members we can view as harmonics, of increasing frequency and decreasing size, on an elastic membrane.

The Cholesky-basis approximation allows us to understand the eigen-basis approximation from another angle. The importance of the eigen-property is diminished and the basis nodes of the Nyström approximation, that were previously understood just as a set quadrature points, are reinterpreted as a set of indices for benchmark climate quantities that are representative of all the simulator outputs. The nodes of both approximations may be compared to the particle swarms that feature in particle filtering and population Monte Carlo methods. There the nodes, or more precisely the kernel functions centred on the nodes, provide a flexible and convenient way to parameterise distributions; here the kernels also define a distribution, for x^*, but do so from inside the likelihood for the system inputs.

The development of the PITC variant of the Cholesky emulator represents the culmination of our work in this chapter, it is the emulator in which we have most confidence with regard to its genuine usefulness to statisticians and modellers. We value it for its parsimonious use of approximation nodes; its interpretability in terms of inducing variables or benchmark values, which we find easier to think about than eigenfunctions of a particular operator; and its sophisticated treatment of the climate variance not accommodated by its basis functions, which renders its approximation to full models, specified using autocovariance functions, more accurate. As such, it is the emulator that we will focus on when we demonstrate the applicability of our methods to real data in the next chapter.

Chapter 5

Analysis of the FAMOUS data

In this penultimate chapter we apply the methods discussed in the previous chapters to a subset of simulations from FAMOUS, the climate simulator introduced in section 1.1.5.1. The chapter consists of two main sections that deal with the emulation of FAMOUS and its calibration given a series of synthetic system data.

The subset of the FAMOUS data we concentrate on here consists of 99 series of varying length, all computed under the same emissions scenario in which carbon dioxide levels increase exponentially for thirty simulator years before decreasing at the same rate back to their initial values.

We preprocess the data by linearly transforming all variables, inputs and outputs, to lie in the interval $[-1, 1]$ such that the most extreme values in the data set are mapped to the faces of the resulting hypercube. We do this primarily so that the emulator's correlation lengths are interpretable as proportions of the ranges for the input parameters that are considered a priori plausible.

Preliminary visual inspection of the series suggests that the effect of increasing the emissions is to reduce the AMOC flux, and the effect of decreasing them again is to allow the flux to return to approximately the value at which it began. This behaviour is visible, as a dip in the output over the first 60000 simulator days, in plot 1.1 of section 1.1.5.1, and 5.3(a)-5.3(c), which depict the fitted emulator and which we will discuss in detail in section 5.1.4. The flux's return is characterised by an over-adjustment, which sees it rise about as far above its initial value as it dipped below, and by increased high- and medium-frequency variation. Ignoring this nonstationarity in the high-frequency behaviour leads to discrepancy statistics far exceeding their anticipated bounds. To be precise, it proves impossible to produce diagnostic plots equivalent to 5.5(b) in which discrepancies for both the longer and shorter series lie mostly within the guidelines informed by the chi-squared distribution, which we discuss shortly.

Our response is to allow for a step change in the weather covariance parameters at the time the emissions begin to decrease, which coincides approximately with the flux's minimum values. We refer to times before and after the step change as past and future; defining sets $\Omega_{t,p}$ and $\Omega_{t,f}$ as those containing these time points respectively, and adopting the subscripts p and f to label the covariance parameters during these periods.

We are unable to include real system data in this example as the simulated emissions scenario is fictional. Even if it were not, relating the simulator's flux to the flux measured by an array of buoys, the primary source of such data, would represent another statistical challenge that would only obscure the emulation and calibration methods we intend to demonstrate. Instead, we gather a time series of 'fake' system data, denoted y^*, simply by removing one simulation from the 99 before we begin the analysis. Later, in section 5.2.2, this fake data is modified in order to mimic the effect of discrepancy between the simulator and the system.

5.1 Emulating FAMOUS

Throughout this chapter we focus exclusively on emulating and calibrating FAMOUS using the PITC variant of the Cholesky-basis model described in section 4.4. We make this decision because, as discussed in sections 4.2 and 4.3 respectively, application of the NIW model is only really practical for small data sets because of its computation-intensive fitting procedure; and the eigen-basis model and FITC Cholesky-basis model produce inferior approximations to specified autocovariance functions because of their crude treatment of approximation residuals, while also, in the case of the eigen-basis model, incurring higher computational costs.

The simulator's input parameter consists of three scalar quantities: a solar constant, which parameterises the energy entering the atmosphere from the sun; a diffusion coefficient, which affects the diffusive mixing of layers of water; and a cloud entrainment coefficient, which affects cloud formation and growth. We label them x_s, x_d, and x_e and order them as elements of the vector

$$x^T = (x_s, x_d, x_e).$$

5.1.1 Defining the covariance structure for the full emulator

We construct the full model for the flux as a sum of climate and weather terms,

$$y(t, x, v) = c(t, x, v) \oplus w(t, x, v),$$

and relate the system and simulator fluxes by considering them as special cases of the same function,

$$y_{sys}(t, x) = y(t, x, v^*), \qquad\qquad y_{sim}(t, x) = y(t, x, \hat{v}).$$

The high-frequency variation of the FAMOUS data is very sensitive, a diagnosis we make on the basis of replicate inputs submitted to CPDN in anticipation of crashes. They reveal that calculations that ought to have been identical produced substantially different outputs. This is thought to have resulted from the simulations being farmed out to machines working at different precision or from corruptions to files as they were transferred between machines. With this in mind, we specify the weather trend as completely uncorrelated between input coordinates and between points in the discrepancy space. Additionally, we specify that the autocovariance function for the climate trend factorises into components as follows,

$$\mathrm{Cov}\left(c(t', x', v')\,,\ c(t'', x'', v'')\right) = k_{cx}((t', x'), (t'', x''))k_{cv}(v', v''), \tag{5.1}$$

$$\mathrm{Cov}\left(w(t', x', v')\,,\ w(t'', x'', v'')\right) = \sigma_w^2 k_{wt}(t', t'')\delta_{x',x''}\delta_{v',v''}. \tag{5.2}$$

We choose a squared exponential autocovariance function, which is very fast to compute, to describe the climate trend because we need to evaluate it very often when calibrating its parameters. For the weather autocovariance function we choose a Matérn function with the intention of achieving greater precision in our specification of the high frequency weather signal. Since we model the weather as uncorrelated between series and as possessing the same variance specification across the input space, the Matérn covariances only need to be computed once for a full series and stored for reference. While we consider just one simulator and one system, parameters for k_{cv}, the covariance function over the discrepancy space, are hopelessly confounded with v^*. For this reason we combine their roles in the parameter ρ^*. Explicitly, the functions appearing in (5.1) and (5.2) are given by

$$k_{cx}((t',x'),(t'',x'')) = \sigma^2_{c\xi} \exp\left[-(\xi' - \xi'')^T \mathbf{L}^{-1}\mathbf{L}^{-T}(\xi' - \xi'')\right],$$

$$k_{cv}(\hat{v}, v^*) = \rho^*,$$

$$k_{wt}(t',t'') = \begin{cases} \sigma^2_{wp} k_{Mat}(|t'-t''|; u_{wp}, \nu_w) & \text{for } t',t'' \in \Omega_{t,p}, \\ \sigma^2_{wf} k_{Mat}(|t'-t''|; u_{wf}, \nu_w) & \text{for } t',t'' \in \Omega_{t,f} \\ \sigma_{wp}\sigma_{wf} k_{Mat}(|t'-t''|; \sqrt{(u^2_{wp}+u^2_{wf})/2}, \nu_w) & \text{for } t' \in \Omega_{t,p}, t'' \in \Omega_{t,f}, \\ \sigma_{wp}\sigma_{wf} k_{Mat}(|t'-t''|; \sqrt{(u^2_{wp}+u^2_{wf})/2}, \nu_w) & \text{for } t' \in \Omega_{t,f}, t'' \in \Omega_{t,p}. \end{cases}$$

where the weather covariance across the transition between past and future periods is informed by the nonstationary Matérn model whose validity as a covariance function is proven in [53].

5.1.2 Preliminary parameter estimation

Our initial specifications of the covariance parameter values for the full model, based on our intuition for the simulator and labelled with the subscript (*init*), are:

$$\mathbf{L}_{(init)} = \begin{pmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 1.5 & 0 & 0 \\ 0 & 0 & 1.5 & 0 \\ 0 & 0 & 0 & 1.5 \end{pmatrix},$$

$$\sigma_{c(init)} = 1, \qquad \nu_w = 0.4,$$

$$\sigma_{wp(init)} = 0.02, \qquad \sigma_{wf(init)} = 0.05,$$

$$u_{wp(init)} = 0.02, \qquad u_{wf(init)} = 0.05.$$

The implication of the value of $\mathbf{L}_{(init)}$ here is that climate values on opposing sides of the simulator input domain are weakly, but not insignificantly, correlated. The smaller entry in the first column means that the climate values separated by a time period equivalent to a quarter of the total period considered have practically negligible correlation. The other covariance parameters are used to describe a weather component approximately 0.05 of the size of the climate variation with temporal autocorrelation such that values of the weather separated by one simulator time step have a correlation of about 0.5.

5.1.2.1 Constructing the basis for the approximate emulator

Given the covariance parameters, construction of the Cholesky-basis consists in selecting the nodes that define the locations of the inducing climate variables. To do so, we use algorithm 3 equipped with an integration grid produced from a uniform Sobol sequence over $[-1, 1]^4$ and equal integration weights.

The full set of 13740 simulator data points is not too large to handle with the algorithm, but the calculation does begin to slow down noticeably as the matrices involved grow to contain more than 1000^2 elements. We respond by partitioning the data arbitrarily into subsets of 600, and then thin each set using the modified Cholesky pivoting algorithm with a cut-off threshold of 1×10^{-4}. This leaves a set of 1249 nodes to which the thinning procedure is applied again to leave 194 nodes.

To illustrate the process, we present in figure 5.1 the maximum values of the pivoting criterion at each iteration of the algorithm. In tests with slightly perturbed covariance parameters, the thinning procedure consistently returns sets of 50 to 300 nodes with the majority of the thinning taking place in the time direction. The result of the thinning can be appreciated from figures 5.2(a) and 5.2(b), which show pairs plots of the candidate node locations and the subset of node locations that are retained in order to define the Cholesky-basis functions.

5.1.3 Adjusting the approximate emulator

5.1.3.1 Adjusting the covariance parameters

The PITC Cholesky-emulator inherits parameters, and the smoothness properties they imply, from the full model. Adjusting them here involves repeatedly fitting different Cholesky-basis emulators derived from different covariance parameter choices for the

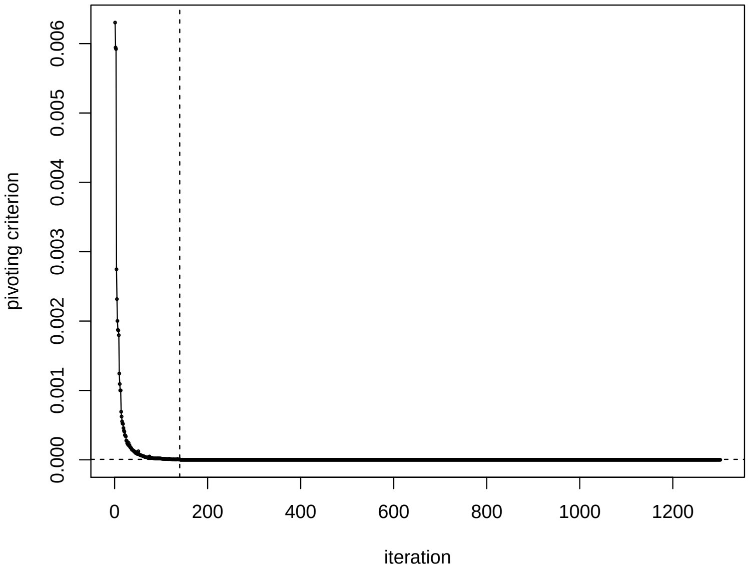


Figure 5.1: The maximum values of the pivoting criterion for the modified Cholesky algorithm at each iteration. The dashed horizontal line marks the criterion cut-off value and the vertical line marks the point at which the cut-off occurs.

full model. To calculate likelihood values for a particular set of covariance parameters we associate the moments implied by the corresponding Cholesky-basis emulator with a multivariate normal distribution, and to moderate the likelihood we allocate $\mathbf{L}^T\mathbf{L}$ a Wishart prior and the variables σ_c, σ_{wp}, σ_{wf}, u_{wp} and u_{wf} gamma priors while holding the weather spikiness parameter, v_w, fixed:

$$\mathbf{L}^T\mathbf{L} \sim \mathrm{W}_{10}\left(\mathbf{\Psi} = \mathbf{L}_{init.}^T\mathbf{L}_{init.}\right), \tag{5.3}$$

$$\sigma_{c0}^2 \sim \mathrm{Gamma}\left(q_l = 0.1, q_u = 2\right), \tag{5.4}$$

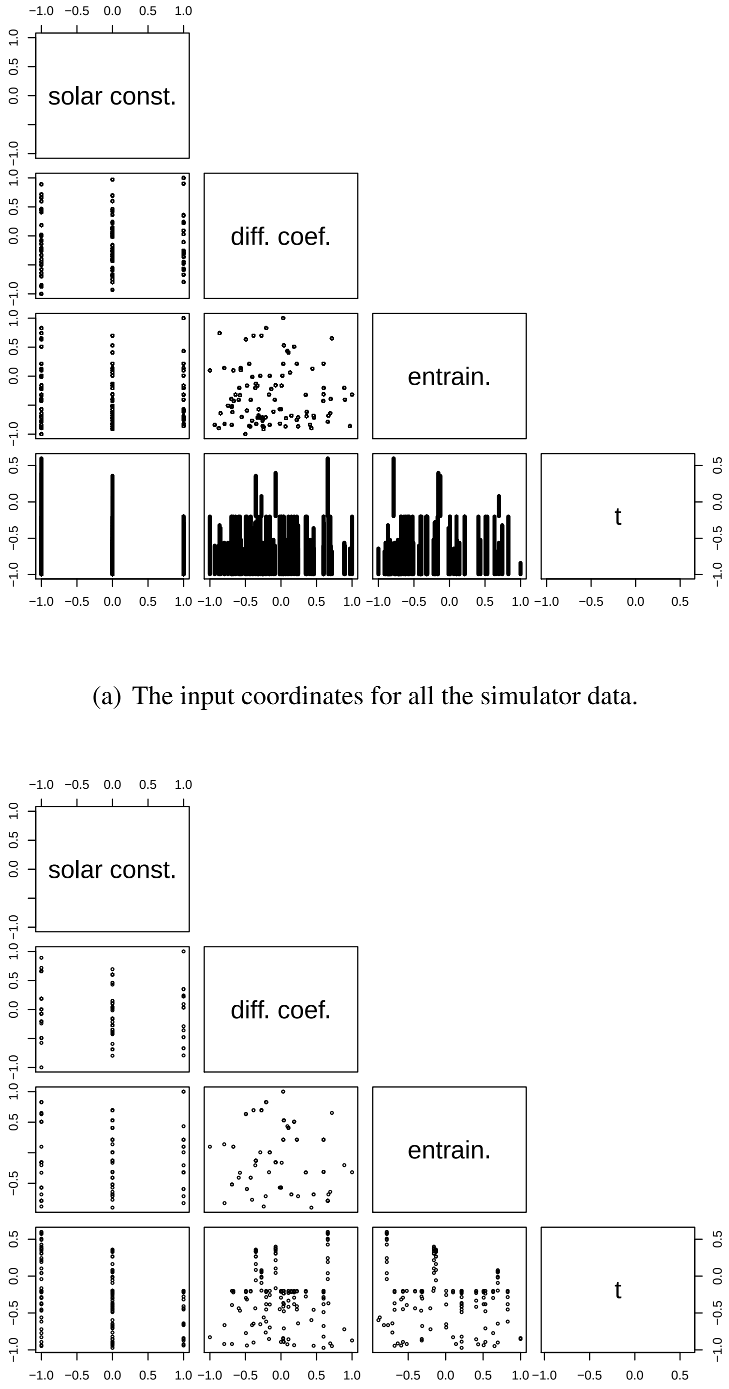
$$\sigma_{wp} \sim \mathrm{Gamma}\left(q_l = 0.01, q_u = 0.2\right), \tag{5.5}$$

$$\sigma_{wf} \sim \mathrm{Gamma}\left(q_l = 0.01, q_u = 0.2\right), \tag{5.6}$$

$$u_{wp} \sim \mathrm{Gamma}\left(q_l = 0.01, q_u = 0.2\right), \tag{5.7}$$

$$u_{wf} \sim \mathrm{Gamma}\left(q_l = 0.01, q_u = 0.2\right). \tag{5.8}$$

Note that in the priors (5.4)-(5.8) we have parameterised the Gamma distributions by their fifth and ninety-fifth percentiles, denoted q_l and q_u, rather than the standard shape and scale statistics, to better communicate the prior information that they encode.



(a) The input coordinates for all the simulator data.

(b) The thinned input coordinates.

Figure 5.2: Pairs plots of the candidate inducing climate variables, 5.2(a), and those that are selected to form the Cholesky basis, 5.2(b).

To determine the parameter values at which to evaluate the posterior and to guide us towards optimal values for them, we hand over a function returning the log-posterior to R's Nelder-Mead optimiser. The optimisation is slow, taking minutes or hours depending on number of basis members used, but it is stable and converges on a solution without identifying any numerical problems. The MAP estimates for the model with the initial node locations are:

$$\mathbf{L}_{(MAP)} = \begin{pmatrix} 0.35 & 0.98 & -0.08 & -0.62 \\ 0 & 2.52 & -0.65 & -0.16 \\ 0 & 0 & 2.38 & -0.60 \\ 0 & 0 & 0 & 1.25 \end{pmatrix},$$

$$\begin{aligned} \sigma^2_{c(MAP)} &= 0.25, & \nu_w &= 0.4, \\ \sigma_{wp(MAP)} &= 0.051, & \sigma_{wf(MAP)} &= 0.090, \\ u_{wp(MAP)} &= 0.02, & u_{wf(MAP)} &= 0.043. \end{aligned}$$

The optimised covariance parameters lead to an emulator that more strongly resists fitting large climate trends insofar as $\sigma^2_{c(MAP)}$ is considerably smaller than $\sigma^2_{c(init)}$. The off-diagonal elements of the optimised parameter $\mathbf{L}_{MAP}$ do not seem particularly large, calling into question whether allowing their values to vary from zero, which significantly increases the dimensionality of the optimisation problem, is really worthwhile.

The correlation length in the direction of the entrainment coefficient is decreased, which leads to a marginal posterior that is slightly less smooth and more tightly concentrated on the true value than that arising from the initial specification of covariance parameters. The correlation lengths in the solar constant, diffusion coefficient and time directions are all increased, which has the effect of producing more gradually changing climates, as well as removing a secondary mode in the marginal for the diffusion coefficient and flattening a large spike in the marginal for the solar constant. In figure 5.8 we have included plots of the posterior arising from the initial covariance parameter specification in order to demonstrate these features.

Repeating the node selection procedure with the adjusted covariance parameters results in a decrease in the number of nodes from 194 to 84. So in this case, the cost of optimising the covariance parameters is partially recouped by suggesting that the number of basis functions consistent with the initial covariance specification was excessively high.

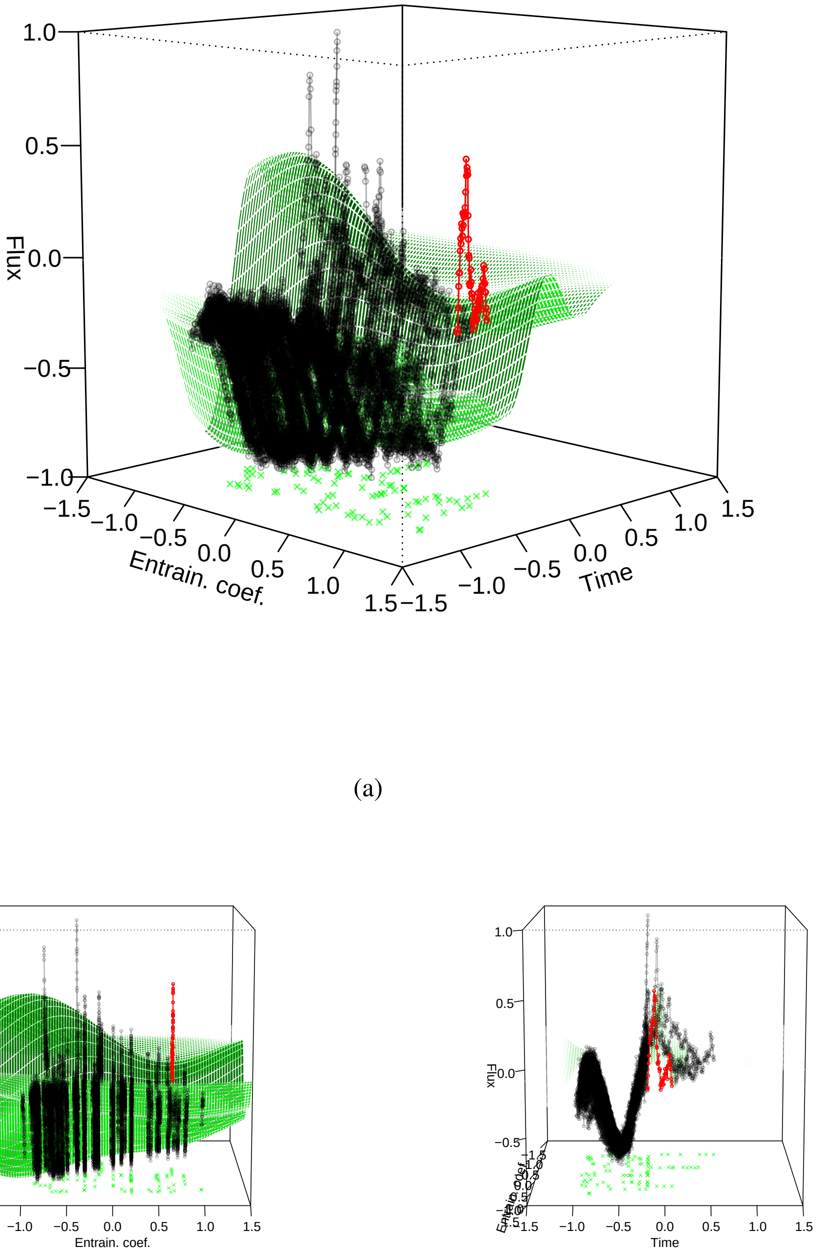
In table 5.1, we have included the calibration inferences resulting from the emulator with the initial specification of the covariance parameters to show the difference in the sizes of the NROYS and HLCS brought about by the optimisation. More important than the sizes, however, is the greater degree of confidence we have in the NROYS and HLCS arising from the better fitting model. Note that all further calculations, including the other numerical experiments described in table 5.1, and all further plots, unless explicitly labelled to the contrary, make use of the optimised covariance parameters.

5.1.4 Fitting the emulator

We now fit the emulator derived from the optimised covariance parameters by running through the set of simulator output time series, sequentially adjusting the expectation and the variance for the emulator's coefficients with the Bayes linear formulae. With this sequential approach, each adjustment involves the construction and inversion of a matrix no larger than the maximum of the length of a particular time series and the number of emulator basis functions. This is the key to making the calculations for the adjustment of the climate function, as well as the likelihood calculations required for the covariance parameter adjustment, tractable. Figures 5.3(a)-5.3(c) show the emulator's expectation of $c(t, (0, 0, x_e))$ given the FAMOUS simulations. It appears from the plots that time and the entrainment input variable account for the greatest part of the variation in the simulated data.

5.1.5 Emulator validation

The first diagnostic we examine is a simple histogram of the emulator's fitted residuals, shown in figure 5.4. Overlaid is the density function for the zero mean normal distribution with standard deviation equal to the standard deviation of the fitted residuals. The correspondence between the histogram and the curve does not suggest that a normal marginal for the weather terms is grossly inappropriate. It may be understood as suggesting that a



(a)

(b)

(c)

Figure 5.3: Plots of FAMOUS outputs for one forcing scenario overlaid with the adjusted expectation for $c(t,(0,0,x_e))$ as calculated with the Cholesky-basis. Time, the inputs, and the output variables were standardised over the full data set, including all scenarios, as a preprocessing step. On the floor of the plot are projections of the basis nodes locations. The series corresponding to the largest LOO discrepancy statistic is highlighted in red.

distribution with slightly higher kurtosis than the normal distribution is more appropriate, but this not a possibility we pursue.

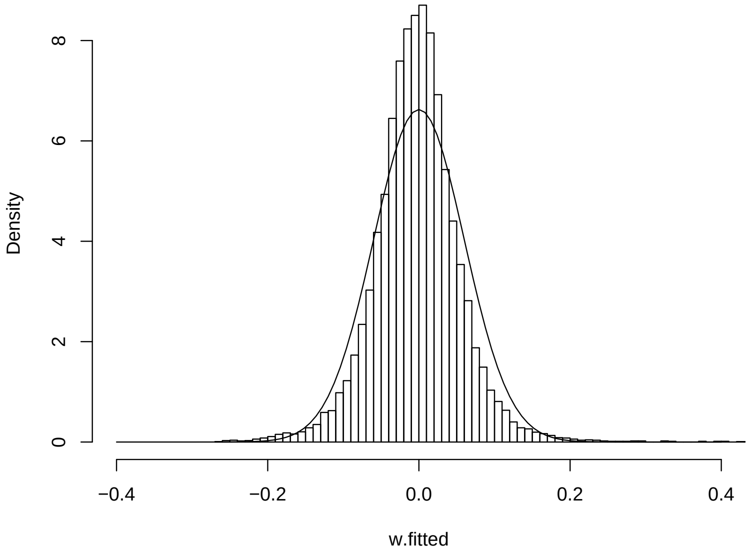


Figure 5.4: A histogram of the differences between the simulator outputs and the expectation for c given the simulator outputs.

Now we turn to examination of the emulator's leave-one-out (LOO) diagnostics. These are produced by running through the simulations, unlearning about them individually using the inverted-update, or downdate, equations and comparing their values to the downdated estimates for them. In figure 5.5(b) we take a look at the discrepancy statistics,

$$d_{loo}(\mathbf{Y}_i) = \frac{1}{|\mathbf{T}_i|}(\mathbf{Y}_i - \mathbb{E}_{\mathbf{Y}_{-i}}(y(\mathbf{T}_i, [\mathbf{X}]_{i,\cdot})))^T \mathrm{Var}_{\mathbf{Y}_{-i}}(y(\mathbf{T}_i, [\mathbf{X}]_{i,\cdot}))^{-1}([\mathbf{Y}]_{i,\cdot} - \mathbb{E}_{\mathbf{Y}_{-i}}(y(\mathbf{T}_i, [\mathbf{X}]_{i,\cdot}))),$$

where $|\mathbf{T}_i|$ is the length of the observed simulator time series with input $[\mathbf{X}]_{i,\cdot}$. The discrepancy for each series is plotted against the series length, and on top of this we have added lines corresponding to the fifth and ninety-fifth percentiles of the scaled chi-squared variables χ^2_n/n whose degree of freedom parameter is equal to the simulation length. Under the assumption that the emulator coefficients and the weather terms are well described by normal distributions with the specified moments, we would expect 0.9 of the LOO statistics to lie within these bounds. We also include in subfigure 5.5(a) the equivalent plot from the unoptimised Cholesky-basis emulator in order to illustrate the improved model fit to which we have referred in the previous subsections.

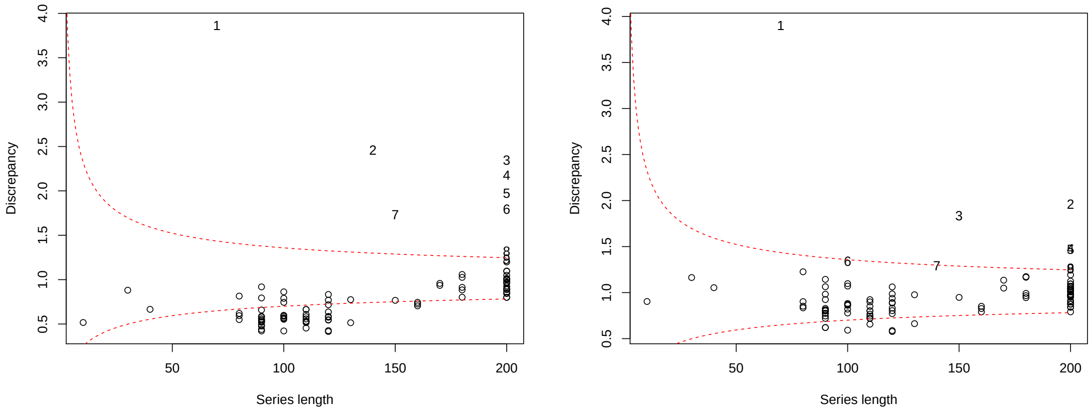
The largest discrepancy statistic, marked with a 1 in figures 5.5(b) and 5.6, and plotted in red in figure 5.3, is particularly extreme. It appears to arise from a post-bounce-back oscillation that is too rough to be accommodated by the climate trend and too large to be accommodated by the weather trend. Fortunately, exploratory re-calculations show that the discrepant series' inclusion or removal from the training data makes negligible difference to our inferences.

Clear trends in the LOO discrepancy statistics are not apparent in figure 5.6 as we vary either the solar constant, the diffusion coefficient or the entrainment coefficient. Figure 5.6(d) is constructed to alert us to the emulator fitting badly to simulations due to the location of its nodes. We would expect the discrepancies to be slightly larger for series that are more tightly constrained by their proximity to the basis functions and slightly smaller away from the nodes, where variation not captured by the basis is compensated for by an error term that is uncorrelated across the input space. These features are not clearly evident in the plot, neither does any other trend suggest that the locations of the series, relative to the locations of the basis functions, are related to the quality of the emulator's fit.

5.2 Calibrating FAMOUS

In our calibration analyses we strive to produce statistics that are concise and intuitively understandable. For this reason we focus our attention on plots, and on membership, of the HLCS and NROYS, and find they produce plenty of information for discussion. In the following sections, we use the term 'experiment' to refer to the examples, or numerical experiments, in which we specify different emulator structures of different values for the simulator discrepancy. Note also that we do not explicitly address observation or measurement error here on the understanding that the variance it introduces may be incorporated, as a nugget term for instance, into the system weather variance.

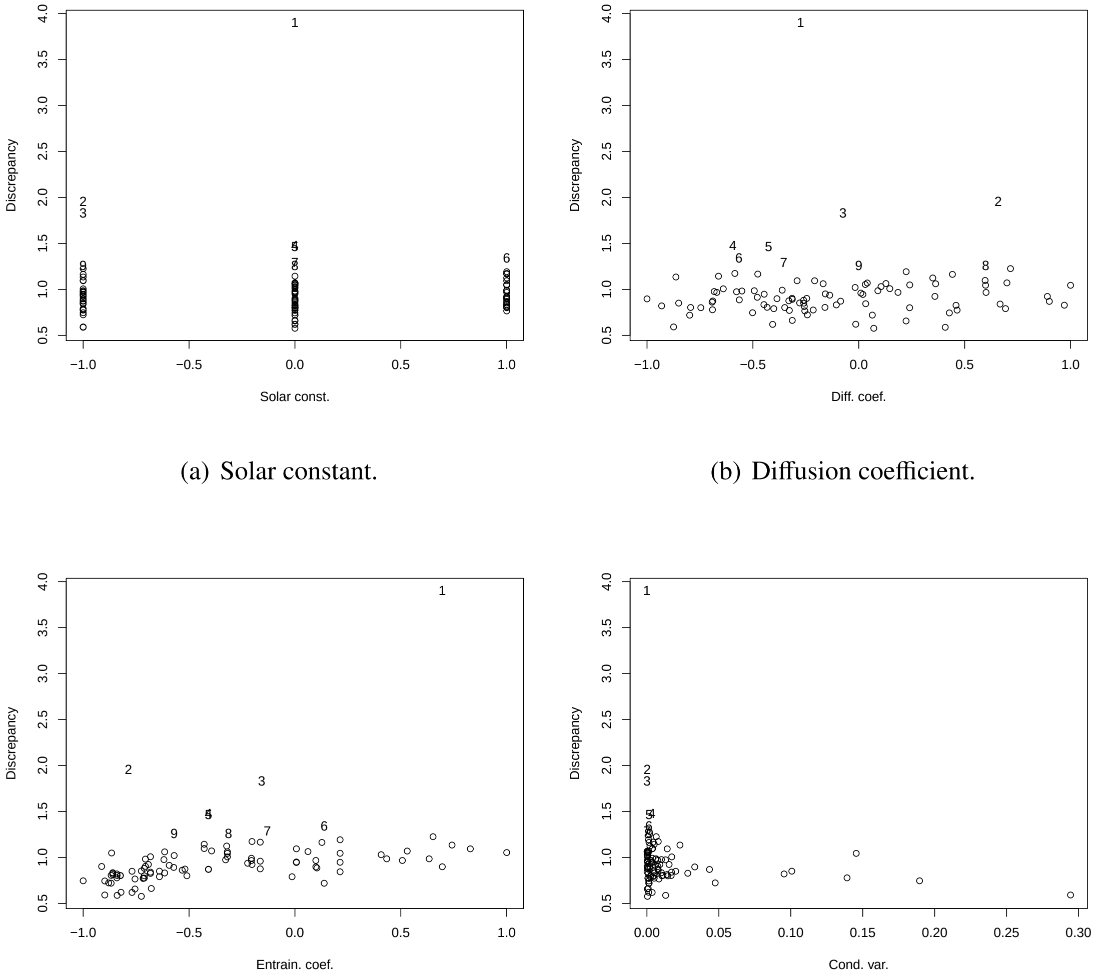
The prior for x^* that we employ here is uniform over $[-1, 1]^3$. This choice has the effect of removing the distinction between the HLCS and HPDCS discussed in section 4.4.2. While this uniform distribution is not explicitly recognised as a prior in the history



(a) Unoptimised emulator.

(b) Optimised emulator.

Figure 5.5: Plots of the LOO discrepancy statistics from the emulators derived from the initial (5.5(a)) and optimised 5.5(b) covariance parameters against the lengths of the corresponding FAMOUS time series. The plots also feature dashed lines, marking the fifth and ninety-fifth percentiles for scaled χ^2-variables, indicating the anticipated range of the discrepancy statistics given that the simulations are normally distributed. The greatest nine discrepancies, which are different for the optimised and unoptimised emulators, are plotted with the corresponding numerals.



(a) Solar constant.

(b) Diffusion coefficient.

(c) Entrainment coefficient.

(d) Node proximity.

Figure 5.6: Plots of the emulator LOO discrepancy statistics for the FAMOUS time series against the simulator inputs in subfigures 5.6(a)-5.6(c), and against the node proximity statistics, (4.116), in 5.6(d). The greatest nine discrepancies are plotted with the corresponding numerals so that they can be identified in all of the plots.

matching context, it plays a role similar to one when we use it to generate points at which to evaluate the implausibility measure, whose values inform the size and shape of the NROYS. The likelihoods that we use to calculate posterior densities are calculated using the multivariate normal density described by the emulator's moments.

The NROYS inclusion criterion we use is the discrepancy statistic

$$d_{ROS}(x) = (y^* - \mathbb{E}_{\mathbf{Y}}(y(\mathbf{T}^*, x)))^T \mathrm{Var}_{\mathbf{Y}}(y(\mathbf{T}^*, x))^{-1}(y^* - \mathbb{E}_{\mathbf{Y}}(y(\mathbf{T}^*, x))).$$

The statistic is compared to the chi-squared distribution with $|\mathbf{T}^*| = 200$ degrees of freedom. We set the implausibility threshold at the ninety-fifth percentile of this distribution, and so informally refer to the NROYS as a 95%-NROYS, despite the set's disassociation from explicit probability statements. Note that although we prefix both NROYS and the HLCS sets with 95%, the meanings of the percentages are not comparable. The 95% of the NROYS can be understood as a significance or a probability concerning the anticipated value of the quantity $y(\mathbf{T}^*, x^*)$, while the 95% of the HLCS describes a posterior probability concerning the location of x^*.

We also append to the prior sample the true input parameters for the simulation that plays the role of the fake system data. The inclusion of this extra sample member allows us to say whether or not the NROYS and HLCS contain the true input value.

5.2.1 Simulator calibration with no simulator discrepancy

In this section we present the results of our calibration procedure for the model in which ρ^* is equal to one, and the fake system data, $y^* = y(\mathbf{T}^*, x^*)$, takes the values of an unmodified FAMOUS simulation.

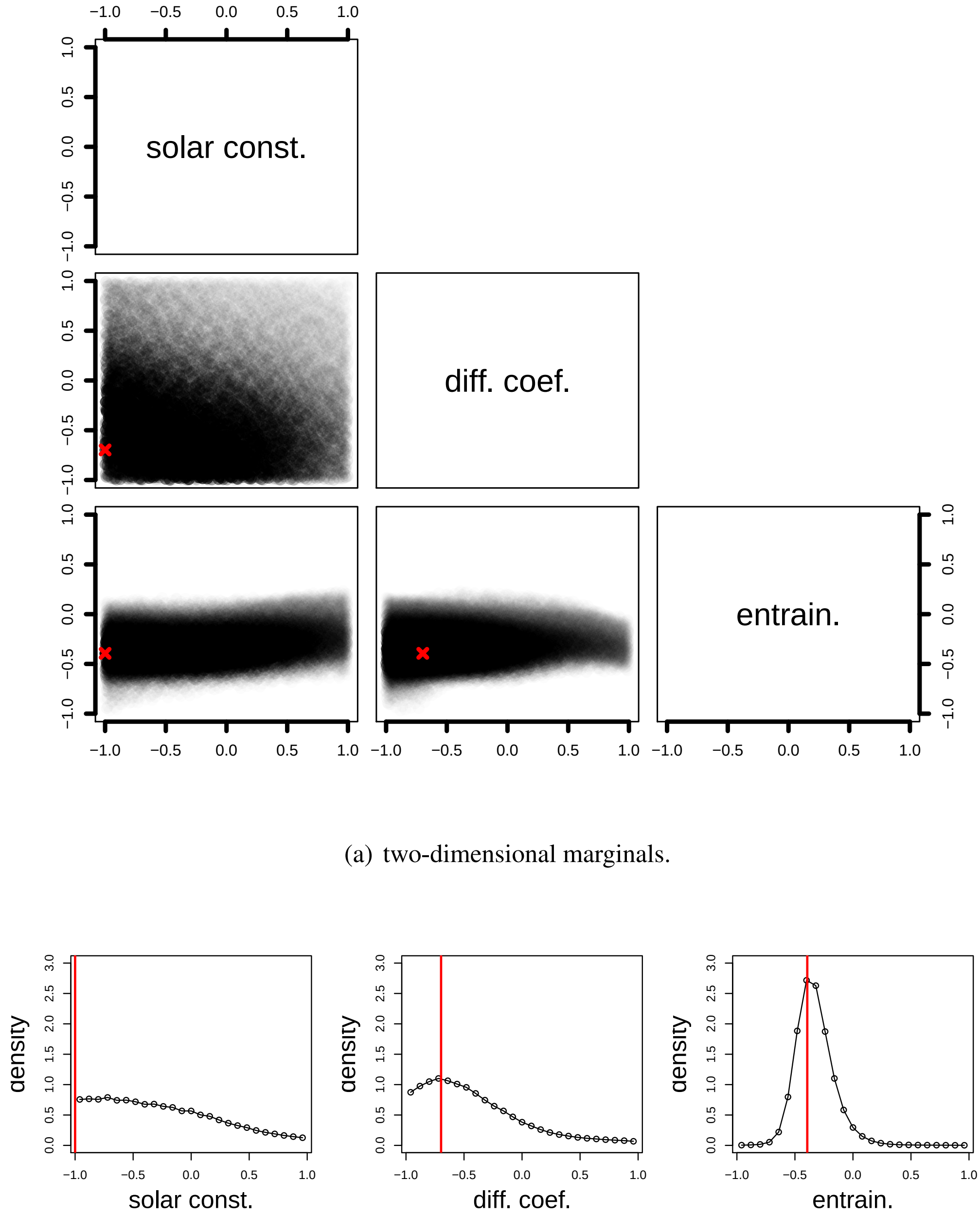
Figure 5.7(a) shows a pairs-plot which was made by generating a sample of 50000 input parameters from a uniform Sobol sequence over $[-1, 1]^3$ and shading them to a degree proportional to the posterior density for x^*. We will refer to such plots as depth plots; they give an impression of the two-dimensional posterior marginal distributions for the components of x^*. Figure 5.7(b), meanwhile, shows approximate one-dimensional marginals produced by binning the sample and summing the posterior densities within each bin. The plots indicate that the emulator is successfully picking up on the effects of each of the input parameters and attributing high posterior density to the true input value for the fake system data.

For interest, we include in 5.8 the equivalent figure produced using the unoptimised emulator. While the location of the entrainment parameter is still well identified here, the inferences for the other parameters are markedly different from those of the optimised emulator. Specifically, the rougher nature of the unoptimised emulator's response surface appears to over-fit to the training data, with the effect that the fake system data is identified with particular training series more than the average of training series in a particular region of the input space. An especially serious result of this effect is the apparent misidentification of the solar constant.

In figure 5.9 we present a depth plot in the same format as 5.7(a), but here the opacity of the plotted points takes only two values: 0.01 if the input parameter falls below the implausibility threshold and zero if it exceeds it. The role of this plot is to provide an impression of the NROYS. The true parameter vector does belong to the NROYS and the HLCS, which is indicative of the emulation and calibration procedures being effective. The NROYS and HLCS are complementary here, in the sense that the significance of the credible region is contingent on the plausibility of the system data having been produced, or explained, by the emulator at all. A very small credible region could be produced, for example, by an emulator that fits very badly to the system data for all values of x^*. When this is the case and we use a likelihood function whose tails decay very quickly, the posterior mass quickly accumulates at the least bad fit. This is how a small HLCS could be interpreted either as a sign of success, insofar as the calibration leads to inferences of high precision, or as a sign of failure, insofar as the emulator bears little relevance to the simulator. For our calibration of the FAMOUS inputs, the large NROYS reassures us that the posterior density is not undermined by a badly fitting emulator and our identification of the approximate location of x^* is defensible.

The NROYS for this example rules out only approximately 35% of the input space. In comparison to other history matching exercises, such as Vernon's [56] for a cosmological simulator, this figure may not seem particularly impressive. Our comparatively large NROYS is a result of the climate variation attributable to the input parameters being small relative to the size of the weather terms.

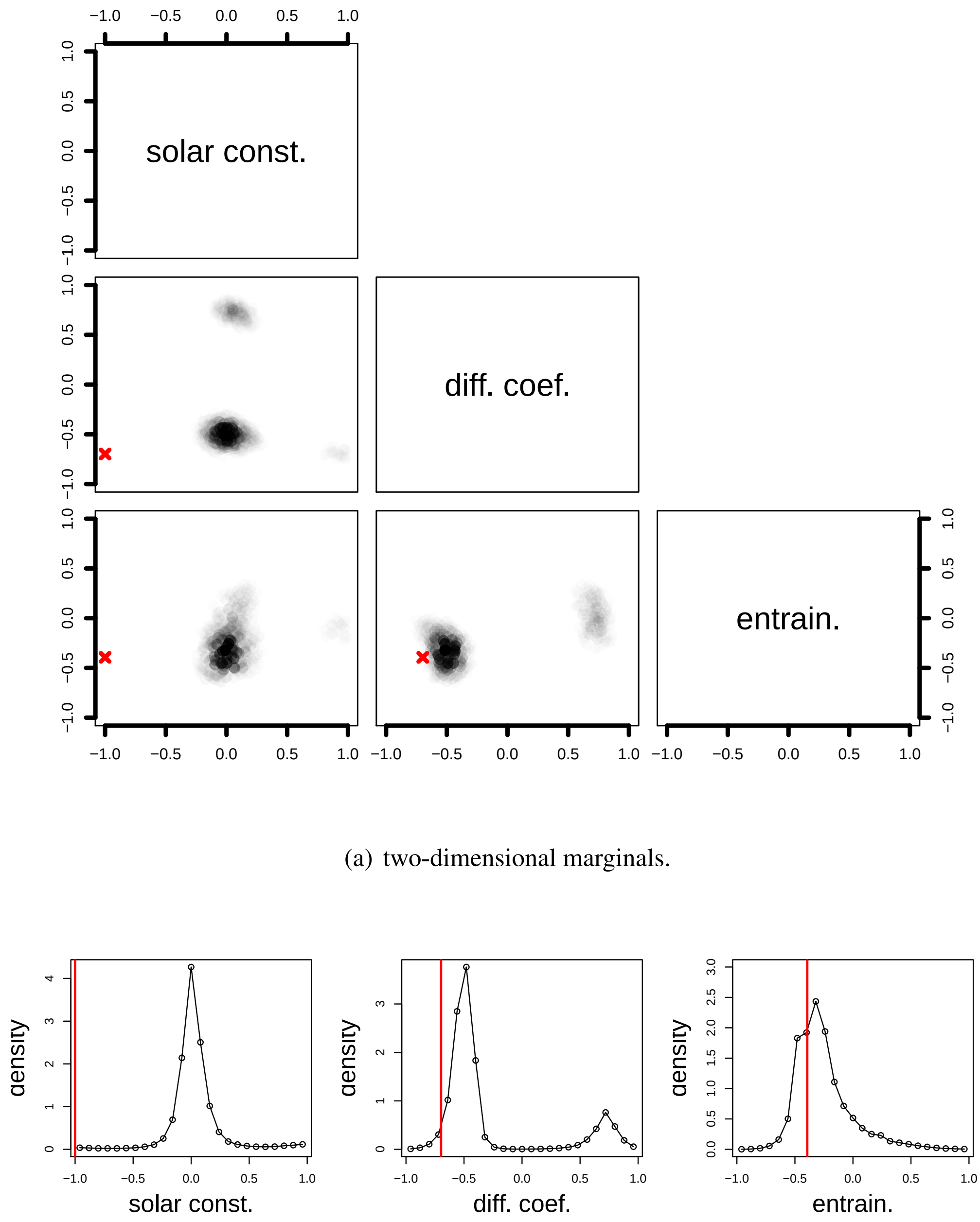
Both the NROYS and HLCS appear to consist of simply connected sets, which is an



(a) two-dimensional marginals.

(b) one-dimensional marginals.

Figure 5.7: Plots illustrating the posterior marginal densities for x^* for the experiment in which there is no simulator discrepancy, so that the simulator climate at x^* is the same as the system climate. Red crosses in subfigure 5.7(a) mark the two-dimensional projection of true input value, while red vertical lines in subfigure 5.7(b) mark its one-dimensional projections.



(a) two-dimensional marginals.

(b) one-dimensional marginals.

Figure 5.8: Plots illustrating the posterior marginal densities for x^* for the experiment in which there is no simulator discrepancy as calculated with the unoptimised Cholesky-basis emulator. Red crosses and lines mark the projected values of the true input value.

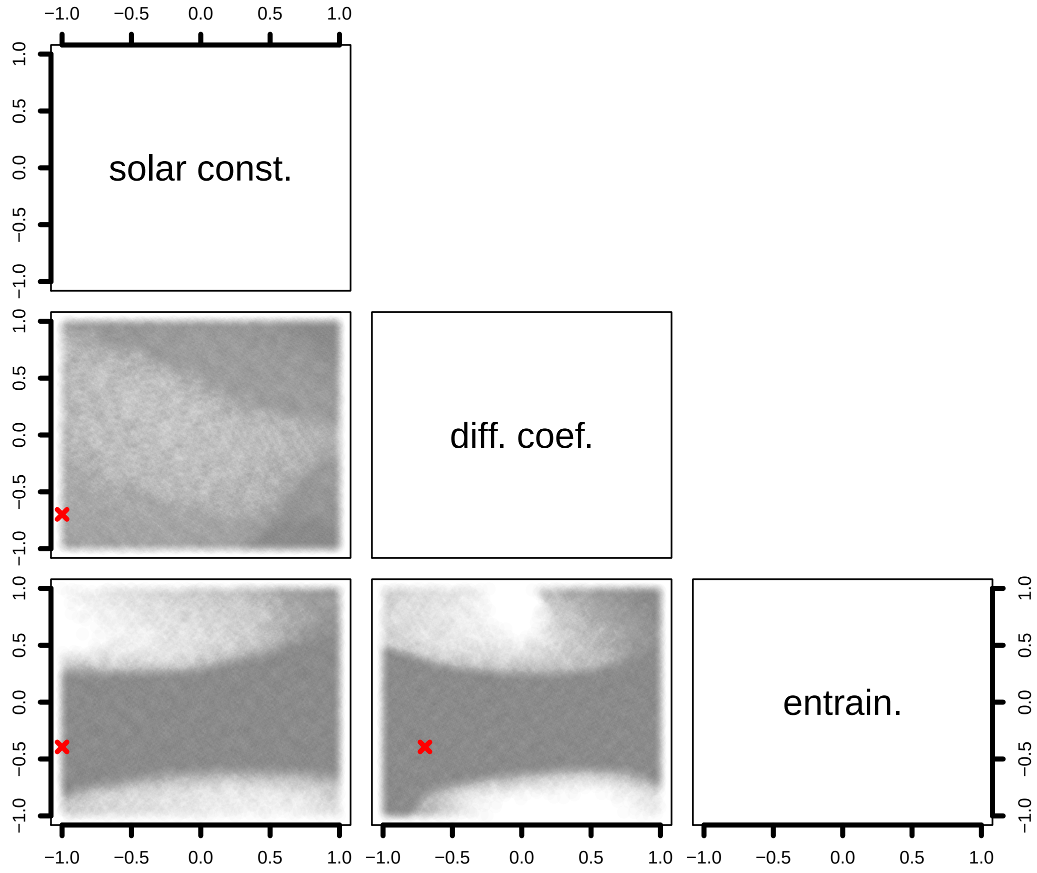


Figure 5.9: A depth plot of the NROYS for the experiment without simulator discrepancy. The red crosses show the projections for the true value of x^*.

observation that cannot be verified using our pointwise evaluations of the discrepancy or likelihood, but is strongly supported by the plots in figures 5.7(a) and 5.9. Furthermore, the marginal posterior densities in figure 5.7(b) also appear to be uni-modal and otherwise well-behaved.

Next, we test the practicality and usefulness of the GCA series approximation technique, described in section 4.4.2, for summarising the emulator's posterior density for x^*. We start by generating a sparse grid integration scheme of 8031 nodes, based on a uniform importance density, which we use to estimate the posterior mean and variance, denoted μ and Σ respectively. With these estimates we define the transformed vector variable, whose x dependence we emphasise in (5.9) but take as implicit in the subsequent expressions,

$$u(x) = \mathbf{\Lambda}^{-1/2}\mathbf{U}^T(x - \mu). \tag{5.9}$$

We then estimate the ten coefficients for the GCA series' third order correction terms by

$$[\beta]_{i_1,i_2,i_3} \approx \sum_j r_j H_{i_1}([u(x_j)]_1) H_{i_2}([u(x_j)]_2) H_{i_3}([u(x_j)]_3) \qquad \text{for } i_1 + i_2 + i_3 = 3,$$

where r_j are the posterior integration weights whose calculation is described in section 4.4.2, and use them to define the GCA density approximation:

$$\pi(x^* \mid y^*, \mathbf{X}, \mathbf{Y}) \approx (2\pi)^{-3/2}|\Sigma|^{-1/2} \exp\left(-\frac{1}{2}u^T u\right)\left(1 + \sum_{i_1+i_2+i_3=3} [\beta]_{i_1,i_2,i_3} \prod_{d=1}^{3} H_{i_d}([u]_d)\right).$$

The GCA series approximation with polynomial correction terms up to Nth order serves to condense our description of a D-dimensional posterior to a set of

$$\binom{N+D}{D} - 1$$

numbers. So here our approximation is derived from 19 estimated parameters. We contrast this to the set of 50000 input coordinates and likelihoods from the Sobol sample, which we would otherwise store as a record of our calibration calculation, and which we would use to approximate properties of the posterior for x^*. In this experiment, pointwise evaluation of the approximate density is also approximately 500 times faster than evaluating the likelihood for x^* using the emulator. The speed of the approximate density function is attributable to the vectorisation possible in implementing the GCA series approximation's calculations, and to its avoidance of any matrix inversions.

While the Cholesky-basis emulator has been constructed to be fast and efficient, and evaluation of the likelihood over the large Sobol sample takes only around a minute, producing estimates with the GCA series approximation is effectively instantaneous. We are aware that the GCA series approximation represents another layer to an already high stack of approximations (we have a non-parametric second-order stationary field emulating the simulator, which is approximated by a finite-basis linear model), but while the speed and storage benefits of the GCA approximation are so great and the posterior is well-behaved, so that the approximation is good, the GCA series remains a valuable tool. To corroborate the quality of the approximation, in figure 5.10 we present plots of its one-dimensional marginals, which we compare to those from the Sobol sample calculation in figure 5.7(b).

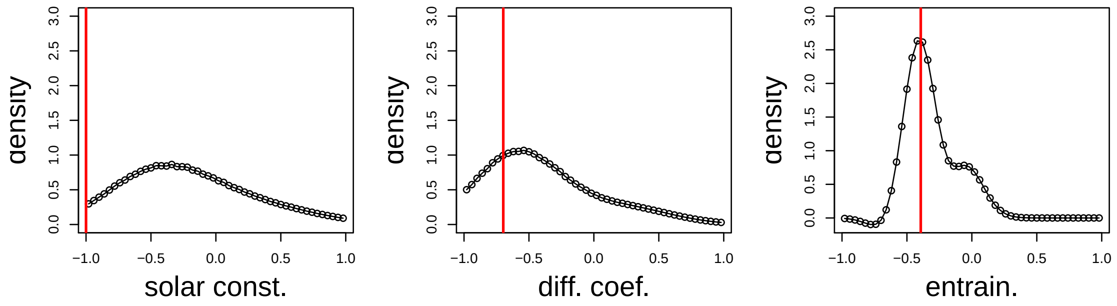


Figure 5.10: The marginal posterior densities for x^* according to the GCA series expansion with third order correction terms.

With the last experiment we perform in this section we compare our calibration calculations to one in which the time series structure of the FAMOUS data is ignored. Specifically, we construct scalar-output emulators for the tenth, one hundredth and two hundredth time steps corresponding to standardised times −0.96, −0.6 and −0.24. These variables approximately relate to the initial or baseline fluxes, to their lowest values and to the strength they return to when the emissions are reduced. Mathematically, the emulators are simply independent linear regression models in the input variables. The standard deviation of the models' *iid* error terms is set at 0.05 while the prior variance for the regression coefficients is the four by four identity matrix. This model for the flux is extremely simple, making it easy to code and fast to execute. If the input dependencies of the time points are characterised by strong, unaligned linearities then we would expect the scalar emulators to pin down x^* fairly well. These provisos are not met by the FAMOUS data however. The first of our findings from the experiment, referred to in table 5.1 as the 'scalar emulators experiment', is that the NROYS resulting from the joint discrepancy measure includes the true input value but excludes only 2% of the candidates. The second is that the HLCS excludes the true input along with 0.38 of the candidates. We judge the three linear emulators to be deficient insofar as they cannot capture a significant amount of the variation in the data, and the information that they do capture, in this case, is misleading.

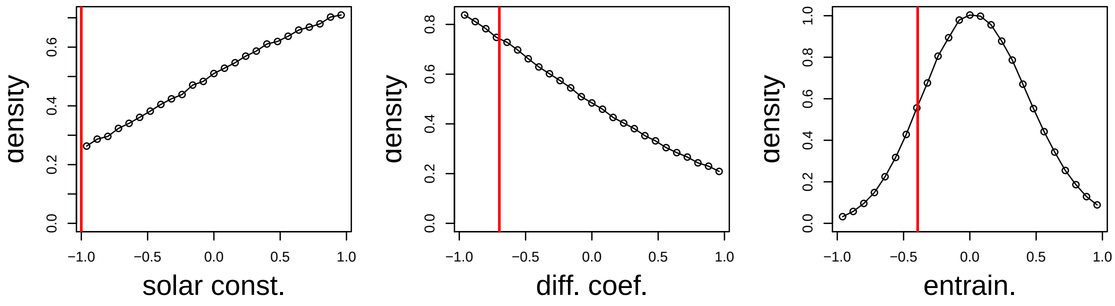


Figure 5.11: Approximate posterior marginals for the components of x^* as calculated with three scalar-output emulators.

5.2.2 Simulator calibration with simulator discrepancy

We now simulate system observations to calibrate to by altering our fake system time series in a way that mimics simulator discrepancy. Specifically, we treat the MAP estimate $\mathbb{E}\left(c(t, x^*_{MAP}, \hat{v}) \mid y^*, \mathbf{Y}, \mathbf{X}\right)$ from the previous experiment as if it were an observation of $c(t, x^*, \hat{v})$, and simulate $c(t, x^*, v^*)$ conditioned on it using (5.10). To this doubly fake climate data we add a weather term sampled from the zero mean multivariate normal distribution (5.11),

$$c(t, x^*, v^*) \sim \mathrm{N}\left(\rho^* \mathbb{E}\left(t, c(x^*_{MAP}), \hat{v}\right), (1 - \rho^{*2})\mathbf{K}_c\right), \tag{5.10}$$

$$w(t, x^*, v^*) \sim \mathrm{N}\left(0, \mathbf{K}_w\right). \tag{5.11}$$

We repeat the experiment three times, with ρ^* equal to 0.99, 0.95 and 0.9, each time using the same seed in our simulations from (5.10) and (5.11) so that the system weather terms are identical and the system climate terms are equivalent to different weighted averages of $\mathbb{E}\left(c(t, x^*_{MAP}, \hat{v}) \mid \mathbf{Y}, \mathbf{X}\right)$ and the same sample from $\mathrm{N}(0, \mathbf{K}_c)$. These artificial system time series are plotted in figure 5.12, demonstrating the degree and type of simulator discrepancy implied by the model. As $\hat{v}$ and v^* move further apart and ρ^* decreases, the simulator data becomes less informative for the climate data and the calibration inference becomes less precise. With figures 5.13-5.15 we show the approximate rate at which precision is lost. We see that as ρ^* falls from 1 to 0.9 the depth plots quickly become

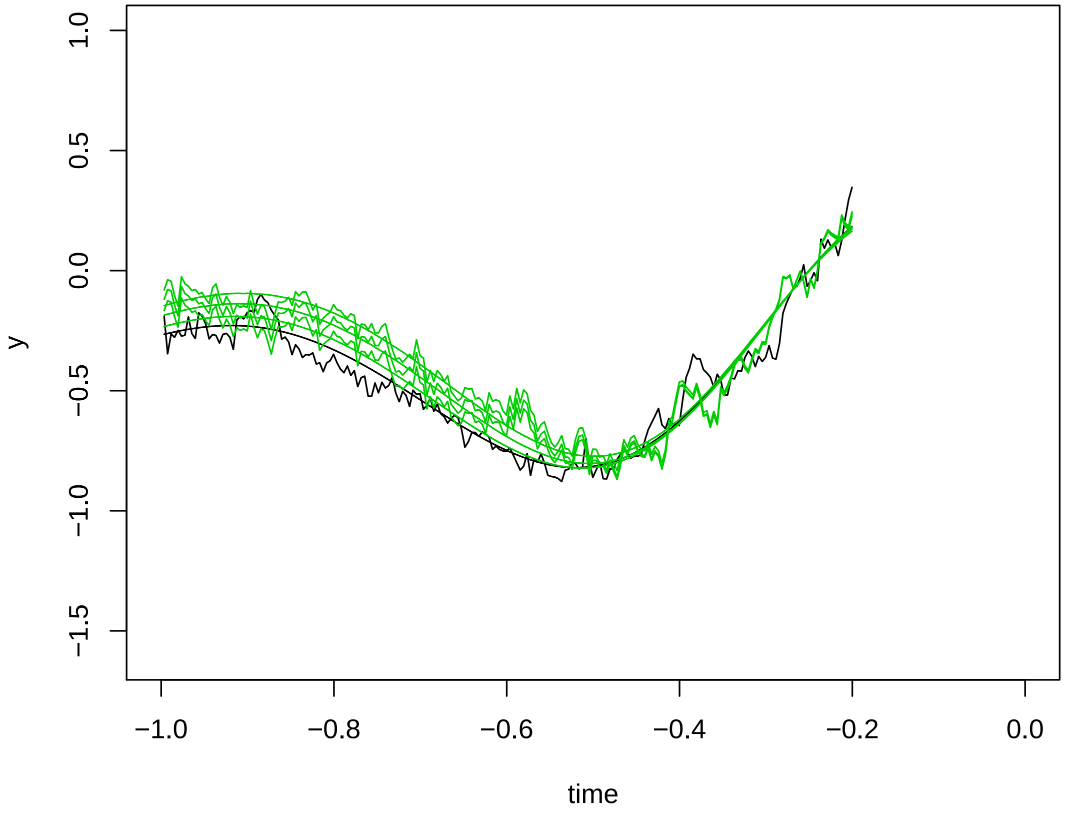


Figure 5.12: The fake system data, which is a FAMOUS simulation removed from the training set, is plotted here as the rough black line; the map estimate $\mathbb{E}\left(c(x^*_{MAP}), t, \hat{v}\right)$ is visible as the smooth black line beneath it. The three fake system time series with decreasing ρ^* are described by the three green lines.

more homogeneous and that the one-dimensional marginals become smoother and flatter. As a quantitative description of the loss of precision, we see, as presented in table 5.1, that the NROYS quickly grows to include all of the input space while the HLCS approximately doubles in size to fill 54% of it.

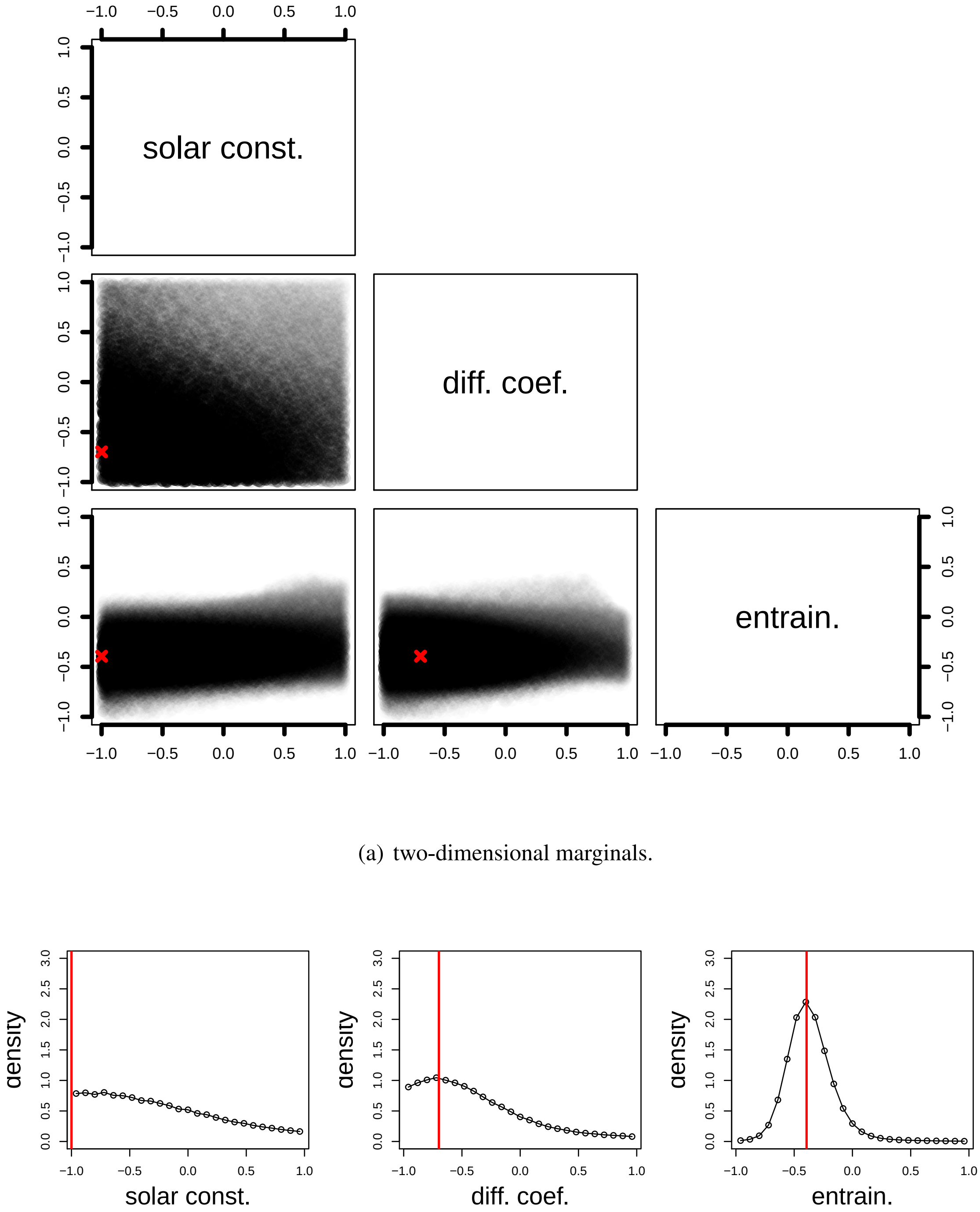
5.2.3 Simulator validation

Now we investigate the extent to which we are able to infer the parameter ρ^* from the simulator data and the fake system observations. Our approach is the same as in previous experiments, only this time we generate a four-dimensional Sobol sample of candidate parameters and scale the last coordinate to the interval $\Omega_\rho = [0.5, 1]$, the implication being that we consider scenarios in which the climates in the system and simulator range from being weakly, to completely correlated. The fake system data we calibrate with here is same as that used in the previous section to investigate the inferability of x^* when $\rho^* = 0.95$. Figures 5.16(a) and 5.16(b) show the approximate marginal densities for the parameters.

To appreciate the effect on our inferences for x^* brought about by treating ρ^* as unknown, we may compare figure 5.7(b) and the first three subfigures in 5.16(b). When we allow ρ^* to vary, the one-dimensional posterior marginals are smoother, the depth plots more homogeneous, and the NROYS and HLCS are larger than our first experiment, in which ρ^* was fixed at one. In this respect, allowing ρ^* to vary from one leads to effects similar to those caused by fixing it at a lower value.

The final subfigure in 5.16(b) shows how the emulator correctly steers us away from the highest values of ρ^*, at which the emulator cannot produce a mean and variance compatible with the system data, and from the lowest values, at which the likelihoods are shrunk by the factor in the multivariate normal density comprising the determinant of the variance for y^*. The implausibility statistic for determining the NROYS, on the other hand, has no equivalent term for penalising large variances, so while it rules out the highest ρ^* values, it provides no basis for refuting low values.

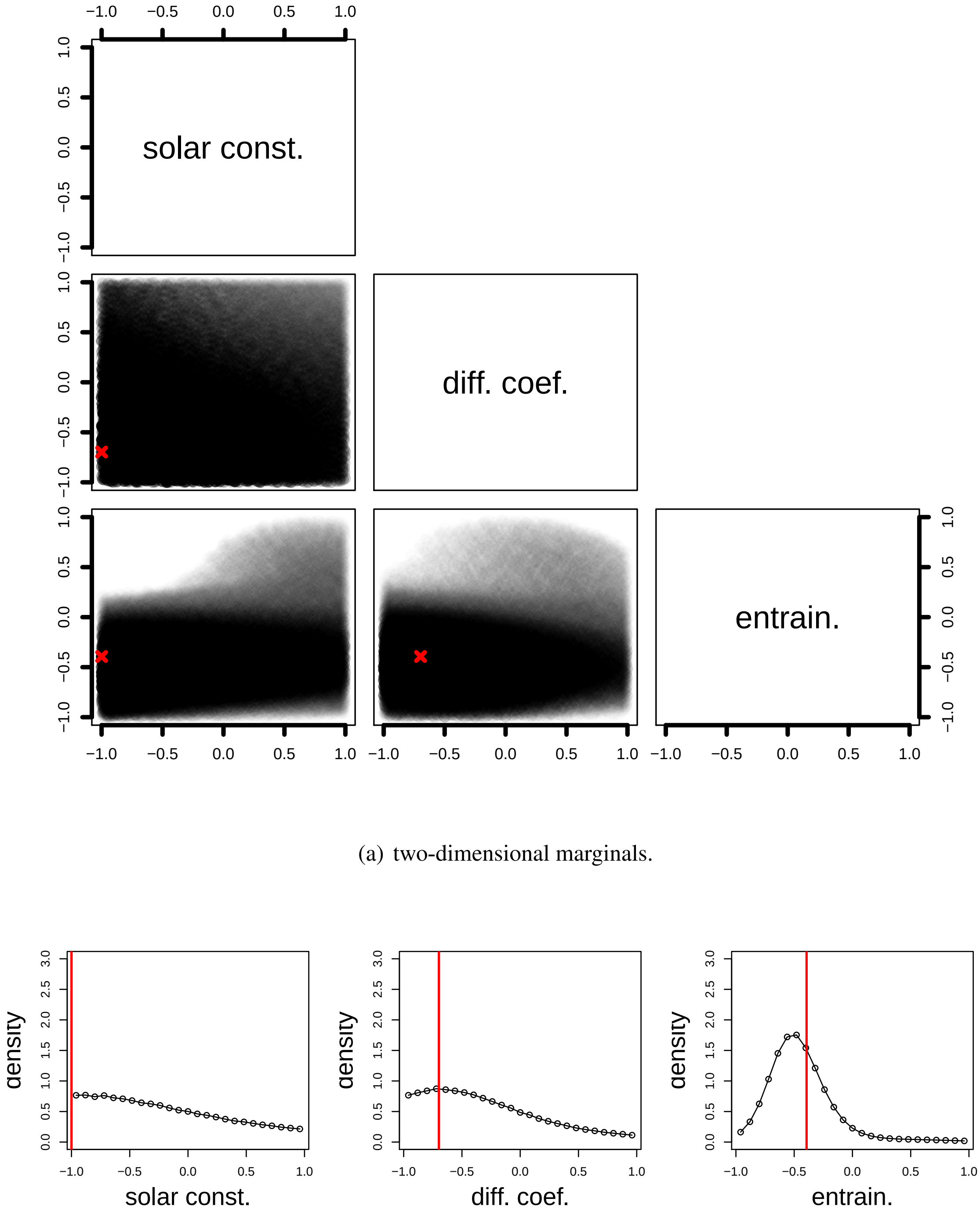
To render the NROYS and HLCS calculated from the prior sample in the higher-dimensional setting comparable to the sets calculated in the previous sections we need to define the marginal NROYS and HLCS. The marginal NROYS for x^* is the set of all input



(a) two-dimensional marginals.

(b) one-dimensional marginals.

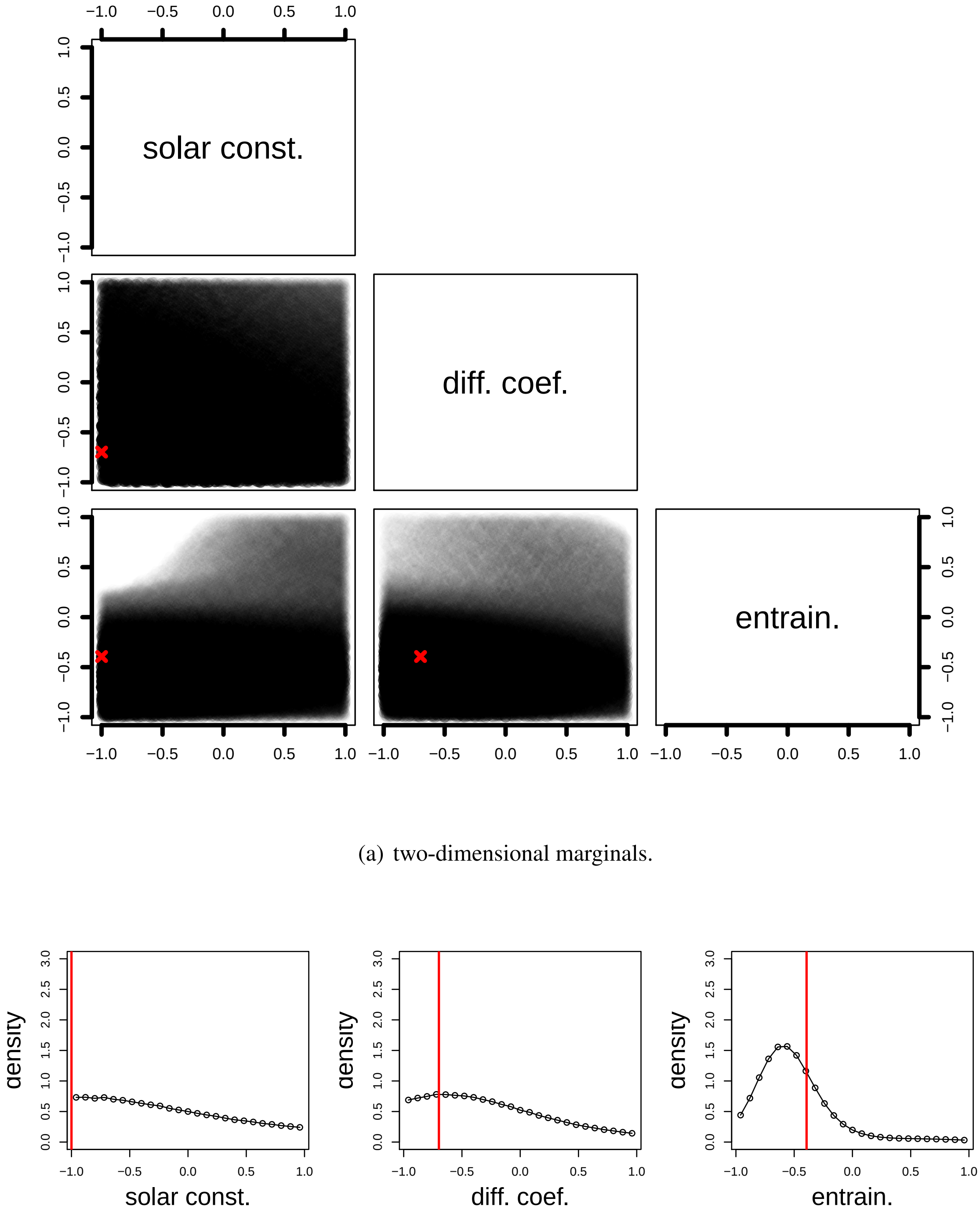
Figure 5.13: Plots illustrating the posterior marginal densities for x^* for the experiment in which $\rho^* = 0.99$. The red crosses and lines show the projections for the true value of x^*.



(a) two-dimensional marginals.

(b) one-dimensional marginals.

Figure 5.14: Plots illustrating the posterior marginal densities for x^* for the experiment in which $\rho^* = 0.95$. The red crosses and lines show the projections for the true value of x^*.



(a) two-dimensional marginals.

(b) one-dimensional marginals.

Figure 5.15: Plots illustrating the posterior marginal densities for x^* for the experiment in which $\rho^* = 0.9$. The red crosses and lines show the projections for the true value of x^*.

parameters such that the emulator's estimate under at least one discrepancy specification in the support of the prior satisfies the implausibility condition. This means defining an implausibility function d_{Imp} taking both x and ρ arguments,

$$d_{Imp}(x,\rho) = (y^* - \mu(x,\rho))^T \Sigma(x,\rho)^{-1}(y^* - \mu(x,\rho)),$$

where

$$\begin{aligned}\mu(x,\rho) &= \mathbb{E}\left(y(x) \mid x, \rho = \rho^*, \mathbf{X}, \mathbf{Y}\right),\\ \Sigma(x,\rho) &= \mathrm{Var}\left(y(x) \mid x, \rho = \rho^*, \mathbf{X}, \mathbf{Y}\right),\end{aligned}$$

so, mathematically, our definition reads,

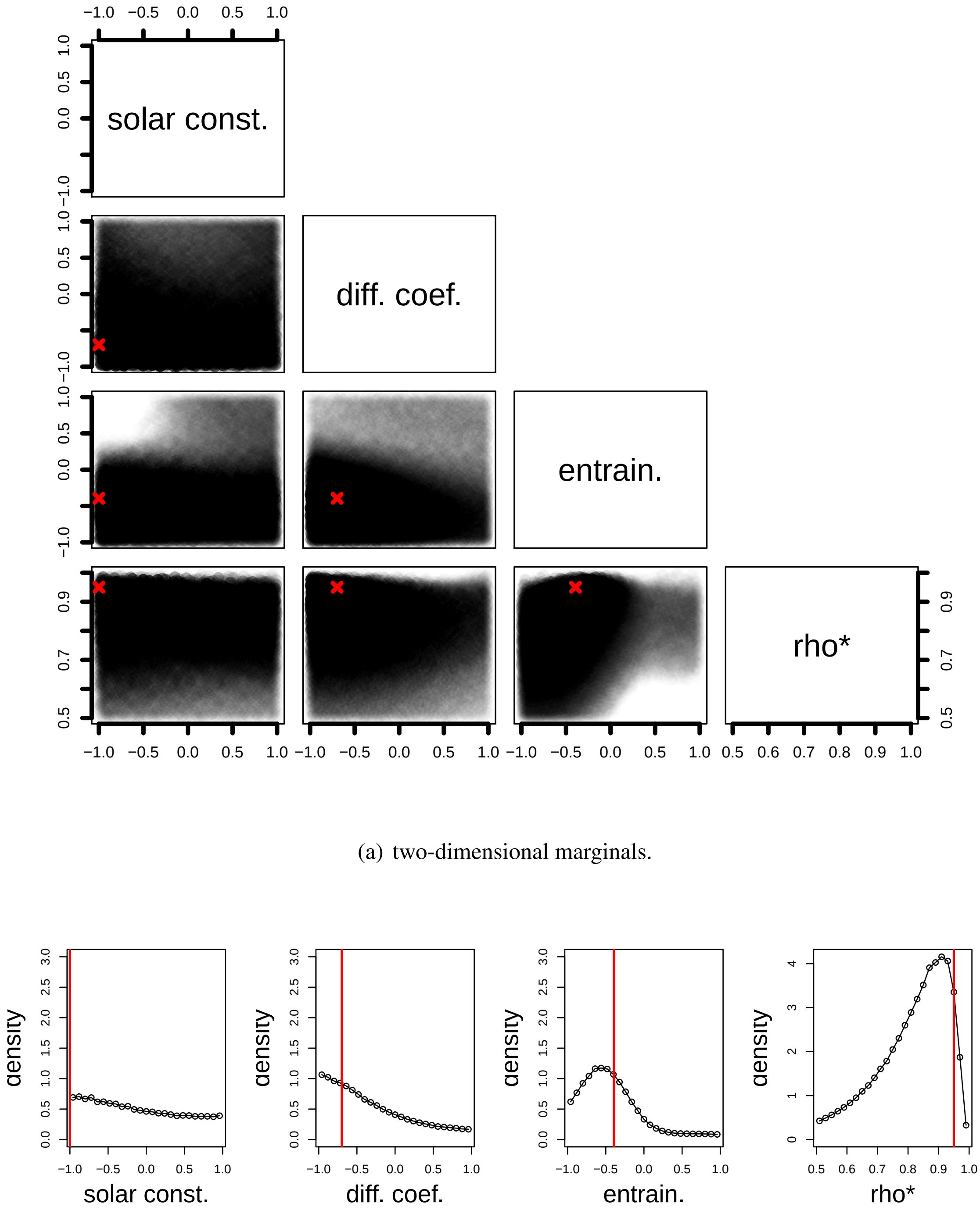
$$\Omega_{NROYS} = \{x \mid \exists\, \rho \in \Omega_\rho,\ d_{Imp}(x,\rho) < h\}.$$

As in the other experiments we set the implausibility threshold h at the ninety-fifth percentile of the chi-squared distribution with 200 degrees of freedom. To approximate the size of the NROYS we start by binning the four-dimensional prior sample, using a three-dimensional equally spaced grid over the input space to define the bin intervals. A bin is designated as 'not yet ruled-out' if the implausibility discrepancy statistic of at least one sample member that falls within the bin also falls below the implausibility threshold.

To define the HLCS we first need to define the marginal likelihood for $x = x^*$,

$$\pi_{x^*|\mathbf{Y}}(x) = \int \pi(y^* \mid x = x^*, v = v^*, \mathbf{Y}, \mathbf{X})\pi(v = v^* \mid x = x^*)\,\mathrm{d}v.$$

The $\alpha\%$-HLCS is then the posterior credible set of input parameters that includes every point at which the marginal likelihood is greater than some value h, and no points at which the marginal likelihood is lower. For our example, approximating the size of the HLCS requires that we sum the normalised likelihoods for the parameters in each bin. We now treat the bins in the same way as we did the prior sample members in the previous HLCS calculations: we order the bins according to the sum of the likelihoods of their members and select the bins with the highest marginal likelihoods such that their posterior mass exceeds 0.95. Again the proposal mechanism compensates for the prior density that is omitted from these calculations. The marginal NROYS and HLCS are the objects described in the rows of table 5.1 qualified with '(Unknown ρ^*)', while the figures in brackets describe the corresponding higher-dimensional sets.



(a) two-dimensional marginals.

(b) one-dimensional marginals.

Figure 5.16: Plots illustrating the posterior marginal densities for x^* and ρ^* for the experiment in which ρ^* is considered unknown. The red crosses and lines show the projections for the true value of x^* and ρ^*.

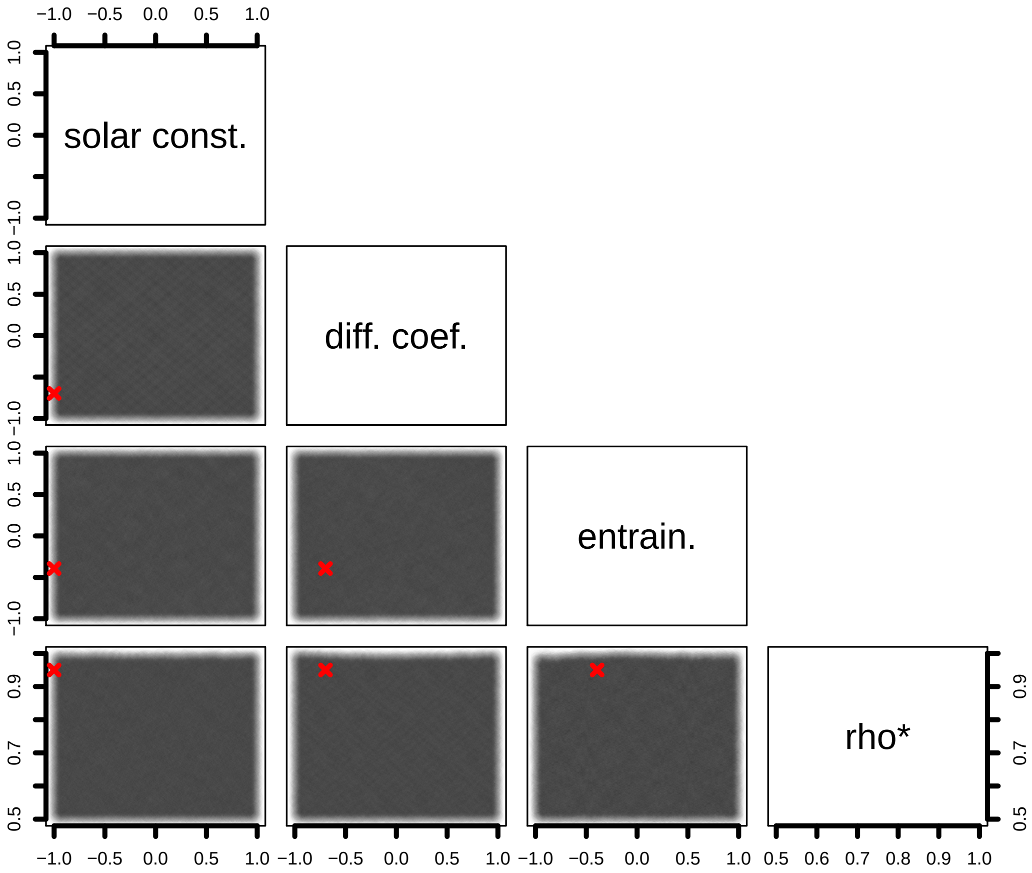


Figure 5.17: The depth plot for the NROYS over the input space for the experiment in which ρ^* is considered unknown. The red crosses show the projections for the true value of x^* and ρ^*.

The marginal NROYS for x^* includes the whole input space; indeed, we ought to have anticipated that the marginal NROYS would not be a useful statistic for this experiment. Because of the way we have defined it, the statistic is effectively determined by the NROYS for the lowest permitted value of ρ^*. At $\rho^* = 0.5$ the discrepancy between the simulator and the system is great enough to diminish the relevance of the simulations to the extent that we cannot infer from them if any inputs are incompatible with the system data.

5.3 Chapter summary

This test of the Cholesky-basis emulator has been highly encouraging. The calculations involved were mostly fast and easy to code, and our diagnostics and plots have proven to be intelligible and informative. The emulator has been able to detect and utilise systematic variation in the simulator output attributable to all the input parameters, variation which is extremely difficult to discern from casual inspection of the simulated time series.

Not being able to run more simulations or to access system data has been frustrating. These limitations meant we were not able to test methods for designing simulation ensembles or for treating simulator discrepancy to a completely satisfactory degree. An opportunity to investigate a genuine instance of simulator discrepancy would only provide anecdotal evidence for the suitability of our treatment of it but, by the nature of this important yet vague concept, anecdotal evidence would still be very valuable.

Conversely, of course, we needed to know the true inputs for the system data in order to tell whether our inferences had been successful. In this respect, using synthetic system data was crucial. Similarly, it is useful to see that the discrepancy parameter ρ^* appears to be inferable given perfect knowledge of, or belief in, the nature of the discrepancy variance structure. Such an observation provides informal, yet important, evidence for the claim that ρ^* may still be a meaningful statistic when the appropriate discrepancy structure is less obvious.

We summarise our conclusions for the chapter's calibration experiments in table 5.1. It shows that we successfully identified sets containing the true parameter values in every experiment with the Cholesky-basis emulator. Optimisation of the covariance parameters proved to substantially improve the emulator's fit to the simulated data and led to a decrease in the size of the NROYS, thus constituting a more secure and precise history match.

While the NROYS and HLCS from the unoptimised emulator still contained the true input parameter, suggesting a level of robustness to covariance parameter misspecification, the observed improvement in the diagnostic plots and the stable behaviour of the optimising algorithm have persuaded us that optimising the emulator's covariance parameters is a good idea. The Nelder-Mead optimisation is slow relative to the other calculations required for this chapter, but the time it takes is still short if we measure it against the time required to perform the FAMOUS simulations, or to code and test more advanced optimisation algorithms.

Our final remarks concern the results presented in the last row of 5.1, which relate to the experiment in which we attempted to infer the value of ρ^*. Interestingly, the approximate posterior marginal density in 5.16(b) does peak distinctly over the true value for ρ^* while the information in the marginals for the components of x^* is mostly retained. The particular point of interest here comes from contrasting this result with authoritative voices, of Goldstein[14] and Robert[46] for example, that advise us to exercise extreme caution, guided by careful deliberation, when treating a model's discrepancy or tolerance parameters as inferable quantities alongside the inputs x^*.

The key feature of our particular problem that makes inference for the discrepancy parameter ρ^* possible and appropriate is the high-dimensionality of the time series data. It means that we are provided with 200 correlated observations of the discrepancy rather than just one, as would be the case when simulating univariate quantities. Furthermore, the high-dimensionality of the data allows us to distinguish between different modes, or directions, of variation; we can identify the weather signal, for example, by its high frequency, and the variation attributable to the input parameters is constrained by the smoothness of the climate surface over the input space to only a small number of directions. The remaining directions of variation are thus directly attributable to the discrepancy. Such untangling of the signal is impossible with a univariate output.

Experiment	$\mathbf{1}(x^* \in \Omega_{NROYS})$	$\lvert\Omega_{NROYS}\rvert_{\pi_{x^*}}$	$\mathbf{1}(x^* \in \Omega_{HLCS})$	$\lvert\Omega_{HLCS}\rvert_{\pi_{x^*}}$
Scalar emulators ($\rho^* = 1$)	1	0.98	0	0.68
Unoptimised Cholesky emulator ($\rho^* = 1$)	1	0.93	1	0.18
Cholesky emulator ($\rho^* = 1$)	1	0.67	1	0.26
Cholesky emulator ($\rho^* = 0.99$)	1	0.86	1	0.35
Cholesky emulator ($\rho^* = 0.95$)	1	0.98	1	0.51
Cholesky emulator ($\rho^* = 0.90$)	1	1	1	0.54
Cholesky emulator (Unknown ρ^*)	1(1)	1(0.99)	1(1)	0.68(0.61)

Table 5.1: A summary of the chapter's calibration experiments. The logical entries indicate whether the true input parameters for the fake system data were in the 95%-NROYS or 95%-HLCS. The decimal entries give the sizes of these sets relative to the prior sample of candidates. For the last row the figures in brackets refer to the sets in four dimensions while those outside the brackets refer to the marginal sets.

Chapter 6

Closing discussion

6.1 Issues addressed

In this thesis we have presented a developed exposition of the roles of covariance parameters and the dual interpretations of smoothing: firstly, as referring to a physical mechanism which obeys, or almost obeys, a law expressed as a linear differential operator; and secondly, as referring to an inherently subjective specification of the similarity between values of a field at separated locations. These interpretations arise naturally when we build our model starting from the specification of penalties or covariances respectively.

Having looked at the ways in which a covariance specification describes and defines the shape of functions, we moved on to study the way it also leads to implications for the significance of a best simulator input and output. Specifically, we identified how the factorisability of a covariance function implies the potential for the separability of beliefs. At this stage we proposed a novel model structure for the discrepancy between systems and simulators that sits comfortably beside the calibration procedures for a simulator's input.

Our work in section 2.3 identified how a model with a factorisable variance structure can be used to assimilate large grids of data in an efficient manner. The technique overcame the problem of inverting huge matrices, but did not remove them from the calculations, which were generally unwieldy in the sort of semi-exploratory environment, namely R, in which we would like to perform further analysis. In section 4.3.2 we sketched out how the problem of dealing with these large arrays would become exponentially more serious as the dimensions involved accumulated, and how we would eventually need to look for ways to compromise grid-based methods.

With the first emulator of chapter 4, the NIW emulator, we introduced a smaller, hidden grid of variables corresponding to basis function coefficients rather than system values. We then tackled the inference for the grid values by dealing only with subsets of them at a time, and by using a Gibbs sampling argument to rationalise the process. Retaining a full set of basis functions for each series allowed us to use the inverse Wishart distribution to collect information regarding the smoothness of the time series. The fitting procedure for this emulator proved to be demanding, although it did bring our attention to the fact that we could emulate the series with a view to calibration by readjusting the emulator in particular parts of the input space identified as being relevant to the system's input parameters.

The evolution of our modelling strategy continued with the development of the Nyström emulators. Again, our approach consisted in describing large, ragged arrays of data using a finite set of basis functions, but this time we chose to focus on constructing a basis for the emulator that would reduce redundancy in our calibration calculations, rather than focusing on the emulator fitting procedure.

Our extended example with the FAMOUS data in chapter 5 represents a proof of concept for the Cholesky basis emulator. The example was rich in detail, which provided many points for discussion. Among the most notable of these were the following findings:

- the calibration procedure using three simple univariate emulators, essentially ignoring the time series structure, revealed almost no information for x^*;
- the Cholesky emulator appears to be robust against relatively uninformed covariance parameter estimates but can be improved significantly if they are tuned to better fit the data;
- the NROYS grows more quickly than the HLCS as the simulator discrepancy is increased, in this way the NROYS loses information more quickly as the simulator diverges from the system;
- the size of the simulator discrepancy or, equivalently, the separation between the system and the simulator in the discrepancy space is potentially inferable when we

calibrate to time series data.

6.2 Issues raised

Many of the practical challenges of emulation and calibration that we have faced have resulted from trying to model too many degrees of freedom and from trying to assimilate too many data simultaneously. We have recognised that the redundancy of high-frequency degrees of freedom is a consequence of the smoothness of a field or function, but we have struggled to estimate appropriate levels of smoothness from data. In sections 2.4.1 and 4.4.3 we sought to make inferences for smoothness parameters; both cases were characterised by high computational demand and conclusions of ambiguous value. These are unsatisfactory findings and motivate renewed research into methods for estimating a field's spectral or wavelet composition. We would be particularly interested in wavelets in multiple dimensions, and in establishing relationships between wavelet methods and the penalty and covariance function smoothing methods discussed in chapter 2. With multiscale methods we would also hope to describe a continuum of signals between climate and weather, and to formulate a more direct relationship between a smooth and the covariance parameters that define it.

Another issue we have identified but not explored adequately is the question of whether and when it is a good idea to smooth a likelihood or posterior density. It would be interesting, for example, to investigate further the relationship between the smoothness of the climate signal and the smoothness of the posterior for x^*. In example 4.2.2 we identified an instance in which we could share information between data by smoothing their likelihoods, and in section 4.4.2 we proposed that a particular basis expansion could provide a convenient vehicle for posterior summary statistics.

Smoothing the likelihood or posterior is much easier than smoothing the output quantities when the output is high-dimensional and so could save us a considerable amount of work in our analysis. Anticipating the smoothness of the likelihood, however, is problematic. Specifying covariance properties for the output may be informed by our intuition for the system and simulator mechanics, but specifying covariance parameters for the likelihood would require intuition for the informativeness of the data, something that we imagine to be very difficult.

Orthogonal density expansions are useful because they allow extremely quick and stable calculations to be made for properties of the distribution. For example, if the expansion basis consists of well-studied, mathematically neat functions, such as enveloped orthogonal polynomials, then many integrals may be computed analytically. No emulator is likely to be fast enough to render the numerical integration of the posterior fast enough to compete with this. One interesting issue here is the selection of an appropriate expansion. Gram-Charlier and Edgeworth series expand around a central distribution, which is modified by a series of correction factors. They allow us to borrow from the library of results for orthogonal polynomials only if the central distribution belongs to one of a set of well-studied special cases. We would like to have had more time to further investigate strategies for choosing expansions, and to assess their robustness to target distributions that differ significantly from the expansion's central distribution. Of course, density estimation with basis functions is already a well-established field, with notable contributions being made by Silverman [51] and Gu [19], and thorough research of the available literature would form a considerable part of our investigations.

We would also like to develop a formal mathematical argument in support of the idea that a smoother likelihood leads to faster or more stable calibration calculations. In [47] Roberts derives a relationship between a measure of roughness of the log of a posterior density and the speed at which an RWM algorithm explores it, the significance lying in the fact that the log posterior for a system's parameters inherits roughness properties directly from those of the climate term. This result represents a tantalizing connection that we might be able to seize upon in such a supporting argument.

6.3 Issues sidelined

A large amount of work has not made it into the thesis. In this last section we identify a selection of issues that are relevant to the simulation of physical systems but do not directly contribute to, or follow from, our main arguments or results.

The design of simulations is a topic we certainly could have pursued further. In particular, we have learnt how sparse grid integration designs are ideally suited to efficiently investigating functions with high-dimensional domains. Zhu's [63] result showing that the eigenfunctions of the squared exponential/normal operator take a form very close to those of Hermite functions leads us, via the work of Dette [10], towards d-optimal designs corresponding to classical Gauss-Hermite quadrature rules. These rules can then be adapted to the high-dimensional context with the sparse grid methodology. From these quadrature rules we can follow trails of research to results for sequences of nested rules[36], which may constitute structured batches or ensembles of designs. The construction of algorithmic adaptive design strategies and tests for their appropriateness will require a great deal of work. It would be interesting to explore this area further though, and in doing so better understand how our sequential basis selection algorithms could be informed by existing adaptive methods.

In the course of our research we have also investigated intelligent random walk metropolis algorithms inspired by RAM,[57], MALA [47] and stochastic Newton [31] algorithms. We can implement these on the posterior density surface defined by the emulator because it is often comprised of relatively simple functions. The complexity introduced by these guided algorithms, although interesting to explore, has not shown itself to result in impressive results in our informal experiments. The greater, and more easily exploited, benefit of the emulator's simplicity is its speed, which allows for a large number of evaluations and the use of simpler algorithms.

Our investigation of derivative-informed random walks and particle methods led us to consider whether we could hope to incorporate particle filters or Kalman filters into our own work with climate simulators. This is an exciting prospect, as with the extended Kalman filter we can start to break into the black box of the simulator code. We can also use filtering methods to calibrate and simulate simultaneously rather than sequentially. To approach these issues we experimented with structurally simple dynamic models such as the Lorenz '96 model and the Ricker model, and found that such filtering methods are highly effective on synthetic examples in which the model is correctly specified, but can perform very badly when the time step mechanism of the simulator and the system are discrepant. The cost of filtering is the reconstruction of the simulator code and the constant interruption of its execution. So despite being enthusiastic, we are also cautious about the application of filtering methods to climate modelling, because we suspect that their benefits will be wasted when the discrepancy is overestimated and that they will be particularly vulnerable to instability when we underestimate it.

We would also like to investigate the validity and the implications of this work's foundational practical assumption: that a system may be decomposed into approximately uncorrelated or independent subsystems on different scales, namely climate and weather. This would require the development of tests for system separability and the proposal of plausible alternatives for the system's dependencies. We see that confirmation of the assumption has implications beyond our emulation and calibration methods: if the simulator's weather is approximately independent of its climate and significantly discrepant from the system's weather, how much benefit is there to simulating the weather at all? Can the weather subsystem be replaced with something simpler, like a small forcing that perturbs the climate? Such perturbations, referred to as stochastic parameterisations, are already often included in simulators to compensate for unmodellable processes, but we could also justify their inclusion on the grounds of computational convenience or on the grounds that the physical mechanisms associated with the weather are poorly understood.

In a similar vein, we can also see a simulator discrepancy as a type of stochastic parameterisation that embodies the cumulative effect of multiple doubts and uncertainties rather than their individual effects. The question of whether these cumulative and contributing uncertainties are consistent with each other is just as important as whether the cumulative and contributing physical processes are.

Following the parallel even further, it is not unusual for modellers to reinterpret a physically-motivated simulator parameter as a tuning parameter, whose value serves to compensate for a perceived deficiency in the simulator. A well-known example of this is the eddy viscosity parameter for an ocean simulator, which may be increased by orders of magnitude in order to compensate for the simulator's under-resolved numerical solver. Similarly, it is not unusual for statisticians to reinterpret a subset of a simulator's output quantities as being unrelated to the physical process they may once have been associated with. We make this point in reference to the selection process for summary statistics for synthetic likelihood and ABC-type methods in which many of the system and simulator outputs are not required to be consistent, so high discrepancies are effectively tolerated, because those outputs are discarded in the analysis.

In both instances only certain quantities are endowed with meaning. The others are understood as artefacts or instruments of parameterisations that compensate for missing physics or missing uncertainties, but which are not readily understood as identifiable, measurable physical processes or reasoning processes themselves. Having recognised these similarities, a holistic treatment of the physical and epistemological parameterisations, and of the input and output discrepancies, appears to be a goal we should be setting ourselves.

These considerations raise some important questions about the way in which we are using scientific theory: are we free to stand by some implications of a theory while ignoring or denying others? To what extent does this undermine the implications that we do stand by? To what extent are we obliged to justify a simulator's shortcomings, or a theory's shortcomings? And to what extent does model fit constitute such a justification?

We anticipate that answering these questions will require discussion and collaboration with other statisticians and scientists, as well as a keen intuition for the behaviour of the particular system we are analysing. To better appreciate the behaviour of physical systems evolving through time and the asymmetries between input- and output-type quantities we ought to invest in the study of dynamical systems. This means committing to research combining sophisticated mathematical treatment of complex models, and carefully considered epistemological deliberation at every step. Dynamical systems, specifically nonlinear dynamical systems, are fascinating because of the richness of the behaviour they exhibit and because of the epistemological limits on the inferences they allow for. A greater understanding of nonlinear dynamics and how their effects might be, at least partially, catered for in statistical analyses represent appealing directions for our own further research.

Appendix A

Notation

A.1 Mathematical notation

In the following tables we present a glossary of notation organised approximately according to the objects' mathematical classes.

A.1.1 Indexing and labelling

We use squared brackets followed by subscripted indices to refer to elements within a rectangular array. When such arrays are inappropriate or inconvenient we use subscripted indices without squared brackets to refer to objects in an ordered list of similar objects. Negative values are used to denote the removal of certain subarrays, or sublists, from a full array or list, while a dot is used to make it clear when no subarrays are removed.

Symbol	Description	Introduced in section
$[\mathbf{A}]_{i_1,i_2,\ldots,i_D}$	The element of the array **A** in the position labelled by the i_1th member of the first index (the i_1th row), the i_2th member of the second index ((the i_2th column)), and so on until the Dth index.	2.1
$\mathbf{A}_i$	The ith array in a list of arrays.	2.1

A.1.2 Array and matrix operations

Symbol	Description	Introduced in section
$\mathbf{A}^T$	The transpose of the matrix **A**.	2
$\mathbf{A}^{-T}$	The transpose of the inverse of **A**.	2
$\mathbf{A}^\dagger$	The Moore-Penrose generalised inverse of the matrix **A**.	2.1
$\otimes$	The Kronecker product operator.	2.3
$\bigotimes$	Shorthand for the Kronecker product of all the elements of its argument.	2.3
$\oplus$	An operator denoting the addition of independent quantities.	1.1.2
$\operatorname{Tr}(\cdot)$	The trace operator.	4.4
$\lvert\mathbf{A}\rvert$	The determinant of the matrix **A**.	2.3
$\lvert\mathbf{A}\rvert_+$	The pseudo-determinant of the matrix **A**.	4.1.1.1
$\operatorname{vec}(\cdot)$	The vec operator.	2.3
$\circ$	The Hadamard or entrywise multiplication operator for arrays.	2.3
$\mathcal{M}(\ldots)$	A tensor multiplication operator.	2.3

A.1.3 Deltas and indicators

Symbol	Description	Introduced in section
$\delta_{x,x'}$	The Kronecker delta function.	4.2
$\delta(x)$	The Dirac delta function.	2
$\mathbf{1}(\cdot)$	The indicator function.	4.4.2

A.1.4 Variables and values

Symbol	Description	Introduced in section
x	A column vector of input parameters.	2.3
x^*	The input parameter giving rise to the system values.	3.1
$\mathbf{X}$	A matrix of simulator inputs formed by stacking transposed input vectors as rows.	2.3
t	A scalar time quantity.	1.1.2
$\mathbf{T}$	A vector of times.	1.1.2
ξ	A concatenation of the time variable and input parameter into a single column vector.	4.3
$\mathbf{\Xi}$	A matrix of output coordinates formed by stacking transposed ξ vectors as rows.	4.3
v	A discrepancy parameter.	3.1
v^*	The discrepancy parameter locating the system function.	3.1
$\hat{v}$	The discrepancy parameter locating the simulator function.	3.1
c	The smooth climate component of the output.	1.1.2
$c(\xi)$	The scalar value of the climate component at ξ.	1.1.2
$c(\mathbf{\Xi})$	The vector of the climate values corresponding to the inputs in the rows of $\mathbf{X}$.	1.1.2
$\mathbf{C}$	A matrix of climate values.	2.3
w	The rough weather component of the output.	1.1.2
$w(\xi)$	The scalar value of the weather component at ξ.	1.1.2
$w(\mathbf{\Xi})$	The vector of the weather values corresponding to the inputs in the rows of $\mathbf{X}$.	1.1.2
$\mathbf{W}$	A matrix of weather values.	4.2
$y(\xi)$	A physically significant output quantity.	1.1.2
$y(\xi)$	A scalar value of the output quantity at ξ.	1.1.2
$\mathbf{Y}$	An array of simulator output values.	2.3
$y_{sys}(\xi)$	Alternate notation for $y(\xi, v^*)$.	3.1

Symbol	Description	Introduced in section
$y_{sim}(\xi)$	Alternate notation for $y(\xi, \hat{v})$.	3.1
z	A measurement of a physically significant output quantity.	1.1.2
β	A column vector of basis function coefficients.	2.1
$\boldsymbol{\beta}$	A matrix of basis function coefficients.	2.1
τ_j	A coefficient function for a derivative contributing to a linear differential operator.	2.1
ζ_j	A weight for a squared derivative contributing to a roughness penalty.	2.2
ϑ	An array of coefficients for a density expansion.	4.4.2
$\mathbf{C}$	A vector of inducing climate variables.	4.4
$\mathbf{N}$	A matrix of basis node locations.	4.3
$\mathbf{P}$	A matrix of integration node locations.	4.4
$\mathbf{K}$	A variance matrix.	2.3
$\mathbf{L}$	An upper triangular correlation length matrix.	4.4.3
N	An upper limit for a sum's index (defined locally).	

A.1.5 Covariance parameters and constructs

Symbol	Description	Introduced in section
$k(\cdot,\cdot)$	A covariance function.	2.2
$k(\cdot)$	An autocovariance function.	2.2
σ^2	A scalar variance parameter.	2.2
u	A scalar correlation length.	2.2
v	A spikiness of differentiability parameter.	2.2
ω	An oscillation frequency parameter.	2.2
λ	An eigenvalue, or a roughness penalty multiplier depending on context.	2.1

Symbol	Description	Introduced in section
θ	A vector of covariance function parameters defined for convenience.	2.4
f	A spectral density.	2.2
H	A reproducing kernel Hilbert space.	4.3.2.3
$\int \ldots \mathrm{D}[f]$	A functional integral.	4.3.2.3
$\phi_i(\cdot)$	An invidual basis function value.	2.1
$\phi(\cdot)$	A column vector basis function values.	2.1
$\boldsymbol{\phi}$	A matrix of basis function values whose rows correspond to individual function inputs, and whose columns correspond to individual basis members.	2.1

A.1.6 Bayes Linear constructs

Symbol	Description	Introduced in section
$\mathbb{E}_z(x)$	The Bayes linear adjusted expectation of x given z.	2
$\mathrm{Var}_z(x)$	The Bayes linear adjusted variance of x given z.	2
$\mathrm{Cov}_z(x\,,\,y)$	The Bayes linear adjusted covariance between x and y given z.	2
$\lfloor x \perp\!\!\!\perp y \rfloor \mid z$	The statement: x is separated from y by z.	3.1

A.1.7 Miscellaneous

Symbol	Description	Introduced in section
π	A probability density function.	2.2.1
ρ	A correlation.	3.1

Symbol	Description	Introduced in section
α	A vector of mixing or visitation probabilities.	4.2
L	A linear differential operator.	2.1
ζ_i	A derivative penalty coefficient.	2.1
ϑ	An array of approximate density coefficients.	4.4.2
$\mathcal{L}$	A loss function.	2.1
Ω	A vector space.	3.1
$\%$	The mod operator.	2.3
$\backslash$	The integer division operator.	2.3

A.2 Abbreviations

Abbreviation	Meaning
AMOC	Atlantic Meridional Overturning Circulation.
AR	AutoRegression.
CDF	Cumulative Distribution Function.
CPDN	Climate Prediction Dot Net.
dof	Degrees of freedom.
EMIC	Earth System Models of Intermediate Complexity.
FAMOUS	FAst Met Office/UK universities Simulator.
FDA	Functional Data Analysis.
GCA	Gram-Charlier A (series).
GCM	General Circulation Model.
GCV	Generalised Cross-Validation.
HadCM3	Hadley Centre Coupled Model (version 3).
HLCS	Highest Likelihood Credible Set.
HPDCS	Highest Posterior Density Credible Set.
iid	Independent and identically distributed.
IPCC	Intergovernmental Panel on Climate Change.

Abbreviation	Meaning
KLD	Kullback-Leibler Divergence.
LDO	Linear Differential Operator.
LOO	Leave-One-Out.
MALA	Mean Adjusted Langevin Approximation.
MAP	Maximum A Posteriori.
MCMC	Markov Chain Monte Carlo.
NIG	Normal Inverse-Gamma.
NIW	Normal Inverse-Wishart.
NROYS	Not Ruled Out Yet Set.
PC	Polynomial Chaos.
PD	Positive Definite.
PDF	Probability Density Function.
PMF	Probability Mass Function.
PSD	Positive Semi-Definite.
RAM	Robust Adaptive Metropolis.
ROS	Ruled Out Set.
RWM	Random Walk Metropolis.

Appendix B

Linear algebra

For matrices **A**, **B**, **C**, **D**, **U** and **V**, when conformity allows:

Theorem B.0.1 (Trace of a Kronecker product)**.**

$$\mathrm{Tr}\,(\boldsymbol{A} \otimes \boldsymbol{B}) = \mathrm{Tr}\,(A)\,\mathrm{Tr}\,(B)\,.$$

Theorem B.0.2 (Determinant of a Kronecker product)**.** *For square* **A** *and* **B** *with n and q rows respectively,*

$$|\boldsymbol{A} \otimes \boldsymbol{B}| = |\boldsymbol{A}|^q |\boldsymbol{B}|^n.$$

Theorem B.0.3 (Vec of a matrix product)**.**

$$\mathrm{vec}\,(\boldsymbol{ABC}) = (\boldsymbol{C}^T \otimes \boldsymbol{A})\mathrm{vec}\,(\boldsymbol{B})\,.$$

Theorem B.0.4 (An identity for Kronecker products defining quadratic forms)**.**

$$\mathrm{vec}\,(\boldsymbol{D})^T\,(\boldsymbol{CA} \otimes \boldsymbol{B}^T)\mathrm{vec}\,(\boldsymbol{D}) = \mathrm{Tr}\left(\boldsymbol{AD}^T\boldsymbol{BDC}\right).$$

Theorem B.0.5 (An identity for matrix multiplication of Kronecker products)**.**

$$(\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{C} \otimes \boldsymbol{D}) = \boldsymbol{AC} \otimes \boldsymbol{BD}.$$

Theorem B.0.6 (The Sherman-Morrison-Woodbury inverse)**.**

$$(\boldsymbol{A} + \boldsymbol{UBV})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{B}^{-1} + \boldsymbol{VA}^{-1}\boldsymbol{U})^{-1}\boldsymbol{VA}^{-1}.$$

Theorem B.0.7 (Matrix determinant lemma)**.**

$$|\boldsymbol{A} + \boldsymbol{U}\boldsymbol{B}\boldsymbol{V}| = |\boldsymbol{B}^{-1} + \boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U}|\,|\boldsymbol{B}|\,|\boldsymbol{A}|$$

Theorem B.0.8 (The trace of a matrix product)**.**

$$\mathrm{tr}(\boldsymbol{A}\boldsymbol{B}) = \sum_{i,j}[\boldsymbol{A} \circ \boldsymbol{B}^T]_{ij}.$$

Theorem B.0.9 (Vec of a Hadamard product)**.**

$$\mathrm{vec}\,(\boldsymbol{D} \circ \boldsymbol{B}) = \left(\bigoplus_k \boldsymbol{d}_k\right)\mathrm{vec}\,(\boldsymbol{B})\,.$$

*Where $\boldsymbol{d}_k$ is a diagonal matrix whose diagonal entries are the elements of the kth column of **D**, and $\boldsymbol{D} \circ \boldsymbol{B}$ is the Hadamard product of **D** and **B** with elements*

$$[\boldsymbol{D} \circ \boldsymbol{B}]_{i,j} = [\boldsymbol{D}]_{i,j}[\boldsymbol{B}]_{i,j}.$$

Theorem B.0.10 (Eigenstructure of the sum of a matrix and the identity)**.** *If the matrix A has an eigenvector u with eigenvalue λ,*

$$\boldsymbol{A}u = \lambda u,$$

then the matrix $(\boldsymbol{A}+c\boldsymbol{I})$, where c is an arbitrary constant, has a corresponding eigenvector, which is also u, with eigenvalue $\lambda + c$:

$$(\boldsymbol{A} + c\boldsymbol{I})u = \boldsymbol{A}u + cu = (\lambda + c)u.$$

Theorem B.0.11 (The Hotelling inverse)**.** *The Hotelling inverse for a partitioned matrix [22]. For non singular A and D,*

$$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{U} \\ \boldsymbol{V} & \boldsymbol{D} \end{pmatrix}^{-1} = \begin{pmatrix} (\boldsymbol{A} - \boldsymbol{U}\boldsymbol{D}^{-1}\boldsymbol{V})^{-1} & -\boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{D} - \boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U})^{-1} \\ -\boldsymbol{D}^{-1}\boldsymbol{V}(\boldsymbol{A} - \boldsymbol{U}\boldsymbol{D}^{-1}\boldsymbol{V})^{-1} & (\boldsymbol{D} - \boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U})^{-1} \end{pmatrix}.$$

Theorem B.0.12 (Eigenstructure of a Kronecker product)**.** *Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ have eigenvalues λ_i, $i = 1, ..., n$, and corresponding right eigenvectors, $x_1, ..., x_n$; and let $\boldsymbol{B} \in \mathbb{R}^{m\times m}$ have eigenvalues μ_j, $j = 1, ..., m$, with eigenvectors $z_1, ..., z_m$. Then the Kronecker product,*

$$\boldsymbol{A} \otimes \boldsymbol{B},$$

has right eigenvectors $x_i \otimes z_j$ corresponding to eigenvalues $\lambda_i\mu_j$.

Theorem B.0.13 (Eigenstructure of a Kronecker sum)**.** *Let* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *be defined as in Theorem B.0.12. Then the Kronecker sum,*

$$\boldsymbol{A} \oplus \boldsymbol{B} = (\boldsymbol{A} \otimes \boldsymbol{I}_m) + (\boldsymbol{I}_n \otimes \boldsymbol{B}),$$

has right eigenvectors $x_i \otimes z_j$ *corresponding to eigenvalues* $\lambda_i + \mu_j$*.*

Theorem B.0.14 (Derivative of a determinant)**.** *[38]*

$$\frac{\partial |\boldsymbol{Y}|}{\partial x} = |\boldsymbol{Y}| \mathrm{Tr} \left(\boldsymbol{Y}^{-1} \frac{\partial \boldsymbol{Y}}{\partial x} \right).$$

Theorem B.0.15 (Derivative of a log determinant)**.**

$$\frac{\partial \log |\boldsymbol{Y}|}{\partial x} = \mathrm{Tr} \left(\boldsymbol{Y}^{-1} \frac{\partial \boldsymbol{Y}}{\partial x} \right).$$

Theorem B.0.16 (Derivative of an inverse)**.**

$$\frac{\partial \boldsymbol{Y}^{-1}}{\partial x} = -\boldsymbol{Y}^{-1} \frac{\partial \boldsymbol{Y}}{\partial x} \boldsymbol{Y}^{-1}.$$

Theorem B.0.17 (Derivative of a trace of an inverse)**.**

$$\frac{\partial \mathrm{Tr}(\boldsymbol{A}\boldsymbol{X}^{-1}\boldsymbol{B})}{\partial \boldsymbol{X}} = -(\boldsymbol{X}^{-1}\boldsymbol{B}\boldsymbol{A}\boldsymbol{X}^{-1})^T$$

Theorem B.0.18 (Matrix chain rule)**.**

$$\frac{\partial g(\boldsymbol{U})}{\partial X_{ij}} = \mathrm{Tr} \left[\left(\frac{\partial g(\boldsymbol{U})}{\partial \boldsymbol{U}} \right)^T \frac{\partial \boldsymbol{U}}{\partial X_{ij}} \right]$$

Definition B.0.19 (Entrywise norms)**.**

$$||\mathbf{A}||_p = \left(\sum_{i,j} |[\mathbf{A}]_{ij}|^p \right)^{1/p}.$$

We get the Frobenius norm when $p = 2$ and the max norm as $p \to \infty$.

Definition B.0.20 (Schatten norms)**.** Use the singular values of a matrix.

$$||\mathbf{A}||_p = \left(\sum_i \lambda_i^p \right)^{1/p}.$$

When $p = 2$ we get the Frobenius norm again. When $p = 1$ we get the trace norm. As $p \to \infty$ we get the spectral norm; the largest singular value.

Theorem B.0.21 (Trace determinant inequality)**.** *For an $n \times n$ positive definite matrix $\boldsymbol{A}$,*

$$|\boldsymbol{A}| \leq \left(\frac{\mathrm{Tr}\,(\boldsymbol{A})}{n} \right)^n,$$

which arises from the 'arithmetic-mean geometric-mean inequality' applied to the eigenvalues of A.

Definition B.0.22 (The Cholesky decomposition)**.** The Cholesky decomposition of a positive definite matrix $\mathbf{K}$ returns an upper right triangular matrix $\mathbf{R}$ such that

$$\mathbf{K} = \mathbf{R}^T\mathbf{R}.$$

$\mathbf{R}$ is not unique, any subset of its rows can be multiplied by minus one to give an alternative factorization. Uniqueness can be achieved by appending to the definition the requirement that the diagonal entries of R are positive.

The Cholesky decomposition of a block matrix is

$$\begin{pmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{pmatrix} = \begin{pmatrix} \mathbf{R}_{11}^T & \mathbf{0} \\ \mathbf{R}_{21}^T & \mathbf{R}_{22}^T \end{pmatrix} \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{21} \\ \mathbf{0} & \mathbf{R}_{22} \end{pmatrix},$$

where

$$\mathbf{R}_{11}^T\mathbf{R}_{11} = \mathbf{K}_{11}, \tag{B.1}$$

$$\mathbf{R}_{22}^T\mathbf{R}_{22} = \mathbf{K}_{22} - \mathbf{K}_{21}\mathbf{K}_{11}^{-1}\mathbf{K}_{12}. \tag{B.2}$$

If $\mathbf{K}$ is the variance matrix for the vector quantity (y_1, y_2) we can identify expressions (B.1) and (B.2) as the marginal variance for y_1 and the adjusted variance for y_2 given y_1. In this light, algorithms for computing the Cholesky decomposition can be seen as sequential variance adjustments for quantities whose prior variance is given by $\mathbf{K}$.

The following pseudo-code for a Cholesky decomposition algorithm is based on that given by Bastos in [2]. The algorithm includes a pivoting subroutine which serves to reorder the rows and columns of the matrix $\mathbf{R}$ according to a criterion Q. At each iteration of the algorithm's for-loop the standard criterion brings the largest diagonal element of the part of the matrix yet to be decomposed to the front of the queue. The resulting reordering of the rows is encoded in the pivot vector ϱ. On iteration j of the for-loop, the pivoting criterion for the Cholesky algorithm is

$$[Q]_i = \begin{cases} [\mathbf{K}]_{i,i} & \text{for } i > (j-1), \\ 0 & \text{otherwise.} \end{cases}$$

Algorithm 4 Cholesky decomposition with pivoting

```
**Initialise** $\mathbf{R} \leftarrow 0$, $\varrho \leftarrow 1:n$
**for** $j = 1, \ldots, n$ **do**
    **procedure** PIVOTING SUBROUTINE
        **if** $\max\, [[Q]_i] < \epsilon^*$ **then**
            Escape For-loop and terminate algorithm
        **end if**
        $q \leftarrow \arg\max_i\ [[Q]_i]$
        $[\mathbf{K}]_{\cdot,j} \leftrightarrows [\mathbf{K}]_{\cdot,q}$
        $[\mathbf{K}]_{j,\cdot} \leftrightarrows [\mathbf{K}]_{q,\cdot}$
        $[\mathbf{R}]_{\cdot,j} \leftrightarrows [\mathbf{R}]_{\cdot,q}$
        $[\varrho]_j \leftrightarrows [\varrho]_q.$
    **end procedure**
    $[\mathbf{R}]_{j,j} \leftarrow \sqrt{[\mathbf{K}]_{j,j}}.$
    $[\mathbf{R}]_{j,(j+1):n} \leftarrow [\mathbf{K}]_{j,(j+1):n}/[\mathbf{R}]^{-1}_{j,j}.$
    $[\mathbf{K}]_{(j+1):n,(j+1):n} \leftarrow [\mathbf{K}]_{(j+1):n,(j+1):n} - [\mathbf{R}]^T_{j,(j+1):n}[\mathbf{R}]_{j,(j+1):n}.$
**end for**
**return** $\mathbf{R}$, $\varrho$.
```

Theorem B.0.23 (Determinant of a block tridiagonal matrix)**.** *The determinant of the block tridiagonal matrix,*

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{A}_1 & \boldsymbol{B}_1 & & & 0 \\ \boldsymbol{C}_1 & \boldsymbol{A}_2 & \boldsymbol{B}_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \boldsymbol{C}_{n-1} & \boldsymbol{A}_{n-1} & \boldsymbol{B}_{n-1} \\ 0 & & & \boldsymbol{C}_{n-1} & \boldsymbol{A}_n \end{pmatrix},$$

is given by the product of determinants:

$$|M| = \prod_{k=1}^{n} |\mathbf{\Lambda}_k|,$$

where

$$\begin{aligned} \mathbf{\Lambda}_1 &= \boldsymbol{A}_1, \\ \mathbf{\Lambda}_k &= \boldsymbol{A}_k - \boldsymbol{B}_{k-1}\mathbf{\Lambda}_{k-1}^{-1}\boldsymbol{C}_{k-1} \quad k = 2, \ldots, n. \end{aligned}$$

B.1 The Thomas algorithm for solving a block tridiagonal matrix

The Thomas algorithm for the inversion of a block tridiagonal matrix requires forward and backward passes over the submatrices. Crucially, inversions and multiplications only take place on the scale of these submatrices and the $\mathbf{\Lambda}$ matrices from the determinant calculation may be reused. Using the same notation as Theorem B.0.23,

$$\begin{pmatrix} \mathbf{A}_1 & \mathbf{B}_1 & & & 0 \\ \mathbf{C}_1 & \mathbf{A}_2 & \mathbf{B}_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \mathbf{C}_{n-1} & \mathbf{A}_{n-1} & \mathbf{B}_{n-1} \\ 0 & & & \mathbf{C}_{n-1} & \mathbf{A}_n \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \vdots \\ \mathbf{x}_n \end{pmatrix} = \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \vdots \\ \vdots \\ \mathbf{d}_n \end{pmatrix}.$$

We define intermediate quantities, d_i', to keep track of the forward sweep,

$$\begin{aligned} \mathbf{d}_1' &= \mathbf{\Lambda}_1^{-1}\mathbf{d}_1, \\ \mathbf{d}_i' &= \mathbf{\Lambda}_i^{-1}\left(\mathbf{d}_i - \mathbf{C}_{i-1}^T\mathbf{d}_{i-1}'\right) \quad \text{for } i = 2, \ldots, n. \end{aligned}$$

And calculate the solution from a backward sweep through them,

$$\begin{aligned} \mathbf{x}_n &= \mathbf{d}_n', \\ \mathbf{x}_i &= \mathbf{d}_i' - \mathbf{\Lambda}_i^{-1}\mathbf{B}_i^T\mathbf{x}_{i+1} \quad \text{for } i = n-1, \ldots, 1. \end{aligned}$$

Significant simplification is possible when the covariance matrix relates to a regularly spaced stationary time series, in which case,

$$\mathbf{A}_i = \mathbf{A}, \qquad \mathbf{B}_i = \mathbf{C}_i^T = \mathbf{B}.$$

B.2 The Levinson algorithm for the inversion of a symmetric Toeplitz matrix

The precise form of the algorithm here is a slightly simplified version of that in the PhD thesis of Tom Bäckström [1]. The result for the determinant is attributable to Musicus [32], who uses alternative notation in his report.

A symmetric Toeplitz matrix $\mathbf{K}_N$ may be parameterized by the vector k that forms its first row like so,

$$\mathbf{K}_N = \begin{pmatrix} k_1 & k_2 & \dots & & k_N \\ k_2 & k_1 & \dots & & k_{N-1} \\ \vdots & \vdots & \ddots & & \vdots \\ & & & k_1 & k_2 \\ k_N & k_{N-1} & \dots & k_2 & k_1 \end{pmatrix}.$$

Given an N-vector $\mathbf{y}$ and the relationship,

$$\mathbf{Kx} = \mathbf{y},$$

we go about calculating $\mathbf{x}$ via a set of intermediate vector variables called forward vectors $\mathbf{f}^{(n)} \in \Re^n$, and two sets of scalars ϵ_n and ε_n. The first vector, which is also a scalar, is set to

$$\mathbf{f}^{(1)} = \frac{1}{k_1},$$

while the next $N-1$ are calculated recursively. So for $n = 2, \dots, N$,

$$\epsilon_n = \sum_{j=1}^{n-1} k_{n+1-j} \mathbf{f}_j^{(n-1)},$$

$$\mathbf{f}^{(n)} = \frac{1}{1-\epsilon_n^2}((\mathbf{f}^{(n-1)}, 0) - \epsilon_n(0, \mathbf{b}^{(n-1)})),$$

where $\mathbf{b}^{(n)}$, the nth backward vector, is the same as $\mathbf{f}^{(n)}$ but with the components in reverse order; and the notation $\mathbf{f}_j^{(n)}$ means the jth component of the nth forward vector.

Having computed the intermediate vectors we use another recursion, building up to the calculation of $\mathbf{x}$ via the vectors $\mathbf{x}^{(n)} \in \Re^n$ which satisfy,

$$\mathbf{K}_n \mathbf{x}^{(n)} = (y_1, y_2, \dots, y_n)^T.$$

We start with,

$$x^{(1)} = \frac{y_1}{k_1},$$

now for $n = 2, \ldots, N$

$$\varepsilon_n = \sum_{j=1}^{n-1} k_{n+1-j} \mathbf{x}_j^{(n-1)},$$

$$\mathbf{x}^{(n)} = (\mathbf{x}^{(n-1)}, 0) + (\mathbf{y}_n - \varepsilon_n) b^{(n)}.$$

The solution we are after is $\mathbf{x}^{(N)} = \mathbf{x}$.

As an added bonus, the determinant of $\mathbf{K}$ can also be calculated from the ϵ_n values from the first recursion:

$$|K| = k_1^N \prod_{j=2}^{N} (1 - \epsilon_j^2)^{N+1-j}.$$

We note that our implementation of the algorithm in R only begins to beat the generic solve function for matrices larger than 500 × 500, and is slower than the Cholesky decomposition until the matrix size nears 900 × 900. Implementation in a lower level language ought to significantly improve this performance.

Appendix C

Deriving properties of the NIW model

C.1 The posterior parameters of the NIW model

Formulating the posterior parameters for the NIW involves the same sort of simple algebra used to derive the posterior parameters for the multivariate normal distribution and the NIG model, but we walk through the calculation here for completeness.

We start with the likelihood for the observations $\mathbf{Y}$,

$$f(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\beta}, \mathbf{H}) = (2\pi)^{-(nq)/2} \left|\mathbf{H}\right|^{-n/2} \left|\mathbf{D}\right|^{-q/2} \exp\left[-\frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}\left(\mathbf{Y}-\mathbf{X}\boldsymbol{\beta}\right)^{T}\mathbf{D}^{-1}\left(\mathbf{Y}-\mathbf{X}\boldsymbol{\beta}\right)\right)\right],$$

and the NIW prior,

$$\pi(\boldsymbol{\beta}, \mathbf{H}) \propto \left|\mathbf{H}\right|^{-(\nu+p+q+1)/2} \exp\left[-\frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}\left(\left(\boldsymbol{\beta}-\mathbf{M}\right)^{T}\mathbf{V}^{-1}\left(\boldsymbol{\beta}-\mathbf{M}\right)+\boldsymbol{\Psi}\right)\right)\right]. \tag{C.1}$$

These two objects are then multiplied to give a function in $(\boldsymbol{\beta}, \mathbf{H})$ proportional to the posterior,

$$\begin{aligned}\pi(\boldsymbol{\beta}, \mathbf{H} \mid \mathbf{X}, \mathbf{Y}) &\propto \left|\mathbf{H}\right|^{-(\nu+n+p+q+1)/2} \\ &\times \exp\left[-\frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}\underbrace{\left(\left(\mathbf{Y}-\mathbf{X}\boldsymbol{\beta}\right)^{T}\mathbf{D}^{-1}\left(\mathbf{Y}-\mathbf{X}\boldsymbol{\beta}\right)+\left(\boldsymbol{\beta}-\mathbf{M}\right)^{T}\mathbf{V}^{-1}\left(\boldsymbol{\beta}-\mathbf{M}\right)+\boldsymbol{\Psi}\right)}_{\mathbf{Q}}\right)\right].\end{aligned} \tag{C.2}$$

Multiplication of the determinant terms is straightforward; we need to invest more effort, however, in unwrapping the object in (C.2) that we have denoted $\mathbf{Q}$, which encodes the posterior's dependency on $\boldsymbol{\beta}$. Firstly, we expand the quadratic forms in $\mathbf{Q}$,

$$\begin{aligned}\mathbf{Q} =& \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{D}^{-1}\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}^T\mathbf{V}^{-1}\boldsymbol{\beta},\\ &- \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{D}^{-1}\mathbf{Y} + \mathbf{Y}^T\mathbf{D}^{-1}\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}^T\mathbf{V}^{-1}\mathbf{M} + \mathbf{M}^T\mathbf{V}^{-1}\boldsymbol{\beta},\\ &+ \mathbf{Y}^T\mathbf{D}^{-1}\mathbf{Y} + \boldsymbol{\beta}^T\mathbf{V}^{-1}\boldsymbol{\beta} + \boldsymbol{\Psi},\end{aligned}$$

and reform them in terms of the new quantities,

$$\mathbf{V}^* = (\mathbf{V}^{-1} + \mathbf{X}^T\mathbf{D}^{-1}\mathbf{X})^{-1}, \tag{C.3}$$

$$\mathbf{M}^* = \mathbf{V}^*(\mathbf{V}^{-1}\mathbf{M} + \mathbf{X}^T\mathbf{D}^{-1}\mathbf{Y}), \tag{C.4}$$

$$\boldsymbol{\Psi}^* = -\mathbf{M}^*(\mathbf{V}^*)^{-1}\mathbf{M}^* + \mathbf{Y}^T\mathbf{D}^{-1}\mathbf{Y} + \boldsymbol{\beta}^T\mathbf{V}^{-1}\beta + \boldsymbol{\Psi}, \tag{C.5}$$

$$\nu^* = \nu + n, \tag{C.6}$$

resulting in,

$$\begin{aligned}\mathbf{Q} =&(\boldsymbol{\beta} - \mathbf{M}^*)^T(\mathbf{V}^*)^{-1}(\boldsymbol{\beta} - \mathbf{M}^*) - \mathbf{M}^*(\mathbf{V}^*)^{-1}\mathbf{M}^* + \mathbf{Y}^T\mathbf{D}^{-1}\mathbf{Y} + \boldsymbol{\beta}^T\mathbf{V}^{-1}\beta + \boldsymbol{\Psi},\\ =&(\boldsymbol{\beta} - \mathbf{M}^*)^T(\mathbf{V}^*)^{-1}(\boldsymbol{\beta} - \mathbf{M}^*) + \boldsymbol{\Psi}^*.\end{aligned}$$

$\mathbf{Q}$ is now expressed as a single quadratic form in $\boldsymbol{\beta}$ plus a constant matrix, allowing us to write the posterior, (C.2), in the same form as (C.1). Consequently, we can identify it with another NIW model whose parameters have been replaced with the starred updates of (C.3)-(C.6):

$$\pi(\boldsymbol{\beta}, \mathbf{H} \mid \mathbf{X}, \mathbf{Y}) \propto |\mathbf{H}|^{-(\nu^*+p+q+1)/2} \exp\left[-\frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}\left((\boldsymbol{\beta} - \mathbf{M}^*)^T(\mathbf{V}^*)^{-1}(\boldsymbol{\beta} - \mathbf{M}^*) + \boldsymbol{\Psi}^*\right)\right)\right].$$

C.2 The normalising constants for the NIW model

C.2.1 The normalising constant for $\pi(\boldsymbol{\beta}, \mathbf{H})$

We start with the expression for the NIW density up to a multiplicative constant $k_{\pi(\boldsymbol{\beta},\mathbf{H})}$,

$$\pi(\boldsymbol{\beta}, \mathbf{H}) = k_{\pi(\boldsymbol{\beta},\mathbf{H})}|\mathbf{H}|^{-(\nu+p+q+1)/2} \exp\left[-\frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}\left((\boldsymbol{\beta} - \mathbf{M})^T\,\mathbf{V}^{-1}\,(\boldsymbol{\beta} - \mathbf{M}) + \boldsymbol{\Psi}\right)\right)\right], \tag{C.7}$$

and proceed to integrate out the parameters $(\boldsymbol{\beta}, \mathbf{H})$ so as to find the value of $k_{\pi(\boldsymbol{\beta},\mathbf{H})}$ for which the integral is equal to one. We begin by integrating out $\boldsymbol{\beta}$,

$$\begin{aligned}\int \pi(\boldsymbol{\beta}, \mathbf{H})\,\mathrm{d}\boldsymbol{\beta} = k_{\pi(\boldsymbol{\beta},\mathbf{H})} \int \exp\left[\frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}\left(\boldsymbol{\beta} - \mathbf{M}\right)^T \mathbf{V}^{-1}\left(\boldsymbol{\beta} - \mathbf{M}\right)\right)\right] \mathrm{d}\boldsymbol{\beta}, \\ \times |\mathbf{H}|^{-(\nu+p+q+1)/2} \exp\left[\frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}\boldsymbol{\Psi}\right)\right].\end{aligned} \tag{C.8}$$

By using theorem B.0.4 in reverse to render the integral in (C.8) an exponential of a negative quadratic form, the solution of the integral is apparent from comparing it with a multivariate normal density:

$$\begin{aligned}\int_{\mathfrak{R}^{pq}} \exp\left[-\frac{1}{2}\left(\mathrm{vec}\left(\boldsymbol{\beta} - \mathbf{M}\right)^T \left(\mathbf{H} \otimes \mathbf{V}\right)^{-1} \mathrm{vec}\left(\boldsymbol{\beta} - \mathbf{M}\right)\right)\right] \mathrm{dvec}\left(\boldsymbol{\beta}\right) &= (2\pi)^{pq/2}|\mathbf{H} \otimes \mathbf{V}|, \\ &= (2\pi)^{pq/2}|\mathbf{H}|^{p/2}|\mathbf{V}|^{q/2}.\end{aligned} \tag{C.9}$$

When we substitute (C.9) into (C.8) we get some cancellation in the powers of $|\mathbf{H}|$, leaving,

$$\begin{aligned}\int \pi(\boldsymbol{\beta}, \mathbf{H})\,\mathrm{d}\boldsymbol{\beta} =& k_{\pi(\boldsymbol{\beta},\mathbf{H})}(2\pi)^{pq/2}|\mathbf{H}|^{p/2}|\mathbf{V}|^{q/2} \times |\mathbf{H}|^{(\nu+p+q+1)/2} \exp\left[\frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}\boldsymbol{\Psi}\right)\right], \\ =& k_{\pi(\boldsymbol{\beta},\mathbf{H})}(2\pi)^{pq/2}|\mathbf{V}|^{q/2} \times |\mathbf{H}|^{(\nu+q+1)/2} \exp\left[\frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}\boldsymbol{\Psi}\right)\right].\end{aligned}$$

Next we need to integrate out $\mathbf{H}$,

$$\int\int \pi(\boldsymbol{\beta}, \mathbf{H})\,\mathrm{d}\boldsymbol{\beta}\,\mathrm{d}\mathbf{H} = k_{\pi(\boldsymbol{\beta},\mathbf{H})}(2\pi)^{pq/2}|\mathbf{V}|^{q/2} \times \int |\mathbf{H}|^{(\nu+q+1)/2} \exp\left[\frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}\boldsymbol{\Psi}\right)\right] \mathrm{d}\mathbf{H}, \tag{C.10}$$

which we do by comparing the integral on the far-right of (C.10) to the density for an inverse-Wishart variable,

$$\int_{\mathbb{M}^+(q,q)} |\mathbf{H}|^{-(\nu+q+1)/2} \exp\left[-\frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}\boldsymbol{\Psi}\right)\right] \mathrm{d}\mathbf{H} = 2^{\nu q/2}\Gamma_q(\nu/2)|\boldsymbol{\Psi}|^{-\nu/2}. \tag{C.11}$$

Substituting (C.11) into (C.10) finally results in the expression,

$$\int\int \pi(\boldsymbol{\beta}, \mathbf{H})\,\mathrm{d}\boldsymbol{\beta}\,\mathrm{d}\mathbf{H} = k_{\pi(\boldsymbol{\beta},\mathbf{H})}(2\pi)^{pq/2}|\mathbf{V}|^{q/2} \times 2^{\nu q/2}\Gamma_q(\nu/2)|\boldsymbol{\Psi}|^{-\nu/2},$$

which we equate to one so that $k_{\pi(\boldsymbol{\beta},\mathbf{H})}$ is revealed to be,

$$k_{\pi(\boldsymbol{\beta},\mathbf{H})} = \frac{|\boldsymbol{\Psi}|^{\nu/2}}{(2\pi)^{pq/2}|\mathbf{V}|^{q/2}\Gamma_q(\nu/2)2^{\nu q/2}}.$$

C.2.2 The normalising constant for $\pi(\mathbf{Y} \mid \mathbf{X})$

To find the normalised marginal density for $\mathbf{Y}$ we consider the mixture of multivariate normal distributions $\pi(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\beta}, \mathbf{H})$ with mixing weights $\pi(\boldsymbol{\beta}, \mathbf{H})$:

$$\pi(\mathbf{Y} \mid \mathbf{X}) = \int \int \pi(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\beta}, \mathbf{H}) \pi(\boldsymbol{\beta}, \mathbf{H}) \, \mathrm{d}\boldsymbol{\beta} \, \mathrm{d}\mathbf{H}. \tag{C.12}$$

We have already computed the product in the integrand of (C.12) in section C.1 when we computed the posterior NIW parameters given $\{\mathbf{X}, \mathbf{Y}\}$. This observation allows us to write the integral using the ratio of the prior and posterior NIW normalising constants,

$$\begin{aligned}
\pi(\mathbf{Y} \mid \mathbf{X}) =& k_{\pi(\boldsymbol{\beta},\mathbf{H})} \int \int (2\pi)^{-(nq)/2} \, |\mathbf{H}|^{-(\nu^*+p+q+1)/2} |\mathbf{D}|^{-q/2} \\
& \times \exp\left[-\frac{1}{2} \mathrm{Tr}\left(\mathbf{H}^{-1}\left((\boldsymbol{\beta} - \mathbf{M}^*)^T (\mathbf{V}^*)^{-1} (\boldsymbol{\beta} - \mathbf{M}^*) + \boldsymbol{\Psi}^*\right)\right)\right] \mathrm{d}\boldsymbol{\beta} \, \mathrm{d}\mathbf{H}, \\
=& (2\pi)^{-(nq)/2} \, |\mathbf{D}|^{-q/2} k_{\pi(\boldsymbol{\beta},\mathbf{H})} k^{-1}_{\pi(\boldsymbol{\beta},\mathbf{H} \mid \mathbf{X},\mathbf{Y})}.
\end{aligned}$$

Cancellation between the NIW normalising constants then results in the expression

$$\pi(\mathbf{Y} \mid \mathbf{X}) = \pi^{-nq/2} |\mathbf{D}|^{-q/2} \frac{\Gamma_q(\nu^*/2)}{\Gamma_q(\nu/2)} \frac{|\mathbf{V}^*|^{q/2}}{|\mathbf{V}|^{q/2}} \frac{|\boldsymbol{\Psi}|^{\nu/2}}{|\boldsymbol{\Psi}^*|^{\nu^*/2}}.$$

C.3 The downdate equations for the NIW model

The NIW downdate equations are used to unlearn about certain data so we can assess how well the model anticipates them, rather than how well it accommodates them, which may be checked by studying the fitted residuals. The downdate equations allow for LOO diagnostics to be computed more quickly than if we were to repeatedly recompute the posterior model parameters using datasets with certain elements left out.

Having adjusted the prior NIW parameters from $\{\mathbf{V}, \mathbf{M}, \boldsymbol{\Psi}, \nu\}$ to $\{\mathbf{V}^*, \mathbf{M}^*, \boldsymbol{\Psi}^*, \nu^*\}$ by the assimilation of input and output quantities $\{\mathbf{X}, \mathbf{Y}\}$, we can derive the posterior parameters that would have resulted from the observation of just $\{[\mathbf{X}]_{-i,\cdot}, [\mathbf{Y}]_{-i,\cdot}\}$ by simply inverting the set of update equations (C.3)-(C.6). The downdated posterior for $(\boldsymbol{\beta}, \mathbf{H})$,

$$\pi(\boldsymbol{\beta}, \mathbf{H} \mid [\mathbf{X}]_{-i,\cdot}, [\mathbf{Y}]_{\cdot}),$$

is also an NIW distribution with parameters, which we label with a bracketed superscript,

$$\mathbf{V}^{(-i)} = \left((\mathbf{V}^{*})^{-1} - [\mathbf{X}]_{i,\cdot}^{T}\mathbf{D}^{-1}[\mathbf{X}]_{i,\cdot}\right)^{-1}, \tag{C.13}$$

$$\mathbf{M}^{(-i)} = \mathbf{V}\left((\mathbf{V}^{*})^{-1}\mathbf{M} - [\mathbf{X}]_{i,\cdot}^{T}\mathbf{D}^{-1}[\mathbf{Y}]_{i,\cdot}\right), \tag{C.14}$$

$$\mathbf{\Psi}^{(-i)} = \mathbf{\Psi}^{*} - \left([\mathbf{Y}]_{i,\cdot} - [\mathbf{X}]_{i,\cdot}\mathbf{M}^{(-1)}\right)^{T}\left(\mathbf{D} + [\mathbf{X}]_{i,\cdot}\mathbf{V}^{(-i)}[\mathbf{X}]_{i,\cdot}^{T}\right)^{-1}\left([\mathbf{Y}]_{i,\cdot} - [\mathbf{X}]_{i,\cdot}\mathbf{M}^{(-i)}\right), \tag{C.15}$$

$$\nu^{(-i)} = \nu^{*} - 1. \tag{C.16}$$

Appendix D

Probability distributions

D.1 A select glossary of distributions

D.1.1 Multivariate Student t-distribution

Notation	$\mathrm{t}_d\left(\mathbf{m}, a\mathbf{V}\right)$
Mean	$\mathbf{m}$
Mode	$\mathbf{m}$
Variance	$a\mathbf{V}/(d-2) \quad d > 2$

PDF

$$f(\mathbf{x} \mid \mathbf{m}, \mathbf{V}, a, d) = \frac{a^{d/2}\Gamma((d+p)/2)}{|\mathbf{V}|^{1/2}\pi^{p/2}\Gamma(d/2)} \left[a + (\mathbf{x}-\mathbf{m})^T\mathbf{V}^{-1}(\mathbf{x}-\mathbf{m})\right]^{-(d+p)/2}.$$

D.1.2 Gamma distribution

Notation	$\text{Gamma}\left(\alpha,\beta\right)$
Mean	$\frac{\alpha}{\beta}$
Mode	$\frac{\alpha-1}{\beta}$
Variance	$\frac{\alpha}{\beta^2}$

PDF

$$\frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp(-\beta x).$$

D.1.3 Inverse gamma distribution

Notation	$\text{Inv-Gamma}\,(\alpha,\beta)$	
Mean	$\frac{\beta}{\alpha-1}$	$\alpha > 1$
Mode	$\frac{\beta}{\alpha+1}$	
Variance	$\frac{\beta^2}{(\alpha-1)^2(\alpha-2)}$	$\alpha > 2$

PDF

$$\frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} \exp(-\beta/x).$$

D.1.4 Log-normal distribution

Notation	$\ln \mathrm{N}\,(\mu, \sigma)$
Mean	$\exp(\mu + \sigma^2/2)$
Median	$\exp(\mu)$
Mode	$\exp(\mu - \sigma^2)$
Variance	$(\exp(\sigma^2) - 1)\exp(2\mu + \sigma^2)$

PDF

$$\pi(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right).$$

The log of variable with a log-normal distribution is normally distributed:

$$Z \sim \mathcal{N}(\mu, \sigma^2) \Leftrightarrow X = \exp(Z) \sim \ln \mathrm{N}\left(\mu, \sigma^2\right).$$

D.1.5 Wishart distribution

Notation	$\mathrm{W}_\nu\,(\mathbf{\Psi})$
Mean	$\nu\mathbf{\Psi}$
Mode	$(\nu - q - 1)\mathbf{\Psi}$
Variance	$\mathrm{Var}\left(H_{ij}\right) = n([\mathbf{\Psi}]_{ij}^2 + [\mathbf{\Psi}]_{ii}[\mathbf{\Psi}]_{jj})$

PDF

$$\pi(\mathbf{H}|\mathbf{\Psi}, \nu) = \frac{1}{|\mathbf{\Psi}|^{\nu/2} 2^{\nu q/2} \Gamma_q(\nu/2)} |\mathbf{H}|^{\nu-q-1)/2} \exp\left[-\frac{1}{2}\mathrm{Tr}(\mathbf{\Psi}^{-1}\mathbf{H})\right].$$

D.1.6 Inverse Wishart distribution

Notation	$\mathrm{IW}_\nu(\mathbf{\Psi})$	
Mean	$\frac{\mathbf{\Psi}}{\nu-q-1}$	$\nu > q+1$
Mode	$\frac{\mathbf{\Psi}}{\nu+q+1}$	

PDF

$$\pi(\mathbf{H}|\mathbf{\Psi},\nu) = \frac{|\mathbf{\Psi}|^{\nu/2}}{2^{\nu q/2}\Gamma_q(\nu/2)}|\mathbf{H}|^{-(\nu+q+1)/2}\exp\left[-\frac{1}{2}\mathrm{Tr}(\mathbf{H}^{-1}\mathbf{\Psi})\right].$$

D.1.7 Central F distribution

Notation	$\mathrm{F}(d_1, d_2)$	
Mean	$\frac{d_2}{d_2-2}$	$d_2 > 2$
Mode	$\frac{d_1-2}{d_1}\frac{d_2}{d_2+2}$	$d_1 > 2$
Variance	$\frac{2d_2^2(d_1+d_2-2)}{d_1(d_2-2)^2(d_2-4)}$	$d_2 > 4$

PDF

$$\pi(x|d_1,d_2) = \frac{\sqrt{\frac{(d_1x)^{d_1}d_2^{d_2}}{(d_1x+d_2)^{d_1+d_2}}}}{xB(d_1/2,d_2/2)}.$$

D.2 Miscellaneous

Theorem D.2.1. *If ϵ is a random variable with mean μ and variance Σ, the expected value of a quadratic form in ϵ is given by*

$$\mathrm{E}(\epsilon^T\Lambda\epsilon) = \mathrm{Tr}(\Lambda\Sigma) + \mu^T\Lambda\mu.$$

If ϵ is normally distributed the quadratic form's variance is

$$\mathrm{Var}(\epsilon^T\Lambda\epsilon) = 2\mathrm{Tr}(\Lambda\Sigma\Lambda\Sigma) + 4\mu^T\Lambda\Sigma\Lambda\mu.$$

Theorem D.2.2 (Gauss Inequality)**.** *For a random variable X admitting a unimodal Lebesgue density with mode ν such that* $\mathrm{E}\left[(X-\nu)^2\right] = \tau^2$,

$$\Pr(|X-\nu| \geq r) \leq \begin{cases} \frac{4\tau^2}{9r^2} & \textit{for all} \quad r \geq \sqrt{4/3}\tau, \\ 1 - \frac{r}{\sqrt{3}\tau} & \textit{for all} \quad r \leq \sqrt{4/3}\tau. \end{cases}$$

Note that there is no requirement for the density to be symmetric nor for it to have finite higher moments.

Theorem D.2.3 (Chebyshev inequality)**.** *For a random variable X with mean μ and variance σ^2,*

$$P\left(|X-\mu| \geq k\sigma\right) \leq k^{-2}.$$

Equivalently,

$$P\left(-k\sigma < X-\mu < k\sigma\right) > 1-k^{-2}.$$

D.2.1 Minimal credible sets

The following argument represents a sketch of the proof that the HLCS of a posterior distribution for a variable x is minimal with respect to its prior distribution. We will write $\pi(x)$ as the prior for x, $l(x)$ as the likelihood of some unspecified data given x, c as the normalising constant $\int_{\Re} l(x)\pi(x)\,\mathrm{d}x$, and h as the lower bound on the likelihood that defines the HLCS.

We start by supposing Ω is an HLCS and Ω' is any other credible set with respect to the posterior. We then define from these three further sets: $I = \Omega \cap \Omega'$, $A = \Omega \cap \Omega'^c$ and $A' = \Omega^c \cap \Omega'$.

It can be shown that the sets satisfy both

$$\Omega = I \cup A \qquad \text{and} \qquad \Omega' = I \cup A'. \tag{D.1}$$

As a consequence, the equality of the integrals that define the credible sets,

$$\alpha = \int_{\Omega} cl(x)\pi(x)\,\mathrm{d}x = \int_{\Omega'} cl(x)\pi(x)\,\mathrm{d}x,$$

implies the equality of the integrals,

$$\int_{A} cl(x)\pi(x)\,\mathrm{d}x = \int_{A'} cl(x)\pi(x)\,\mathrm{d}x.$$

Since $A \subseteq \Omega$ and $A' \nsubseteq \Omega$, the value of $l(x)$ in the set A is greater than h while the value of $l(x)$ in A' is not greater than h. This allows us to write the inequality

$$h\int_{A} \pi(x)\,\mathrm{d}x \leq \int_{A} l(x)\pi(x)\,\mathrm{d}x = \int_{A'} l(x)\pi(x)\,\mathrm{d}x < h\int_{A'} \pi(x)\,\mathrm{d}x,$$

which means that

$$\int_{A} \pi(x)\,\mathrm{d}x < \int_{A'} \pi(x)\,\mathrm{d}x.$$

And because the sets satisfy (D.1) we can also say that

$$\int_{\Omega} \pi(x)\,\mathrm{d}x < \int_{\Omega'} \pi(x)\,\mathrm{d}x.$$

This result represents the statement that Ω, the HLPS, is smaller with respect to the prior than other different sets.

Bibliography

[1] Bäckström, T. *Linear predictive modelling of speech - constraints and line spectrum pair decomposition*. PhD thesis, Aalto University, Finland, 2004.

[2] Bastos, L. S., and O'Hagan, A. Pivoting cholesky decomposition applied to emulation and validation of computer models. Tech. rep., MUCM project, Sheffield University, 2007.

[3] Bayarri, M. J., Berger, J. O., Cafeo, J., Garcia-Donato, G., Liu, F., Palomo, J., Parthasarathy, R. J., Paulo, R., Sacks, J., and Walsh, D. Computer model validation with functional output. *Ann. Statist. 35*, 5 (2007), 1874–1906.

[4] Bayarri, M. J., Berger, J. O., Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, J., Lin, C.-H., and Tu, J. A framework for validation of computer models. *Technometrics 49*, 2 (May 2007), 138–154.

[5] Bernardo, J., and Smith, A. *Bayesian Theory*. Wiley Series in Probability and Statistics. John Wiley & Sons Canada, Limited, 2007.

[6] Brown, P. J. Multivariate calibration. *Journal of the Royal Statistical Society. Series B (Methodological) 44*, 3 (1982), pp. 287–321.

[7] Browne, W. MCMC algorithms for constrained variance matrices. *Computational Statistics and Data Analysis 50*, 7 (2006), 1655–1677.

[8] Bungartz, H.-J., and Griebel, M. Sparse grids. *Acta Numerica 13* (May 2004), 147–269.

[9] Craig, P. S., Goldstein, M., Seheult, A. H., and Smith, J. A. Bayes linear strategies for history matching of hydrocarbon reservoirs. In *Bayesian Statistics 5* (1996),

J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, Eds., Clarendon Press, Oxford, pp. 69–95.

[10] Dette, H., and Trampisch, M. A general approach to d-optimal designs for weighted univariate polynomial regression models. *Journal of the Korean Statistical Society 39*, 1 (2010), 1 – 26.

[11] Fearnhead, P., and Prangle, D. Constructing Summary Statistics for Approximate Bayesian Computation: Semi-automatic ABC. *ArXiv e-prints* (Apr. 2010).

[12] Gamerman, D. *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. Texts in statistical science. Chapman & Hall, 1997.

[13] Ganachaud, A., and Wunsch, C. Large-scale ocean heat and freshwater transports during the world ocean circulation experiment. *Journal of Climate 16* (Feb. 2003), 696–705.

[14] Goldstein, M. External bayesian analysis for computer simulators. *Bayesian Statistics 9*, 1996 (2010).

[15] Goldstein, M., and Rougier, J. Probabilistic formulations for transferring inferences from mathematical models to physical systems. *SIAM J. Sci. Comput. 26*, 2 (2004), 467–487 (electronic).

[16] Goldstein, M., and Rougier, J. Reified Bayesian modelling and inference for physical systems. *J. Statist. Plann. Inference 139*, 3 (2009), 1221–1239.

[17] Goldstein, M., and Wooff, D. *Bayes Linear Statistics, Theory and Methods*. Wiley, 2007.

[18] Gu, C. *Smoothing Spline ANOVA Models*. IMA Volumes in Mathematics and Its Applications. Springer, 2002.

[19] Gu, C., and Qiu, C. Smoothing spline density estimation: Theory. *The Annals of Statistics 21*, 1 (1993), pp. 217–234.

[20] Heiss, F., and Winschel, V. Likelihood approximation by numerical integration on sparse grids. *Journal of Econometrics 144*, 1 (2008), 62 – 80.

[21] Higdon, D., Gattiker, J., Williams, B., and Rightley, M. Computer model calibration using high-dimensional output. *J. Amer. Statist. Assoc. 103*, 482 (2008), 570–583.

[22] Hotelling, H. Some new methods in matrix calculation. *Ann. Math. Statistics 14* (1943), 1–34.

[23] Jenkins, G., and Watts, D. *Spectral analysis and its applications*. Holden-Day series in time series analysis. Holden-Day, 1969.

[24] Kaufman, C., Bingham, D., Habib, S., Heitmann, K., and Frieman, J. Efficient emulators of computer experiments using compactly supported correlation functions, with an application to cosmology. *ArXiv e-prints* (Jul. 2011).

[25] Kennedy, M. C., and O'Hagan, A. Bayesian calibration of computer models. *J. R. Stat. Soc. Ser. B Stat. Methodol. 63*, 3 (2001), 425–464.

[26] Kimeldorf, G. S., and Wahba, G. A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics 41*, 2 (1970), pp. 495–502.

[27] Lawrence, N., Seeger, M., and Herbrich, R. Fast sparse gaussian process methods: The informative vector machine. In *Advances in Neural Information Processing Systems 15* (2003), S. Becker, S. Thrun, and K. Obermayer, Eds., MIT Press, pp. 625–632.

[28] Levine, R., Yu, Z., Hinley, W., and Mitao, J. Implementing random scan Gibbs samplers. *Comput. Stat. 20*, 1 (2005), 177–196.

[29] MacKay, D. Hyperparameters: Optimize, or integrate out? In *Maximum Entropy and Bayesian Methods*, G. Heidbreder, Ed., vol. 62 of *Fundamental Theories of Physics*. Springer Netherlands, 1996, pp. 43–59.

[30] Mackay, D. J. C. *Introduction to Gaussian Processes*. Cambridge University. Unpublished technical report.

[31] Martin, J., Wilcox, L., Burstedde, C., and Ghattas, O. A stochastic newton mcmc method for large-scale statistical inverse problems with application to seismic inversion. *SIAM Journal on Scientific Computing 34*, 3 (2012), A1460–A1487.

[32] Musicus, B. R. Levinson and fast choleski algorithms for toeplitz and almost toeplitz matrices. Tech. rep., Research laboratiry of electronics, MIT, 1984.

[33] Naylor, J. C., and Smith, A. F. M. Applications of a method for the efficient computation of posterior distributions. *Journal of the Royal Statistical Society. Series C (Applied Statistics) 31*, 3 (1982), pp. 214–225.

[34] O'Hagan, A. Polynomial chaos: A tutorial and critique from a statistician's perspective. *SIAM/ASA Journal of Uncertainty Quantification* (2013), (submitted.).

[35] O'Hagan, A., and Forster, J. *Kendall's Advanced Theory of Statistics: Volume 2B: Bayesian Inference*. Kendall's Advanced Theory of Statistics. Hodder Arnold, 2004.

[36] Patterson, T. N. L. The optimum addition of points to quadrature formulae. *Mathematics of Computation 22*, 104 (1968), pp. 847–856+s21–s31.

[37] Pearson, K. *On Lines and Planes of Closest Fit to Systems of Points in Space*. University College, 1901.

[38] Petersen, K. B., and Pedersen, M. S. The matrix cookbook. `http://www2.imm.dtu.dk/pubdb/p.php?3274`, Nov. 2012. Version 20121115.

[39] Pinheiro, J. C., and Bates, D. M. Unconstrained parametrizations for variance-covariance matrices. *Statistics and Computing 6* (1996), 289–296. 10.1007/BF00140873.

[40] Plessis, J. L. d., and Merwe, A. J. V. D. Inferences in multivariate bayesian calibration. *Journal of the Royal Statistical Society. Series D (The Statistician) 43*, 1 (1994), pp. 45–60.

[41] Quiñonero Candela, J., and Rasmussen, C. E. A unifying view of sparse approximate gaussian process regression. *J. Mach. Learn. Res. 6* (dec 2005), 1939–1959.

[42] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2010. ISBN 3-900051-07-0.

[43] Ramsay, J. O., and Silverman, B. W. *Functional data analysis*. Springer-Verlag, New York, USA, 1997.

[44] Rasmussen, C. E., and Williams, C. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[45] Ratmann, O., Andrieu, C., Wiuf, C., and Richardson, S. Model criticism based on likelihood-free inference, with an application to protein network evolution. *Proceedings of the National Academy of Sciences 106*, 26 (2009), 10576–10581.

[46] Robert, C. P., Mengersen, K., and Chen, C. Model choice versus model criticism. *Proceedings of the National Academy of Science 107* (Jan. 2010), 5.

[47] Roberts, G. O., and Rosenthal, J. S. Optimal scaling for various metropolis-hastings algorithms. *Stat. Sci. 16*, 4 (2001), 351–367.

[48] Rougier, J. Efficient emulators for deterministic functions. *Journal of Computational and Graphical Statistics 17*, 4 (2008), 827–843.

[49] Santner, T. J., Williams, B. J., and Notz, W. I. *The design and analysis of computer experiments*. Springer Series in Statistics. Springer-Verlag, New York, 2003.

[50] Scheidegger, A. adaptmcmc: Implementation of a generic adaptive monte carlo markov chain sampler. `http://CRAN.R-project.org/package=adaptMCMC`, 2012. R package version 1.1.

[51] Silverman, B. W. On the estimation of a probability density function by the maximum penalized likelihood method. *The Annals of Statistics 10*, 3 (1982), pp. 795–810.

[52] Soetaert, K., Petzoldt, T., and Setzer, R. W. Solving differential equations in r: Package desolve. *Journal of Statistical Software 33*, 9 (2010), 1–25.

[53] Stein, M. Nonstationary spatial covariance functions. http://www-personal.umich.edu/~jizhu/jizhu/covar/Stein-Summary.pdf, 2005. Unpublished technical report.

[54] Stein, M. L. *Interpolation of Spatial Data*. Springer-Verlag, New York, USA, 1999.

[55] Tresp, V. A Bayesian committee machine. *Neural Computation 12*, 11 (Nov. 2000), 2719–2741.

[56] Vernon, I., Goldstein, M., and Bower, R. G. Galaxy Formation: a Bayesian Uncertainty Analysis. *Bayesian analysis 5*, 4 (2010), 619–669.

[57] Vihola, M. Robust adaptive metropolis algorithm with coerced acceptance rate. *ArXiv e-prints* (Nov. 2010).

[58] Wahba, G. *Spline Models for Observational Data*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, 1990.

[59] Wolfram, S. Expansion of the modified bessel function of the second kind for specific values. http://functions.wolfram.com/03.04.03.0004.01, 2001.

[60] Wood, S. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature 466*, 7310 (2010), 1102–1104.

[61] Xiu, D., and Karniadakis, G. E. The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput. 24*, 2 (Feb. 2002), 619–644.

[62] Ympa, J. Sparsegrid. http://CRAN.R-project.org/package=SparseGrid, 2011.

[63] Zhu, H., Williams, C., Rohwer, R., and Morciniec, M. Gaussian regression and optimal finite dimension linear models. In *Neural Networks and Machine Learning* (1998), C. Bishop, Ed., Springer-Verlag, Berlin, p. pp. 97.