Summer 7-15-2020

Methods of Uncertainty Quantification for Physical Parameters

Kellin Rumsey
University of New Mexico

Kellin Rumsey

Candidate

Mathematics and Statistics

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Gabriel Huerta, Chairperson

Lauren Hund

Ronald Christensen

Trilce Estrada

Methods of Uncertainty Quantification for Physical Parameters

by

Kellin N. Rumsey

B.S., Mathematics, University of Arizona, 2016
M.S., Statistics, University of New Mexico, 2018
M.S., Computer Science, University of New Mexico, 2020

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctorate of Philosophy
Statistics

The University of New Mexico

Albuquerque, New Mexico

July, 2020

Dedication

For Jaxon and Nala

"Only those who attempt the absurd will achieve the impossible. I think it's in my basement... let me go upstairs and check." – M.C. Escher

Acknowledgments

Without the guidance of my advisors Gabriel Huerta and Lauren Hund, I would not be where I am today. There is not enough room on this page to type "thank you" as many times as you deserve. I also thank Professor Ronald Christensen and Professor Trilce Estrada for their helpful feedback and comments regarding this thesis. I thank Justin Newcomer and Derek Tucker for funding parts of this research and, more importantly, for their support. Special thanks also goes out to Jaimie Lin, Kyle Henke, Zach Sturat, Huan Yu, Malik Barrett, Andrew Taylor, Amy Umaretiya, Danny Ries, my parents and siblings (and niece) and many others for their advice, friendship and for keeping me sane during this journey.

This work was supported by a Sandia National Laboratories Laboratory Directed Research and Development (LDRD) grant. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND 2020-7154 T

This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

Methods of Uncertainty Quantification for Physical Parameters

by

Kellin N. Rumsey

B.S., Mathematics, University of Arizona, 2016

M.S., Statistics, University of New Mexico, 2018

M.S., Computer Science, University of New Mexico, 2020

Ph.D. Statistics, University of New Mexico, 2020

Abstract

Uncertainty Quantification (UQ) is an umbrella term referring to a broad class of methods which typically involve the combination of computational modeling, experimental data and expert knowledge to study a physical system. A parameter, in the usual statistical sense, is said to be physical if it has a meaningful interpretation with respect to the physical system. Physical parameters can be viewed as inherent properties of a physical process and have a corresponding true value. Statistical inference for physical parameters is a challenging problem in UQ due to the inadequacy of the computer model. In this thesis, we provide methods for UQ for physical parameters in the presence of model discrepancy which allow us to save time, identify overfitting, incorporate physical constraints, diagnose challenging problems and provide more robust answers to the inverse problem.

Contents

List of Figures

List of Tables

Glossary

$\mathbb{1}(b)$	The indicator function; outputs 1 if b is true 0 otherwise.
$\mathcal{O}(\cdot)$	Big-O notation. $f(n) = \mathcal{O}(g(n))$ if $\limsup_{n\to\infty} \lvert\frac{f(n)}{g(n)}\rvert < \infty$
I_p	The $p \times p$ identity matrix.
J_p	The $p \times p$ matrix of all ones.
$\lVert\boldsymbol{v}\rVert_2$	The L_2 vector norm
x, y, etc.	A scalar.
$\boldsymbol{x}$, $\boldsymbol{y}$, etc.	A vector. Typically a row vector (where it matters).
$\boldsymbol{x}^\top$	Transpose of a vector
$\boldsymbol{X}$, $\boldsymbol{Y}$, etc	A matrix.
$\boldsymbol{X}_{ij}$	The element of $\boldsymbol{X}$ in row i and column j.
$\boldsymbol{X}^\top$	Transpose of a matrix
$\boldsymbol{X}^{-1}$	Inverse of a matrix
Diag($\boldsymbol{a}$)	A diagonal matrix with $(ii)^{th}$ element a_i.
$\odot$	The Hadamard product. If A and B are $n \times m$ matrices, then $(A \odot B)_{ij} = A_{ij}B_{ij}$

Glossary

$\times$	The Cartesian product. If A and B are spaces then $A \times B = \{(a, b) \vert a \in A, b \in B\}$.
$N(\mu, \sigma^2)$	The normal distribution with mean μ and standard deviation σ.
$\text{Unif}(a, b)$	The uniform distribution on the interval (a, b).
$\text{Gamma}(a, b)$	The gamma distribution with mean a/b.
$\text{Beta}(a, b)$	The two parameter Beta distribution with shape parameters a and b.
$t(v)$	The t distribution with v degrees of freedom.
$C_+(\mu, \sigma)$	The half-Cauchy distribution with location μ and scale σ.
$\log N(\mu, \sigma)$	The log normal distribution with log-mean μ and log-sd σ.
$GP(m(\cdot), C(\cdot, \cdot))$	A Gaussian process with mean function $m(\cdot)$ and covariance function $C(\cdot, \cdot)$.
$\mathcal{F}(x \vert \psi_1, \cdots \psi_r)$	Indicates the probability density function of a RV X having distribution $X \sim \mathcal{F}(\psi_1 \cdots \psi_r)$.

Chapter 1

Introduction & Background

"The only true wisdom is in knowing that you do not know." – Socrates

1.1 Overview

There is no generally accepted definition of Uncertainty Quantification (UQ), perhaps due to its separate and simultaneous development in the fields of applied mathematics, statistics and engineering/physics. Nonetheless, many authors have attempted to define UQ, and some notable examples are given as follows [38].

> *"UQ is precisely the quantification of one's lack of knowledge concerning (in science and engineering) a physical reality." – J. Tinsley Oden*

> *"Combining computational models, physical observations and possibly expert judgment to make inferences about a physical system." – David Higdon*

> *"UQ is about providing bounds on our knowledge of system behavior and on confidence in our predictions." – Omar Knio*

The key component which can be found at the intersection of almost every definition of UQ, is that physics-inspired computational modeling, statistical inference and often experimental data are combined in an effort to learn about a physical system.

Although rigorously defining UQ is a challenging task, we subscribe to the the old adage *if it looks like UQ, swims like UQ and quacks like UQ, then it probably is UQ*. By this, we mean that UQ can often be recognized in terms of the problems that it attempts to solve even though the methods used to solve these problems may vary. A few examples of physical systems which may be of interest include (i) the effect of global climate on glaciers and sea ice [70, 82, 122], (ii) the behavior of a tropical storm [2], (iii) the spread of a global pandemic [111] and (iv) the response of a material under extreme conditions [21, 132]. The primary goal of UQ is to learn about these systems (and many others) and to characterize or quantify precisely what we know and do not know about them. To accomplish this, UQ may involve (i) forward propagation of uncertainty [62, 89, 97], (ii) inverse problems [87, 117, 148], (iii) response surface modeling [9, 66], (iv) design of computer experiments [113, 133], (v) verification and validation [109], (vi) sensitivity analysis [134] and (vii) dimension reduction [37, 48, 152].

In this thesis, we will primarily focus on the inverse problem, in which experimental data is combined with a computational model and expert knowledge in order to learn about physical parameters that govern the physical system of interest. This is widely recognized as a challenging problem, especially in comparison with so-called *forward UQ*, which involves making predictions about the physical system at previously unobserved settings [87, 110].

The remainder of this thesis is summarized as follows. In Chapter 1 we give a thorough summary of the UQ background that will be relevant to our work. Topics include computer model emulation, model calibration, design of computer experiments and Bayesian regularization. This is intended to be a contribution in its own right, because comprehensive summaries of UQ methodologies are presently lacking and we include many foundational and modern references. In Chapter 2, we propose an extension of the Local Approximate Gaussian process framework of [66] and demonstrate its superior computational properties for inherently sequential settings. We suggest that the emulators developed in this setting are ideal for Bayesian model calibration problems when the training set is too large for a standard Gaussian process to be tractable. In Chapter 3, we return our attention to the primary objective of physical parameter inference in the context of model calibration. We propose a novel new metric for overfitting and a related regularization prior for the case where measurement uncertainties are included as computer model inputs. In Chapter 4, we take a detailed look at the inverse problem when the parameter of interest has a physical interpretation. While it is known that the predominant model calibration framework can fail in this setting, we show that even modern approaches for tackling identifiability are insufficient for this problem. We propose a new modularization framework for Bayesian model calibration and demonstrate its value as a tool for diagnosing the identifiability of physical parameters with respect to the computer model. In Chapter 5, we discuss methods for performing cross validation when the data has spatial or temporal structure as in model calibration. We demonstrate how these cross validation strategies can be used to construct empirical priors for the model discrepancy function and modularization is discussed in this setting.

1.2 Motivation: Compressibility of Tantalum

Consider a class of dynamic materials experiments in which a strong and sudden force is imposed on a material of interest. There are a variety of impulses that may be under consideration including explosives [7], guns [6], lasers [45] and pulsed magnetic fields [5]. The relevant boundary condition generates extreme pressures and induces a directional stress wave which then propagates through the material of interest. The way in which a material responds when subjected to extreme pressures is related to the *compressibility* of the material and is a matter of profound scientific interest. The broad "parameter" here is the function which describes the pressure-density relationship of the material. For a given material, manufactured in a controlled environment, this functional parameter can be viewed as an inherent property of nature and must possess some "true value". Rather than trying to estimate an infinite dimensional parameter, the discussion can be simplified going forward by specifying a parametric form for the pressure-density relationship. This parametric form, known as the *equation of state* (EoS), reduces the estimation problem to that of estimating a finite set of EoS parameters, denoted $(\alpha_1, \alpha_2, \cdots \alpha_p)$. Many physically motivated EoS forms exist, including the Vinet [157] or Mie Gruneisen [75] equations, but the particular selection should be made with regard for the material of interest and with the use of expert opinion.

As a proof of concept, we will consider a set of previously published measurements on tantalum which were generated using Sandia National Laboratories' Z-Machine, the world's largest electromagnetic wave generator [20, 136]. These experiments represent a useful testbed for model calibration methodology, because they have been previously analyzed using state-of-the-art analytic techniques, with Bayesian model calibration and again with a Bayesian *effective sample size* (ESS) calibration procedure [20, 21]. In the simplified version of the experimental set up, shown in the left panel of Figure 1.1, a strong magnetic field is generated as a boundary condition. Aluminum (Al) acts as an electrode and leads to a stress wave which propagates through the system from left to right, and the velocity of this stress wave is measured at the interface of the tantalum and lithium fluoride (LiF) samples. In this example, Al and LiF are chosen for their desirable material properties, but also because these materials behavior under extreme pressures are well understood which isolates the pressure-density relationship for tantalum as the primary unknown. The field data, shown in the right panel of Figure 1.1, consists on $p = 9$ different experiments where the color is used to represent experiments with (nearly) identical design variable settings. In [21], it is noted that the peak pressure reached in these experiments is $70 - 240GPa$ which is on the order of the pressure in the Earth's inner core.

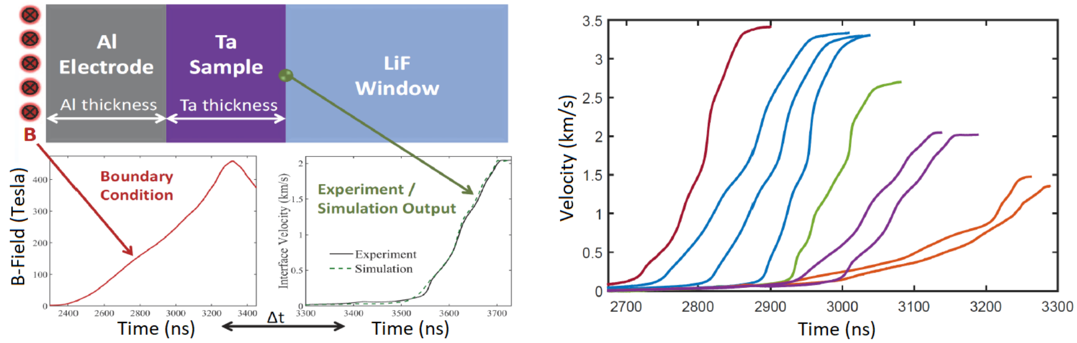


Figure 1.1: (left) A simplified model of the experimental setup. A magnetic field boundary condition results in a stress wave which propagates through the system as a function of time. The velocity of this stress wave is measured at the Ta LiF interface. The velocity of the stress wave is influenced by the thickness of the tantalum and aluminum samples, the thickness of the samples (which are measured with small error) and the scaling of the boundary condition. (right) The experimental data consists of the measured velocity curve for each of the $p = 9$ experiments.

For tantalum, prior (i.e. expert) information indicates that the pressure (P) density (ρ) relationship may be appropriately modeled using the Vinet equation of state

$$P(\rho) = 3B_0 \left(\frac{1-\xi}{\xi^2}\right) \exp\left\{\frac{3}{2}(B_0' - 1)(1-\xi)\right\} \tag{1.1}$$

where

$$\xi(\rho) = (\rho_0/\rho)^{1/3}. \tag{1.2}$$

The parameters (B_0, B_0', ρ_0) are the *bulk modulus*, the *bulk modulus pressure derivative* and the *initial (ambient) density* of tantalum. We refer to these as physical parameters, since they have a physical interpretation and they presumably have "true" meaningful values which are inherent properties of the system. Alternatively, we can summarize the compressibility of tantalum using the pressure strain relationship $P(s)$, which is defined again by eq. (1.1) and eq. (1.3) but with strain defined as

$$s = 1 - \frac{\rho}{\rho_0}. \tag{1.3}$$

This implies that compressibility can be summarized with just the bulk modulus (B_0) and its pressure derivative (B_0'). Thus we might say that (B_0, B_0', ρ_0) are all physical parameters, but only (B_0, B_0') are of primary scientific interest.

The ALEGRA wave propagation code [129] can be used to simulate functional outputs which can then be compared to the velocimetry curves shown in Figure 1.1 and used for calibration. The computer model for a single experiment can be represented as $\eta(x, \boldsymbol{\alpha}, \boldsymbol{\gamma})$ where x denotes time (the single design variable) and $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\gamma})$ represent a set of calibration parameters. As described in Section 1.4.3, we partition the calibration parameters as $\boldsymbol{\alpha} = (B_0, B_0')$ and $\boldsymbol{\gamma} = (\rho_0, \tau_{Al}, \tau_{Ta}, B_{\text{scale}}, \Delta t)$ to represent physical and nuisance parameters respectively. The parameters τ_{Al} and τ_{Ta} represent the thickness of the aluminum and tantalum samples. Although these thickness parameters can be measured, they are measured with error and even a small change in thickness can significantly impact the output of the computer model. Finally, the parameter B_{scale} is a constant magnetic field scaling term associated with the boundary condition and Δt describes the constant time-offset between the experiments. In practice, we can align the functional data prior to calibration and fix $\Delta t = 0$ for all experiments. As a final comment, we note that some of the parameters (i.e. ρ_0) have physical interpretations and an argument could be made that these parameters belong in $\boldsymbol{\alpha}$. Our choice is based on the parameters whose values are of scientific interest with respect to the compressibility of tantalum. To simplify notation, we write

$$\begin{aligned}(B_0, B_0') &\coloneqq (\alpha_1, \alpha_2) = \boldsymbol{\alpha}\\ (\rho_0, \tau_{Al}, \tau_{Ta}, B_{\text{scale}}) &\coloneqq (\gamma_1, \gamma_2, \gamma_3, \gamma_4) = \boldsymbol{\gamma}.\end{aligned} \tag{1.4}$$

Although there are $p = 9$ experiments, we can assume that B_0 and B_0' are constant across all experiments. Since the tantalum samples were all cut from the same plate, we will also assume that ρ_0 is constant across each experiment, but we assume that the remaining three nuisance parameters have values which are experiment dependent. This implies that the full set of calibration parameters is defined as

$$\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \cdots \boldsymbol{\gamma}_9) \tag{1.5}$$

where $\boldsymbol{\gamma}_j = (\gamma_1, \gamma_{2j}, \gamma_{3j}, \gamma_{4j})$. Thus there are a total of 27 calibration parameters in this model. The model likelihood is expressed as the product of $p = 9$ independent likelihoods, where the likelihood for each experiment is given by eq. (1.26).

Since the ALEGRA wave propagation code is expensive to run, we are not given full access to the model. Rather, we are given a set of $d = 5000$ input-output pairs for each experiment, where the matrix of design points $\boldsymbol{X}_j$ is a generated using a Latin hypercube design over the appropriate space. See [21] for specifics on the generation of the design points or Section 1.5.1 for a general discussion of Latin hypercube sampling.

1.3 The Computer Model and Emulation

Many important scientific applications use mathematical models to describe complex physical processes [20, 67, 76, 111, 161]. As an example, consider the behavior of a hurricane off the east coast of North America. For obvious reasons, scientists are interested in understanding the evolution of these tropical storms. A better understanding of how these storms move and change in space and time could lead directly to strategies that save lives and money. Hurricanes are rare events, and thus studying these phenomena is expensive and time consuming. In the age of computation power, computer models (or deterministic simulators), based on mathematics and the physics of a process, have proved invaluable to the study of physical phenomena.

Throughout this thesis, we will denote the *computer model* by $\eta(\boldsymbol{x}) \; : \; \mathcal{X} \to \mathbb{R}$, where the input space $\mathcal{X}$ is typically a subset of $\mathbb{R}^p$. In later sections and chapters, it will be useful to partition the input vector $\boldsymbol{x}$ into several different "types" of inputs, but we make no such distinction here. We also assume that the output of the computer model, i.e. the response variable, is a real number, but we acknowledge that multivariate responses are possible and not uncommon [11, 76]. Although the terms *computer model* and *simulator* are often used interchangeably, we prefer the former since it does not obscure the deterministic nature of the model. In practice, computer models often require a large amount of computation time. The models are seldom linear or even convex functions of the (possibly high dimensional) input space. Finally, the details of the computer model itself may be proprietary or classified. Instead of having full access to the black-box computer model, we are often given a set of input-output pairs, $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{\eta})$, where $\boldsymbol{X}$ is a $d \times p$ matrix of inputs and $\boldsymbol{\eta}$ is a d-dimensional vector of outputs. Explicitly, the j^{th} element of $\boldsymbol{\eta}$ is $\eta_j = \eta(\boldsymbol{x}_j)$ where $\boldsymbol{x}_j$ is the j^{th} row of $\boldsymbol{X}$. It is sometimes convenient to informally write $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \boldsymbol{x}_d)$ and $\boldsymbol{x}_j \in \boldsymbol{X}$ with the intent that $\boldsymbol{x}_j$ is a row of $\boldsymbol{X}$.

An *emulator*, also called a *statistical surrogate* or *metamodel*, can be viewed as a proxy for the computer model and is denoted by $\hat{\eta}(\boldsymbol{x})$. The emulator should be cheap to evaluate and capable of making predictions at all $\boldsymbol{x} \in \mathcal{X}$, even if the particular inputs are not in the training set $\mathcal{D}$. To reflect the deterministic nature of the simulator, a reasonable emulator should be an *interpolator*, which is to say that $\hat{\eta}(\boldsymbol{x}) = \eta(\boldsymbol{x})$ for every $\boldsymbol{x} \in \boldsymbol{X}$.

For example, the least squares response surface, comprised of adding together terms of the form $\prod_{j=1}^{p} x_j^{a_j}$, with each $a_j \in \{0, 1, 2, \cdots a\}$, will not be an interpolator unless the order a is exceptionally large, in which case the out of sample predictions are typically worthless due to Runge's phenomenon [43].

1.3.1 Nearest Neighbor Emulators

Perhaps the simplest example of an interpolator is the *nearest neighbor* (NN) emulator. For location $\boldsymbol{x}$ and distance function $d(\cdot, \cdot)$, we can define $d_j = d(\boldsymbol{x}, \boldsymbol{x}_j)$, for $j = 1, 2, \cdots d$. Let $d_{(1)}, \cdots d_{(d)}$ denote the sorted distances, so that $d_{(1)} \leq d_{(2)} \leq \cdots \leq d_{(d)}$. Finally, we define $N_K(x_0)$ to be the K-nearest neighbor index set, such that $j \in N_K(x_0)$ if and only if $d_j \leq d_{(K)}$. With respect to these definitions, the NN emulator is given by

$$\hat{\eta}_{NN}(\boldsymbol{x}) = \eta_j, \ j \in N_1(\boldsymbol{x}). \tag{1.6}$$

A more sophisticated choice, it would seem, is to use the information provided by the K nearest neighbors of $\boldsymbol{x}_0$, with $K > 1$. Taking the average prediction of the K nearest neighbors of $\boldsymbol{x}_0$ yields the classical K-NN predictor

$$\hat{\eta}_{KNN}(\boldsymbol{x}) = \frac{1}{K} \sum_{j \in N_K(\boldsymbol{x})} \eta_j \tag{1.7}$$

The corresponding prediction surface typically underwhelms in practice, and also loses the interpolation property. The interpolation property can be restored, while simultaneously "smoothing" the surface and improving performance, by taking a weighted average

$$\hat{\eta}_{sKNN}(\boldsymbol{x}) = \sum_{j \in N_K(\boldsymbol{x})} w_j \eta_j, \tag{1.8}$$

with weights

$$w_j \propto \frac{1}{d_j}, \quad \sum_{j \in N_K(\boldsymbol{x})} w_j = 1. \tag{1.9}$$

The interpolation property is satisfied here by adopting the convention that $w_j = 1$ and $w_{j'} = 0$ $(j \neq j')$ whenever $d_j = 0$.

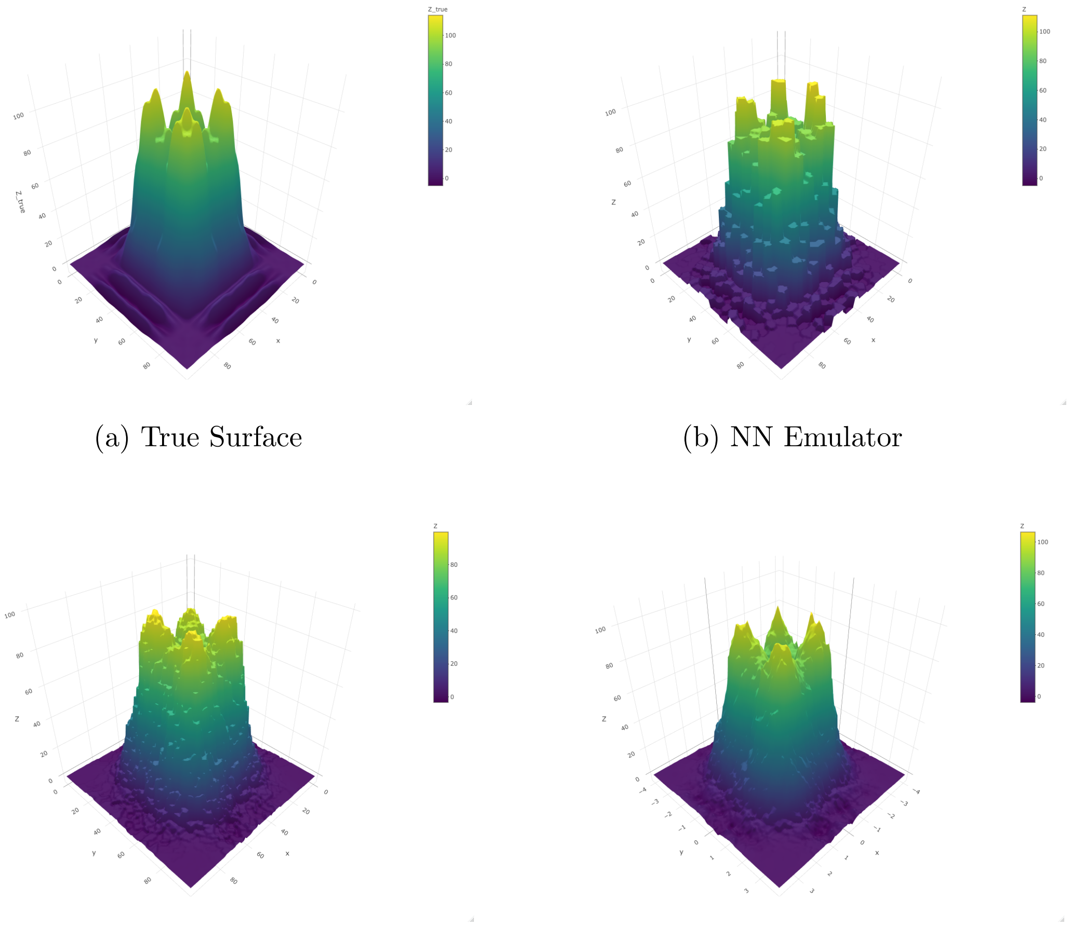
An Illustration

For illustrative purposes, consider the surface defined by the Gramacy-Lee function [68]

$$\begin{aligned} \eta(x_1, x_2) &= 100 w(x_1) w(x_2), \text{ where} \\ w(x) &= \exp\left(-(x-1)^2\right) + \exp\left(-0.8(x+1)^2\right) - 0.05 \sin\left(8(x+0.1)\right), \end{aligned} \tag{1.10}$$

for $\boldsymbol{x} \in \mathcal{X} = [-4, 4]^2$. Training data $\mathcal{D}$ is generated by taking $\boldsymbol{X}$ to be a two dimensional maximin *Latin hypercube sample* (LHS) with $d = 400$ points [98, 113]. We then request $T = 10,000$ predictions at each location on a rectangular grid covering the input space $\mathcal{X}$. Figure 1.2a shows the true Gramacy-Lee surface, while Figures 1.2b, 1.2c and 1.2d illustrate the emulated surfaces using the emulators described in this section (with $K = 5$).

Nearest neighbor emulation is a convenient choice for its simplicity but will typically be outperformed by other methods. In particular, nearest neighbor methods are known to struggle in high dimensions, where astronomically large amounts of training data are required to obtain reasonable accuracy [43]. Although this “curse of dimensionality” is not completely unique to KNN emulation, it is particularly susceptible due to its simplicity. There are also theoretical limitations with the NN family of emulators such as non-differentiability and the inability to ever predict a value of the response which is beyond the range of the previously observed outputs $\boldsymbol{\eta}$.



(a) True Surface

(b) NN Emulator

(c) K-NN Emulator ($K = 10$)

(d) Smooth K-NN Emulator ($K = 10$)

Figure 1.2: The Gramacy-Lee surface and emulated surfaces using Nearest Neighbor based approaches.

1.3.2 Gaussian Process Emulation

By far, the Gaussian Process (GP) is the most popular choice for emulation in the realm of computer experiments. It is flexible, accurate, efficient in making predictions and can be readily fit using most statistical software. For our purposes, the GP can be viewed as a distribution over the family of real-valued functions $\hat{\eta} : \mathbb{R}^p \to \mathbb{R}$, such that every finite collection of outputs follow a multivariate normal distribution. That is to say, for any finite collection $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \boldsymbol{x}_p$, the output vector $\hat{\boldsymbol{\eta}} = (\hat{\eta}_1, \hat{\eta}_2, \cdots \hat{\eta}_p)^\top$, where $\hat{\eta}_j = \hat{\eta}(\boldsymbol{x}_j)$, is distributed as

$$\hat{\boldsymbol{\eta}} \sim N\left(\boldsymbol{\mu},\ \boldsymbol{\Sigma}\right).$$

The mean vector $\boldsymbol{\mu}$ has components μ_j and the $(jk)^{th}$ entry of the covariance matrix is parametrized as

$$\boldsymbol{\Sigma}(\boldsymbol{x}_j, \boldsymbol{x}_k) = \phi R(\boldsymbol{x}_j, \boldsymbol{x}_k \mid \kappa) + \tau \mathbb{1}(\boldsymbol{x}_j = \boldsymbol{x}_k). \tag{1.11}$$

In this thesis, we will typically specify

$$R(\boldsymbol{x}_j, \boldsymbol{x}_k \mid \kappa) = \exp\left\{-\kappa \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2\right\}, \tag{1.12}$$

which is known as the *isotropic Gaussian correlation function*, but other correlation functions such as the Matérn are sometimes desirable alternatives [41]. This specification of the covariance structure specifies that outputs should be more highly correlated when the corresponding inputs are close to each other. In particular, the correlation parameter κ controls the strength of the correlation as a function of distance, while ϕ controls the variance. The parameter τ is often called the *nugget*, and the GP is an interpolator whenever $\tau = 0$. In practice, it is often best for numerical reasons to set τ equal to some small positive number, such as the square root of machine epsilon.

Although it can be a challenging problem, the parameters ϕ and κ can be estimated using a variety of methods, typically falling into the categories of maximum likelihood or empirical Bayes [13, 124]. Conditional on the observed data and on the estimated parameters, the Gaussian process emulator can be written as

$$\hat{\eta}_{GP}(\boldsymbol{x}) = E\left(\hat{\eta}(\boldsymbol{x}) | \hat{\eta}_1, \cdots, \hat{\eta}_d\right) = r^\top(\boldsymbol{x}) R^{-1} \hat{\boldsymbol{\eta}}, \tag{1.13}$$

where $r^\top(\boldsymbol{x})$ is the d-vector whose j^{th} component is $R(\boldsymbol{x}, \boldsymbol{x}_j \mid \kappa)$ and R is the matrix whose $(jk)^{th}$ element is $R(\boldsymbol{x}_j, \boldsymbol{x}_k \mid \kappa)$. Since $R^{-1}\hat{\boldsymbol{\eta}}$ can be computed and stored in advance, a prediction can be obtained for a new input $\boldsymbol{x}$ with an amortized cost of $\mathcal{O}(d)$, the time required for a single vector-vector multiply.

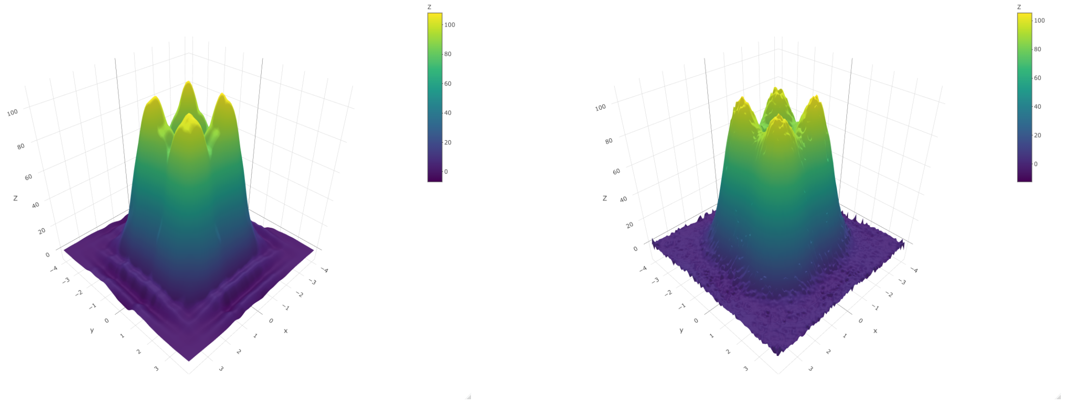
Another benefit of the GP, is that prediction uncertainties can be readily obtained. Assuming that $\boldsymbol{\mu} = 0$ and $\tau = 0$, we have that the standard error is

$$\mathrm{SE}\left(\hat{\eta}_{GP}(\boldsymbol{x})\right) = \sqrt{\frac{\hat{\boldsymbol{\eta}}^\top R^{-1} \hat{\boldsymbol{\eta}} \left(1 - r^\top(\boldsymbol{x}) R^{-1} r(\boldsymbol{x})\right)}{d-2}}. \tag{1.14}$$

Computation of this standard error requires multiplication of the matrix R^{-1} with the vector $r(\boldsymbol{x})$ and, since $r(\boldsymbol{x})$ is dependent on the input, the amortized cost is at least $\mathcal{O}(d^2)$, even if R^{-1} and $\hat{\boldsymbol{\eta}}^\top R^{-1} \hat{\boldsymbol{\eta}}$ are stored in memory.

An Illustration

Figure 1.3a shows the emulation surface corresponding to $d = 50$ input-output pairs for the Gramacy-Lee function defined in eq. (1.10). Visual inspection suggests that the GP emulator is a large improvement over the nearest neighbor emulators of Section 1.3.1.



(a) GP Emulator

(b) LA-GP Emulator

Figure 1.3: The Gramacy-Lee emulated surfaces for the GP and LA-GP emulators.

1.3.3 Local Approximate GP Emulation

The GP emulator is typically viewed as the gold standard of computer model emulation. A GP is rarely beaten in out of sample predictions, is capable of trivially satisfying the interpolation property and is capable of making predictions with $\mathcal{O}(d)$ amortized complexity. Once the parameters have been estimated and R has been inverted, predictions can be obtained in linear time (with respect to the size d of the training set $\mathcal{D}$) but obtaining R^{-1} can itself be cost prohibitive when d is large. In essence, this training process requires a minimum of $\mathcal{O}(d^3)$ time, which can limit the use of standard Gaussian processes for large training sets, which are typically required for high-dimensional problems.

The local approximate Gaussian process (LA-GP) is, at first glance, a relatively simple idea dating back to at least 1991 [41]. In its most basic form, known as local kriging, the size of the training set is reduced to $c \ll d$ using the c nearest neighbors of a prediction location $\boldsymbol{x}_{\text{new}}$. This is a reasonable approach, because data points far from $\boldsymbol{x}_{\text{new}}$ typically have very little impact on the predictive distribution of $\hat{\eta}(\cdot)$ at $\boldsymbol{x}_{\text{new}}$. It has been demonstrated, however, that this local nearest neighbor approach leads to a suboptimal local design [142,154]. With this in mind, [65,66] have developed criteria for greedy selection of the $\boldsymbol{x}_{\text{new}}$ neighborhoods which yield more accurate predictions without increasing the asymptotic complexity of the procedure. Under this approach, the predictive equations given in eq. (1.13) and eq. (1.14) remain the same, where the training set is $\mathcal{C} = (\boldsymbol{X}_c, \boldsymbol{\eta}_c)$ of dimension c, rather than the full training set $\mathcal{D}$.

Although predictions in the LA-GP framework can technically be obtained in $\mathcal{O}(c)$ time, prediction at each new location also requires $\mathcal{O}(c^3)$ time in order to build the neighborhood and train the local GP. In summary, to make T predictions, an ordinary GP has cost $\mathcal{O}(d^3{+}Td)$ while the LA-GP requires $\mathcal{O}(Tc^3)$ time. Moreover, if the set of prediction locations $\boldsymbol{X}_{\text{new}} = (\boldsymbol{x}_{\text{new},1}, \boldsymbol{x}_{\text{new},2}, \cdots \boldsymbol{x}_{\text{new},T})$ is known in advance, then the prediction process is "pleasantly parallel", and the cost of T can be reduced or eliminated.

An Illustration

Figure 1.3b shows the emulation surface for the Gramacy-Lee example using the LA-GP with $c = 60$. Visually, the LA-GP surface is a massive improvement over the NN approaches but is slightly less smooth than the standard GP approach. Table 1.1 shows the root mean squared error (RMSE), defined in eq. (2.5), for all of the approaches discussed in this section. As expected, the KNN-smooth emulator is the best of the nearest neighbor-based methods. The Gaussian process demonstrates much better prediction accuracy, with the LA-GP emulator close behind.

The SLAP-GP(0.99) and LEAP-GP(400) emulators, described in Chapter 2, are also included for comparison. The predictive performance of these emulators is comparable to the LA-GP emulator, although they require significantly less time to produce predictions.

Table 1.1: Comparison of RMSE for 7 different emulators of the Gramacy-Lee surface.

NN	KNN	KNN-Smooth	GP	LA-GP	SLAP-GP(0.99)	LEAP-GP(400)
5.22	4.82	3.79	1.61	3.59	3.67	3.49

1.3.4 Alternative Approaches to Emulation

Although the focus of this thesis is on the GP emulator, and it's relatives LA-GP, SLAP-GP and LEAP-GP, we note that a wide number of alternatives exist for emulation. In this subsection, we briefly outline a few of the more common alternatives.

Sparse Gaussian Processes

The local approximate Gaussian processes of [66] are just one way to address the computational challenges of GP emulation. Alternatively, one can choose to address these issues using a family of approaches known as sparse Gaussian processes [69, 85, 139]. For instance, we can parameterize the covariance of a GP regression model using the location of $c \ll d$ pseudo-inputs. An early effort by [139] gives a gradient-descent based implementation, which has since been improved [69]. These methods are similar in flavor to the local approximate GP strategies with two primary differences. The reduced training set $\mathcal{C}$ is global rather than local and the training set $\mathcal{C}$ can consist of input-output pairs $(\boldsymbol{x}, \eta(\boldsymbol{x}))$ which are not a part of the original training set $\mathcal{D}$ (called pseudo-inputs). An obvious benefit of this class, is that it must be trained a single time and thus leads to fast predictions like the standard Gaussian process. Although training a sparse-GP is more feasible than training a standard GP, it is nonetheless a challenging and time-consuming problem, especially when compared to the fast neighborhood construction algorithm of the LA-GP framework. If parallelization during prediction is possible, we feel that the LA-GP framework should be the preferred option.

Multivariate Adaptive Regression Splines

Multivariate adaptive regression splines (MARS) is another popular emulator which achieves fast predictions and can be trained quickly. It is often a good choice when GP emulators are impractical for computational reasons. The original proposal is due to [54], with a Bayesian extension presented in [42]. The model can be written as

$$\eta_{\text{MARS}}(\boldsymbol{x}) = \beta_0 + \sum_{k=1}^{K} \beta_k \prod_{i=1}^{p} (x_i - t_{k,i})_+^{o_k} \tag{1.15}$$

Functions of the form $(\pm(x - t))_+ = \max(0, \pm(x - t))$ are called *hinge functions* and t is called a *knot*. The MARS training algorithm selects knot locations and appropriate hinge function combinations automatically using a forward-backward selection process. The primary advantage of MARS (and Bayesian MARS) is computational, requiring only $\mathcal{O}(K^3)$ time to form the emulator, where K is generally much smaller than d and grows sub-linearly with d. Additionally, MARS does not require or assume that the response surface is smooth and is therefore better suited to handle this case than a GP based emulator. On the other hand, BMARS is not an interpolator, lacks a closed form for the prediction variance and is generally less accurate than some other alternatives [144]. A notable modern extension of BMARS is emulation with *Bayesian adaptive smoothing splines* (BASS), which allows for the use of large amounts of data and facilitate more efficient MCMC sampling [51]. An implementation of this approach can be found in the `BASS` package in R [49, 50].

Polynomial Chaos Expansions

Whereas the Gaussian process is the predominant emulator in the statistics community, polynomial chaos expansion (PCE) is the principal surrogate model in applied mathematics (at least in the not so distant past) [110]. The basic idea is to write

$$\eta_{\text{PCE}}(\boldsymbol{x}) = \sum_{j=0}^{L} w_j \psi_j(\xi) \tag{1.16}$$

where ξ is a random realization of the random variable Ξ, called the *germ*. The germ distribution is a modeling choice, and the ψ_j terms are orthogonal polynomials of order j with $\psi_0 = 1$ which must be constructed with respect to this choice. The w_j terms are weights (called mode strengths) which depend on the polynomials and the germ distribution. In the discussion [110], the emulation properties of PCE are critiqued on a variety of fronts. In particular, the truncation parameter L must be large to yield an adequate approximation of $\eta(\cdot)$, yet large values of L can lead to instability of the predictions. The PCE can also be heavily reliant on certain hard to justify modeling assumptions. While the PCE has some notable advantages in related areas, such as the forward propagation of uncertainty, we argue that GP based approaches are more suitable for most model calibration problems. Finally, we note that a recent study comparing the performance of the models and found that both models provided excellent predictions of the outputs, although the GP model was more accurate and more capable of handling various stochastic challenges [88].

Deep Neural Networks

Some authors (i.e. [74, 99, 141, 147]) have had success in building surrogate models using deep convolutional neural networks [91, 126]. Deep neural networks (DNN) are often used in climate, planetary and financial applications, where the size of the input space is often tremendous. The *curse of dimensionality* states that the volume of an input space $\mathcal{X}$ grows exponentially as the input dimension p grows linearly [43]. In these settings, an astronomical number of training examples $(\boldsymbol{x}_j, \eta(\boldsymbol{x}_j))$ must be obtained for $\mathcal{D}$ to be "space-filling". This big data paradigm is exactly where DNN methods thrive. While predictions with a DNN can be very impressive, there are a number of drawbacks including (i) considerable computational resources are required to be feasible, (ii) non-interpolating predictions, (iii) does not produce reliable prediction variances and (iv) can readily overfit. For most applications, with a small to moderate number of inputs, we suggest that DNN has many drawbacks compared with other reasonable choices and may not be worth the trouble. When the input dimension is very large, and high-performance resources are available, DNN emulation may be a suitable option.

1.4 Model Calibration

Consider a deterministic and unknown *true process* $\zeta(\boldsymbol{x})$ which governs some physical phenomenon. The inputs $\boldsymbol{x}$ are called *design variables* which are observable and usually controllable, to some extent, by the experimenter. These inputs can include things like time, the settings of a machine and the thickness of a material sample. We will denote the space of these parameters by $\mathcal{X}$ which is a subset of the Euclidean space with appropriate dimension.

Based on our current understanding of the physics which governs the physical system, a computer model $\eta(\boldsymbol{x}, \boldsymbol{\theta})$ is specified with the hope that it can serve as a suitable proxy for the unknown $\zeta(\boldsymbol{x})$. This computer model is often replaced with an emulator $\hat{\eta}(\cdot, \cdot)$, as described in Section 1.3, but we will largely ignore the distinction here. The inputs $\boldsymbol{\theta}$ are called *calibration parameters*, which may be tuning parameters designed to add flexibility to the model or they may describe properties of the physical system which are inherent, unknown and uncontrollable. As an example of the latter, consider the dynamic material properties example described in Section 1.2, where the material properties bulk modulus B_0 and the corresponding first pressure derivative B_0' of tantalum are calibration parameters with physical interpretations and unknown true values. Many authors, such as [21, 77], have discussed this distinction although it rarely leads to a difference in how the calibration parameters are treated. In the present context, it will be useful to make this distinction explicit, partitioning the calibration parameters as $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\gamma})$. In this case, the parameters $\boldsymbol{\alpha}$ are called *physical parameters*, indicating that they describe some inherent truth about the physical process, typically having a physical interpretation and some unknown true value. The remaining calibration parameters, denoted $\boldsymbol{\gamma}$, are referred to as *nuisance parameters*, suggesting that they are not of scientific interest. Occasionally, it will be useful partition the physical parameters further, writing $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ to distinguish between the physical parameters which are of scientific interest ($\boldsymbol{\alpha}$), and those which are not ($\boldsymbol{\beta}$). The input space for the calibration parameters will be written as $\Theta = A \times \Gamma$, where $\boldsymbol{\alpha} \in A \subset \mathbb{R}^p$ and $\boldsymbol{\gamma} \in \Gamma \subset \mathbb{R}^q$. The notation $\boldsymbol{\theta}_\star = (\boldsymbol{\alpha}_\star, \boldsymbol{\gamma}_\star)$ is used to denote the "true values" of the calibration parameters, although this notion must be carefully defined for the application, as there is no consensus to what the "true value" means in the context of model calibration.

A computer model $\eta(\boldsymbol{x}, \boldsymbol{\theta})$ is said to be *perfect* if there exists a $\boldsymbol{\theta} \in \Theta$ such that $\eta(\boldsymbol{x}, \boldsymbol{\theta}) = \zeta(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$. The calibration parameters $\boldsymbol{\theta}$ are said to be *distinguishable* with respect to the simulator if $\boldsymbol{\theta} \neq \boldsymbol{\theta}'$ implies that $\eta(\boldsymbol{x}, \boldsymbol{\theta}) \neq \eta(\boldsymbol{x}, \boldsymbol{\theta}')$ for at least one $\boldsymbol{x} \in \mathcal{X}$. For example, in the borehole function (eq. (2.6)), the parameters H_u and H_l only appear as a difference $H_u - H_l$ and are therefore not distinguishable. By defining $\Delta H = H_u - H_l$, as in eq. (3.11), the calibration parameters can be made distinguishable with respect to the computer model. Unless stated otherwise, we will typically assume that the computer model is imperfect and the calibration parameters are distinguishable.

The specific goal or purpose of model calibration will depend on the application. The primary goal of a researcher will typically fall into one of two categories: prediction or estimation. In the former, the quantity of interest is $\zeta(\boldsymbol{x}_{\text{new}})$ for a particular input $\boldsymbol{x}_{\text{new}}$, a set of inputs or even all inputs in $\mathcal{X}$. In the latter, we are interested in the so-called inverse problem, where the quantity of interest is some generic function $g(\boldsymbol{\theta}_\star)$. In general, the inverse problem is more challenging than the forward prediction problem for reasons discussed in Section 1.4.4, Chapter 3 and Chapter 4. Although the specific goal of model calibration may drastically vary, we will informally define the purpose as that of finding a set of calibration parameters $\boldsymbol{\theta}$ such that

$$\eta(\boldsymbol{x}, \boldsymbol{\theta}) \approx \zeta(\boldsymbol{x}).$$

To accomplish this goal, we will typically have a set of n observations from the true process, possibly recorded with error. The observations $\boldsymbol{y} = (y_1, y_2, \cdots y_n)^\top$

correspond to design variables $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \boldsymbol{x}_n)$. Collectively, the data $(\boldsymbol{X}, \boldsymbol{y})$ are called the *field data* or the *experimental data* and it is assumed that

$$y_i = \zeta(\boldsymbol{x}_i) + \epsilon_i,$$

where ϵ_i is an error term which incorporates uncertainty in the measurement of the observed data. More generally, we can write

$$\boldsymbol{y} = \boldsymbol{\zeta} + \boldsymbol{e}$$

where $\boldsymbol{\zeta} = (\zeta(\boldsymbol{x}_1), \cdots \zeta(\boldsymbol{x}_n))^\top$ and $\boldsymbol{e} = (\epsilon_1, \epsilon_2, \cdots \epsilon_n)^\top$ is an error vector with arbitrary structure.

1.4.1 Least Squares Calibration

Perhaps the simplest method of model calibration is to select the calibration parameter values as those that minimizes the sum of squares between the observed data and the computer model output. That is, the estimated calibration parameters are

$$\hat{\boldsymbol{\theta}}_{LS} = \underset{\boldsymbol{\theta} \in \Theta}{\arg\min} \sum_{i=1}^{n} \left(y_i - \eta(\boldsymbol{x}_i, \boldsymbol{\theta})\right)^2 . \tag{1.17}$$

If we assume that

$$\begin{aligned} & y_i = \eta(\boldsymbol{x}_i, \boldsymbol{\theta}) + \epsilon_i, \ i = 1, 2, \cdots n \\ & \epsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2) \end{aligned} \tag{1.18}$$

then $\hat{\boldsymbol{\theta}}_{LS}$ can also be obtained as the maximum likelihood estimator of θ [28]. Equation (1.18) implies the existence of a perfect computer model, and is an unrealistic modeling assumption in practice, often leading to poor predictions and underestimation of the uncertainty.

1.4.2 L_2 Calibration

In the work of Tuo and Wu [148, 149], the L_2 calibration procedure is proposed as an alternative the Bayesian model calibration framework of Kennedy and O'Hagan [87] (described in Section 1.4.3). Tuo and Wu begin by defining the true value of the calibration parameters to be

$$\begin{aligned} \boldsymbol{\theta}_{L_2} &= \underset{\boldsymbol{\theta}\in\Theta}{\arg\min} \left\|\zeta(\cdot) - \eta(\cdot,\boldsymbol{\theta})\right\|_{L_2}^2 \\ &= \underset{\boldsymbol{\theta}\in\Theta}{\arg\min} \int_{\mathcal{X}} \left(\zeta(\boldsymbol{x}) - \eta(\boldsymbol{x},\boldsymbol{\theta})\right)^2 d\boldsymbol{x} \end{aligned} \tag{1.19}$$

This is an important step because, as described in [117, 149], the predominant BMC framework of [87] leaves the calibration parameters undefined.

Since the true process $\zeta(\boldsymbol{x})$ is unknown, the first step is to construct a proxy $\hat{\zeta}(\boldsymbol{x})$ based on the data $(\boldsymbol{y}, \boldsymbol{X})$. In [149], it is assumed that $y_i = \zeta(\boldsymbol{x})$ and a kernel interpolator is used for $\hat{\zeta}(\cdot)$. We will focus on the L_2 calibration proposed in [148][1], which treats the data as stochastic, i.e. $y_i = \zeta(\boldsymbol{x}_i) + \epsilon_i$. In this more realistic scenario, $\hat{\zeta}(\cdot)$ is defined as the nonparametric regressor in the reproducing kernel Hilbert space [158, 160], generated by the kernel $\Psi(\cdot,\cdot)$

$$\hat{\zeta}(\cdot) = \underset{f\in\mathcal{N}_\Psi(\mathcal{X})}{\arg\min} \left\{ \frac{1}{n}\sum_{i=1}^{n} \left(y_i - f(\boldsymbol{x}_i)\right)^2 + \lambda \left\|f\right\|_{\mathcal{N}_\Psi(\mathcal{X})}^2 \right\}. \tag{1.20}$$

The kernel function $\Psi(\cdot,\cdot)$ can be viewed as a covariance function, such as (but not limited to) $\Psi(\cdot,\cdot) = \phi R(\cdot,\cdot)$, where R is given in equation eq. (1.12). The λ parameter is used to control the smoothness of the approximation and can be chosen using certain model selection criterion, such as generalized cross validation [158].

[1] Although [148] was published before [149], the latter is the logical prequel.

Along with eq. (1.20), the L_2 calibration procedure can now be defined as

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{L_2} &= \underset{\boldsymbol{\theta}\in\Theta}{\arg\min} \left\| \hat{\zeta}(\cdot) - \hat{\eta}(\cdot, \boldsymbol{\theta}) \right\|_{L_2}^2 \\ &= \underset{\boldsymbol{\theta}\in\Theta}{\arg\min} \int_{\mathcal{X}} \left(\hat{\zeta}(\boldsymbol{x}) - \hat{\eta}(\boldsymbol{x}, \boldsymbol{\theta}) \right)^2 d\boldsymbol{x}\end{aligned} \tag{1.21}$$

where $\hat{\eta}(\cdot, \cdot)$ is any emulator of $\eta(\cdot, \cdot)$ which is a sufficiently good approximation.

Tuo and Wu show that, under suitable conditions (see [148] for an extensive list), the estimator $\hat{\theta}_{L_2}$ is consistent and semiparametric efficient for the "true value" $\boldsymbol{\theta}_{L_2}$ defined in eq. (1.19). These authors also prove that the least squares calibration estimates $\hat{\boldsymbol{\theta}}_{LS}$ are consistent, but not semiparametric efficient, for $\boldsymbol{\theta}_{L_2}$. On the other hand, the Bayesian model calibration framework of [87] is shown to be neither consistent nor efficient for this value. This result is not surprising, because $\boldsymbol{\theta}_{L_2}$ does not coincide with the "true value" of $\boldsymbol{\theta}$ in the BMC framework.

1.4.3 Bayesian Model Calibration

We use the term Bayesian model calibration (BMC) to refer to the method of model calibration originally proposed by Kennedy and O'Hagan in their landmark 2001 paper [87]. In the model calibration literature, it is often called the KOH (or KO) model. Since 2001, the BMC framework has been widely used for a variety of applications in a diverse collection of scientific fields [3, 4, 11, 16, 21, 66, 76, 77, 95, 111, 132, 161]. Despite its wide use, it has been at times heavily critiqued [3, 4, 24, 148, 149] and numerous extensions, modifications and alternatives have been proposed [21, 24, 76, 94, 117, 148, 149].

Prior to 2001, model calibration primarily referred to the process of finding calibration parameter values such that the corresponding model output matched the data, in some sense, as closely as possible [14, 22]. The least squares and L_2 calibration methods described in the previous subsections provide examples of this idea, with varying levels of sophistication. The calibrated values are then treated as known, and used to predict the behavior of the process going forward. This so-called "plug-in" approach treats the calibration inputs as known, when in reality they are only estimated imperfectly. The BMC framework marked the first comprehensive treatment of uncertainty in model calibration, incorporating (i) parameter uncertainty, (ii) model inadequacy, (iii) residual variability, (iv) observation error and (v) code uncertainty.

In the BMC framework, we begin by linking the field data $\boldsymbol{y}$ to the computer model $\eta(\cdot, \cdot)$ by specifying

$$\begin{aligned} y_i &= \zeta(\boldsymbol{x}_i) + \epsilon_i \\ \zeta(\boldsymbol{x}_i) &= \eta(\boldsymbol{x}_i, \boldsymbol{\theta}) + \delta(\boldsymbol{x}). \end{aligned} \tag{1.22}$$

The ϵ_i terms represent observation error and are typically treated as independent and identically distributed Gaussian random variables, although certain applications may require a heteroskedastic model [21]. The $\delta(\cdot)$ term is called the model discrepancy function (the terms bias function or inadequacy function are sometimes used instead) and represents the difference between the true process and the computer model evaluated at the "true values" of $\boldsymbol{\theta}$. The discrepancy function, which explicitly acknowledges that the computer model is imperfect, is largely unique to the BMC framework, and was perhaps the primary contribution of [87]. Prior information about the unknown discrepancy function $\delta(\cdot)$ is represented in the form of a Gaussian process (see Section 1.3.2).

$$\begin{aligned} \epsilon_i &\overset{\text{iid}}{\sim} N(0, \sigma^2), \ i = 1, 2, \cdots n \\ \delta(\cdot) &\sim GP(m(\cdot), \Sigma(\cdot, \cdot)) \end{aligned} \tag{1.23}$$

where $m(\cdot)$ is a prior mean function and $\Sigma(\cdot, \cdot)$ is a prior covariance function. It is clear that the parameters governing $\delta(\cdot)$ and $\boldsymbol{\theta}$ are almost completely unidentifiable [162], in the sense that for every $\boldsymbol{\theta}_0 \in \Theta$ there exists a particular function $\delta_0(\cdot)$ such that $\zeta(\boldsymbol{x}) = \eta(\boldsymbol{x}_i, \boldsymbol{\theta}_0) + \delta_0(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$. For this reason, the prior distribution for $\delta(\cdot)$ must be carefully constrained. A common and intuitive approach is to set $m(\boldsymbol{x}) = 0$, effectively requiring that the computer model is unbiased on average across $\mathcal{X}$. Thus we specify

$$\begin{aligned} m(\boldsymbol{x}) &= 0 \\ \Sigma(\boldsymbol{x}, \boldsymbol{x}') &= \phi R(\boldsymbol{x}, \boldsymbol{x}' \mid \kappa), \end{aligned} \tag{1.24}$$

where $R(\cdot,\cdot|\cdot)$ is defined in eq. (1.12). The absence of a nugget leads implicitly to the assumption that the sum of the discrepancy function and computer model is an interpolator of the true process, with any observational error being attributed to the ϵ terms. The BMC framework presented here has three model parameters which we collectively refer to as $\boldsymbol{\psi} = (\sigma, \phi, \kappa)$. The final stage of the BMC model specification is to define prior distributions for the calibration and model parameters

$$\boldsymbol{\theta} \sim \pi_1(\boldsymbol{\theta}) \qquad \boldsymbol{\psi} \sim \pi_2(\boldsymbol{\psi}) \tag{1.25}$$

Collectively, eq. (1.22), eq. (1.23) and eq. (1.25) describe the BMC framework. Equation (1.24) is an important addendum for practical purposes and will be used throughout this thesis. Note that equations eq. (1.22), eq. (1.23) and eq. (1.24) imply that the log-likelihood can be written as

$$\begin{aligned} L(\boldsymbol{\theta}, \boldsymbol{\psi}|\boldsymbol{y}) =& (2\pi)^{-n/2} \left|\phi\boldsymbol{R} + \sigma^2\boldsymbol{I}_n\right|^{-1/2} \times \\ & \exp\left\{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\eta})^\top \left(\phi\boldsymbol{R} + \sigma^2\boldsymbol{I}_n\right)^{-1}(\boldsymbol{y}-\boldsymbol{\eta})\right\} \end{aligned} \tag{1.26}$$

and the resulting posterior distribution is

$$\pi(\boldsymbol{\theta}, \boldsymbol{\psi}|\boldsymbol{y}) \propto L(\boldsymbol{\theta}, \boldsymbol{\psi}|\boldsymbol{y})\pi_1(\boldsymbol{\theta})\pi_2(\boldsymbol{\psi}). \tag{1.27}$$

Posterior samples can be obtained from eq. (1.27) using a wide variety of techniques, including Markov Chain Monte Carlo (MCMC) methods or variational inference [19, 31, 63, 72].

Some Practical Considerations

Bayesian model calibration involves a large number of challenges, both computational and theoretical in nature. Fully Bayesian attempts at model calibration have been made (i.e. [161]) in certain application spaces, with varying amounts of success, but many authors opt to use a combination of empirical and stagewise techniques to reduce the burden [16, 21, 87, 94, 111].

For instance, the lack of joint identifiability between the calibration parameters and the discrepancy function lead to the constraining assumptions in eq. (1.24). In practice, these constraints may still fall short. In particular, we are typically unable to jointly infer both $\boldsymbol{\theta}$ and the correlation parameter κ. It is common practice, possibly due to the convincing arguments found in [94], to fix the correlation parameter κ to a reasonable value prior to running MCMC over the rest of the model. This is often done empirically by first using a simpler calibration method (i.e. least squares calibration). The resulting residuals, once smoothed to remove the effect of observation error, can be viewed as an empirical discrepancy function and κ can be estimated. In some cases, a subject matter expert is often able to give a reasonable estimate of the measurement error σ^2, and thus a plug-in approach or an informative prior will be used for this parameter as well [3, 21, 111, 132].

It is also common or necessary in many applications to use an emulator $\hat{\eta}(\cdot, \cdot)$ in place of the full computer model $\eta(\cdot, \cdot)$. It is possible to account for the emulation error in the model (see again [161]), especially if the emulator is another Gaussian process. In many cases, the emulation error will be sufficiently small and can be implicitly absorbed by the model discrepancy term while having little to no effect on the posterior distribution of eq. (1.27) [94]. Thus, as long as significant resources are invested in the construction of a good emulator, we can safely replace $\boldsymbol{\eta}$ in eq. (1.26) with $\hat{\boldsymbol{\eta}}$.

1.4.4 Comparison of Calibration Procedures

Suppose that forward prediction is the goal and that $\zeta(\boldsymbol{x}_{\text{new}})$ is the quantity of interest, and consider the estimators

$$\hat{\zeta}_{LS} = \hat{\eta}(\boldsymbol{x}_{\text{new}}, \hat{\boldsymbol{\theta}}_{LS}), \qquad \hat{\zeta}_{L_2} = \hat{\eta}(\boldsymbol{x}_{\text{new}}, \hat{\boldsymbol{\theta}}_{L_2}), \qquad \hat{\zeta}_{BMC\dagger} = \hat{\eta}(\boldsymbol{x}_{\text{new}}, \hat{\boldsymbol{\theta}}_{BMC}), \tag{1.28}$$

where $\hat{\boldsymbol{\theta}}_{BMC}$ is some reasonable frequentist point estimator derived from the BMC posterior distribution of $\boldsymbol{\theta}$. In this setting, the estimator $\hat{\zeta}_{L_2}$ is consistent, efficient and generally the superior estimator of this form. The estimate $\hat{\zeta}_{BMC\dagger}$ is unstable, unreliable and usually inconsistent for $\zeta(\boldsymbol{x}_{\text{new}})$. There are two major caveats to this critique of BMC, however. The first point is that BMC is a Bayesian procedure, and attacking its frequentist properties may be misleading. The second and more important caveat, is that comparison of the estimators in eq. (1.28) is akin to comparing apples and Porsches. In the BMC framework, the estimator $\hat{\zeta}_{BMC\dagger}$ was never intended to be a predictor of $\zeta(\boldsymbol{x}_{\text{new}})$. For a fair comparison, the L_2 calibration estimator defined above should be compared to the BMC estimator

$$\hat{\zeta}_{BMC} = \hat{\eta}(\boldsymbol{x}_{\text{new}}, \hat{\boldsymbol{\theta}}_{BMC}) + \hat{\delta}(\boldsymbol{x}_{\text{new}}). \tag{1.29}$$

In this case, a third paper by Tuo and Wu [150] proves the consistency of this estimator and shows that the BMC framework demonstrates more robust behavior than L_2 calibration in making these forward predictions. Moreover, the BMC framework has a variety of other benefits for forward prediction, such as the ability to incorporate expert knowledge and to comprehensively account for relevant uncertainties.

In the case of the inverse problem, where $g(\boldsymbol{\theta}_\star)$ is the quantity of interest, special care must be taken in defining the "true value" $\boldsymbol{\theta}_\star$ so that the problem maintains any meaning. If the true value of the calibration parameters coincides with the value defined by Tuo and Wu (i.e. $\boldsymbol{\theta}_\star = \boldsymbol{\theta}_{L_2}$), then L_2 calibration is an effective solution. The BMC framework, on the other hand, is known to struggle in this arena, where the lack of identifiability between $\boldsymbol{\theta}$ and $\delta(\cdot)$ renders it difficult or impossible to recover the true values [3]. In the original framework, [24] show that by adding specific information regarding the form of $\delta(\cdot)$, this problem can be resolved. In the important work [117], the BMC framework is extended using prior distributions for $\delta(\cdot)$ which are orthogonal to the gradient of the computer model. This has the effect of restoring consistency of the estimator $\hat{\boldsymbol{\theta}}_{BMC}$ for the parameter $\boldsymbol{\theta}_{\mathcal{L}}$ (where $\mathcal{L} = L_2$ or some other loss function belonging to a broadly defined class), while maintaining the other benefits of Bayesian model calibration, including the superiority and robustness of forward prediction. We reiterate that, if forward prediction is the goal or if the calibration parameters have physical interpretation, then the Bayesian calibration procedure described in [117] is seldom worth the effort.

In the case of the inverse problem where calibration parameters have physical interpretations, $\boldsymbol{\theta}_\star$ is an inherent property of the physical process and may not coincide with $\boldsymbol{\theta}_{L_2}$. In this case, no calibration procedure can guarantee reliable inference for $g(\boldsymbol{\theta}_\star)$. This issue is the primary topic of this thesis and is discussed in great detail in Chapter 3 and especially Chapter 4 .

1.5 Related Background

1.5.1 Latin Hypercube Designs

Latin hypercube sampling is a very useful tool when dealing with black-box computer model [98, 113, 135]. A *black-box* model is a function $\eta(\boldsymbol{x}) \ : \ \mathcal{X} \to \mathbb{R}$ which has no explicit description but can be evaluated to obtain input-output pairs $(\boldsymbol{x}, \eta(\boldsymbol{x}))$. In our discussion of emulators (Section 1.3), we are given a set of input-output pairs $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{\eta})$ and asked to find a suitable representation of the otherwise unknown function. In practice, the computer model $\eta(\cdot)$ is often time-consuming to run, and therefore the number of evaluations, denoted d, may be small. It is paramount then to use our limited allowance of computer model evaluations as effectively as possible.

A good set of evaluation points, called the *design*, should satisfy at least the following two properties. First, a good design should be *space-filling* in the sense that the information obtained is spread out over the entire space. For instance, consider the one-dimensional input space $\mathcal{X} = [0, 1]$. In the absence of any other information, certainly the best choice of d design points is to take $x_1, x_2, \cdots x_d$ to be evenly spaced between 0 and 1. It would be a waste of resources to load up on points closer to 1 or 0. The second property is that a good design should be *non-collapsing.* Since computer models are deterministic, it would be a waste of precious resources to replicate design points. If one of the design variables has little or no effect on the output of $\eta(\boldsymbol{x})$, then two design points which differ only in this parameter are said to *collapse*. That is to say the output of $\eta(\cdot)$ evaluated at these two design points will give (almost) the same value and can therefore be viewed as a repeated evaluation. Therefore a good design maintains the property that no two design points should ever share a coordinate value. A Latin hypercube sample (LHS) is a non-collapsing design which generally has, or can be made to have, good space-filling properties.

An LHS of d points in p dimensions can be defined as a $d \times p$ matrix $\boldsymbol{X}$ where each column is a permutation of the set $\{0, \frac{1}{d-1}, \frac{2}{d-1}, \cdots 1\}$. The rows $\boldsymbol{x}_j = (x_{j1}, \cdots x_{jp})$, $j = 1, 2, \cdots d$ define the d design points. It is an immediate consequence of this definition that any LHS design is non-collapsing. Not all LHS designs will be space-filling, however, but a randomly selected design can be evaluated according to a separation criterion. The maximin criteria, in which the minimal distance between any two design points is maximized, generally yields a good space-filling design [135]. The `lhs` package in R offers a variety of routines which construct LHS designs with respect to a number of different criteria [26].

The LHS, as defined above, is only useful if the input space happens to be $\mathcal{X} =$

$[0,1]^p$. More generally, we can assume that the input space is a rectangular subspace of $\mathbb{R}^p$, i.e.

$$\mathcal{X}_R = [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_p, b_p],$$

and the notion can be naturally extended. Let $\boldsymbol{X}$ be an LHS design and let $\boldsymbol{X}'$ be the design with rows

$$\boldsymbol{x}_j' = (\boldsymbol{b} - \boldsymbol{a}) \odot \boldsymbol{x}_j + \boldsymbol{a},$$

where $\boldsymbol{a} = (a_1, a_2, \cdots a_p)$, $\boldsymbol{b} = (b_1, b_2, \cdots b_p)$ and $\odot$ denotes the Hadamard product. Then $\boldsymbol{X}'$ is said to be an LHS design over the space $\mathcal{X}_R$.

If other cases, the input space may not be compact or some regions of the input space may be viewed as more important than others. To extend the concept of an LHS design in this setting, a (prior) distribution for each design parameter must be specified in terms of the CDF $F_k(x_k)$, $k = 1, 2, \cdots p$ [112]. The matrix $\boldsymbol{X}'$ with $(jk)^{th}$ element

$$\boldsymbol{X}_{j,k}' = F_k^{-1}\left(\boldsymbol{X}_{j,k}\right)$$

is said to be an LHS sample with respect to the distributions $F_1, \cdots F_p$. Note that the case of rectangular regions can be seen as a special case of this setting, where the distributions are uniform over the interval $[a_k, b_k]$. Figure 1.4 illustrates this procedure using a maximin LHS design with $d = 13$ two-dimensional design points (left). In the middle panel, this same design is transformed over the space $[-1, 2] \times [-2, 0]$. On the right, the LHS design is transformed over the space $\mathbb{R}^2$ with respect to the distribution $\boldsymbol{x} \sim N(\boldsymbol{0}, I_2)$, so that $F_1(x) = F_2(x) = \Phi(x)$. For the extension of LHS design to $\mathbb{R}^p$ where the prior distribution has dependent components, we refer the reader to [165].

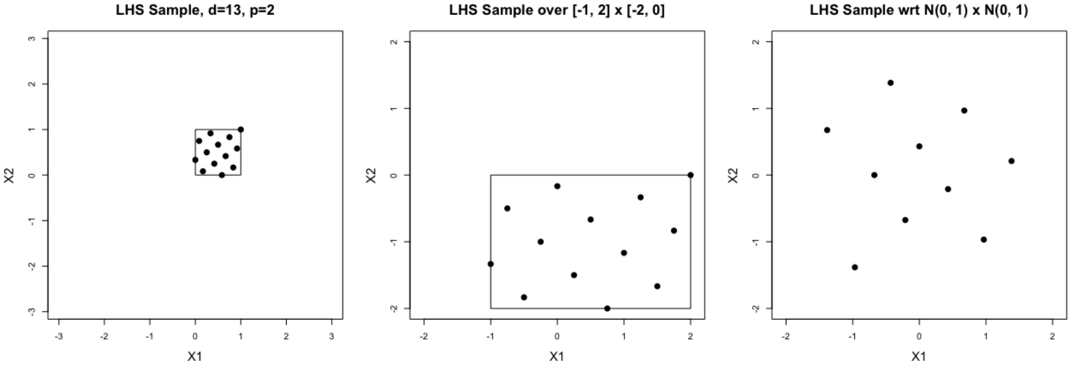


Figure 1.4: In the left panel, a maximin LHS design with $d = 13$ and $p = 2$ is shown. In the middle panel, this same LHS design is taken over the space $[-1, 2] \times [-2, 0]$. In the right panel, the same LHS design is taken over the space $\mathbb{R}^2$ with respect to the standard bivariate normal distribution.

1.5.2 Markov Chain Monte Carlo

In the BMC framework of Section 1.4.3, the joint posterior distribution of the calibration parameters $\boldsymbol{\theta}$ and the model parameters $\boldsymbol{\psi}$ is defined in eq. (1.27). This posterior distribution is not analytically tractable, and thus numerical tools will be required to approximate the posterior. In this thesis, we primarily focus on the class of *Markov Chain Monte Carlo* (MCMC) methods [58,63]. For a historical review of MCMC methods, see [127].

MCMC is a general method for drawing samples of θ from a posterior distribution $\pi(\theta|\mathcal{D})$, in which the samples $\theta^1, \theta^2, \cdots \theta^M$ form a Markov chain (a Markov chain is a sequence of random variables for which the distribution of θ^t depends only on the most recent value θ^{t-1}). Rather than obtaining independent samples from $\pi(\theta|\mathcal{D})$, MCMC methods produce a sequence of correlated samples which converge to the target distribution as $M \to \infty$. In practice, the first M_0 iterations are referred to as the *warm-up* phase (or burn-in) and these samples are discarded. The process of *thinning* refers to the discarding of further samples, keeping every k^{th} sample for some positive integer k. If M is sufficiently large, then the remaining $(M - M_0)/k$ values can be viewed as approximate samples from the posterior distribution.

A large number of MCMC methods have been developed and used by practitioners with notable examples including *Gibbs sampling* [56,60], *slice sampling* [108], Metropolis algorithms [73, 100, 115] and Hamiltonian based approaches [80]. Although different applications may benefit from different MCMC approaches, the results in this thesis make use of (almost exclusively) the *Metropolis-Hastings algorithm* and, when necessary, the adaptive extension of [72]. The Metropolis Hastings (MH) algorithm can be stated as follows.

1. Initialize θ^0 to be a starting point for which $\pi(\theta^0|\mathcal{D}) > 0$. In theory, any choice of θ^0 with positive posterior support will suffice, but choosing a reasonable starting value is crucial to the practical success of MH. See [58] for a detailed discussion.

2. For $m = 1, 2, \cdots M$

    2a. Sample a *candidate value* θ^* from a *proposal distribution* at time m, $P_m(\theta^*|\theta^{m-1})$.

    2b. Calculate the ratio

    $$\alpha = \frac{\pi(\theta^*|\mathcal{D})\ P_m(\theta^{m-1}|\theta^*)}{\pi(\theta^{m-1}|\mathcal{D})\ P_m(\theta^*|\theta^{m-1})} \tag{1.30}$$

    2c. Set

    $$\theta^m = \begin{cases} \theta^*, & \text{with probability } \min(\alpha, 1) \\ \theta^{m-1}, & \text{otherwise} \end{cases}$$

In many applications, the proposal distribution $P_m(\cdot|\cdot)$ is symmetric, satisfying the condition $P_m(a|b) = P_m(b|a)$ for all a, b and m. In this case, the ratio defined in eq. (1.30) simplifies to

$$\alpha = \frac{\pi(\theta^*|\mathcal{D})}{\pi(\theta^{m-1}|\mathcal{D})},$$

and the resulting algorithm is known simply as the *Metropolis Algorithm.* A good proposal distribution should be (i) easy/fast to sample from and (ii) facilitate easy/fast evaluation of eq. (1.30). The proposal distribution defines a random walk through the parameter space Θ, and the size of each step taken is also crucial for success. If the steps are too small, then the random walk will move too slowly and many iterations will be required. If the steps are too large, then the candidate values will rarely be accepted and the random walk spends much of its time standing still. For this reason, proposal distributions are often equipped with a tuning parameter λ which must be carefully selected in order to balance these extremes. For example, suppose that $\theta \in \mathbb{R}^p$ and consider the multivariate normal proposal distribution defined by

$$\theta^*|\theta^{m-1} \sim N\left(\theta^{m-1}, \lambda I\right).$$

This symmetric proposal distribution is a common choice for its simplicity, leading to efficient candidate generation and evaluation of the acceptance ratio. It also requires just a single parameter which must tuned to obtain reasonable acceptance rates. In more challenging problems however, additional flexibility will be needed, and it is common to replace λI with a more general covariance matrix C. The appropriate structure for C is difficult to ascertain and naive attempts at tuning may be impractical. By allowing the covariance matrix to evolve over time, the structure can be effectively tuned during the warmup period of the Markov chain.

A popular implementation of this idea is now known as the *adaptive Metropolis* (AM) Algorithm and can be summarized as follows. The proposal distribution at time m is defined to be multivariate normal with mean θ^{m-1} and covariance C_m. Initialize θ^0 as before, and also choose an initial covariance matrix C_0 according to our best available prior knowledge. The covariance matrix at time m is set to

$$C_m = \begin{cases} C_0, & m \leq m_0 \\ \lambda \left(\mathrm{Cov}(\theta^0, \theta^1, \cdots \theta^{m-1}) + \tau I_p \right), & m > m_0 \end{cases} \tag{1.31}$$

where τ is a small positive value which helps control the condition number C_m (much like the nugget in Section 1.3.2), m_0 controls the length of the pre-adaptive stage and λ is a scaling parameter which can be set to $(2.4)^2/p$ [59] or manually tuned for additional flexibility. The $\mathrm{Cov}(\theta^0, \cdots \theta^{m-1})$ denotes the empirical covariance matrix and can be computed as

$$\mathrm{Cov}(\theta^0, \theta^1, \cdots \theta^k) = \frac{1}{k} \left(\sum_{i=0}^{k} \theta^i {\theta^i}^\top - (k+1)\bar{\theta}_k \bar{\theta}_k^\top \right),$$

where $\bar{\theta}_k = \frac{1}{k+1} \sum_{i=0}^{k} \theta^i$. Since computing the empirical covariance matrix can be expensive when m is large, an efficient implementation must make use of the following recursive equations for $m > m_0$

$$\begin{aligned} \bar{\theta}_m &= \frac{(m-1)\bar{\theta}_{m-1} + \theta^m}{m} \\ C_m &= \frac{m-2}{m-1} C_{m-1} + \frac{\lambda}{m-1} \left((m-1)\bar{\theta}_{m-2}\bar{\theta}_{m-2}^\top - m\bar{\theta}_{m-1}\bar{\theta}_{m-1}^\top + \theta^m {\theta^m}^\top + \tau I \right) \end{aligned} \tag{1.32}$$

The `MHadaptive` package in R provides a simple implementation of the AM algorithm [32]. This `MHadaptive` package was used for many of the simpler analyses described in this paper, but many of the more computationally intensive applications required writing MCMC implementations from scratch so that flexibility could be added as needed (see Section 1.5.3, for instance). In some high dimensional applications (i.e. climate modeling) these traditional approaches may be impractical and more creative alternatives such as Multiple Very Fast Simulated Annealing and Delayed Rejection Adaptive Metropolis algorithms may be more suitable [155].

1.5.3 Fast Matrix Algebra for BMC

In Section 1.4.3, the BMC framework is presented as a multivariate normal likelihood with covariance matrix

$$\boldsymbol{\Sigma} = \phi \boldsymbol{R}_\kappa + \sigma^2 \boldsymbol{I}_n.$$

To evaluate the likelihood in eq. (1.26), both the determinant and inverse of this matrix are needed, requiring $\mathcal{O}(n^3)$ complexity. If MCMC is used to sample from the posterior, then these model parameters can be viewed as a Markov chain $\boldsymbol{\psi}_m = (\phi_m, \kappa_m, \sigma_m^2)$, $m = 1, 2, \cdots M$, and the covariance matrix itself becomes stochastic $\boldsymbol{\Sigma}_m$. Since the inverse and determinant must be found M times, the cost of MCMC can be prohibitively large in practice. In this section we show that by setting κ to a fixed value, as recommended in Section 1.4.3, the time spent performing matrix algebra can be substantially reduced in practice. In particular, we require a cubic-time pre-computation phase which will allow us to calculate the determinant in linear time and the inverse with a speedup of approximately two.

We begin by finding the eigenvalue decomposition of $\boldsymbol{R}$, that is

$$\boldsymbol{R} = \boldsymbol{Q}\boldsymbol{D}\boldsymbol{Q}^\top \tag{1.33}$$

where $\boldsymbol{D} = \text{Diag}(r_1, r_2, \cdots r_n)$ is the diagonal matrix of eigenvalues r_i and $\boldsymbol{Q}$ is an orthogonal $n \times n$ matrix whose columns are the eigenvectors of $\boldsymbol{R}$. Note that the existence and form of the decomposition in eq. (1.33) are guaranteed since $\boldsymbol{R}$ is a correlation matrix [64]. Next, we define $\tau = \sigma^2$ and write

$$\begin{aligned} |\boldsymbol{\Sigma}_m| = |\boldsymbol{\phi}_m \boldsymbol{R} + \tau_m \boldsymbol{I}| = \tau_m^n \left| \frac{\phi_m}{\tau_m} \boldsymbol{R} + \boldsymbol{I} \right| &= \tau_m^n \prod_{i=1}^n \left(\frac{\phi_m}{\tau_m} r_i + 1 \right) \\ &= \prod_{i=1}^n \left(\phi_m r_i + \tau_m \right). \end{aligned} \tag{1.34}$$

Thus the determinant of each $\boldsymbol{\Sigma}_m$ can be computed in $\mathcal{O}(n)$ time with a memory cost of $\mathcal{O}(n)$.

Using the eigenvalue decomposition shown in eq. (1.33), we define $\tilde{\boldsymbol{R}}_m = \phi_m \tau_m^{-1} \boldsymbol{R}$ and $\tilde{D}_m = \phi_m \tau_m^{-1} \boldsymbol{D} = \mathrm{Diag}(\phi_m \tau_m^{-1} r_1, \phi_m \tau_m^{-1} r_2, \cdots, \phi_m \tau_m^{-1} r_n)$ which allows us to write the inverse as

$$\begin{aligned}
\boldsymbol{\Sigma}_m^{-1} &= (\boldsymbol{\phi}_m \boldsymbol{R} + \tau_m \boldsymbol{I})^{-1} \\
&= \frac{1}{\tau_m}\left(\tilde{\boldsymbol{R}}_m + \boldsymbol{I}\right)^{-1} \\
&= \frac{1}{\tau_m}\left(\boldsymbol{Q}\tilde{\boldsymbol{D}}_m \boldsymbol{Q}^\top + \boldsymbol{I}\right)^{-1} \\
&= \frac{1}{\tau_m}\left(\boldsymbol{Q}\left(\tilde{\boldsymbol{D}}_m + \boldsymbol{I}\right)\boldsymbol{Q}^\top\right)^{-1} \\
&= \frac{1}{\tau_m}\left(\boldsymbol{Q}\left(\tilde{\boldsymbol{D}}_m + \boldsymbol{I}\right)^{-1}\boldsymbol{Q}^\top\right).
\end{aligned} \tag{1.35}$$

Since $\left(\tilde{\boldsymbol{D}}_m + \boldsymbol{I}\right)$ is a diagonal matrix, the inverse

$$\left(\tilde{\boldsymbol{D}}_m + \boldsymbol{I}\right)^{-1} = \mathrm{Diag}\left(\left(\frac{\phi_m r_1}{\tau_m} + 1\right)^{-1}, \left(\frac{\phi_m r_2}{\tau_m} + 1\right)^{-1}, \cdots, \left(\frac{\phi_m r_n}{\tau_m} + 1\right)^{-1}\right) \tag{1.36}$$

can be computed in linear time. Next, we can decompose $\left(\tilde{\boldsymbol{D}}_m + \boldsymbol{I}\right)^{-1} = \boldsymbol{W}\boldsymbol{W}^\top$ which allows us to write

$$\begin{aligned}
\boldsymbol{\Sigma}_m^{-1} &= \boldsymbol{Q}\boldsymbol{W}\boldsymbol{W}^\top\boldsymbol{Q}^\top \\
&= (\boldsymbol{Q}\boldsymbol{W})(\boldsymbol{Q}\boldsymbol{W})^\top
\end{aligned} \tag{1.37}$$

Since W is diagonal, the product $\boldsymbol{Q}\boldsymbol{W}$ can be obtained in quadratic time. Taking advantage of fast algorithms for the multiplication of a matrix with its transpose, the entire inverse can be computed in $\mathcal{O}(n^2 + n^3/2)$ time with a memory cost of $\mathcal{O}(n^2)$ [64]. Although the cost of this approach is asymptotically the same as the naive implementation, we have found it to drastically reduce computation time in practice.

Although it is not possible to directly obtain the inverse of $\boldsymbol{\Sigma}_m$ in $\mathcal{O}(n^2)$ time, we can often find a suitable approximation to the inverse in near-quadratic time. Starting from the final equality of eq. (1.35), we note that the $(ij)^{th}$ element of $\boldsymbol{\Sigma}_m^{-1}$ is given by

$$(\boldsymbol{\Sigma}_m)_{ij}^{-1} = \frac{1}{\tau_m}\sum_{k=1}^{n} \boldsymbol{Q}_{ik} d_k \boldsymbol{Q}_{jk}, \tag{1.38}$$

where d_k is the $(kk)^{th}$ element of $\left(\tilde{\boldsymbol{D}}_m + \boldsymbol{I}\right)^{-1}$. Defining $q_{ijk} = \boldsymbol{Q}_{ik}\boldsymbol{Q}_{jk}$, this becomes

$$(\boldsymbol{\Sigma}_m)_{ij}^{-1} = \sum_{k=1}^{n} \frac{q_{ijk}}{r_k\phi_m + \tau_m}. \tag{1.39}$$

One simple approach for approximating the inverse is to exploit the fact that the eigenvalues typically become very small in magnitude. Formally, there usually exists $N_0 \ll n$ such that $r_i < \epsilon_{\text{tol}}$ for $i > N_0$ for reasonably small values of ϵ_{tol}. This implies that the $(ij)^{th}$ component of $\boldsymbol{\Sigma}_m^{-1}$ can be approximated as

$$\left(\boldsymbol{\Sigma}_m^{-1}\right)_{ij} \approx \sum_{k=1}^{N_0} \frac{q_{ijk}}{r_k\phi_m + \tau_m} + \frac{1}{\tau_m}\boldsymbol{Z}_{ij}, \tag{1.40}$$

where $\boldsymbol{Z}_{ij} = \sum_{k=N_0+1}^{n} q_{ijk}$ can be pre-computed for all $1 \leq i \leq j \leq n$. Thus, we readily obtain an algorithm which operates in $\mathcal{O}(n^2 N_0)$ time with an $\mathcal{O}(n^2)$ memory requirement. This computation can be optimized in practice by taking $\boldsymbol{Q}_0$ to be the first N_0 columns of $\boldsymbol{Q}$ and $\boldsymbol{W}_0$ to be the $N_0 \times N_0$ upper left block of $\boldsymbol{W}$ and computing

$$\boldsymbol{\Sigma}_m^{-1} \approx (\boldsymbol{Q}_0\boldsymbol{W}_0)(\boldsymbol{Q}_0\boldsymbol{W}_0)^\top + \frac{1}{\tau_m}\boldsymbol{Z}. \tag{1.41}$$

1.5.4 Bayesian Regularization

Consider the model $y(\boldsymbol{x}) = \eta(\boldsymbol{x}, \boldsymbol{\theta}) + \boldsymbol{e}$, such that $E(\boldsymbol{e}) = \boldsymbol{0}$. We assume that $\boldsymbol{x} = (x_1, x_2, \cdots x_p)$ is an observable covariate, $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots \theta_p)^\top$ is a set of unknown parameters and $\boldsymbol{\eta}(\cdot,\cdot)$ is an arbitrary mean function governed by the parameters $\boldsymbol{\theta}$. Given data $\boldsymbol{y} = (y_1, y_2, \cdots y_n)^\top$ and $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \boldsymbol{x}_n)^\top$ and a suitable loss function $\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{X})$ the parameters $\boldsymbol{\theta}$ can be estimated as

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{y}, \boldsymbol{X}). \tag{1.42}$$

Linear regression is a well-known special case where $\eta(\boldsymbol{x}) = \boldsymbol{x}\boldsymbol{\theta}$ and the ordinary least squares (OLS) estimators of $\boldsymbol{\theta}$ are given by

$$\hat{\boldsymbol{\theta}}_{OLS} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|_2^2 = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y}. \tag{1.43}$$

When $p \ll n$ the OLS estimators often perform very well, but when p is very large (compared to n) the estimators are often plagued by multicollinearity, leading to high variance and poor efficiency. Regularization is a general term for a class of methods used to address this problem. A regularized estimator of $\boldsymbol{\theta}$ is an estimator of the form

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \{\mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{y}, \boldsymbol{X}) + \text{pen}_\lambda(\boldsymbol{\theta})\} \tag{1.44}$$

where $\text{pen}_\lambda(\boldsymbol{\theta})$ is a *penalty function* used to enforce a constraint on the parameters $\boldsymbol{\theta}$. Perhaps the most well known form of regularization is *Ridge Regression* [79], in which the penalty function is

$$\text{pen}_\lambda^{\text{ridge}}(\boldsymbol{\theta}) = \lambda\|\boldsymbol{\theta}\|_2^2 = \lambda \sum_{i=1}^{p} \theta_i^2.$$

In the case of linear regression, the ridge estimator of $\boldsymbol{\theta}$ can be obtained in closed form as

$$\hat{\boldsymbol{\theta}}_{\text{ridge}} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda I_p)^{-1}\boldsymbol{X}^\top \boldsymbol{y}. \tag{1.45}$$

Ridge regression has the effect of pulling the OLS estimator back towards the origin, introducing bias in exchange for reduced variance. Although Ridge Regression works best when there are a large number of parameters, it has been proved in the case of linear regression that there always exists a $\lambda > 0$ such that the MSE of the Ridge estimators is strictly smaller than that of the OLS estimator [44]. The ridge penalty can be generalized as

$$\mathrm{pen}_{\lambda}^{\mathrm{bridge}}(\boldsymbol{\theta}) = \lambda \|\boldsymbol{\theta}\|_p^p = \lambda \sum_{i=1}^{p} |\theta_i|^p. \tag{1.46}$$

The resulting estimator is sometimes called the Bridge(b) estimator of $\boldsymbol{\theta}$ [55]. The Bridge regression framework contains Ridge regression ($b = 2$) and the LASSO [146] ($b = 1$) as special cases. Ridge regression and LASSO are both widely used, each method having advantages and disadvantages for certain scenarios. Another common regularization method is the so-called *elastic net* [166], in which the advantages of both Ridge and LASSO are harnessed by defining the penalty function

$$\mathrm{pen}_{\lambda}^{\mathrm{EN}}(\boldsymbol{\theta}) = \lambda_1 \sum_{i=1}^{p} |\theta_i| + \lambda_2 \sum_{i=1}^{p} \theta_i^2. \tag{1.47}$$

A large number of highly specialized penalty functions have been developed for a wide variety of applications. These regularization methods have a convenient Bayesian interpretation by letting $\exp\left(-\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{y})\right)$ denote the model likelihood and by specifying the prior distribution

$$\pi(\boldsymbol{\theta}) \propto \exp\left(-\mathrm{pen}_{\lambda}(\boldsymbol{\theta})\right). \tag{1.48}$$

Conditional on any hyperparameters $\boldsymbol{\lambda}$, the posterior distribution of $\boldsymbol{\theta}$ becomes

$$\begin{aligned} \pi(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{y}) &\propto \exp\left(-\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{y})\right) \exp\left(-\mathrm{pen}_{\lambda}(\boldsymbol{\theta})\right) \\ &= \exp\left\{-\left[\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{y}) + \mathrm{pen}_{\lambda}(\boldsymbol{\theta})\right]\right\}, \end{aligned} \tag{1.49}$$

and thus maximizing the posterior distribution is equivalent to minimizing the sum of the loss and penalty functions as in eq. (1.44) [18]. For many particular instances of the penalty function, this leads to a surprisingly straightforward Bayesian interpretation. For instance, the Bridge(b) penalty corresponds to the prior distribution

$$\pi(\boldsymbol{\theta}) \propto \exp\left\{-\lambda \sum_{i=1}^{p} |\theta_i|^p\right\},$$

which immediately implies that the Ridge regression penalty ($b = 2$) corresponds to independent normal priors for each $\theta_1, \theta_2, \cdots \theta_p$ with mean 0 and variance $(2\lambda)^{-1}$. Similarly, the LASSO penalty can be viewed as the product of independent Laplacian priors for each θ, with mean 0 and variance $2\lambda^{-1}$. Similar yet less recognizable interpretations exist for nearly every imaginable penalty function. In some cases, the Bayesian execution of these principles can be improved with careful attention to detail such as in the Bayesian LASSO and the Bayesian elastic net [93, 114].

Other Bayesian regularization priors, such as the *g-prior* [164] and the *spike and slab prior* [81] have received attention recently for variable selection problems, rivaling the LASSO and the elastic net. The spike and slab prior in particular is theoretically enticing and is provably optimal under certain conditions, but the combinatorial search space renders it computationally unusable for most practical problems [119]. Instead, many practitioners rely on computationally tractable prior distributions which attempt to approximate the behavior of the spike and slab prior. For instance, we consider the flexible class of priors known as *Global-Local Gaussian scale mixtures*, a class which contains many other well known regularization priors such as Ridge regression and the Bayesian LASSO [15, 17]. For $i = 1, \cdots p$, this class is defined as

$$\begin{aligned} &\theta_i \mid (\tau, \psi_i) \overset{\text{ind}}{\sim} N(0,\ \tau\psi_i) \\ &\tau \sim g \qquad \text{and} \qquad \psi_i \sim g_i. \end{aligned} \tag{1.50}$$

Different choices of the prior distributions g and g_i lead to different behavior, giving this class its flexibility. A popular new prior contained in this class is the *Horseshoe prior* [27] defined by the choice

$$\tau \sim g \qquad \text{and} \qquad \psi_i \sim C_+(0, \sigma_i) \tag{1.51}$$

where $C_+(0, \sigma)$ denotes the half-Cauchy prior, with scale parameter (also the median) σ_i. Assuming that the parameters are on a standardized scale, then each σ_i is usually taken to be 1, but these hyper-parameters can be adjusted for additional flexibility if needed. There is no consensus on the best choice of prior for the global parameter τ, although $C_+(0, a)$ seems to be a common default. Following [57], an early choice by [27] suggest setting $a = 1$, while a later work by [119] suggest letting a scale with the variance of the data. If we are willing to guess a priori at the number of non-zero nuisance parameters, [116] derive a formula for choosing a and demonstrate its potential superiority via simulations. Finally, [152] consider truncation of the half Cauchy prior to the interval $[1/p, 1]$ in order to avoid collapsing to the degenerate $\tau = 0$ case, which leads to total shrinkage of all parameters. The *Dirichlet Laplace prior* is another notable member of this class which has gained momentum since it was published in 2015 [17]. A comprehensive comparison of these approaches can be found in [15] and [120].

These ideas are important in Chapter 3 where we develop a highly specialized Bayesian regularization prior for nuisance parameters with a given structure. We rely heavily on the penalty interpretation of regularization in Bayesian settings.

Chapter 2

Efficient GP Emulation for MCMC Applications

"Time is that which we want most, but use worst." – William Penn

2.1 Overview

In Section 1.3.2, the Gaussian process emulator was presented as a gold standard for computer model emulation. Unfortunately, training a GP requires inverting a dense $d \times d$ matrix, rendering the GP emulator infeasible for many problems, where large training sets are required to precisely emulate a computer model. The local approximate Gaussian process (LA-GP) approach described in section Section 1.3.3 solves this problem by selecting a subset of $c \ll d$ examples from the training data $\mathcal{D}$ [65, 66]. These neighborhoods are selected using a greedy selection criterion based on the new input $\boldsymbol{x}_{\text{new}}$. The downside of this approach is that the selection of the neighborhood and LA-GP training requires $\mathcal{O}(c^3)$ time, but this step must be repeated each time a new prediction is desired, and thus the true cost of producing T predictions with an LA-GP emulator is $\mathcal{O}(Tc^3)$. The nature of the LA-GP allows for straightforward parallelization of the predictions, so long as the set of predictions

$\boldsymbol{X}_{\text{new}} = (\boldsymbol{x}_{\text{new},1}, \boldsymbol{x}_{\text{new},2}, \cdots \boldsymbol{x}_{\text{new},T})$ is known in advance.

In the context of Bayesian model calibration, the inputs required for prediction at each time step t are often part of a *Markov chain* and thus, by definition, the prediction set $\boldsymbol{X}_{\text{new}}$ cannot be known in advance. If the prediction set is not known in advance, then the predictions cannot be computed in parallel and, depending on the magnitude of T, emulation with the LA-GP can be considerably slower at making predictions than a standard GP (provided the standard GP can be fit at all).

In this chapter, we present two modifications to the LA-GP which are demonstrably more efficient than either the standard or local approximate Gaussian process, at the cost of additional memory requirements and an often negligible loss of accuracy. These methodologies are referred to as the **S**equence of **L**ocal **Ap**proximate Gaussian processes (SLAP-GP) and the **L**ocalized **E**nsemble of Local **Ap**proximate Gaussian processes (LEAP-GP). Details for these methods will be discussed in the remainder of this chapter, but Table 2.1 gives a look at the asymptotic complexity required for training and prediction along with the asymptotic memory complexity for each emulator. The prediction and memory requirements are broken into the cases with and without uncertainty. Although the case with uncertainty is important for forward uncertainty propagation [89] and in fully Bayesian calibration [161], we are typically interested in obtaining a cheap-to-evaluate surrogate for the computer model (for reasons explained in Section 1.4.3) and will focus on the case without uncertainty.

2.2 The Prediction Hub

The primary strategy of both the SLAP-GP and LEAP-GP emulators is to reuse our previous work whenever possible, trading time for memory. For example, suppose we are asked to make a prediction at time t corresponding to the input $\boldsymbol{x}_{\text{new},t} = (0, 0)$. Then at time $t+1$ we are asked to make a prediction at the location $\boldsymbol{x}_{\text{new},t+1} = (\epsilon_1, \epsilon_2)$.

Table 2.1: Asymptotic complexities for training, prediction, prediction with uncertainty and total (training + prediction with uncertainty) for 4 different emulators. Results assume that $c = \mathcal{O}(\sqrt{d})$ and that parallel computation is not possible.

		No uncertainty		With uncertainty	
Method	Train	Pred	Memory	Prediction	Memory
GP	d^3	Td	d	Td^2	d^2
LA-GP	0	$Td^{1.5}$	0	$Td^{1.5}$	0
SLAP-GP	0	$T_\star d^{1.5}$	$T_\star d^{0.5}$	$T_\star d^{1.5}$	$T_\star d^{0.5}$
LEAP-GP	$Hd^{1.5}$	$Td^{0.5}$	$Hd^{0.5}$	Td	Hd

In the LA-GP framework, we would be expected to explore all d candidate points searching for a near-optimal neighborhood $\mathcal{C}_{t+1}$ and would then be required to invert a $c \times c$ matrix many times. If the ϵ are sufficiently small, one could reasonably expect that reusing the results from the previous time step will lead to a negligible loss of accuracy and a significant amount of time savings.

To accomplish this, we define a *prediction hub* $\mathcal{H}$ to be a mathematical object containing all of the relevant information needed to make future predictions. A hub must contain its own coordinates, the estimated correlation parameter κ and an index set containing the indices $J = (j_1, j_2, \cdots j_c)$ corresponding to the neighborhood selected for prediction. This is all the information required to make a prediction using equation 1.13, but we can make an additional memory for time exchange and store the vector $\boldsymbol{\psi} = R^{-1}\boldsymbol{\eta}$ without increasing the asymptotic memory requirements. We will usually be dealing with a set of prediction hubs, denoted $\boldsymbol{\mathcal{H}} = (\mathcal{H}_1, \mathcal{H}_2, \cdots \mathcal{H}_H)$, where each hub is viewed as the tuple $\mathcal{H}_h = (\boldsymbol{x}_h, J_h, \kappa_h, \boldsymbol{\psi}_h)$. If uncertainty estimates for the prediction are desired, then we must also store the matrix K_h^{-1}, leading to the additional memory cost seen in Table 2.1. Additionally, we will store the scalar value $\nu_h = \boldsymbol{\eta}^\top K_h^{-1}\boldsymbol{\eta}$ in order to avoid unnecessary matrix-vector and vector-vector multiplications in the future. Thus, if uncertainty for a prediction is desired, the prediction hubs will be defined as $\mathcal{H}_h = (\boldsymbol{x}_h, J_h, \kappa_h, \boldsymbol{\psi}_h, K_h^{-1}, \nu_h)$.

2.2.1 Building a Prediction Hub

We begin here with a high-level review of the steps required to make a prediction, at location $\boldsymbol{x}_{\text{new}}$, using LA-GP and training data $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{\eta})$. The first stage is to find a reduced training set $\mathcal{C} = (\boldsymbol{X}_c, \boldsymbol{\eta}_c)$, such that $\boldsymbol{X}_c = (\boldsymbol{x}_{j_1}, \boldsymbol{x}_{j_2}, \cdots \boldsymbol{x}_{j_c})$ and $\boldsymbol{\eta}_c = (\eta_{j_1}, \eta_{j_2}, \cdots \eta_{j_c})$. For LA-GP, this stage is executed in two steps. First, a small number of points are chosen via a nearest neighbor strategy. Next, each of the remaining pairs $(\boldsymbol{x}_j, \eta_j)$ are evaluated with respect to some criteria, and the best pair is greedily accepted and added into $\mathcal{C}$. This process is repeated until c pairs have been chosen and $\mathcal{C}$ is complete. In [65, 66], the authors give a number of possible criteria and discuss the tradeoffs extensively, ultimately suggesting that the *active learning Cohn* (ALC) technique be used as a default [36]. Once the reduced training data $\mathcal{C}$ has been selected, a Gaussian process is constructed as described in Section 1.3.2. This entire process takes $\mathcal{O}(c^3)$ time, and a prediction at location $\boldsymbol{x}_{\text{new}}$ can be obtained using eq. (1.13) at a cost of $\mathcal{O}(c)$.

Upon completion of these steps, the prediction hub can be formally constructed by merely storing all of the relevant information. The location of the hub is given by $\boldsymbol{x}_{\text{new}}$ and the neighbor index set $J = (j_1, j_2, \cdots j_c)$ is given by the ALC process described above. The correlation parameter κ is estimated when the Gaussian process is constructed, and the vector $\boldsymbol{\psi}$ must be computed in order to make a prediction. By storing each of these values in memory, represented as $\mathcal{H} = (\boldsymbol{x}_{\text{new}}, J, \kappa, \boldsymbol{\psi})$, future predictions at some location $\boldsymbol{x}_{\text{newer}}$ can be obtained in $\mathcal{O}(c)$ time by using eq. (1.13) directly. In other words, the lengthy LA-GP process can be skipped entirely. Since κ is a scalar, $\boldsymbol{x}_{\text{new}}$ is a vector of length p, and J and $\boldsymbol{\psi}$ are vectors of length c, the cost of storing a single prediction hub in memory is order $\mathcal{O}(c+p)$. This cost is fairly small compared to storing $\mathcal{D}$, which requires $\mathcal{O}(pd)$ memory. Finally, we note that a single prediction hub can be stored using $8(p+1)+12c$ bytes. In the borehole example first described in Section 2.3.1, this equates to 840 bytes per prediction hub. For the borehole example, this implies that roughly 342 prediction hubs can be stored at the same cost as storing the training data $\mathcal{D}$.

2.2.2 Searching for a Prediction Hub

In Section 2.2.1, we discuss how a prediction hub is built and stored in memory. In this subsection, we discuss the details of searching through a set of prediction hubs and using the extracted hub for prediction. Suppose we are equipped with a non-empty set of prediction hubs $\boldsymbol{\mathcal{H}} = (\mathcal{H}_1, \mathcal{H}_2, \cdots \mathcal{H}_H)$ and we wish to make a prediction at some input location $\boldsymbol{x}_{\text{new}}$. As described in the previous subsection, each prediction hub can be used to make a prediction with the input $\boldsymbol{x}_{\text{new}}$. We use the notation

$$\hat{\eta}(\boldsymbol{x}_{\text{new}}|\mathcal{H}_h)$$

to denote a prediction of $\eta(\boldsymbol{x}_{\text{new}})$ using $\mathcal{H}_h$ and eq. (1.13). Of course, each hub will lead to a different prediction and the prediction will generally be worse when the hub location $\boldsymbol{x}_h$ is far from the location of interest $\boldsymbol{x}_{\text{new}}$. Although multiple hubs can be aggregated to produce a prediction, we will limit ourselves to a single hub.

The simplest strategy for hub selection is to choose the hub which is nearest to the new location or, more formally, we select $\mathcal{H}_{h_\star}$ such that

$$h \neq h_\star \;\Rightarrow\; d(\boldsymbol{x}_{\text{new}}, \boldsymbol{x}_{h_\star}) < d(\boldsymbol{x}_{\text{new}}, \boldsymbol{x}_h).$$

We can improve this method by choosing the hub for which the response at $\boldsymbol{x}_{\text{new}}$ and the response at $\boldsymbol{x}_h$ are most strongly correlated. That is, we want to find h that maximizes $\text{Cor}(\eta(\boldsymbol{x}_{\text{new}}), \eta(\boldsymbol{x}_h)$. Based on the isotropic Gaussian correlation function from eq. (1.12), this can be accomplished by selecting $h_\star$ such that

$$h \neq h_\star \;\Rightarrow\; \kappa_{h_\star} d(\boldsymbol{x}_{\text{new}}, \boldsymbol{x}_{h_\star})^2 < \kappa_h d(\boldsymbol{x}_{\text{new}}, \boldsymbol{x}_h)^2. \tag{2.1}$$

This can be accomplished by directly computing $\kappa_h d(\boldsymbol{x}_{\text{new}}, \boldsymbol{x}_h)^2$ for $h = 1, 2, \cdots H$ and selecting h with the smallest corresponding value. If the number of hubs H is not too large, then this may be the best approach. When the number of hubs is large, especially with respect to the input dimension p, it may be worthwhile to maintain a KD-Tree data structure for the hub locations. A KD-Tree is a space partitioning data structure which can allow for a nearest neighbor search in logarithmic time [12, 40]. The memory cost of maintaining a KD-Tree is linear in H, but it will allow us to significantly decrease the time spent searching for the nearest hub. If the number of hubs is very large and the dimension of the input space is not, then we highly recommend the use of a KD-Tree during the hub extraction phase.

In summary, to make a prediction at location $\boldsymbol{x}_{\text{new}}$ using an existing hub, the total computational cost is either $\mathcal{O}(c + H)$ using linear search and $\mathcal{O}(c + \log H)$ using a KD-Tree, although the latter approach also requires an additional memory cost of $\mathcal{O}(H)$.

2.3 SLAP-GP Emulation

Sequential Local Approximate GP emulation initially works equivalently to the LA-GP, creating prediction hubs sequentially and as-needed. When a new prediction location $\boldsymbol{x}_{\text{new}}$ is sufficiently close to the coordinates of an existing hub, the hub is extracted and used to make the prediction. This allows us to avoid the more expensive procedure of building a LA-GP and allows for prediction in $\mathcal{O}(c)$, rather than $\mathcal{O}(c^3)$ time. This approach shares many of the same benefits as the standard LA-

GP framework, including the fact that it requires no formal training phase. Assuming that the overhead is relatively small, the SLAP-GP framework is guaranteed to be at least as fast as the corresponding LA-GP emulator, with efficiency gains becoming more prominent as the number of predictions T increases. In particular, the cost of obtaining T predictions using the SLAP-GP emulator is $\mathcal{O}(T_\star c^3 + (T - T_\star)c)$, where the first term represents the number of predictions which are made by building a new LA-GP and the second term represents predictions coming from a previously constructed hub. Since $T_\star \leq T$, this is guaranteed to be no worse asymptotically than the LA-GP emulator. Under mild conditions, there exists a constant upper bound on $T_\star$ so that the asymptotic complexity of SLAP-GP is $\mathcal{O}(Tc)$ as $T \to \infty$.

In order to formalize this idea, we will need to define what it means for a prediction location $\boldsymbol{x}_{\text{new}}$ to be sufficiently close to an existing hub. We attack this problem by defining the prediction boundary of $\mathcal{H} = (\boldsymbol{x}, J, \kappa, \boldsymbol{\psi})$, with respect to a parameter $\rho \in (0, 1)$, to be the set

$$B(\mathcal{H}) = \{\boldsymbol{x}_0 \mid R(\boldsymbol{x}_0, \boldsymbol{x}|\kappa) \geq \rho\} \, .$$

Taking $d(\cdot, \cdot)$ to be a generic distance function, the boundary can be rewritten as

$$B(\mathcal{H}) = \left\{ \boldsymbol{x}_0 \;\middle|\; d(\boldsymbol{x}_0, \boldsymbol{x}) \leq \sqrt{\frac{-\log\rho}{\kappa}} \right\} . \tag{2.2}$$

Written like this, the term $\sqrt{\frac{-\log\rho}{\kappa}}$ can be viewed as the *radius* of a prediction hub. Now, consider prediction at a new location $\boldsymbol{x}_{\text{new}}$.

- If $\boldsymbol{x}_{\text{new}}$ is not contained in the boundary of any current prediction hub, then we use LA-GP to make a prediction and create a new hub at location $\boldsymbol{x}_{\text{new}}$.
- If $\boldsymbol{x}_{\text{new}}$ is contained in the boundary of $K \geq 1$ prediction hubs, use any number of these hubs to make a prediction at $\boldsymbol{x}_{\text{new}}$. A new prediction hub will not be created.

This can also be viewed as a modified K-NN prediction problem, using

$$\kappa_h d(\boldsymbol{x}_{\text{new}}, \boldsymbol{x}_h)^2 \tag{2.3}$$

as a notion of distance between $\boldsymbol{x}_{\text{new}}$ and a prediction hub location $\boldsymbol{x}_h$. For efficiency and simplicity, we propose using just the "closest" prediction hub, even if $K > 1$, by adapting 1.6 as

$$\hat{\eta}(\boldsymbol{x}_{\text{new}}) = \hat{\eta}(\boldsymbol{x}_{\text{new}} \mid \mathcal{H}_h), \text{ where } h' \neq h \;\Rightarrow\; \kappa_h d(\boldsymbol{x}_{\text{new}}, \boldsymbol{x}_h) < \kappa_{h'} d(\boldsymbol{x}_{\text{new}}, \boldsymbol{x}_{h'}). \tag{2.4}$$

Alternatively, one could use multiple prediction hubs, extending the idea behind equation 1.8, which may lead to improved predictions in some scenarios. This differs from a true nearest neighbor algorithm in the sense that a prediction hub must be created when there does not exist a current hub which is sufficiently close to $\boldsymbol{x}_{\text{new}}$. As discussed in Section 2.2.2, our implementation also uses a K-D tree to allow for efficient extraction of the nearest hub.

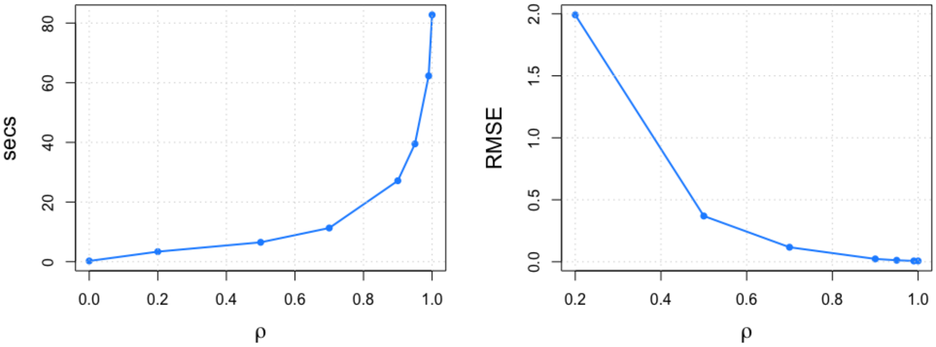


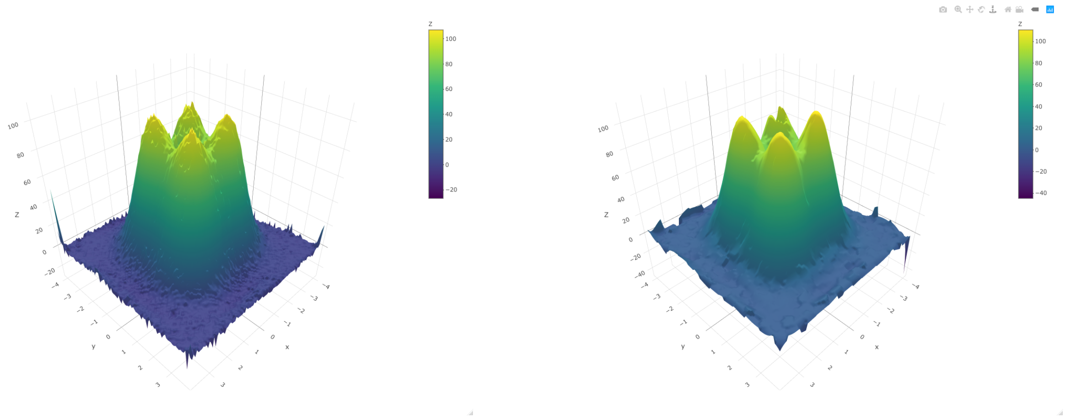
Figure 2.1: Time-accuracy tradeoff for the SLAP-GP

An intuitive advantage of this approach is that it relies on just a single tuning parameter ρ, which can be selected in terms of a time/accuracy tradeoff. When $\rho = 1$, the SLAP-GP is equivalent to the LA-GP emulator with additional overhead.

The case where $\rho = 0$ is akin to fitting a single LA-GP for the first prediction and using it globally for all predictions thereafter. For values of $\rho \in (0, 1)$, the SLAP-GP emulator can be viewed as an LA-GP where suboptimal neighborhoods are sometimes used for prediction. The degree and frequency of sub-optimality which we will allow is determined by the parameter ρ. For illustration, we return to the Gramacy-Lee function defined in eq. (1.10), where the training data consists of $d = 10,000$ Latin hypercube samples, $(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \boldsymbol{x}_{10000})$, over the input space $\mathcal{X} = [-4, 4]^2$. A separate Latin hypercube sample of size $T = 1000$ is taken to form the prediction set $\boldsymbol{X}_{\text{new}}$. Using $c = 100$, the SLAP-GP emulator was used to predict the output at each of the new locations using a variety of ρ values between 0 and 1. For each run, we recorded the run time and the *root mean squared error*, defined as

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(\eta(\boldsymbol{x}_{\text{new},t}) - \hat{\eta}(\boldsymbol{x}_{\text{new},t})\right)^2}. \qquad (2.5)$$

Figure 2.1 illustrates this time/accuracy tradeoff as a function of ρ. The emulation surface constructed for the SLAP-GP is shown in Figure 2.2a and is directly comparable to the surfaces in Section 1.3.1 and Section 1.3.2.



(a) SLAP-GP Emulator with $\rho = 0.99$ (b) LEAP-GP Emulator with $H = 400$

Figure 2.2: The Gramacy-Lee emulated surfaces for the SLAP-GP(0.99) and LEAP-GP(400) emulators. See Section 1.3.2 for details.

One remaining issue with the SLAP-GP emulator, is that the interpolation property is not guaranteed for all $\boldsymbol{x}_j \in \boldsymbol{X}$. To see this, suppose that $\boldsymbol{x}_{\text{new}} \in \boldsymbol{X}_{\text{new}}$ is equal to some $\boldsymbol{x}_j \in \boldsymbol{X}$, and suppose that a prediction is made using a pre-existing prediction hub $\mathcal{H}_h$. If $\boldsymbol{x}_j$ is one of the c points used to build the LA-GP for this hub, or equivalently if $j \in J_h$, then the interpolation property will hold. Thus the SLAP-GP emulator will be an interpolator with high probability, whenever ρ is close to 1. The interpolation property can be permanently restored with the following steps.

i) If $\boldsymbol{x}_{\text{new}} = \boldsymbol{x}_j$ for some $\boldsymbol{x}_j \in \boldsymbol{X}$ and $\mathcal{H}_h$ has been extracted for prediction, then check to see if $j \in J_h$.

ii) If $j \in J_h$, then the interpolation property will hold.

iii) If $j \notin J_h$, then modify the prediction hub $\mathcal{H}_h$ so that $\boldsymbol{x}_j$ is used for prediction. This is just a $\mathcal{O}(c^2)$ operation, if the partition inverse equations are used [8, 65].

This simple procedure guarantees that the interpolation property will hold for the SLAP-GP emulator, and has no effect on the asymptotic complexity of the algorithm. With that said, this can add substantially to the overhead, and we choose not to include it in our implementation, opting instead for a value of ρ which is reasonably close to 1.

2.3.1 Comparison: The Borehole Function

In this section, we compare the SLAP-GP and LA-GP emulators using the well-known borehole function [103, 145]. The details of the simulation follow closely from the work of [66] on LA-GP and of [85] on compactly supported covariance emulation.

The response is given by

$$\eta(\boldsymbol{x}) = \frac{2\pi T_u \left(H_u - H_\ell\right)}{\ln(r/r_w)\left(1 + \frac{2LT_u}{\ln(r/r_w) r_w^2 K_w} + \frac{T_u}{T_l}\right)}, \tag{2.6}$$

where the eight inputs are constrained to lie in the space $\mathcal{X}$ defined by

$$r_w \in [0.05, 0.15] \quad r \in [100, 5000] \quad T_u \in [63070, 115600] \quad T_l \in [63.1, 116]$$

$$H_u \in [990, 1110] \quad H_l \in [700, 820] \quad L \in [1120, 1680] \quad K_w \in [9855, 12045].$$

We generate a maximin LHS of size 4500, using the first 4000 locations for training and the remaining locations for testing. To compare different approaches, we record the runtime of each emulator as well as the normalized root mean squared error (NRMSE) and the *Nash Sutcliffe efficiency* (NSE). The NRMSE is defined as

$$NRMSE = \frac{1}{SD(\boldsymbol{X}_{\text{new}})}\sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(\eta(\boldsymbol{x}_{\text{new},t}) - \hat{\eta}(\boldsymbol{x}_{\text{new},t})\right)^2},$$

where $SD(\boldsymbol{\eta}_{\text{new}})$ is the standard deviation of the true responses corresponding to the test set $\boldsymbol{X}_{\text{new}}$, putting RMSE on a normalized scale. The NSE is a statistic, which can be viewed as an analogue to the R^2 statistic for linear models, which is defined as

$$NSE = 1 - \frac{\sum_{\boldsymbol{x}\in\boldsymbol{X}_{\text{new}}}\left(\hat{\eta}(\boldsymbol{x}) - \eta(\boldsymbol{x})\right)^2}{\sum_{\boldsymbol{x}\in\boldsymbol{X}_{\text{new}}}\left(\hat{\eta}(\boldsymbol{x}) - \bar{\eta}\right)^2}.$$

Writing SLAP-GP(ρ) to denote the SLAP-GP emulator with parameter ρ, the results of the borehole experiment are given in Table 2.2 and clearly demonstrate the time/accuracy tradeoff. For example, the SLAP-GP(0) emulator requires just 0.33 seconds to make $T = 50$ predictions but does so with suboptimal accuracy. The LA-GP emulator takes about 342 times as long to make the predictions but gains an order of magnitude in terms of NRMSE. The SLAP-GP(0.95) represents a reasonable tradeoff between these two extremes, as it is twice as fast as LA-GP with a NRMSE value of 0.02216. We also note that the speedup gained by setting

$\rho < 1$ is expected to increase as the number of predictions T grows. Finally, we note that SLAP-GP(1) and LA-GP are equivalent emulators, as expected, in terms of accuracy. The SLAP-GP(1) emulator requires an additional 7 seconds to make $T = 500$ predictions, representing an overhead of roughly 6%.

Table 2.2: Comparison of LA-GP and SLAP-GP(ρ) emulators using the borehole exemplar.

Method	Pred (secs)	NSE	NRMSE
SLAP-GP(0)	0.33	0.97371	0.14025
SLAP-GP(0.5)	0.33	0.97371	0.14025
SLAP-GP(0.8)	3.52	0.99301	0.07810
SLAP-GP(0.9)	19.78	0.99761	0.04640
SLAP-GP(0.95)	66.76	0.99947	0.02216
SLAP-GP(0.99)	117.58	0.99984	0.01215
SLAP-GP(1)	119.91	0.99984	0.01213
LA-GP	112.96	0.99984	0.01213

2.4 LEAP-GP Emulation

One of the convenient features of the full GP emulator, is that once the GP has been trained, the computation required for future predictions is trivial. The LA-GP and the SLAP-GP emulators avoid a formal training phase, but pay for it later during prediction, since building the neighborhood and inverting the correlation matrix must be done for each new prediction location. By allocating some initial time for training, we obtain faster predictions which is ideal for many online or exploratory analyses. The Localized Ensemble of Approximate (LEAP) Gaussian processes is an emulator which uses many of the same ideas as SLAP-GP but does much of the work in advance so that faster predictions may be obtained down the road. While SLAP-GP builds up the set of prediction hubs as needed, the LEAP-GP emulator creates a set of H prediction hubs in advance. An additional benefit of this method, is that the initial set of hubs can be simultaneously selected, thus restoring the potential for parallel computation which makes the LA-GP so attractive in many other domains. So although the training time for LEAP-GP is theoretically $\mathcal{O}(Hc^3)$, the impact of H can be eliminated or reduced with parallelization. This allows us to choose a much larger value of H than we might otherwise be able to afford, although we note that H may still be limited in practice due to memory requirements, which are $\mathcal{O}(Hc)$ regardless of parallel computation.

All that remains is to determine an acceptable value for H as well as the coordinates of each hub $\mathcal{H}_h$, $h = 1, 2, \cdots H$. Assuming that $c \propto \sqrt{d}$, one possible option is to construct a hub at every single location in the training set $\mathcal{D}$. This strategy has the desirable feature of trivially satisfying the interpolation property and can be trained as efficiently as $\mathcal{O}(d^{1.5})$ with parallel computation. Assuming that resources are available to train this emulator and that a K-D Tree is maintained for the prediction hubs, then each future prediction can be executed in an impressive $\mathcal{O}(d^{0.5} + \log_2 d)$ time. If training is done sequentially, the complexity of the procedure is $\mathcal{O}(d^{2.5})$, which is still an improvement over the standard GP.

Nonetheless, it would be convenient to develop an algorithm for training a LEAP-GP emulator which requires only quadratic sequential training time. This can be accomplished by setting $c \propto \sqrt{d}$ as well as $H \propto \sqrt{d}$. Depending on parallelization, the cost of training such an emulator is anywhere from $\mathcal{O}(d^{1.5})$ to $\mathcal{O}(d^2)$. A further advantage of this strategy is that the additional memory required is linear in the size of the training data $\mathcal{D}$. Since the training data must be stored anyways, the asymptotic memory cost remains the same as both the GP and LA-GP algorithms. Now we must consider placement of the prediction hubs. The general goal should be to have a prediction hub as close as possible to every reasonable prediction location $\boldsymbol{x}_{\text{new}}$. Assuming that the set of training locations $\boldsymbol{X}$ are representative of the input space, we can seek to solve a proxy for this problem instead. Specifically, we propose using the *partitioning around medoids* (PAM) algorithm with a slight modification.

PAM is a clustering algorithm, which is similar in spirit to the k-means algorithm, where each cluster is represented by its medoid [86, 138, 151]. By definition, the medoid is required to be a member of the dataset, a property which will be useful for enforcing the interpolation property (at least partially). Part of what makes PAM so appealing in this application is it's quadratic runtime in the number of data points, thereby maintaining our goal of a $\mathcal{O}(d^2)$ training algorithm. PAM refers to the efficient algorithm which attempts to find a set of H data points from the set $(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \boldsymbol{x}_d)$ which minimizes a clustering cost function. Typically this cost function is given by

$$\text{cost} = \sum_{h=1}^{H} \sum_{\boldsymbol{x} \in N_h} d(\boldsymbol{x}, \boldsymbol{m}_h), \tag{2.7}$$

where $\boldsymbol{m}_h$ is the medoid of the h^{th} cluster, and $\boldsymbol{x} \in N_h$ if and only if $d(\boldsymbol{x}, \boldsymbol{m}_h) \leq d(\boldsymbol{x}, \boldsymbol{m}_{h'})$ for all $h' \neq h$. To better suit the current application, we propose modifying this cost function to

$$\text{cost} = \sum_{h=1}^{H} \max_{\boldsymbol{x} \in N_h} d(\boldsymbol{x}, \boldsymbol{m}_h), \tag{2.8}$$

This cost function attempts to minimize the maximum distance from any point in the training set to its nearest hub, which is precisely our goal. This cost function can be calculated at least as fast as the cost function in 2.7, and therefore we maintain the desired $\mathcal{O}(d^2)$ training requirement.

Unfortunately, the LEAP-GP emulator described above is unlikely to be an interpolator, unless we are pathologically lucky. Recall that the LEAP-GP emulator will only interpolate at location $\boldsymbol{x}_j \in \boldsymbol{X}$ if $j \in J_h$ for at least one $h \in \{1, 2, \cdots H\}$. Thus $\mathcal{J} = \bigcup_{h=1}^{H} J_h$ is the set of all indices j such that the training point $(\boldsymbol{x}_j, \eta_j)$ is involved in the construction of at least one prediction hub, and the metric

$$\text{frac}_{\mathcal{J}} = \frac{|\mathcal{J}|}{d}, \tag{2.9}$$

gives the proportion of training locations for which interpolation is possible. If $\text{frac}_{\mathcal{J}}$ is not relatively close to 1, then the LEAP-GP emulator may need to be re-trained with a larger value of H and/or c.

The Gramacy-Lee surface, defined in eq. (1.10), is emulated using the data described in Section 1.3.1 and using LEAP-GP with $H = 400$. The emulated surface is shown in Figure 2.2b and is directly comparable to the surfaces in Section 1.3.1 and Section 1.3.2. The RMSE value for this example is also given in Table 1.1.

Table 2.3: Comparison of LA-GP, SLAP-GP(ρ) and LEAP-GP(H) emulators using the borehole exemplar.

Method	Train (secs)	Pred (secs)	NSE	NRMSE	$\text{frac}_{\mathcal{J}}$
LEAP-GP(64)	24.05	0.10	0.99759	0.04658	0.66375
LEAP-GP(200)	67.38	0.19	0.99797	0.04290	0.97325
LEAP-GP(500)	163.98	0.39	0.99891	0.03153	0.99975
LEAP-GP(1000)	327.11	0.83	0.99907	0.02909	1.00000
LEAP-GP(4000)	916.67	3.62	0.99931	0.02507	1.00000
SLAP-GP(0.95)	0.00	66.76	0.99947	0.02216	–
LA-GP	0.00	112.96	0.99984	0.01213	–

2.4.1 Comparison: The Borehole Function

Using the same data as in Section 2.3.1, we trained a LEAP-GP emulator using $c = 64$ and various values of H between $\sqrt{d} \approx 64$ and $d = 4000$. We report the time required to make $T = 500$ predictions, the NRMSE and the NSE as well as the time required for training and the statistic $\text{frac}_{\mathcal{J}}$ given in equation 2.9. The results are shown in Table 2.3. We were unable to fit the standard GP for comparison due to cost, but with a crude approximation we estimate that the training procedure would have taken weeks or months to complete using the `GPfit` package in R [96]. The SLAP-GP emulator with $\rho = 0.95$ and the full LA-GP emulator are also included in this table for comparison. Although both the SLAP-GP(0.95) and LA-GP emulators take less time overall, the LEAP-GP emulator will become faster as the number of predictions grows large. A simple linear extrapolation suggests that LEAP-GP(1000) will be faster than LA-GP for $T > 1459$ and will be faster than SLAP-GP for $T > 2481$. Finally, we note that these are the results of training LEAP-GP sequentially, and the use of parallel programming could easily reduce the runtime even further.

2.4.2 Combining LEAP-GP and SLAP-GP

Both the SLAP-GP and LEAP-GP emulators offer the user a simple choice between accuracy and efficiency. Typically, significant time savings are possible at the cost of a near negligible loss in accuracy. In the context of Bayesian model calibration, the discrepancy function can incorporate a small amount of emulation error with little or no effect on the overall results. The LEAP-GP framework is convenient because it can be trained in $\mathcal{O}(d^2)$ time with reasonably good accuracy, especially if parallelization is used in training. Moreover, by moving the work forward, we end up with very efficient prediction capabilities, convenient when running a large number of MCMC repetitions, or even when writing and debugging code.

Table 2.4: Runtime and accuracy results for the emulator which results from combining SLAP-GP(ρ) and LEAP-GP(H).

H	ρ	Train (s)	Pred (s)	NSE	NRMSE	# Hubs
64	0.95	24.05	51.44	0.99946	0.02309	279
64	0.98	24.05	111.81	0.99984	0.01249	539
1,000	0.95	327.11	4.77	0.99912	0.02951	1,016
1,000	0.98	327.11	97.20	0.99979	0.01433	1,406
4,000	0.95	916.67	4.68	0.99931	0.02609	4,000
4,000	0.98	916.67	65.70	0.99971	0.01706	4,260

The SLAP-GP algorithm on the other hand, is capable of prediction accuracy which rivals the LA-GP framework directly, since when $\rho = 1$ they produce equivalent predictions. Thus, in some cases, it may be worthwhile to use a combination of the two approaches. Concretely, we propose using LEAP-GP to train the emulator, shifting as much of the work to the front end as possible. The SLAP-GP structure, preferably with a large value of ρ, can then be used on top of the trained LEAP-GP emulator to improve prediction accuracy. Some selected results are shown in 2.4. Note that by training LEAP-GP, with $H = 4000$, in advance, the resulting emulator is nearly twice as fast for 500 predictions compared to LA-GP. A similar speedup was achieved with the SLAP-GP(0.95) emulator, but the NRMSE of the combined algorithms is 0.017 compared to 0.022 for SLAP-GP and 0.013 for LA-GP.

2.5 Conclusions & Future Work

In this chapter, we propose a series of modifications to the LA-GP framework of [65, 66] for use in applications where parallel prediction is not possible. In many model calibration applications, particularly those with a large number of parameters, sampling from the posterior distributions is time consuming, requiring many thousands (or even millions) of MCMC iterations. In this setting, it is desirable that computer model output can be obtained as efficiently as possible and even a constant speedup can be a huge welcome. Our approaches offer the user flexible choices throughout including the ability to trade time for accuracy and memory in SLAP-GP. A fixed amount of time can also be allocated for training prior to prediction using LEAP-GP, leading to improved accuracy and speed of prediction. Best results are obtained by combining the proposed procedures, training a LEAP-GP emulator using the allocated training time and then improving the emulator as necessary using the SLAP-GP framework during the prediction phase.

Future work will involve a more thorough comparison of these procedures to other emulation strategies, such as sparse Gaussian processes and MARS emulators (see Section 1.3). Our current implementation for LEAP-GP can also be drastically improved by allowing for the training phase to be conducted in parallel. This is a straightforward step, but necessary for a proper comparison of results. At present these methods are limited to the use of the isotropic Gaussian correlation function, so another reasonable modification is to allow for the use of other correlation structures, such as power-exponential, Matérn, and non-isotropic variants. Further theoretical justification or empirical evidence is needed to assess some of the modeling choices. For instance, we propose partitioning around medoids (PAM) for selecting the initial hub structure in LEAP-GP which performs well in the applications we have seen, but the properties of this structure should be considered and compared with other possible options.

Finally, we note that LEAP-GP and SLAP-GP are intended for making fast sequential predictions, but a more general framework could be described allowing for emulation in a broader sense. These methods could theoretically be used for other problems in UQ, such as forward uncertainty propagation, sequential contour estimation or optimization and fully Bayesian (rather than stage-wise) model calibration (i.e. [161]), but a more general analysis which explicitly incorporates variance prediction will be needed.

Chapter 3

Dealing with Nuisance Parameters in BMC

“Your assumptions are your windows on the world. Scrub them off every once in a while, or the light won’t come in.” – Isaac Asimov

3.1 Overview

In our application, it is useful to partition the calibration parameters into the physical parameters of interest $\boldsymbol{\alpha}$, and a set of nuisance parameters $\boldsymbol{\gamma}$ whose estimated values are not of scientific interest. To complete the specification of the Kennedy and O’Hagan model (see section 1.4.3, [87]), we must assign a prior distribution on the calibration parameters. Using this partition of calibration parameters, we examine priors of the form

$$\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\gamma}) \sim \pi_\alpha(\boldsymbol{\alpha})\pi_\gamma(\boldsymbol{\gamma}).$$

For simplicity, we assume throughout this chapter that the calibration parameters have been *standardized* so that each θ_i has prior mean and variance of 0 and 1 respectively. Based on the application, we use subject matter knowledge for the physical parameters to build a weakly informative prior $\pi_\alpha(\boldsymbol{\alpha})$. The main goal of this chapter is to explore, compare and develop choices for $\pi_\gamma(\boldsymbol{\gamma})$ for applications consisting of several experiments such as the tantalum example described in section 1.2. When there are a large number of these nuisance parameters, the potential for overfitting with the presence of model discrepancy is increased. We consider the case where some of the nuisance parameters represent measurement uncertainties, such as the thickness or mass of a material sample, where the generating probability distribution, called the measurement error model, is often known. With this additional problem structure in mind, we can attempt to *identify* the overfitting of nuisance parameters and *reduce* the overfitting when it occurs. This information is useful for exploration of the relationship between discrepancy and parameter inference and can sometimes be used to *diagnose* the presence and effect of model discrepancy form on the parameters of interest.

Sections 1.4.3, 1.5.2, 1.5.4 and 1.2 are highly relevant to the contents of this chapter. The rest of this chapter is organized as follows. In section 3.2, we formally define nuisance parameters and overfitting. We propose a criteria called *probability of prior coherency* and a specialized regularization prior called *moment penalization* to identify and reduce the overfitting of nuisance parameters. In section 3.3, these methodologies are used to analyze physical parameters using a variety of applications and synthetic examples, and the results are compared to the standard approach. The results are further explored and extended in section 3.4.

3.2 Overfitting, Nuisance Parameters and Regularization

In the compressibility of tantalum example, measurement uncertainty inputs to the model (i.e. material thickness measurements) are expected to behave like draws from a known generating distribution (i.e. the *measurement error model*) and any systematic bias in these parameters can be easily identified. In section 3.2.1, we describe three scenarios of overfitting in the context of nuisance parameters and BMC and section 3.2.2 describes a criteria which is capable of recognizing overfitting. In section 3.2.3, we build on these ideas and develop a *moment penalization* prior which is capable of reducing overfitting in this context. In section 3.2.4, we discuss how the methods developed in this section can be useful as diagnostic tool for checking the violation of model discrepancy assumptions.

3.2.1 Overfitting for Nuisance Parameters

Overfitting can have many meanings depending on the application and methodology, thus we need to pin down what is meant by overfitting for nuisance parameters in the BMC framework. For illustration, we consider the material thickness parameters from the material property calibration problem of section 1.2. The exclusive source of uncertainty for these parameters is measurement error. For $k = 1, 2$, let $(\gamma_{k1}, \gamma_{k2}, \cdots \gamma_{kp})$ denote each set of nuisance parameters across p experiments. Since the parameters have been standardized to have mean 0 and variance 1, assuming normally distributed errors implies that

$$(\gamma_{k1}, \gamma_{k2}, \cdots \gamma_{kp}) \sim N(0, \boldsymbol{I}_p),\ \ k = 1, 2 \tag{3.1}$$

where $\boldsymbol{I}_p$ is the $p \times p$ identity matrix. The measurement error model specified in eq. (3.1) can be viewed as a modeling constraint on each set of nuisance parameters. For instance, if the measurement device is well registered, the set of $p = 9$ tantalum thickness measurements should look like independent draws from a standard normal distribution. We now outline three scenarios of overfitting for these nuisance parameters, which we will refer to as overdispersion, underdispersion and collective bias.

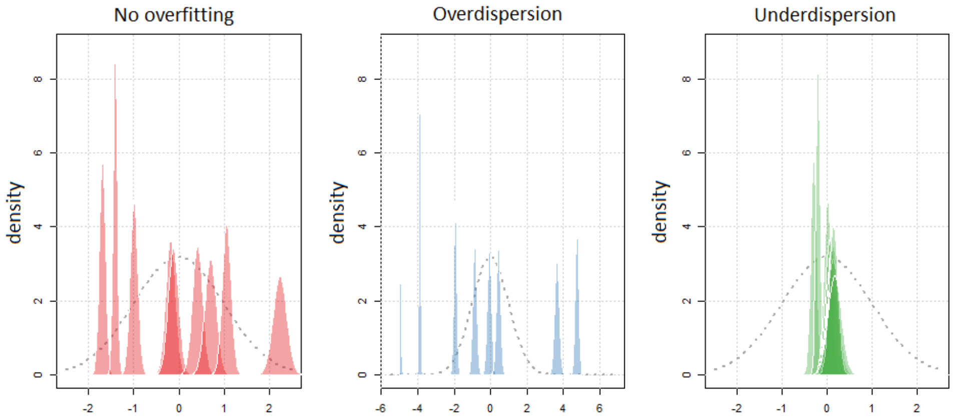


Figure 3.1: In the background of each panel is an (unnormalized) standard normal distribution, representing the expected generating distribution for the $p = 9$ nuisance parameters. The hypothetical posterior distributions shown on the left are consistent with our prior knowledge of the problem structure. The posterior distributions in the middle and right panel are not coherent with our understanding of the problem and may be the result of overfitting.

Suppose that BMC has been performed and we inspect the nuisance posteriors of the thickness parameters. If the posterior mean for some of these parameters is, for example, more than ≈ 5 prior standard deviations away from the prior mean, we should be concerned that these errors are too large given what we know about the measurement device. This *overdispersion* of nuisance parameter estimates is a classic form of overfitting, and can readily occur if prior information is ignored, such as in maximum likelihood estimation or with the use of non-informative priors. On the other hand, suppose we obtain nuisance posteriors which are tightly centered around 0 for each of the nuisance parameters. The *underdispersion* of nuisance parameter estimates here should be a similar concern given our prior information of the measurement process. The specification of independent normal priors on each thickness parameter does nothing to address this, since the prior is maximized when each of the thickness uncertainties is 0. This type of overfitting can lead to biased posteriors in the physical parameters of interest, and a good prior should be able to handle such cases when necessary. Figure 3.1 illustrates these cases using a set of $p = 9$ hypothetical posterior densities. In the left panel there is no evidence of overfitting, as the nuisance parameter posterior means appear like independent draws from a standard normal distribution (dotted gray). The middle panel exhibits the behavior of overdispersion, as several of the posterior distributions are too far from the center. The right panel gives evidence of underdispersion, where the spread of the nuisance parameter estimates is too small according to our expectations.

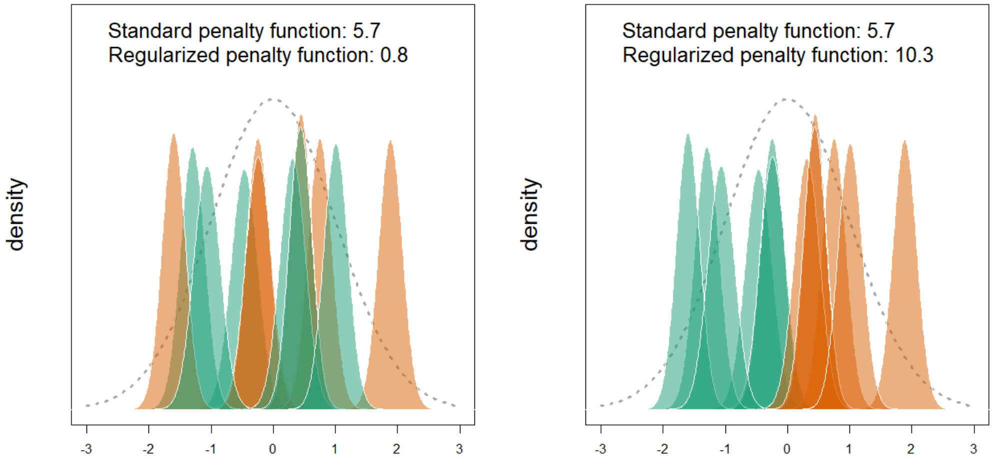


Figure 3.2: Hypothetical posterior densities for two sets of $p = 6$ nuisance parameters. The left panel is consistent with the expected generating distribution (dotted gray curve). The right panel cannot be explained by the generating function and may be the result of overfitting (i.e. collective bias). The standard prior distribution is incapable of distinguishing between these two cases, assigning the same penalty (negative log prior density) to each case. In this setting, a regularization prior should be able to penalize the case on the right.

When model discrepancy leads to systematic bias, a third form of overfitting, which we refer to as collective bias, can readily occur. Suppose that we perform BMC and upon inspection we see that each of the aluminum thickness parameters have negative posterior estimates and each of the tantalum thickness parameters are estimated to be positive. A hypothetical depiction of this type of overfitting is illustrated in fig. 3.2, which shows posterior distributions for 2 groups of $p = 6$ nuisance parameters. In this Figure, each color (orange or green) represents a different group assignment (i.e. tantalum or aluminum thickness). The arrangement on the left is reasonable with respect to the known problem structure, while the arrangement on the right may be an indication of systematic bias caused by a violation of the model discrepancy identifiability constraint. The standard approach of assigning independent standard normal priors for each nuisance parameter is unable to distinguish between these two cases, but a good regularization prior should be capable of penalizing the case on the right.

3.2.2 A Metric for Overfitting

Now that we have defined overfitting of nuisance parameters, we can develop a metric for identifying the existence and severity of this overfitting. To visualize the *collective* behavior of a high dimensional set of nuisance parameters, it is useful to assess the behavior of the first two moments. Such as in the measurement error example, we assume that there are $p \geq 2$ nuisance parameters which have been standardized and are a priori normally distributed. We define,

$$M_\gamma = \frac{1}{p}\sum_{j=1}^{p}\gamma_j \qquad\qquad V_\gamma = \frac{1}{p-1}\sum_{j=1}^{p}(\gamma_j - M_\gamma)^2 \tag{3.2}$$

to be the mean and variance respectively of the nuisance parameter set. Since the nuisance parameters are assumed to be standard normal, we know that M_γ and V_γ are independent and the joint distribution with respect to the prior is[1]

$$\pi_{M_\gamma,V_\gamma}(m,v) = N(m \mid 0, 1/p) \times \left[(p-1)\chi^2(v(p-1) \mid p-1)\right] \tag{3.3}$$

[1] $N(x|\mu,\sigma^2)$ denotes a normal density with mean μ and variance σ^2 and $\chi^2(x|\nu)$ represents a chi-square density with ν degrees of freedom

Now let $\hat{M}_\gamma$ and $\hat{V}_\gamma$ denote posterior estimates (i.e. posterior means) of these quantities. Even if the posterior estimate for each nuisance parameter γ_k is marginally reasonable, it is worthwhile to check that the estimates $\hat{M}_\gamma$ and $\hat{V}_\gamma$ are also coherent with our prior knowledge of the problem structure. If

$$\pi_{M_\gamma,V_\gamma}(m,v) > \pi_{M_\gamma,V_\gamma}(m',v')$$

then we say that the pair (m,v) is *more coherent with the prior* than the pair (m',v'), and we write $(m,v) \succ_c (m',v')$. Let

$$\Gamma_{\hat{M}_\gamma,\hat{V}_\gamma} = \Big\{(m,v) \mid (\hat{M}_\gamma,\hat{V}_\gamma) \succ_c (m,v)\Big\} \tag{3.4}$$

be the set of all pairs (m,v) for which the point $(\hat{M}_\gamma,\hat{V}_\gamma)$ is more coherent with the prior than (m,v). We now define the *probability of prior coherency* of $(\hat{M}_\gamma,\hat{V}_\gamma)$ to be the probability that the posterior estimates are more coherent with the prior structure than a point (m,v), which is distributed according to the density in eq. (3.3).

$$\begin{aligned} p_c(\hat{M}_\gamma,\hat{V}_\gamma) &= \int_{\Gamma_{\hat{M}_\gamma,\hat{V}_\gamma}} \pi_{M_\gamma,V_\gamma}(m,v)\ dmdv \\ &\approx \frac{1}{L}\sum_{\ell=1}^{L} \mathbb{1}\Big((\hat{M}_\gamma,\hat{V}_\gamma) \succ_c (m_\ell,v_\ell)\Big). \end{aligned} \tag{3.5}$$

The second line of eq. (3.5) is a Monte Carlo approximation for some large integer L, where $\mathbb{1}(\cdot)$ is the indicator function and (m_ℓ,v_ℓ) are random draws from the distribution defined in eq. (3.3). For this definition of p_c, we have that $p_c \sim \text{Unif}(0,1)$ whenever equation eq. (3.1) holds [28, p. 397]. An alternative definition of prior coherency is given in Section 3.4.1 which allows for direct computation and eliminates the need for Monte Carlo, but uniformity of the metric under eq. (3.1) will no longer hold.

Figure 3.3 illustrates this metric for 4 different simulated datasets consisting of $p = 10$ nuisance parameters. The large orange point with a black outline represents

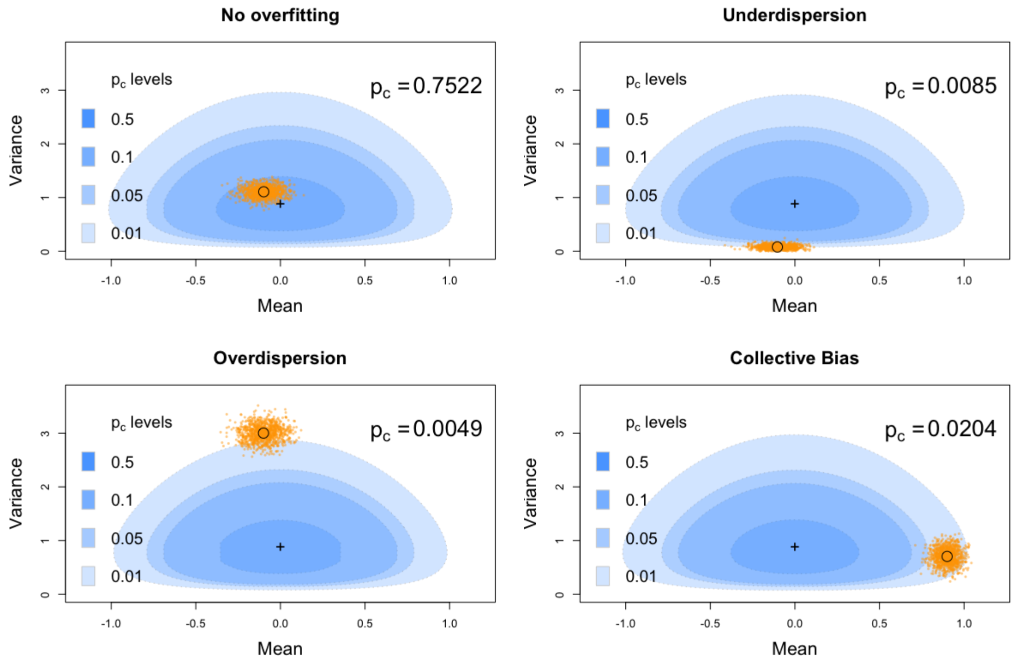


Figure 3.3: Diagnostic plots for four different simulated scenarios. The top-left panel illustrates a scenario where the posterior distributions are coherent with our knowledge of the problem structure. The top-right and bottom-left panels indicate underdispersion and overdispersion respectively. Compare these three panels to the hypothetical posteriors displayed in fig. 3.1. The bottom-right panel is the diagnostic plot corresponding to the collective bias, as shown in the right panel of fig. 3.2 (green posteriors). Probability of prior coherency is computed using the Monte Carlo approximation given in eq. (3.5).

the posterior estimates $(\hat{M}_\gamma, \hat{V}_\gamma)$ in each case, and the surrounding orange scatter represents posterior draws of (M_γ, V_γ). The p_c values were computed using the Monte Carlo approximation in eq. (3.5). The top left panel indicates nuisance posteriors which are consistent with the prior structure. The bottom left panel indicates overdispersion of the nuisance posteriors, where the posteriors are collectively too far from their prior means. The top right panel indicates underdispersion, as $\hat{V}_\gamma$ is far smaller than expected. These three cases can be compared to their corresponding panels in fig. 3.1. The bottom right panel indicates a systematic and collective bias of the nuisance parameters, similar to the case depicted in fig. 3.2 (green). Figure 3.3 demonstrates that small values of p_c may indicate that overfitting is occurring. We emphasize that this criteria is only appropriate when the necessary problem structure exists, such as when the nuisance parameters represent measurement uncertainties. In the absence of suitable problem structure, we suggest the use of other less specialized regularization methods, such as the Dirichlet-Laplace or Horeshoe priors [17,27]. More details on these methods in the context of model calibration can be found in section 1.5.4.

When the posterior distribution for a set of nuisance parameters has low probability of prior coherency, we need to consider possible explanations. First and foremost, it is important that the prior distributions are chosen carefully and reasonably. In our material property applications, experts are confident in their ability to provide reasonable (not necessarily informative) priors. Assuming that the prior distributions are reasonable, consider the following argument. If the model is well-specified, then nuisance parameter posteriors should be relatively coherent. Similarly, this suggests that severe incoherency of the nuisance parameters can be caused by a misspecified model. Although high prior coherency is not guaranteed, even if the model is well-specified, we find that low prior coherency is best explained by model misspecification and overfitting.

3.2.3 Regularization: The Moment Penalization Prior

Building on the idea of prior coherency, in this subsection we develop a prior which attempts to reduce overfitting by pulling the posterior estimate $(\hat{M}_\gamma, \hat{V}_\gamma)$ into a region of higher prior coherency. Since $\hat{M}_\gamma$ and $\hat{V}_\gamma$ are standardized versions of the first and second moment, we refer to the prior described in this section as the *moment penalization (MP) prior*. As before, we assume that all nuisance parameters have been standardized and are normally distributed, so that the structure given in eq. (3.3) holds. Section 3.4.6 discusses an extension of the methods discussed here for uniformly distributed nuisance parameters.

It is common in the Bayesian regularization literature to treat a prior distribution as a *penalty*, and it will be useful and informative for us to follow that approach here [15, 27]. In this framework a penalty function $\text{pen}_\lambda(\boldsymbol{\gamma})$ is chosen to either reward or penalize a candidate solution $\boldsymbol{\gamma}$ based on some criteria, where the hyperparameter(s) λ controls the magnitude of the penalty. The corresponding prior distribution becomes $\exp(-\text{pen}_\lambda(\boldsymbol{\gamma}))$. In the current setting, we want to penalize nuisance parameter candidates which have low probability of prior coherency. The probability of prior coherency will be large when the constraints $M_\gamma = 0$ and $V_\gamma = 1$ are approximately satisfied. Using simple squared loss penalty functions, we obtain

$$\text{pen}_\lambda(\boldsymbol{\gamma}) = \lambda_1(M_\gamma - 0)^2 + \lambda_2(V_\gamma - 1)^2$$

which leads to the following prior

$$\pi_\gamma^{MP}(\boldsymbol{\gamma}) \propto \exp\left[-\lambda_1 M_\gamma^2\right]\, \exp\left[-\lambda_2(V_\gamma - 1)^2\right]. \tag{3.6}$$

The probability of prior coherency can be made large by increasing the penalty terms λ_1 and λ_2, forcing the nuisance posteriors to behave in a manner which is consistent with the prior structure. As a starting place, it would be convenient to have "default" hyperparameter values which allow the MP prior to behave similarly to the independent prior of equation eq. (3.1). According to our knowledge of the problem structure, or equivalently under the SI prior, we have $Var(M_\gamma) = 1/p$ and $Var(V_\gamma) = 2/(p-1)$. Thus we can reparameterize the prior by setting,

$$\lambda_1 = \frac{\omega_1}{2Var(M_\gamma)} = \frac{p\omega_1}{2}, \qquad \lambda_2 = \frac{\omega_2}{2Var(V_\gamma)} = \frac{(p-1)\omega_1}{4}.$$

Now when $\omega_1 = \omega_2 = 1$, the variance of each Gaussian kernel, which define the new prior, is equal to the corresponding implied variance under the SI prior. We write

$\boldsymbol{\gamma} \sim MP(\omega_1, \omega_2)$ to denote that the joint prior density of the nuisance parameter vector is

$$\pi_\gamma^{MP}(\boldsymbol{\gamma}) \propto \exp\left[-\frac{p\omega_1}{2} M_\gamma^2\right] \; \exp\left[-\frac{(p-1)\omega_2}{4}(V_\gamma - 1)^2\right], \qquad \boldsymbol{\gamma} \in \mathbb{R}^p, \; \omega_1 > 0, \; \omega_2 > 0. \tag{3.7}$$

If $\boldsymbol{\gamma} \sim MP(1,1)$, then we say that the nuisance parameter vector has a *standard moment penalization (SMP) prior*. The general objective of the moment penalization prior is to reward posterior solutions for which the constraints $M_\gamma = 0$ and $V_\gamma = 1$ are approximately satisfied. The size of the reward or penalty given to a particular solution is a function of the hyper-parameters ω_1 and ω_2. The desired effect is that when $\omega_1 = \omega_2 = 1$, the MP prior closely mimics the behavior of the standard informative prior of eq. (3.1). Setting ω_1 or ω_2 between 0 and 1 leads to a prior which is less informative than the standard informative prior which is an undesirable characteristic in the present context. On the other hand, setting $\omega_1 = \omega_2 = \infty$ should place all of the prior density on the set

$$\Gamma_{\infty,p} = \{(\gamma_1, \cdots \gamma_p) \mid M_\gamma = 0, \; V_\gamma = 1\}. \tag{3.8}$$

When $p = 2$, for example, this set contains just two points

$$\Gamma_{\infty,2} = \left\{ \left(\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\right), \; \left(\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right) \right\}.$$

As p increases, the marginal constraints on each nuisance parameter γ_j become less restrictive. For example, the set $\Gamma_{\infty,3}$ contains infinitely many points with the marginal constraints $|\gamma_j| \leq \sqrt{4/3}$. This bound increases with p, so that there are no marginal restrictions on each γ_j as $p \to \infty$. In section 3.4.2, we show how the normalizing constant can be approximated efficiently for any p and prove that this normalizing constant is both finite and positive for all $\omega_1, \omega_2 > 0$. We also develop an efficient rejection sampler [29] for generating draws from the MP prior for small to moderate values of p. Using this rejection sampler, we draw $100,000$ independent samples from the $MP(1,1)$, $MP(5,5)$ and $MP(1,20)$ priors, for dimensions of $p=2$, $p=3$ and $p=5$. In section 3.2.3, samples from these 9 distributions are summarized with a two-dimensional histogram of (γ_1, γ_2) and a marginal histogram of γ_1, with the standard normal density curve overlaid (solid line) for reference.

When the dimension is small, the marginal distribution of each γ_j can be distinctly non-Gaussian and even bimodal. As the dimension p grows, the marginal distributions become unimodal and bell-shaped. In particular, we have been unable to find any combination of ω_1 and ω_2 which leads to a multi-modal distribution for $p \geq 5$.

Another feature of the MP prior, is that it induces correlation for each pair of nuisance parameters, especially for large ω_1 and ω_2. We view this dependence as a necessary trade-off to encourage the desired constraints, and we note that this induced dependence is not uncommon in regularization frameworks. For example, the magnitude of regression coefficients are positively correlated under the well-known horseshoe prior (eq. (1.51)) [27]. Moreover, it can be formally shown that $-\frac{1}{p-1} \leq \text{Cor}(\gamma_j, \gamma_{j'}) \leq 0$ for all $j \neq j'$, and thus this correlation vanishes as $p \to \infty$ for any fixed ω_1, ω_2.

In the extreme case, we may want to enforce the constraints $M_\gamma = 0$ and $V_\gamma = 1$ exactly, by setting $\omega_1 = \omega_2 = \infty$. Although the MP prior can approximate this case by choosing large but finite values of ω_1 and ω_2, it will not be exact and may lead to computational challenges. In section 3.4.4, we discuss an alternate implementation which intuitively and efficiently enforces these constraints and requires no choice for the hyper-parameters (implicitly, $\omega_1 = \omega_2 = \infty$). In fig. 3.16, the marginal distribution for each γ_j is shown to be approximately standard normal as $p \to \infty$, where the approximation is already quite good for $p = 10$. Moreover, the correlation of any two nuisance parameters is exactly $-1/p$, which quickly tends to 0 for large p.

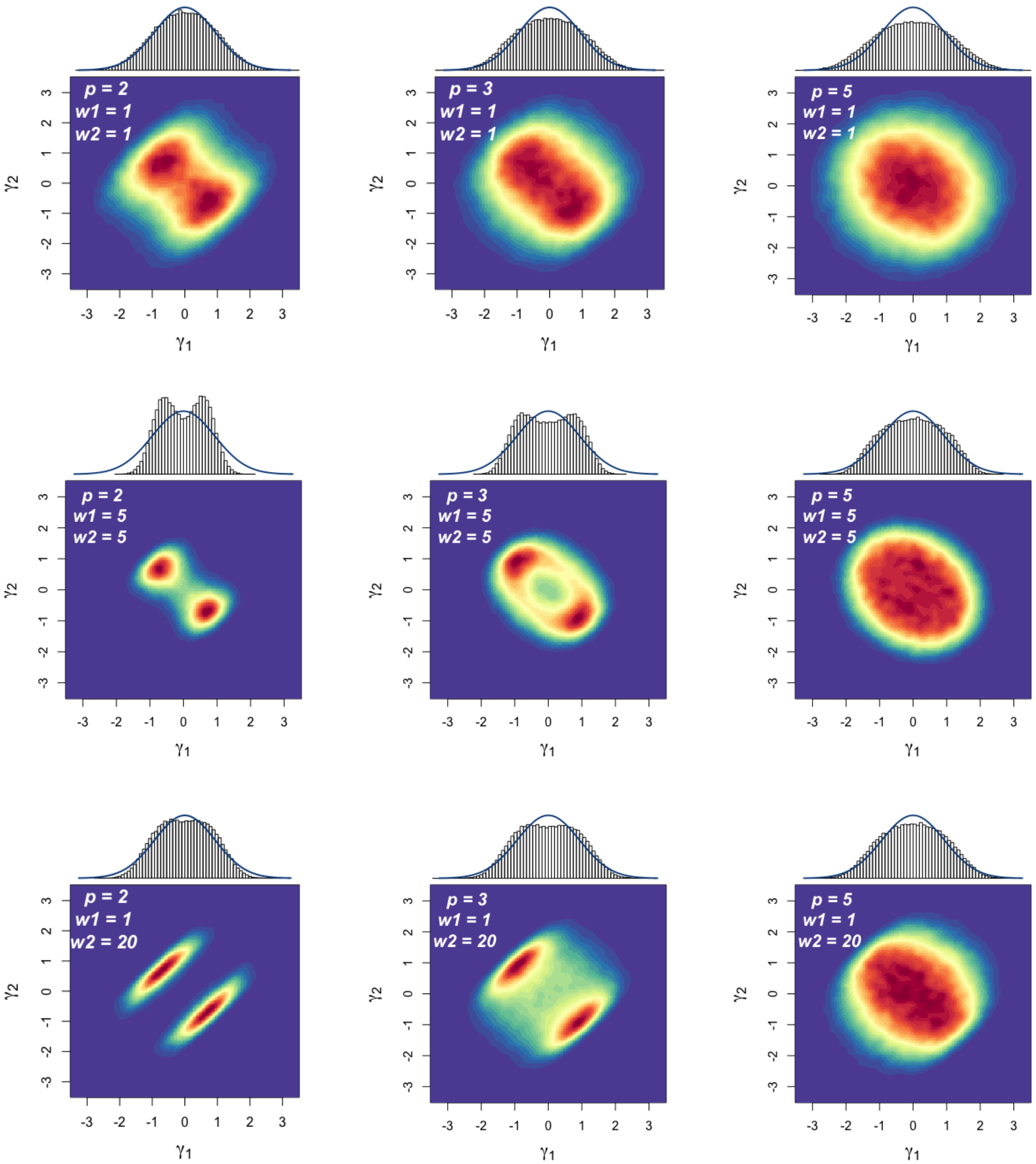


Figure 3.4: Two dimensional histograms of (γ_1, γ_2) based on 10^5 draws from the moment penalization prior using a rejection sampler (see section section 3.4.2 for details). Marginal histograms are also given for the draws of γ_1, with a standard normal density (solid line) shown for reference. The dimension p is equal to 2, 3 and 5 in the first, second and third columns respectively. The parameters (ω_1, ω_2) are set to $(1, 1)$, $(5, 5)$ and $(1, 20)$ in the first, second and third rows respectively. For small p, the distribution can be bimodal and the variables are correlated. As p gets large, the distributions become bell-shaped and the correlation tends to zero.

3.2.4 The MP Prior as a Diagnostic Tool

In this subsection, we discuss how moment penalization can be useful as a diagnostic tool for understanding the relationship between calibration parameters and model discrepancy. In the current framework, there are three components which directly affect the predictions: the physical calibration parameters α, the nuisance parameters γ and the model discrepancy $\delta(\cdot)$. If all components (this includes the assumptions about the structure of the measurement uncertainties) of the model are well specified, then the nuisance posteriors should be relatively coherent. Although the contrapositive of this statement is not equivalent, due to its probabilistic nature [35], we assert that a misspecified model is a possible (and often a probable) explanation for severe incoherency. As we will show in section 3.3, moment penalization can often be used to to reduce the bias of the nuisance parameter estimates and lead to better estimation of these parameters. If the nuisance parameter estimates change under moment penalization then, keeping everything else fixed, the computer model output will change as well. To account for this change and still fit the data well, either the physical calibration parameters or the discrepancy function (or both) will also need to change.

a) The inferred physical parameters stay the same and the inferred model discrepancy changes. In this scenario, the same inferences are obtained for the parameters of scientific interest, regardless of the estimated discrepancy function. This is a positive result, since we have found no evidence that α is unidentifiable with either the nuisance parameters γ or the discrepancy function $\delta(\cdot)$.

b) The inferred physical parameters may change, but the model discrepancy stays the same. This also indicates some level of identifiability between α and $\delta(\cdot)$. One way that this scenario can occur, is if the inferred model discrepancy is completely correct (so that the potential unidentifiability is between α and γ). In this case, it is reasonable to assume that better estimation of γ will lead to better estimation of α, although this can depend on the form of the model discrepancy.

c) Both the physical parameter and model discrepancy inference changes. All of our inferences have changed upon applying moment penalization, rendering it very difficult to pin point which component is to blame. At the very least, we have identified evidence of unidentifiability, leading us to question the reliability of the inferences obtained here. We must try to improve our understanding of the problem so that the model, or model discrepancy assumptions, can be improved. [24].

In order to identify and address overfitting, the methodology described in this chapter gives primacy, in a sense, to the measurement error model. In our applications of interest, we are willing to assume that a subject matter expert can correctly specify the structure (or generating distribution) of a set of nuisance parameters. If these implicit assumptions are violated, then the probability of prior coherency may indicate overfitting, when no such overfitting is actually occurring. In section 3.4.5, we conduct a series of simulations in order to assess the sensitivity of these methods to violations in the measurement uncertainty model. To summarize the findings of section 3.4.5, both heavy tails and underestimation of the true variance can lead to false conclusions, producing p_c values which point towards overfitting when no such overfitting is actually occuring. Our analysis indicates that the probability of prior coherency is fairly robust to tail behavior, unless the tails are exceptionally heavy. Prior coherency is less robust to misspecification of the variance, with possibly severe implications when the true variance exceeds the specified variance by more than about 20%. Taking a conservative approach, the consequences of overestimating the variance are much less problematic.

3.3 Examples

In this section, we show how probability of prior coherency and moment penalization can be used as a diagnostic tool for understanding the relationship between physical parameters and model discrepancy. In section 3.3.1, we demonstrate our approach using synthetic data for a benchmark example adapted from [24]. In section 3.3.2, the methods are applied to the dynamic material property application described in section 1.2. A third example, based on the well-known Borehole function, and an accompanying simulation study can be found in section 3.3.3.

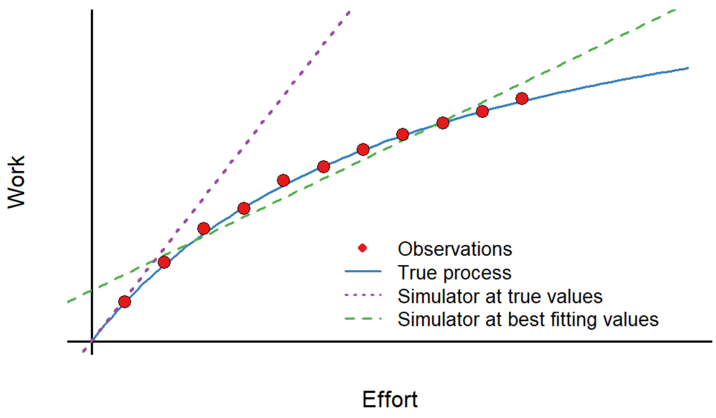


Figure 3.5: A simple machine. The true process $\zeta(x_i)$ (solid blue), the simulator evaluated at the true efficiency values (dotted purple), the simulator evaluated at best fit parameters (dashed green) and the experimental data (red circles). The true discrepancy function is not unbiased across x, so the true parameter values are difficult to infer.

3.3.1 A Simple Machine

We revisit the idea of the simple machine introduced by [24] which is used to illustrate the systematic parameter bias which can occur with the absence of information about the model discrepancy form. We will show that the moment penalization prior can be used as a diagnostic tool to recognize that violation of the model discrepancy assumption is leading to poor inference for the calibration parameters. Consider a collection of p simple machines, which deliver work depending on the amount of effort (x) that we put into it. The true process describing the effort-work relationship is

$$\zeta_j(x) = G_j + \frac{Ax}{1 + x/20} \tag{3.9}$$

The physical parameter of interest here is the efficiency of the machine, denoted by A. The denominator of the second term accounts for loss of work due to friction, and the G_j parameters represent nuisance parameters which we refer to as base-efficiency. The base efficiency parameters, which are not a part of the original formulation [24], play the role of measurement uncertainties in the material property example. Therefore we assume that the G_j can be estimated a priori in an unbiased manner with uncertainty $\sigma_G = 0.05$. To introduce model discrepancy, the simulator for each machine ignores the loss of work due to friction.

$$\eta_j(x, A, G_j) = G_j + Ax \tag{3.10}$$

Setting the true value of efficiency to $A = 0.65$, we simulate experimental data for $p = 10$ simple machines at input locations $x_1, x_2, \cdots x_{11}$ spaced evenly across the interval $[1, 4]$ according to the following data generating mechanism.

$$\begin{aligned} &y_{ij} = \zeta_j(x_i) + \epsilon_{ij} \\ &\boldsymbol{G} \sim N(0, 0.05^2\boldsymbol{I}_{10}) \\ &\epsilon_{ij} \overset{iid}{\sim} N(0, 0.01^2) \end{aligned}$$

The calibration parameters are standardized using hypothetical prior information.

$$\begin{aligned} A \sim N(0.65, 0.3^2) \quad &\Rightarrow \quad \alpha = \frac{A - 0.65}{0.3} \\ G_j \sim N(0, 0.05^2) \quad &\Rightarrow \quad \gamma_j = \frac{G_j - 0}{0.05}, \quad j = 1, 2, \cdots 10 \end{aligned}$$

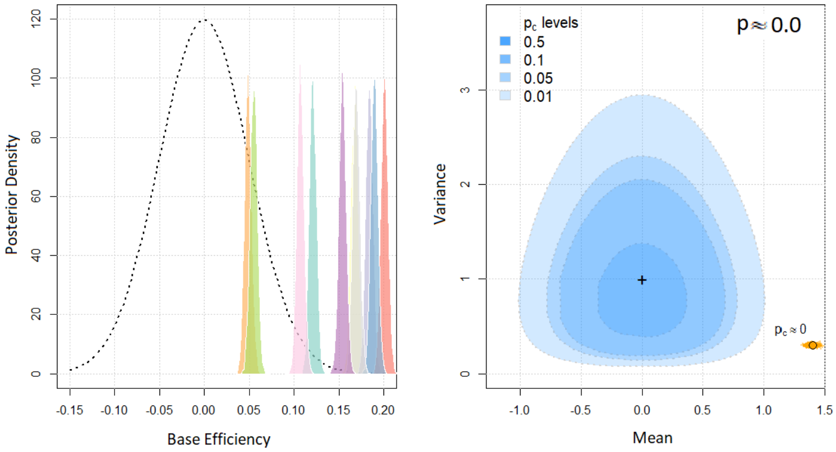


Figure 3.6: In order to better fit the data in the presence of model discrepancy, the inferred nuisance parameters have collectively shifted to the right. These posterior distributions are incoherent with our knowledge of the generating distribution (dotted black line). The probability of prior coherency is effectively 0, indicating strong evidence for overfitting.

Figure 3.5 shows the true process (dashed line), a single realization of the data (circles) and the simulator evaluated at the true parameter values (solid line) and evaluated at the best fitting parameter values (dotted line). As an extreme example of a poor model discrepancy assumptions, we will assume that our computer model is perfect and that there is no model discrepancy. Since our assumptions about model discrepancy are blatantly violated, in order to better fit the data, the inferred base-efficiency (intercept) parameters will be driven upwards and the inferred efficiency (slope) parameter will be driven downwards. We begin the analysis by specifying independent standard normal priors for each of the calibration parameters and we perform BMC. The posterior distributions for the $p = 10$ nuisance parameters

Without information about the model discrepancy form we will be unable to recover the true value of efficiency. The base efficiency parameters will have a systematic upward bias, driving the estimate for the efficiency (the slope) even farther

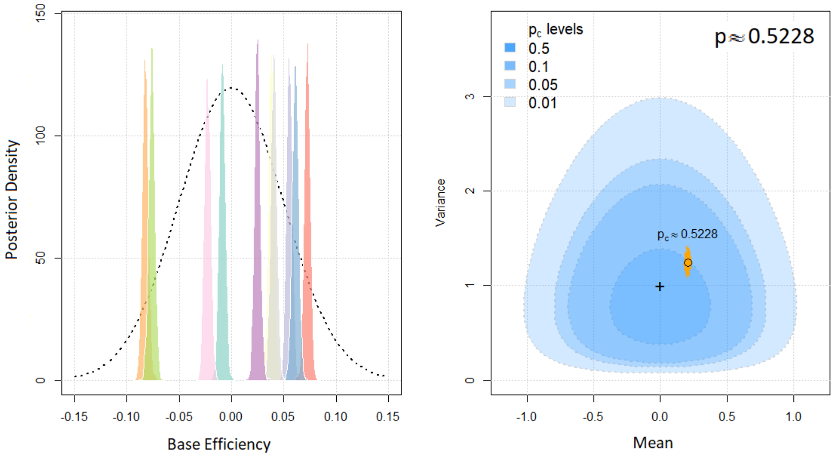


Figure 3.7: Under moment penalization, the collective bias (i.e. overfitting) of the nuisance parameters has been reduced. The inferred base efficiency parameters now behave like draws from the expected generating distribution (dotted gray curve), and the probability of prior coherency is now reasonable.

in the wrong direction. We then assume independent standard normal priors for the physical and nuisance parameters and perform BMC. Diagnostic plots for the nuisance parameters are shown in fig. 3.6. The posterior means are all positive indicating systematic bias, and the Monte Carlo estimate of p_c is 0 demonstrating clear evidence of overfitting. To rectify this, we perform BMC again using a MP(5, 5) prior. The diagnostic plots shown in fig. 3.7 illustrate that the MP prior has pulled the nuisance parameter estimates into a region of higher prior coherency ($p_c = 0.52$), and drastically reduced the bias of these estimates. In this particular example, improving the estimation of the nuisance parameters leads to better estimation of the physical parameter of interest A.

Posterior distributions of the efficiency A under the SI and MP priors are shown in fig. 3.8. As expected, the posterior distribution is biased even after successful regularization of the nuisance parameters and is missing the true value $A = 0.65$ by several posterior standard deviations. Nonetheless, the sensitivity of the posterior for

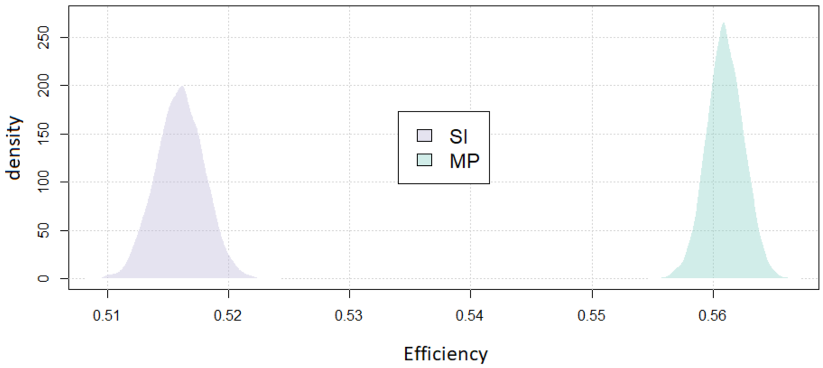


Figure 3.8: Posterior distributions for the efficiency parameter A in the simple machine example. The posterior distribution under moment penalization is much closer to the true value of 0.65, but both posteriors miss the true value. The drastic shift in the posterior distribution of A indicates that the model is sensitive to the estimation of the nuisance parameters.

A to regularization of the nuisance parameters is valuable information, and indicates that a violation of the model discrepancy assumptions is leading to poor inference. This result should cause us to question our results and consider gathering more information about model discrepancy or attempt to improve the model.

3.3.2 Compressibility of Tantalum

In this subsection, we use the probability of prior coherency and moment penalization prior to analyze the tantalum data described in section 1.2. We begin by assigning iid standard normal priors for the 27 nuisance parameters which correspond to the (i) aluminum thickness, (ii) tantalum thickness and (iii) magnetic field scaling parameter for each of the $p = 9$ experiments. The final nuisance parameter, the density of tantalum, is fixed at a nominal value specified by the subject matter expert. After performing BMC, the probability of prior coherency was computed for each of the three sets of nuisance parameters. The posterior distributions for the

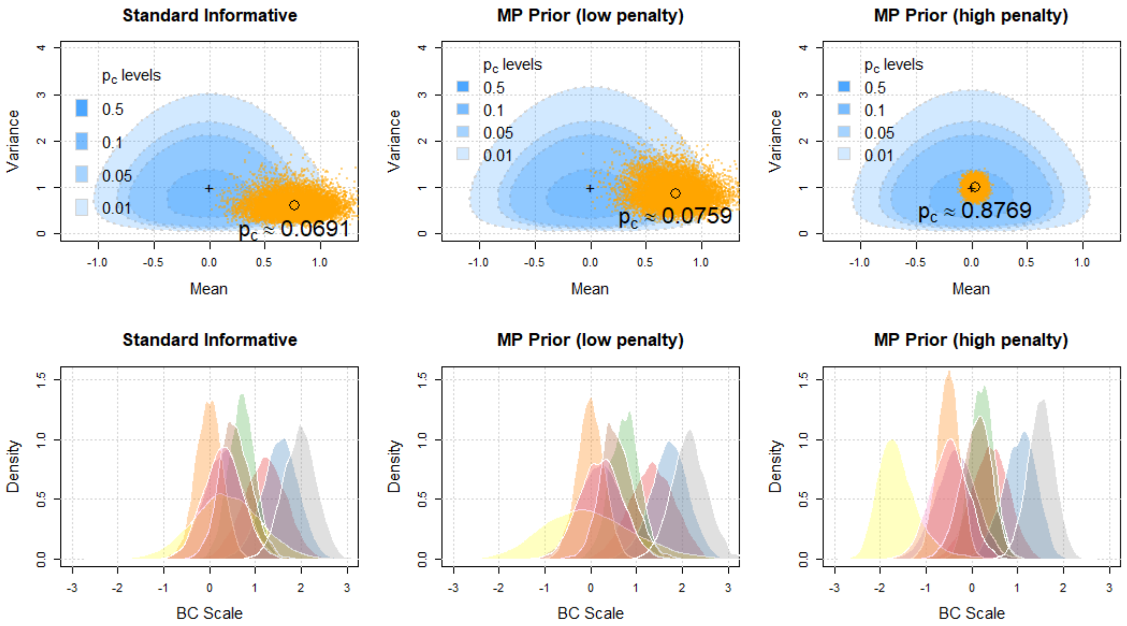


Figure 3.9: Diagnostic plots (top) and posterior distributions (bottom) for the 9 boundary condition scaling nuisance parameters. The SI prior leads to posteriors which are collectively shifted to the right, and low prior coherency ($p_c = 0.0691$). The MP (low penalty) prior produces similar results to the SI prior, by construction. The MP (high penalty) prior forces the posterior solution into a region of high coherency ($p_c = 0.8769$), leading to posterior distributions which are consistent with our knowledge of the generating distribution.

nuisance parameters corresponding to aluminum and tantalum thickness were found to be reasonably coherent with the prior, with $p_c = 0.79$ and $p_c = 0.82$ respectively. The posteriors for the magnetic field scaling parameters, however, showed some evidence of overfitting with $p_c = 0.069$. The posterior distribution for each of these nine magnetic field scaling parameters is shown in the bottom-left panel of fig. 3.9. The posteriors illustrate the collective bias behavior discussed in section 3.2, and the posterior mean is positive for all nine of these parameters. Our interpretation is that the prior incoherency is most likely a product of overfitting, and should be addressed with moment penalization.

We repeat the BMC procedure twice more, assigning moment penalization priors to each of the three sets of nuisance parameters using a low penalty case ($\omega_1 = \omega_2 = 1$) and a high penalty case ($\omega_1 = 20 = \omega_2 = 20$). In the low penalty (standard MP prior) case the analysis is, by construction, very similar to the SI case. The posteriors (bottom-middle) are qualitatively very similar to before, and the prior coherency for the magnetic field scaling parameters changes only slightly ($p_c = 0.0759$). On the other hand, the high penalty MP prior is able to force the posterior distributions into a region of high prior coherency ($p_c = 0.88$). The posterior distributions (bottom-right) for the nine magnetic field scaling parameters now looks consistent with our expectations of the generating model; four of the nine nuisance parameter posterior means are negative and only five are positive.

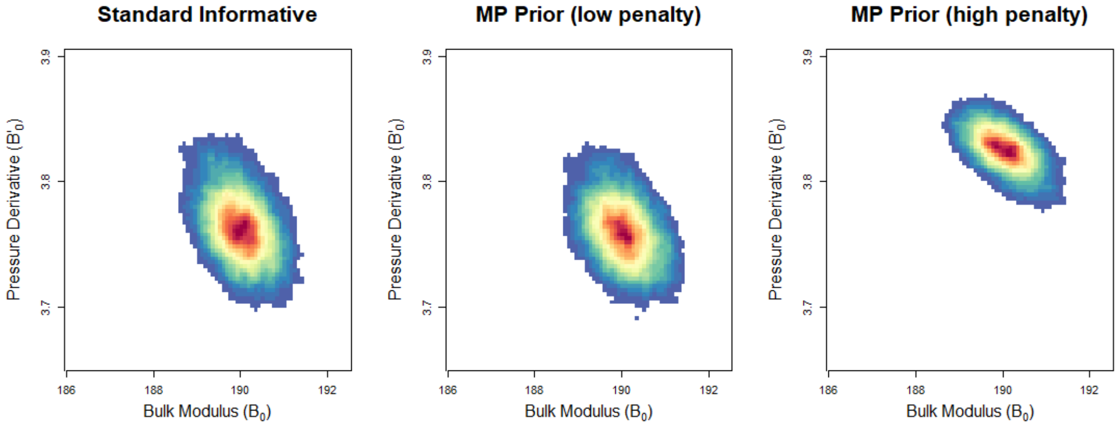


Figure 3.10: Bivariate posterior distributions for the physical parameters (B_0, B_0') with the baseline model (ρ_0 fixed to nominal value). The inferred values of the physical parameters is sensitive to the treatment of the nuisance parameters, indicating a lack of identifiability between physical parameters, nuisance parameters and model discrepancy.

In the next step of our analysis, we examine the posterior distributions for the physical parameters (B_0, B_0'). The bivariate posterior of these physical parameters, shown in fig. 3.10, looks nearly identical under the SI and SMP priors, as expected.

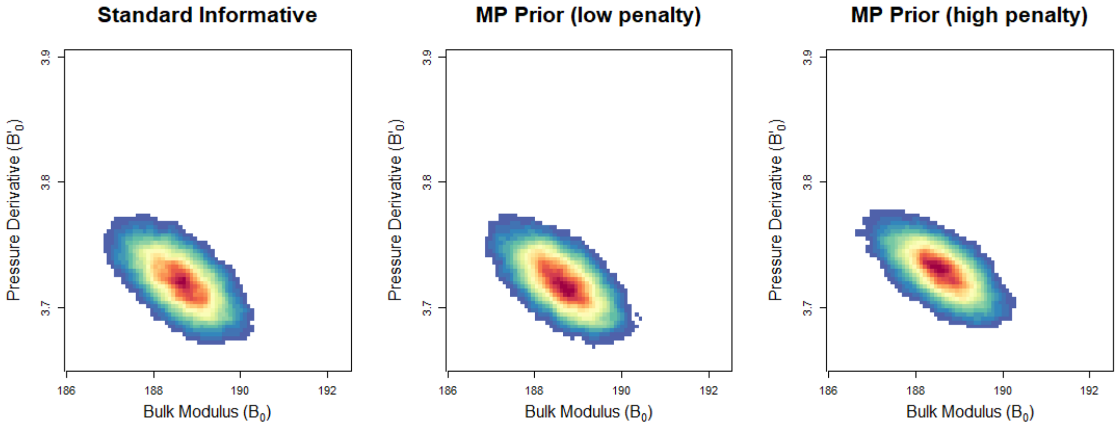


Figure 3.11: Bivariate posterior distributions for the physical parameters (B_0, B_0') with the baseline model (ρ_0 treated as a calibration parameter). There is no longer any evidence of overfitting for the measurement uncertainty parameters, and the posterior distributions are no longer sensitive to the treatment of nuisance parameters.

Application of moment penalization with high penalty, on the other hand, leads to an upward shift in the posterior distribution for both of the physical parameters. In general, there is no way to be sure that the posterior distribution under moment penalization is more reliable than under the SI prior. In fact, the true values may not be contained in either posterior, such as in section 3.3.1.

Since model discrepancy and poor identifiability assumptions are the most likely cause of overfitting, we can attempt to rectify the problem at its root by adjusting the model. For instance, we can obtain a more flexible class of computer models by treating the initial density of tantalum (ρ_0) as a calibration parameter. Rather than fixing ρ_0 at a nominal value, it is equipped with an informative normal prior, centered at a the same nominal value and with variance specified by the subject matter expert. Since the tantalum used in each of the 9 experiments are cut from the same plate, the density is assumed to be constant across all 9 experiments and thus moment penalization cannot be applied to this calibration parameter directly. Now equipped with an informative prior for ρ_0, we repeat the BMC procedure using independent standard normal priors for each of the measurement uncertainty nuisance parameters. With this broader class of computer models, the probability of prior coherency is now 0.86, 0.90 and 0.66 for the aluminum thickness, tantalum thickness and magnetic field scaling parameters respectively. Thus the p_c metric no longer detects overfitting for the 27 measurement uncertainty parameters. For completeness, we perform BMC again using the low penalty and high penalty MP prior. We note that the posterior distributions of B_0, B_0', shown in fig. 3.11, are now far less sensitive to moment penalization.

Once the material properties have been calibrated, these physical parameters can be used to predict the behavior of the material in different settings. For instance, the pressure-density relationship can be modeled, conditional on these parameters,

Table 3.1: Posterior mean predictions of tantalum density at pressures 100-500 GPa. The ALA is a state-of-the-art analytic method and the predictions are used as “ground truth” for comparison. The baseline model fixes the initial tantalum density at a nominal value and the extended model allows this parameter to vary, leading to a more flexible class of models. The bottom row is the estimated mean square prediction error and is used to facilitate comparison. The high penalty MP prior leads to predictions which most closely agree with the ALA results. Likewise, the extended model leads to better agreement of the predictions than the baseline model.

		Baseline model			Extended model		
Pressure (GPa)	ALA	SI	MP (low)	MP (high)	SI	MP (low)	MP (high)
100	22.36	22.41	22.41	22.38	22.22	22.21	22.18
200	26.02	26.26	26.26	26.18	26.08	26.07	26.03
300	29.16	29.38	29.39	29.25	28.80	29.22	29.21
400	31.88	32.10	32.11	31.93	31.05	31.95	31.94
500	34.34	34.55	34.56	34.33	32.98	34.42	34.41
MSPE	–	0.995	1.010	0.708	0.773	0.760	0.675

using the physically motivated Vinet equation-of-state [157]. Each calibration can be used to produce a posterior distribution for the density of tantalum at a given (potentially extreme) pressure. Table 3.1 shows the predicted density of tantalum, i.e. the posterior mean, at $100, 200, 300, 400$ and 500 GPa for each of the 6 calibrations. For comparison, the tantalum pressure-density relationship according to an *average Lagrangian analysis* (ALA), a state-of-the art analytic method, is also given for comparison [20], [130]. Using the ALA predictions as a ground truth, the 6 calibration procedures can be easily compared. To facilitate comparison, the mean squared prediction error for each calibration is given in the bottom row of table 3.1. The baseline model refers to the computer simulator with the initial density of tantalum fixed at a nominal value, where the extended model refers to the more flexible class of models. Note that the predictions for the large penalty MP calibration are in best agreement with the ALA predictions. Moreover, the diagnostic based on probability of prior coherency and moment penalization led us to the extended model, which leads to better predictions under all 3 priors compared to their baseline model counterparts.

3.3.3 The Borehole Function

This section serves to provide a third example to illustrate the use of probability of prior coherency and moment penalization in a Bayesian model calibration context. We will compare the results of BMC using SI and MP priors for two scenarios involving (i) correct specification of the model discrepancy prior and (ii) a naive and faulty specification of the model discrepancy prior. When the model discrepancy prior is well specified, the MP prior has little effect on the posterior distribution of the physical parameter and p_c does not identify overfitting. When the model discrepancy assumptions are violated, we are able to identify overfitting and the posterior is sensitive to the use of moment penalization. Thus the diagnostic capabilities of

MP and p_c are demonstrated even though inference does not improve under moment penalization. In addition, we conduct an extensive simulation study comparing moment penalization to the SI prior and to other methods including Z-regularization (section 3.4.4) and the well-known but non-specialized horseshoe prior [27].

The borehole function models water flow through a borehole. This function is commonly used in computer experiment literature due its simplicity and capacity for quick evaluation [1, 102, 103, 145]. The borehole function can be written as

$$\text{borehole}(x, \boldsymbol{\theta}) = \frac{2\pi T_u \Delta H}{\ln(r/r_w)\left(1 + \frac{2LT_u}{\ln(r/r_w) r_w^2 K_w} + \frac{T_u}{T_l}\right)}, \tag{3.11}$$

and the true process will be defined as $\zeta(x) = \text{borehole}(x, \boldsymbol{\theta}_\star)$ for some true value $\boldsymbol{\theta}_\star$. To simplify the problem, most of the inputs are treated as fixed and known.

- $r = 2,230$ is the *radius of influence* in meters.
- $T_u = 90,000$ is the *transmissivity of the upper aquifer* in meters squared per year.
- $T_l = 90$ is the *transmissivity of the lower aquifer* in meters squared per year.
- $\Delta H = 300$ is the *potentiometric head differential* in meters.

In addition, we take $x = L$ to be a known input or design variable where L is the length of the borehole measured in meters. This input can take values in the range $[1120, 1680]$. For each borehole, there are two unknown calibration parameters which we denote $\boldsymbol{\theta} = (\alpha, \gamma)$. The input r_w is the radius of the borehole in meters which is known up to measurement uncertainty for each borehole and is treated as the nuisance parameter. As is standard in the Borehole function literature [145], we specify a normal prior for r_w as

$$r_w \sim N(0.1, 0.0161812^2).$$

To put the calibration parameters on a standard scale, for each borehole we define

$$\gamma = \frac{r_w - 0.1}{0.0161812} \sim N(0, 1).$$

The physical parameter of interest for this problem is K_w, which represents the hydraulic conductivity of the borehole in meters per year. For the sake of consistency with our application of interest, we assume that K_w is the same for every borehole. Again we use standard prior information from the Borehole function literature, and specify a normal prior for K_w of the form

$$K_w \sim N(10950, 632.2^2).$$

Or equivalently,

$$\alpha = \frac{K_w - 10950}{632.2} \sim N(0, 1).$$

In the spirit of [163], we use the following function as a low fidelity computer simulator

$$\eta(x, \boldsymbol{\theta}) = \frac{2\pi T_u \Delta H}{\ln(r/r_w)\left(1 + \frac{1.4LT_u}{\ln(r/r_w)r_w^2 K_w} + \frac{T_u}{T_l}\right)}, \tag{3.12}$$

which is nearly the same as the Borehole function except for the constant 2 in the second denominator term being replaced by 1.4. This low fidelity simulator introduces model discrepancy, since the true process and the simulator can never agree for all inputs.

To construct a dataset, we simulate $p = 20$ different boreholes each of which yields $n = 10$ observations corresponding to borehole lengths x_i, spaced evenly across the input range $[1120, 1680]$. The true value of hydraulic conductivity is fixed at the prior mean, corresponding to $\alpha_\star = 0$. True values of the nuisance parameters are simulated as $\boldsymbol{\gamma}_\star \sim N(0, \boldsymbol{I}_{20})$. We define $\boldsymbol{\theta}_{\star,j} = (\alpha_\star, \gamma_{\star,j})$ and simulate field data as follows:

$$y_{ij} = \text{borehole}(x_i, \boldsymbol{\theta}_{\star,j}) + \epsilon_{ij}, \qquad \epsilon_{ij} \sim N(0, 0.01^2), \qquad i = 1, \cdots n,\ j = 1, \cdots p.$$

Following section 1.4.3, the data y_{ij} are modeled as

$$\begin{aligned} y_{ij} &= \eta(x_i, \boldsymbol{\theta}_j) + \delta_j(x_i) + \epsilon_{ij} \\ \delta_j(\cdot) &\sim GP(\mu_j, \Sigma_j) \\ \epsilon_{ij} &\overset{iid}{\sim} N(0, 0.01^2) \\ \alpha &\sim N(0,1) \\ \boldsymbol{\gamma} &\sim \pi(\boldsymbol{\gamma}). \end{aligned}$$

The covariance structure of each discrepancy function $\delta_j(\cdot)$ is specified as

$$\Sigma_j(x_i, x_{i'}) = \phi_j \exp(-\kappa_j (x_i - x_{i'})^2).$$

In addition to the prior distributions for α and $\boldsymbol{\gamma}$, we need to specify prior distributions for the discrepancy function hyper parameters $(\boldsymbol{\mu}, \boldsymbol{\phi}, \boldsymbol{\kappa})$. As many authors have discussed [3,21,111], a fully Bayesian treatment of these hyper parameters often leads to severe lack of identifiability between δ and $\boldsymbol{\theta}$. In particular, the μ_j and κ_j parameters will be fixed at known values. In section 3.3.3 we will apply the usual zero mean discrepancy assumption ($\mu_j = 0$) and in section 3.3.3 we will fix the μ_j values at their “true values”. The first approach is an inappropriate assumption for this problem and will lead to faulty inference. The second approach, while typically not feasible in practice, is used for the sake of comparison. To handle the remaining discrepancy parameters, a maximum a posteriori (MAP) estimate is obtained for the discrepancy function corresponding to each of the simulated boreholes, and a Gaussian process is fit to this empirical discrepancy function via maximum likelihood to produce point estimates $\hat{\kappa}_j$ and $\hat{\phi}_j$. For identifiability reasons and to improve the mixing time of the MCMC, the estimated κ_j values are treated as fixed and known throughout the analysis. The ϕ_j variance parameters are assigned weakly informative half Cauchy priors with a scale parameter (also the median) of $\hat{\phi}_j$. The posterior distribution of $(\alpha, \boldsymbol{\gamma}, \boldsymbol{\phi})$ is sampled using a Gibbs sampling scheme with Metropolis-Hastings steps.

Mean of the discrepancy function is known

We begin with the case where the constant bias of the computer model is somehow known and thus μ_j can be fixed at $\mu_{j\star}$. Although this is unrealistic in practice, it represents a case where the prior distribution for each model discrepancy function is reasonably good. These true value are obtained as

$$\mu_{j,\star} = \int_{1120}^{1680} \left(\zeta(x) - \eta(x, \boldsymbol{\theta}_{\star,j})\right) dx, \ \ j = 1, 2, \cdots 20. \tag{3.13}$$

Using the calibration procedure described in this section, posterior distributions are obtained for α and $\boldsymbol{\gamma} = (\gamma_1, \cdots \gamma_{20})$ using the SI prior. The posterior distribution for α (shown in fig. 3.12) is reasonable, yielding a posterior mean which lies just 0.55 posterior standard deviations below the true value $\alpha_\star = 0$. The probability of prior coherency for the nuisance parameters is $p_c = 0.83$, providing no evidence of

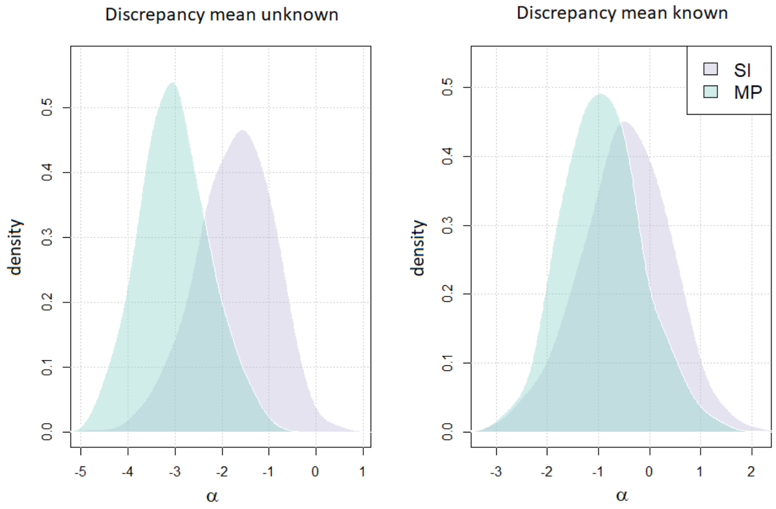


Figure 3.12: (left) Discrepancy mean is unknown, so we make a naive $\mu_j = 0$ assumption. The inference is sensitive to moment penalization indicating a violation of assumptions. As a result, the posterior is biased (true value $\alpha_\star = 0$). (right) When the discrepancy means μ_j are known, the posterior inference is much less sensitive to moment penalization and the posterior inference for α is much improved.

overfitting for the nuisance. The BMC procedure is repeated using a $MP(5,1)$ prior for the nuisance parameters $\boldsymbol{\gamma}$, and we obtain another solution which is consistent with the known problem structure ($p_c = 0.52$). The posterior distribution for the hydraulic conductivity is relatively consistent with the previous calibration results, showing only a slight shift in the posterior distribution. Under both priors, the true value $\alpha_\star$ is captured in the corresponding 80% posterior credible interval. In this example, the nuisance parameters do not seem to be overfit under the SI prior and physical parameter inference is not sensitive to nuisance parameter treatment as a result.

There is no evidence of overfitting under the independent standard normal prior specification ($p_c = 0.83$). Moreover, the posterior inference for hydraulic conductivity is much improved, having a posterior mean $\hat{\alpha}$ which lies just 0.55 posterior standard deviations below the true value. We repeat the BMC using a MP(5,1) prior and obtain a solution which is consistent with the problem structure ($p_c = 0.52$).The posterior distribution for α for both of these priors can be seen in the right panel of fig. 3.12. The posterior inference for α is much less sensitive to moment penalization in this case. Coupled with the high prior coherency of the non-regularized solution, the right panel of fig. 3.12 indicates that the physical parameter inference is not being adversely affected by any violations of the model discrepancy.

Mean of the discrepancy function is unknown

In practice, we will be unable to ascertain the true values $\mu_{j\star}$ with any level of confidence, and will be forced to fix them to some nominal value. A common choice (as discussed in section 1.4.3) is to set the each value equal to zero, $\mu_j = 0$, asserting that each model is unbiased on average across the design variable space. This assumption is not valid in the present setting and the resulting inference may be unreliable.

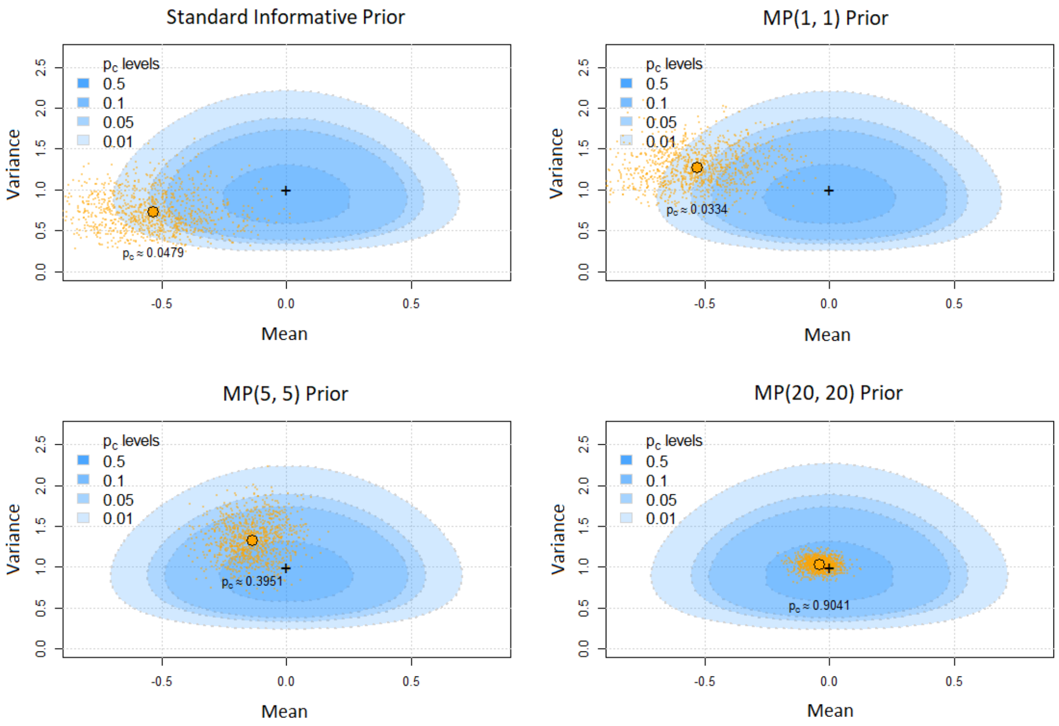


Figure 3.13: Diagnostic plot for the Borehole example under four different priors. Each μ_j, $j = 1, 2 \cdots 20$ is fixed at zero, an assumption which does not hold in this scenario.

We begin by performing BMC using independent standard normal priors for each of the nuisance parameters. The probability of prior coherency for this solution is small ($p_c = 0.0479$), indicating a lack of coherency with the known problem structure. The diagnostic plot, shown in the top left panel of fig. 3.13, indicates that violation of the model discrepancy assumptions may be leading to overfitting in the form of collective bias. We repeat the BMC process using a standard moment penalization prior ($\omega_1 = \omega_2 = 1$). The prior coherency and diagnostic plot (top right) show similar results for both cases. These diagnostic plots indicate that $\hat{M}_\gamma$, the posterior mean of M_γ, is approximately -0.5, indicating a left shift in the nuisance posteriors. To correct for this, we can increase the penalty associated with the first moment. Setting $\omega_1 = 5$, we assign a $MP(5, 5)$ prior on the nuisance parameters and repeat the BMC procedure another time. The diagnostic plot is shown in the bottom left panel of fig. 3.13 and shows that the posterior estimate $(\hat{M}_\gamma, \hat{V}_\gamma)$ is now consistent with our prior expectations. For comparison, a $MP(20, 20)$ prior is also assigned and BMC is performed a fourth time. The diagnostic plot in the bottom right panel of fig. 3.13 demonstrates that the posterior distribution of (M_γ, V_γ) can be shrunk aggressively to the point $(0, 1)$ for large values of ω_1 and ω_2.

Under moment penalization, estimation of the nuisance parameters is drastically improved. The mean squared error, defined here as

$$MSE = \frac{1}{20}\sum_{j=1}^{20}(\hat{\gamma}_j - \gamma_{\star,j})^2,$$

for posterior means $\hat{\gamma}_j$, is 7.2 larger under the SI prior ($MSE = 0.331$) compared to the MP(5,5) prior ($MSE = 0.046$). This does not lead to better physical parameter inference in this case, however, as seen in the left panel of fig. 3.12. Nonetheless, the posterior distribution for α is very sensitive to the treatment of nuisance parameters, and is indicative of a poor computer model and poor modeling assumptions.

Simulation study

The discussion of this section is based on the analysis of a single dataset. To illustrate these ideas more generally, we conducted a small simulation study for the borehole example, setting $p = 10$, $n = 5$ and $\alpha_\star \in \{0, -1, 2\}$.

To assess the performance of the moment penalization and other priors, we performed a small simulation study for the Borehole example, by setting $p = 10$, $n = 5$ and $\alpha_\star \in \{0, -1, 2\}$. To mimic the measurement error model structure, we sampled 100 different nuisance parameter sets $\boldsymbol{\gamma} \sim N(0, \boldsymbol{I}_{10})$ and simulated a corresponding dataset for each value of $\alpha_\star$. We assume that the mean of the discrepancy function is unknown, using $\mu_j = 0$ as the identifiability constraint.

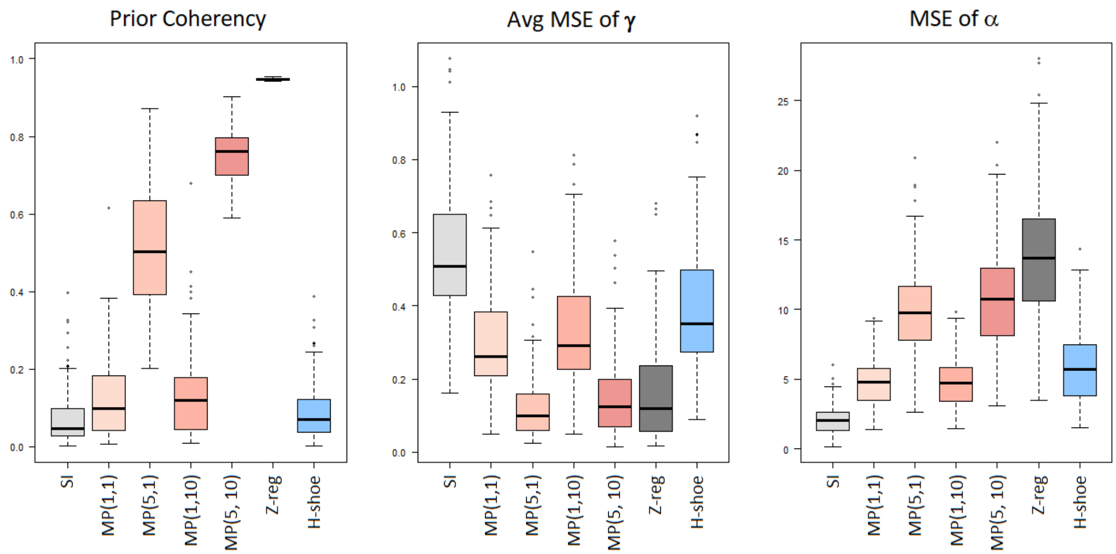


Figure 3.14: Boxplots showing the distribution of p_c, MSE_α and $\overline{MSE_{\boldsymbol{\gamma}}}$ for 7 different priors and 100 simulations. True value of physical parameter is $\alpha_\star = 0$.

For each simulated dataset, we calibrate the model with 7 different priors for the nuisance parameters. Along with the standard informative prior, we also consider four moment penalization priors consisting of all combinations of $\omega_1 \in \{1, 5\}$ and $\omega_2 \in \{1, 10\}$. Additionally, the results are compared to the Z-regularization prior (see section 3.4.4) and the Horseshoe priors (see section 1.5.4). For each simulation, we compute the prior coherency (p_c), the MSE of γ_k

$$\overline{MSE_{\boldsymbol{\gamma}}} = \frac{1}{p}\sum_{j=1}^{p}(\hat{\gamma}_j - \gamma_{\star,j})^2$$

and the MSE of α

$$MSE_\alpha = (\hat{\alpha} - \alpha_\star)^2,$$

where $\hat{\alpha}$ and $\hat{\gamma}_j$ denote a posterior mean.

Figure 3.14 shows the distribution of these metrics for each of the 7 priors in the $\alpha_\star = 0$ case. These boxplots illustrate that the moment penalization prior is improving the prior coherency of the nuisance parameter estimates as expected. In particular, the $MP(5,1)$ and $MP(5,10)$ priors enforce a reasonable level of prior coherency in agreement with the process described in fig. 3.13. The Z-regularization prior always leads to a prior coherency of 0.95, since the mean and variance constraints are achieved exactly. The full results of this simulation study are summarized in table 3.2.

Table 3.2: Summary of the Borehole simulation results. Reported value is the median across 100 simulations. Bold value indicates "best" value in the column.

	Prior Coherency			Avg MSE of γ			MSE of α		
		$\alpha_\star$			$\alpha_\star$			$\alpha_\star$	
	0	-1	2	0	-1	2	0	-1	2
SI	0.05	0.02	0.19	0.51	0.52	0.51	**2.07**	2.03	**2.07**
MP(1,1)	0.10	0.06	0.27	0.26	0.38	0.15	4.81	**1.97**	13.21
MP(5,1)	0.50	0.45	0.59	**0.10**	**0.12**	**0.06**	9.79	6.5	18.71
MP(1,10)	0.12	0.06	0.34	0.29	0.41	0.17	4.76	2.28	13.29
MP(5,10)	0.76	0.73	0.83	0.12	0.15	0.08	10.72	7.29	19.42
Z-reg	**0.95**	**0.95**	**0.95**	0.12	0.15	0.09	13.66	10.18	22.26
H-shoe	0.07	0.04	0.18	0.35	0.48	0.27	5.73	2.57	14.64

3.4 Extensions and Analytic Results

3.4.1 An Analytic Approximation of p_c

In equation 3.5, we propose estimating the probability of prior coherency using Monte Carlo. The use of MC is required since the integral in equation 3.5 is taken over the set $\Gamma_{\hat{M}_\gamma,\hat{V}_\gamma}$ (3.4), leading to the "egg shaped" contours seen in Figure 3.3. Leveraging the independence of V and M for iid normal $\gamma_1, \gamma_2, \cdots \gamma_p$, Monte Carlo can be avoided by altering the definition of p_c. In particular, we define $\tilde{p}_c$ equivalently to p_c in equation 3.5 with the set $\Gamma_{\hat{M}_\gamma,\hat{V}_\gamma}$ replaced by

$$A_{\hat{M}_\gamma,\hat{V}_\gamma} = \left\{(m,v) \mid \left(\pi_M(m) < \pi_M(\hat{M}_\gamma)\right) \wedge \left(\pi_V(v) < \pi_V(\hat{V}_\gamma)\right)\right\}. \tag{3.14}$$

Assuming that $p > 2$, computation of $\tilde{p}_c$ now proceeds as follows,

$$\begin{aligned}\tilde{p}_c &= \int_{A_{\hat{M}_\gamma,\hat{V}_\gamma}} \pi_{\hat{M}_\gamma,\hat{V}_\gamma}(m,v)dmdv \\ &= 2\Phi\left(-\sqrt{p}\,|\hat{M}|\right)\left(1 + F\left((p-1)\hat{V}_-\right) - F\left((p-1)\hat{V}_+\right)\right),\end{aligned} \tag{3.15}$$

where $\Phi(\cdot)$ denotes the standard normal CDF and $F(\cdot)$ denotes the CDF of a chi-square random variable with $p-1$ degrees of freedom. The quantities $\hat{V}_+$ and $\hat{V}_-$ are fully defined by the properties (i) $\hat{V}_+ > \hat{V}_-$, (ii) $\pi_V(\hat{V}_\gamma) = \pi_V(\hat{V}_-) = \pi_V(\hat{V}_+)$ and (iii)

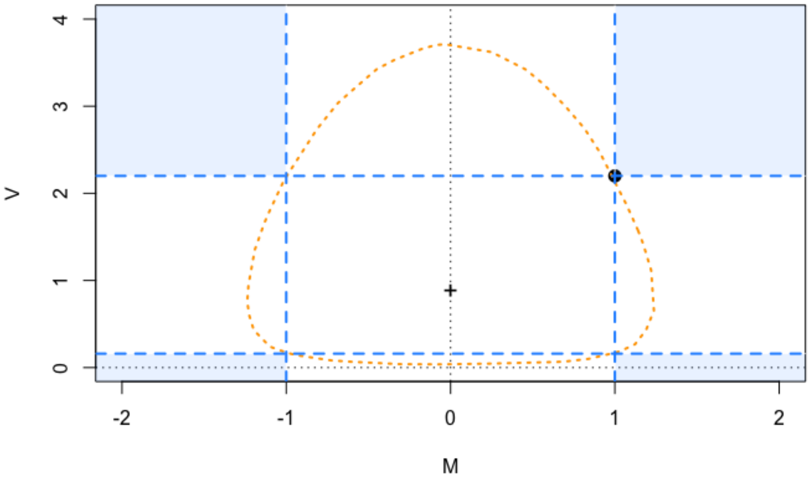


Figure 3.15: A visual illustration of the sets $\Gamma_{\hat{M}_\gamma,\hat{V}_\gamma}$ and $A_{\hat{M}_\gamma,\hat{V}_\gamma}$ when $\hat{M}_\gamma = 1$ and $\hat{V}_\gamma = 2.2$ (solid circle). The set $\Gamma_{\hat{M}_\gamma,\hat{V}_\gamma}$ (equation 3.4) is represented by the exterior of the "egg-shaped" contour (dotted orange line) and the set $A_{\hat{M}_\gamma,\hat{V}_\gamma}$ is represented by the shaded region extending from the corners of the rectangular contour (dashed blue line). This figure also illustrates that $A_{\hat{M}_\gamma,\hat{V}_\gamma} \subset \Gamma_{\hat{M}_\gamma,\hat{V}_\gamma}$.

either $\hat{V}_\gamma = \hat{V}_-$ or $\hat{V}_\gamma = \hat{V}_+$. Equation 3.15 can be calculated efficiently and without the need for Monte Carlo, since the CDF of the normal and chi-square distributions are readily available in many statistical softwares.

On the other hand, we note that $A_{\hat{M}_\gamma,\hat{V}_\gamma}$ is a proper subset of $\Gamma_{\hat{M}_\gamma,\hat{V}_\gamma}$ and therefore $\tilde{p}_c < p_c$ almost surely. Since p_c can be viewed as a p-value with respect to the hypothesis $(\gamma_1, \gamma_2, \cdots \gamma_p) \sim N(\mathbf{0}, I_p)$, it has the desirable property that $p_c \sim \text{Unif}(0,1)$ when the generating distribution is correctly specified [28]. It is clear that $\tilde{p}_c$ cannot possess this property. Moreover, since $\tilde{p}_c$ is always strictly smaller than p_c, it becomes likely that overfitting may be falsely identified. As an example, we conducted a short simulation study and found that in 10000 simulations with $(\gamma_1, \gamma_2, \cdots \gamma_5) \sim N(\mathbf{0}, I_5)$, the distribution of $\tilde{p}_c$ is approximately $\text{Beta}(0.7, 2.2)$ and $P(\tilde{p}_c < 0.05) \approx 0.2$.

3.4.2 The Normalizing Constant

In this section we show that the normalizing constant for the MP prior can be efficiently approximated for the special case $\omega_1 = \omega_2$, and more generally we develop an upper and lower bound for the normalizing constant. By showing that these bounds are both positive and finite, we are able to guarantee that the MP prior is proper for positive parameter values. Let $\boldsymbol{x} = (x_1, x_2, \cdots x_p) \sim N\left(\mathbf{0}, \frac{1}{\tau} I_p\right)$ and define $M = \frac{1}{p}\sum_{i=1}^p x_i$ and $V = \frac{1}{p-1}\sum_{i=1}^p (x_i - M)^2$. Note that the joint probability density function of $\boldsymbol{x}$ can be written as

$$\begin{aligned} h(\boldsymbol{x}|\tau) &= \left(\frac{2\pi}{\tau}\right)^{-p/2} e^{-\frac{\tau}{2}\sum_{i=1}^p x_i^2} \\ &= \left(\frac{2\pi}{\tau}\right)^{-p/2} e^{-\frac{\tau}{2}(pM^2 + (p-1)V)} \qquad (3.16) \\ &= \left(\frac{2\pi}{\tau}\right)^{-p/2} e^{-\frac{p\tau}{2}M^2} e^{-\frac{(p-1)\tau}{4}2V}. \end{aligned}$$

We also recognize that $U = (p-1)\tau V \sim \chi^2(p-1)$.

Now let $\boldsymbol{x} = (x_1, x_2, \cdots x_p) \sim MP(\omega_1, \omega_2)$. Then the normalizing constant for the MP prior is given by

$$\begin{aligned}
k(\omega_1,\omega_2) &= \int_{\mathbb{R}^p} e^{-\frac{p\omega_1}{2}M^2} e^{-\frac{(p-1)\omega_2}{4}(V-1)^2} d\boldsymbol{x} \\
&= \left(\frac{2\pi}{\omega_1}\right)^{p/2} \left(\frac{2\pi}{\omega_1}\right)^{-p/2} \int_{\mathbb{R}^p} e^{-\frac{p\omega_1}{2}M^2} \left[e^{-\frac{(p-1)\omega_1}{4}2V} e^{\frac{(p-1)\omega_1}{4}2V} \right] e^{-\frac{(p-1)\omega_2}{4}(V-1)^2} d\boldsymbol{x} \\
&= \left(\frac{2\pi}{\omega_1}\right)^{p/2} \int_{\mathbb{R}^p} h(\boldsymbol{x}|\omega_1) e^{-\frac{(p-1)\omega_1}{4}\left[\omega_2 (V-1)^2 - 2\omega_1 V\right]} d\boldsymbol{x}.
\end{aligned} \tag{3.17}$$

With respect to $h(\boldsymbol{x}|\omega_1)$, this can now be written as the expected value

$$k(\omega_1,\omega_2) = \left(\frac{2\pi}{\omega_1}\right)^{p/2} \mathbb{E}\left\{ \exp\left(-\frac{(p-1)\omega_1}{4} \left[\omega_2 \left(\frac{U}{(p-1)\omega_1} - 1 \right)^2 - \frac{2}{p-1} U \right] \right) \right\}, \tag{3.18}$$

which can be computed with reasonable efficiency via Monte Carlo, by recalling that $U \sim \chi^2(p-1)$. In the special case where $\omega \equiv \omega_1 = \omega_2$, the equation for the normalizing constant simplifies to

$$\begin{aligned}
&k(\omega,\omega) = \\
&\left(\frac{2\pi}{\omega}\right)^{p/2} \mathbb{E}\Bigg\{ \exp\left(-\frac{\left(U - (p-1)\omega(2-\sqrt{3})\right)\left(U - (p-1)\omega(2+\sqrt{3})\right)}{4(p-1)\omega} \right) \Bigg\}.
\end{aligned} \tag{3.19}$$

Since the function $\left(x - a(2-\sqrt{3})\right)\left(x - a(2+\sqrt{3})\right)$ has a minimum value of $-3a^2$, we can bound $k(\omega,\omega)$ above by

$$k(\omega,\omega) \leq \left(\frac{2\pi}{\omega}\right)^{p/2} \exp\left(\frac{3(p-1)\omega}{4}\right) < \infty.$$

It is trivial to see that $k(\omega,\omega) > 0$, therefore $k(\omega_1,\omega_2)$ is both positive and finite whenever $\omega_1 = \omega_2$. By defining $\omega_{(1)} = \min(\omega_1,\omega_2)$ and $\omega_{(2)} = \max(\omega_1,\omega_2)$, it is easy to see that

$$k_{(1)} \equiv k(\omega_{(2)}, \omega_{(2)}) \leq k(\omega_1, \omega_2) \leq k(\omega_{(1)}, \omega_{(1)}) \equiv k_{(2)},$$

with $k_{(1)} = k(\omega_1, \omega_2) = k_{(2)}$ if and only if $\omega_1 = \omega_2$. Since $k_{(1)}$ and $k_{(2)}$ must be finite and positive, it follows that $k(\omega_1, \omega_2)$ is also finite and positive for all values of $\omega_1, \omega_2 > 0$.

3.4.3 A Rejection Sampler

In this section, we develop a simple rejection sampling scheme for sampling from the MP prior by giving a simple formula for choosing the envelope constant and the optimal variance for the proposal distribution (among a particular class of proposal distributions). Consider the distribution defined in eq. (3.16) as a proposal distribution. To sample $\boldsymbol{x} \sim MP(\omega_1, \omega_2)$ via the Accept-Reject algorithm [29], we need to find a constant c such that $c \geq \frac{\pi_{MP}(\boldsymbol{x})}{h(\boldsymbol{x}|\tau)}$ for all $\boldsymbol{x} \in \mathbb{R}^p$. We begin by writing the ratio

$$\frac{\pi_{MP}(\boldsymbol{x})}{h(\boldsymbol{x}|\tau)} = \frac{1}{k(\omega_1, \omega_2)} \left(\frac{2\pi}{\tau}\right)^{p/2} e^{-\frac{p}{2}(\omega_1 - \tau)M^2} e^{-\frac{(p-1)}{4}\left(\omega_2(V-1)^2 - 2\tau V\right)}.$$

The first exponential term is bounded above by 1, so long as we choose $\tau \leq \omega_1$. Since the function $a(x-1)^2 - 2bx$, for $a, b > 0$, has a minimum value of $\frac{-b(b+2a)}{a}$, the second exponential term can also be bounded above. Together, we have that

$$c_\tau(\omega_1, \omega_2) = \frac{1}{k(\omega_1, \omega_2)} \left(\frac{2\pi}{\tau}\right)^{p/2} \exp\left(\frac{(p-1)\tau(\tau + 2\omega_2)}{4\omega_2}\right), \tag{3.20}$$

is greater than $\frac{\pi_{MP}(\boldsymbol{x})}{h(\boldsymbol{x}|\tau)}$ for all $\boldsymbol{x} \in \mathbb{R}^P$, as long as $\tau \leq \omega_1$. In eq. (3.20), $k(\omega_1, \omega_2)$ can be estimated using eq. (3.18) or replaced with $k_{(1)}$ using eq. (3.19). In either case, $c_\tau(\omega_1, \omega_2)$ can be viewed as an upper bound on the number of samples (from $h(\boldsymbol{x}|\tau)$) required to obtain a single draw from $\pi_{MP}(\boldsymbol{x})$. Alternatively, $k(\omega_1, \omega_2)$ can be ignored altogether (i.e. set to 1), by replacing $\pi_{MP}(\boldsymbol{x})$ with the unnormalized density throughout the algorithm.

The sampling scheme proposed above works for any positive value $\tau < \omega_1$, but the efficiency of the sampler will depend on this choice. For optimal efficiency (with respect to this simple class of proposal distributions), the precision should be chosen by the rule

$$\tau_\star = \arg\min_{\tau \in (0,\omega_1]} \tau^{-p} a^{\tau(\tau+2\omega_2)}, \quad a = e^{(p-1)/(2\omega_2)}.$$

This sampler is quite efficient for moderate values of p, ω_1 and ω_2, but loses efficiency as these values (especially p) become large.

When ω_1 and/or ω_2 is large, the efficiency of the sampler can be improved by expanding the class of proposal distributions, at the cost of added complexity. Consider the proposal distribution $\boldsymbol{x} \sim N(\mathbf{0}, \Sigma)$, with

$$\Sigma = \frac{1}{\tau}\left(\rho J_p + (1-\rho) I_p\right),$$

where J_p is a $p \times p$ matrix of all ones, I_p is the identity matrix and $\rho \in [-\frac{1}{p}, 0]$ is a correlation parameter. A similar derivation to the one above shows that the constant

$$c_{\tau,\rho} = \frac{(2\pi)^{p/2}}{k(\omega_1,\omega_2)} \left(\frac{(1-\rho)^{p-1}((p-1)\rho+1)}{\tau^p}\right)^{1/2} \exp\left(\frac{(p-1)\tau(\tau+2\omega_2(1-\rho)}{4\omega_2(1-\rho)^2}\right), \tag{3.21}$$

is guaranteed to be an upper bound for $\frac{\pi_{MP}(\boldsymbol{x})}{h(\boldsymbol{x}|\tau,\rho)}$, so long as the constraint

$$\frac{\tau}{\rho(p-1)+1} < \omega_1 \tag{3.22}$$

is satisfied. For best results, τ and ρ should be chosen to minimize $c_{\tau,\rho}$ of eq. (3.21) subject to the constraint in eq. (3.22). This added complexity can lead to a more efficient sampler for a wide variety of ω_1 and ω_2 values, but will still be inefficient for large p. For large values of the parameters in high-dimensions, the Z-regularization approach of section 3.4.4 may be preferable.

3.4.4 Moment Penalization in the Limit

The moment penalization prior is able to consider a candidate solution $\boldsymbol{\gamma}$ and reward it for having mean $M_\gamma \approx 0$ and variance $V_\gamma \approx 1$. By increasing ω_1 and ω_2, we can place as much prior density as we desire arbitrarily close to the set

$$\Gamma_{\infty,p} = \{(\gamma_1, \cdots \gamma_p) \mid M_\gamma = 0 \text{ and } V_\gamma = 1\}.$$

When conducting posterior inference, we can make the restriction $\boldsymbol{\gamma} \in \Gamma_\infty$ by sending $\omega_1 \to \infty$ and $\omega_2 \to \infty$, choosing to focus only on solutions which satisfy the mean and variance constraints exactly. In practice however, this is a set of measure zero and the moment penalization prior can only restrict to solutions which nearly, but not exactly, satisfy these constraints. This subsection will focus on an alternative prior distribution for $\boldsymbol{\gamma}$ which places all of its prior density on the set $\Gamma_{\infty,p}$. The following prior specification will be referred to as *Z-Regularization*.

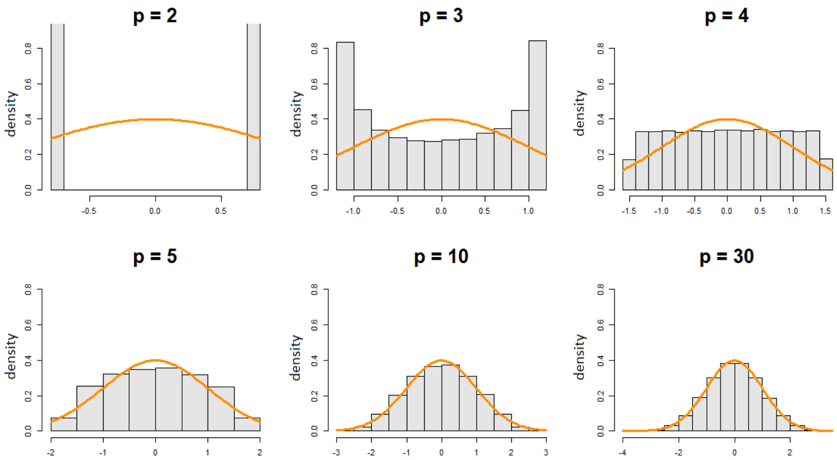


Figure 3.16: Marginal prior (emprical) distribution on each γ_k under *Z*-regularization. Orange curve shows the $N(0, 1)$ distribution for reference.

Consider a set of p latent variables $\boldsymbol{Z}$ such that

$$\begin{aligned} Z_1, \cdots Z_p &\overset{iid}{\sim} N(0,1) \\ \gamma_k &= \frac{Z_k - M_Z}{S_Z}. \end{aligned} \tag{3.23}$$

where M_Z and S_Z are the mean and standard deviation respectively of $Z_1, \cdots Z_p$. By construction, we have $M_\gamma = 0$ and $V_\gamma = 1$ with probability 1. Figure 3.16 shows empirically the marginal prior for each nuisance parameter under Z-regularization. As the number of nuisance parameters p grows large, the induced marginal priors become approximately standard normal.

Relaxing the constraints

The Z-regularization prior, by default, enforces the constraints $M_\gamma = 0$ and $V_\gamma = 1$ almost surely. As an alternative to moment penalization, we can relax the constraints by introducing a *relaxation variable* ζ, with a single *relaxation parameter* σ_R^2. We refer to the following as *relaxed Z-regularization*

$$\begin{aligned} Z_1, \cdots Z_p &\overset{iid}{\sim} N(0,1),\ \zeta \sim N(0, \sigma_R^2) \\ \zeta &\perp\!\!\!\perp Z_j,\ j = 1, \cdots p \\ \gamma_k &= \frac{Z_k - \tilde{M}_Z}{\tilde{S}_Z}, \end{aligned} \tag{3.24}$$

where $\tilde{M}_z = (\sum_{i=1}^p Z_i + \zeta)/p$ and $\tilde{S}_Z^2 = (\sum_{i=1}^p (Z_i - \tilde{M}_Z)^2)/(p - 1 + \sigma_R^2/p)$. Note the correction factor of σ_R^2/p in the denominator of $\tilde{S}_Z^2$ which ensures that $E(\tilde{S}_Z^2) = 1$. By construction, setting the relaxation parameter $\sigma_R^2 = 0$ reduces to the strict Z-regularization discussed in the previous subsection. With respect to the relaxed Z-regularization prior, the induced mean and variance of M_γ (defined in eq. (3.2)) can be written as

$$E(M_\gamma) \approx 0 \qquad \text{and} \qquad \text{Var}(M_\gamma) \approx \frac{\sigma_R^2}{p^2}\,\frac{p - 1 - \sigma_R^2/p}{p - 3 - \sigma_R^2/p},$$

illustrating that the constraints are relaxed monotonically with σ_R^2.

3.4.5 Violations of the Measurement Error Model

Both the probability of prior coherency and the moment penalization prior depend on having strong prior information about the structure of the problem. In particular, we rely heavily on the assumption that a set of nuisance parameters $\gamma_1, \cdots \gamma_p$ are iid with mean μ_0, variance σ_0^2 and distribution f_0. In this section, we consider the potential consequences which can occur when these assumptions are violated.

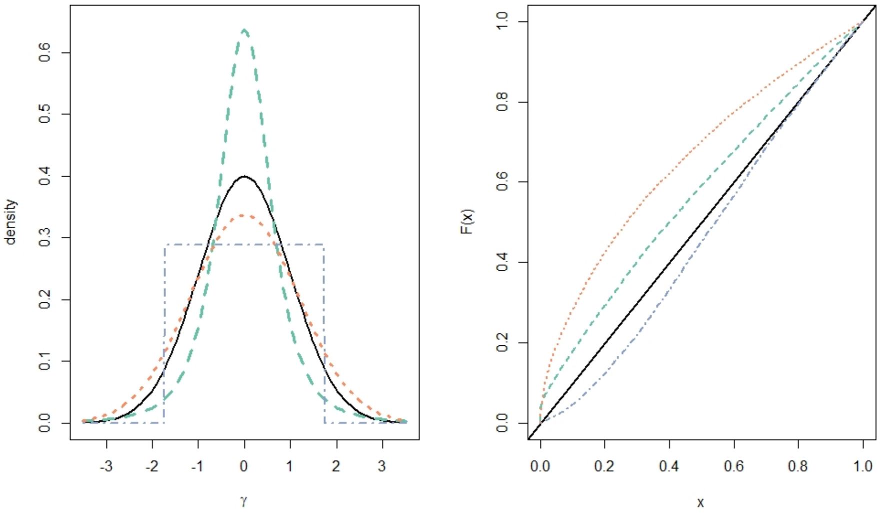


Figure 3.17: The left panel shows four potential "true" models for measurement uncertainties: i) standard normal - solid black, ii) normal with mean 0 and variance 1.4 - dotted orange, iii) scaled t distribution with mean 0, variance 1 and 3 degrees of freedom - dashed green, iv) a uniform distribution with mean 0 and variance 1 - dot/dashed purple. The right panel shows the CDF of p_c (eq. (3.5)) for each of the 4 potential models. Deviation from the identity line indicates sensitivity to violation of the implicit assumptions for p_c.

There are three basic ways in which the measurement error model can be misspecified. First off, there can be a violation if the true mean of the γ_j values is not

μ_0. In the present context, the nuisance parameters refer to measurement errors and the assumption that these are measured without bias is paramount. We believe that the assumption is both necessary and reasonable for many applications. If the nuisance parameters are measured with bias, then unidentifiability will be introduced in a unreconcilable manner.

In the context of this chapter, the second and third assumptions are far more suspect and may be easily violated. The second assumption is that the variance σ^2 is known by an expert to be σ_0^2. In the material science applications of interest, the subject matter expert is typically able to specify σ_0^2 with a high degree of (subjective) confidence. We can envision scenarios however, where the true variance σ^2 differs substantially from the specified value. The third assumption is that the distribution of each γ_j, denoted by f_0, is known. In the main text we focus primarily on the case where the measurement uncertainties are assumed to follow a normal distribution, and we will continue to do so here. Thus we should also explore the effect of heavy (or short) tails as a form of misspecification. In a related note, if there is good reason to specify a distribution for γ_j which is *not* normal, section 3.4.6 shows how the probability of prior coherency can be adjusted to account for this assumption.

Let us assume that a set of nuisance parameters $\boldsymbol{\gamma} = (\gamma_1, \cdots \gamma_p)$ is generated by the following measurement error model

$$\boldsymbol{\gamma} \sim N(\mathbf{0}, I_p).$$

Under this model, the probability of prior coherency, defined in eq. (3.5), will have a uniform distribution on the interval $(0, 1)$. The question we are trying to answer is: *what happens if the true model for the measurement errors differs from our assumptions in some way?* Figure 3.17 shows the CDF of p_c under 4 different "true" measurement error models:

i) standard normal (i.e. model is *correctly* specified) - solid black line

ii) normal with mean 0 and variance 1.4 - dotted orange

iii) scaled t distribution with mean 0, variance 1 and 3 degrees of freedom - dashed green

iv) uniform distribution with mean 0 and variance 1 - dot/dashed purple.

For each case, the sensitivity of p_c to violations of our assumptions can be qualitatively measured by the deviation from the identity line, which represents the cdf of p_c when the model is correctly specified. From this figure, it can be deduced that underestimation of the variance or the tail heaviness leads to a metric (p_c) which is more susceptible to false positives, in the sense that overfitting may be incorrectly identified. The figure also indicates that overestimation of the tail heaviness (or the variance, as we will shortly see) leads to a fairly small loss of *power*, in the sense that we are less likely to identify overfitting when it is present.

To examine these results more thoroughly, let us suppose that the "true" model for the measurement uncertainties follows a *generalized normal distribution* with probability density function

$$f(\gamma \mid \mu, \sigma, \xi) = \frac{\xi}{2a\Gamma(1/\xi)} \exp\left(-\left(\frac{|x|}{a}\right)^{\xi}\right), \; a = \sigma\sqrt{\frac{\Gamma(1/\xi)}{\Gamma(3/\xi)}}.$$

This is a symmetric distribution with mean 0, standard deviation σ and excess kurtosis

$$\kappa = \frac{\Gamma(5/\xi)\Gamma(1/\xi)}{\Gamma(3/\xi)^2} - 3.$$

Special cases of this distribution include the normal distribution ($\xi = 2$), the Laplace distribution ($\xi = 1$) and a uniform distribution from $-a$ to a (as $\xi \to \infty$). When the shape parameter ξ is small ($\xi < 2$), the tails are heavier than that of the normal distribution, and any arbitrarily large kurtosis can be obtained for some small positive ξ. Using this as a "true" generating model for the measurement errors, fig. 3.18 shows the resulting distribution of p_c for various values of κ and σ.

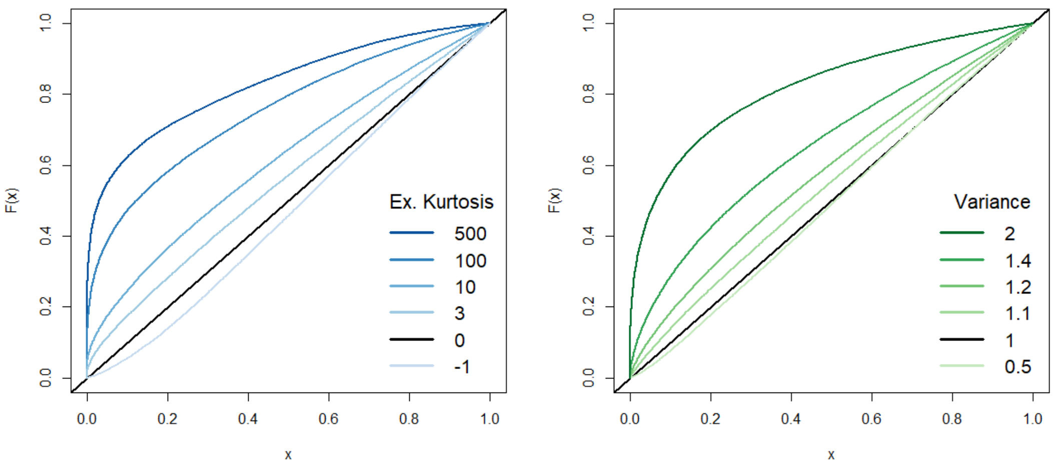


Figure 3.18: This figure shows the CDF of p_c when the “true” model for measurement uncertainty is generalized normal. In the left panel, we fix $\sigma = 1$ and vary ξ to obtain different tail behaviors. Positive excess kurtosis leads to an increase in the false positive rate and negative excess kurtosis leads to a (negligible) loss of power. In the right panel, we fix $\xi = 2$ and vary σ to emulate misspecification of the magnitude of the measurement uncertainties. When $\sigma^2 > 1$ the false positive rate increases and when $\sigma^2 < 1$ there is a (negligible) loss of power.

- **When the form of the distribution is misspecified.** The formula for p_c given in eq. (3.5) implicitly assumes that the form of the measurement model is normal ($\xi = 2$, $\kappa = 0$). Fixing $\sigma = 1$, the left panel of fig. 3.18 shows the empirical CDF of p_c for a collection of distributions with different tail behavior. When $\kappa < 0$ there seems to be a negligible loss of power. As the tails of the distribution become increasingly heavy ($\kappa \to \infty$), the probability of prior coherency will be small with high probability, whether or not overfitting is actually occurring. While this effect can be severe for extreme values of κ, deviation from the expected distribution is fairly small even for moderately large values of kurtosis such as $\kappa = 10$.

- **When the variance of the distribution is misspecified.** When we compute p_c, we assume that an expert is willing to specify the variance of the measurement errors as σ_0^2 (assume $\sigma_0^2 = 1$ without loss of generality). If the true variance, σ^2, is larger than 1, then the false positive rate will increase (possibly drastically). The right panel of fig. 3.18 indicates only slight deviations from the expected distribution so long as $\sigma^2/\sigma_0^2 \overset{\sim}{<} 1.2$. Specification of a variance which *exceeds* the true variance leads to a loss of power, which seems to be almost completely negligible even for $\sigma^2/\sigma_0^2 = \frac{1}{2}$.

Both heavy tails and underestimation of the true variance can lead to false conclusions, producing p_c values which point towards overfitting when no such overfitting is actually occurring. Our analysis indicates that the probability of prior coherency is fairly robust to tail behavior, unless the tails are exceedingly heavy. Prior coherency is less robust to misspecification of the variance, with possibly severe implications when the true variance exceeds the specified variance by more than about 20%. Taking a conservative approach, the consequences of overestimating the variance are much less problematic.

3.4.6 Dealing with Non-normality

So far, we have focused on the case where the nuisance parameters are assumed to be normally distributed, a common assumption when the nuisance parameters represent measurement uncertainty. It is worth considering an extension of the moment penalization prior to instances where the common nuisance prior is not normal. We consider, for a moment, the case where the nuisance parameters have a uniform prior across some interval (a, b). As before, we standardize the parameters so that the prior mean and variance are 0 and 1 respectively.

$$\gamma_k \overset{iid}{\sim} \text{Unif}(-\sqrt{3}, \sqrt{3}),\ \ k = 1, \cdots p.$$

Again we define M_γ and V_γ to be the mean and variance of $\boldsymbol{\gamma}$. If we want to compute the probability of prior coherency (p_c) from Section 3.2, there are a few things we must consider. First, the distribution of M_γ is no longer normal and the distribution of V_γ is no longer chi square. The nuisance mean M_γ now follows a *Bates distribution* [10], with a tractable density function. The variance term V_γ does not follow a well known distribution. For moderately large p, we appeal to the Central Limit Theorem instead. As $p \to \infty$ we have the following, where $\xrightarrow{d}$ represents convergence in distribution.

$$M_\gamma \xrightarrow{d} N\left(0,\ \frac{1}{p}\right) \qquad\qquad V_\gamma \xrightarrow{d} N\left(1,\ \frac{4}{5p}\right).$$

Another problem with moving away from the normality assumption, is that the resulting M_γ and V_γ are no longer independent. Still these quantities are asymptotically uncorrelated and, for large enough p, replacing equation eq. (3.3) with the product of two normal distributions can become a reasonable approximation for the joint distribution of (M_γ, V_γ).

$$\tilde{\pi}_{M_\gamma,V_\gamma}(m,v) = N(m \mid 0, 1/p) \times N(v \mid 1, 4/(5p))$$

By simulating the nuisance parameters independently from the standardized uniform distribution, we can obtain a large number L of draws (m_ℓ, v_ℓ) from the true joint distribution of (M_γ, V_γ). If the approximation is accurate, the resulting p_c values (equation eq. (3.5)) should be uniformly distributed. Using this approach, we find that the joint distribution is reasonably well approximated for about 10 nuisance parameters. The moment penalization prior can be extended to handle uniform distributions by changing $\lambda_2 = \frac{5p\omega_2}{8}$, but the results may be sensitive to p. The approximation can be improved by using a Bates distribution for M_γ, and estimating $Var(V_\gamma)$ via simulation.

3.5 Conclusions & Future Work

In this chapter, we have provided a framework for identification and reduction of overfitting in the context of measurement uncertainty parameters within BMC. We also show how this can be used as a diagnostic tool to validate the posterior inference on physical parameters. When the form of the model discrepancy is unknown, BMC can lead to overfitting, especially when the dimensionality of the nuisance space is large. The probability of prior coherency metric that was introduced in section 3.2.2 is capable of *identifying* a wide range of cases where overfitting of nuisance parameters occurs. Using this criteria, moment penalization can be used to constrain the inference in a reasonable and meaningful way, restricting nuisance parameter solutions to regions of high prior coherency. This prior is flexible enough to roughly mimic the standard informative prior on one hand ($\omega_1 = \omega_2 = 1$) and strictly enforce the mean and variance constraints on the other ($\omega_1 = \omega_2 = \infty$). By varying these parameters, and constructing the diagnostic plot of section 3.2.3, we can *diagnose* exactly how the overfitting is occurring in a given problem.

The ability to diagnose discrepancy assumption violations is important in applications such as, but not limited to, dynamic material property calibration, where the suitable problem structure often exists. Standard methods of model calibration incorporate information on these measurement uncertainties through informative priors. We have shown that the standard assignment of priors for these nuisance parameters ignores valuable information on the expected distribution of posterior estimates. As the use of statistical methods such as BMC continues to expand in these fields, specialized regularization methods such as the one developed here will become essential for robust inference, especially for high dimensional data with a large number of nuisance parameters.

Although our discussion primarily focuses on the context of Bayesian model calibration, future work will involve application of these methods to other frameworks where measurement uncertainties are present. Additional efforts could be geared towards developing specialized regularization methods for BMC when the nuisance parameters lack the probabilistic structure required for moment penalization. We also suspect that probability of prior coherency could be generalized, allowing for the use of higher moments and accounting for the case where the measurement error model has heavy tails. Finally, we suggest that the MP assumptions can be relaxed by using pseudo-Bayesian procedures such as the *c*-posterior methodology of [101] or the modularization posterior described in Chapter 4 and Chapter 5.

Chapter 4

A Modularization Framework for BMC

"The hardest thing of all is to find a black cat in a dark room, especially if there is no cat." – Confucius

4.1 Overview

In the classic Bayesian model calibration framework of Kennedy and O'Hagan [87], it is well understood that the calibration parameters $\boldsymbol{\theta}$ and the model discrepancy $\delta(\cdot)$ are not jointly identifiable. For instance, an example is provided in [3] where credible intervals for both the calibration parameter θ and the discrepancy function $\delta(x)$ fail to capture the true values, yet the credible interval for the true process $\zeta(x)$ captures the true value anyways, with remarkable precision. The authors of [24] provide another example, one that we will revisit shortly, demonstrating that the true value of a calibration parameter θ cannot be recovered for any amount of data, unless the discrepancy function is known to have a particular form. These ideas were formalized in the last decade with a pair of papers by Tuo and Wu [148, 149], which show that the prior distribution for the discrepancy function $\delta(\cdot)$ in BMC becomes a permanent fixture of the posterior distribution for $\boldsymbol{\theta}$, often leading to bizarre estimates for the calibration parameters. They precisely define the "true value" of a calibration parameter in general and show that the BMC procedure generally leads to *inconsistent* estimators of the calibration parameters, providing a rigorous explanation for the earlier results shown by [3, 24, 87] and others. In comparison, they prove that the simple least squares calibration procedure leads to consistent estimators and they propose a new calibration procedure, which they refer to as L_2 calibration and prove that it is consistent and semiparametric efficient. In the defense of BMC, the same authors produced a third paper in 2018 [150] showing that BMC often leads to superior estimation of the physical response $\zeta(\boldsymbol{x})$ and demonstrate that the BMC estimator of $\zeta(\cdot)$ is consistent.

From this discussion, it is clear that the question of which model calibration procedure method should be used depends explicitly on the goal of model calibration. In this chapter, we will show that all of the previously discussed calibration frameworks can fail when calibration parameters have physical interpretations. In other words, the novel work by Tuo and Wu, though important, does not address the problem of unidentifiability in the present context. Thus we propose a new *modularization* framework for BMC and demonstrate its value as a tool for diagnosing the identifiability of physical parameters. The methods and accompanying discussion found in Section 1.4 are crucial to this chapter.

4.2 Calibrating Physical Parameters

In L_2 calibration (see Section 1.4.2), the "purpose" of model calibration is defined to be "that of finding... the parameter value which minimizes the discrepancy between the true process and the computer output under the L_2 norm". It is shown that the L_2 calibration, and not BMC, is able to accomplish this goal. The primary issue is that the "true values" of the calibration parameters are ambiguously defined in BMC. By defining the calibration parameters mathematically, Tuo and Wu are able to produce a calibration procedure which accomplishes their desired goal [148].

In our applications of interest, the goal of model calibration differs from both of these perspectives. We define the purpose of model calibration as that of obtaining accurate and robust estimation (with uncertainty) for a small set of physical parameters, whose true values are of scientific interest. This goal is loftier than either of the previously discussed goals and is challenging or impossible in the general case. In a sense, this objective is similar to the objective of L_2 calibration and can be viewed as trying to minimize an unknown loss function. On the other hand, this is necessary (but not sufficient) for accurate extrapolative predictions. In other words, if the true value of the parameters governing the physical process cannot be accurately estimated, then predictions of the true process beyond the range of observed data cannot be generally trusted.

4.2.1 An Impossible Problem

We begin by defining the objective mathematically. Consider a generic loss function $\mathcal{L}(\boldsymbol{\theta})$ and define

$$\boldsymbol{\theta}_{\mathcal{L}} = \underset{\boldsymbol{\theta}\in\Theta}{\arg\min}\, \mathcal{L}(\boldsymbol{\theta}). \tag{4.1}$$

We will refer to $\boldsymbol{\theta}_{\mathcal{L}}$ as the *optimal value* of $\boldsymbol{\theta}$. Partitioning the calibration parameters as described in section 1.4.3, we write $\boldsymbol{\theta}_{\mathcal{L}} = (\boldsymbol{\alpha}_{\mathcal{L}}, \boldsymbol{\gamma}_{\mathcal{L}})$. We will assume that every calibration procedure is consistent for some loss function $\mathcal{L}$, where consistency here means that the estimator of $\boldsymbol{\theta}$ converges in probability to $\boldsymbol{\theta}_{\mathcal{L}}$ as the design points $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \boldsymbol{x}_n)$ become dense in $\mathcal{X}$. For example, least squares and L_2-calibration both correspond to the L_2 loss function given in eq. (1.21). Now suppose that $\boldsymbol{\alpha}$ has a physical interpretation and some unknown "true" value $\boldsymbol{\alpha}_\star$. In general, there is no reason to suppose that $\boldsymbol{\alpha}_\star = \boldsymbol{\alpha}_\mathcal{L}$, except for the unrealistic case where the model is a perfect representation of the true process. With respect to the loss function $\mathcal{L}$, we define the goal of calibration, in this setting, as that of finding the parameter value

$$\boldsymbol{\theta}_\star = (\boldsymbol{\alpha}_\star, \boldsymbol{\gamma}_{\mathcal{L},\star}), \quad \text{with} \quad \boldsymbol{\gamma}_{\mathcal{L},\star} = \underset{\boldsymbol{\gamma}\in\Gamma}{\arg\min}\, \mathcal{L}(\boldsymbol{\alpha}_\star, \boldsymbol{\gamma}). \tag{4.2}$$

In other words, $\gamma_{\mathcal{L},\star}$ is the best fitting value of the nuisance parameter (with respect to $\mathcal{L}$) which can be found while holding $\boldsymbol{\alpha}$ at its true value. In general, it will be the case that $\boldsymbol{\theta}_\star \neq \boldsymbol{\theta}_\mathcal{L}$ and thus the $\mathcal{L}$-consistent calibration procedure will lead to inconsistent estimation of the true value of the physical parameter. By definition, $\boldsymbol{\theta}_\star$ is a suboptimal solution in the sense that $\mathcal{L}(\boldsymbol{\theta}_\mathcal{L}) < \mathcal{L}(\boldsymbol{\theta}_\star)$ which is precisely what makes this problem so difficult. Searching for an optimal value can be challenging, but at least this defining property allows us to recognize the optimal solution once we have found it. On the other hand, there are infinitely many suboptimal values and it may be impossible to discern which suboptimal point represents divine truth. If estimating $\boldsymbol{\theta}_\mathcal{L}$ is like finding a needle in a haystack, then estimating $\boldsymbol{\theta}_\star$ is like finding a particular needle in a needle factory.

4.2.2 A Simple Machine

The futility of the situation can be made clear with an example. Consider a simple machine which produces *work* as a function of a single input *effort*, denoted by x. The *work* produced by the machine is given by the true process

$$\zeta(x) = \frac{\alpha_\star x}{1 + x/10}, \ 0 \leq x \leq 10. \tag{4.3}$$

The work output of the machine is proportional to the *effort* x put into it, except for a loss of work due to *friction* which is accounted for in the denominator. The *efficiency* of the machine is denoted by $\alpha_\star$ which is a physically meaningful parameter which describes the nature of the physical system and whose value we wish to estimate.

For a particular machine, suppose that the true efficiency is $\alpha_\star = 2$. A physical experiment is conducted and the field data $\boldsymbol{y} = \{y_1, y_2, \cdots y_n\}$ is collected as

$$\begin{aligned} & y_i = \zeta(x_i) + \epsilon_i \\ & \epsilon_i \overset{\text{iid}}{\sim} N\left(0, 0.5^2\right). \end{aligned} \tag{4.4}$$

In the original formulation of this problem [24], a naive simulator was used which ignored friction altogether (eq. (3.10)). Let us assume that the researchers of this machine have recognized the inadequacy of this model, and choose to account for friction using a piecewise linear computer model

$$\eta(x, \boldsymbol{\theta}) = \begin{cases} \alpha x, & x < \gamma \\ \\ \alpha\gamma + \beta(x - \gamma), & x \geq \gamma, \end{cases} \tag{4.5}$$

where $\boldsymbol{\theta} = (\alpha, \beta, \gamma)$ denotes the calibration parameters. This computer model is imperfect, as the mechanism for loss of work due to friction is still not correctly understood. The efficiency parameter α at least partially retains its physical interpretation as the *initial* efficiency of the machine. The other calibration parameters γ and β are not of interest to the researchers and are thus referred to as nuisance parameters.

L_2 Calibration

Following [148, 149] (Section 1.4.2), we start by determining the optimal value of $\boldsymbol{\theta}$ under L_2 loss

$$\begin{aligned} \boldsymbol{\theta}_{L_2} &= \arg\min_{\theta} \int_X (\zeta(x) - \eta(x, \boldsymbol{\theta}))^2 dx \\ &= \arg\min_{\theta} \left\{ \int_0^\gamma \left(\frac{2x}{1 + x/10} - \alpha x \right)^2 dx + \right. \\ & \qquad \left. \int_\gamma^{10} \left(\frac{2x}{1 + x/10} - \alpha\gamma - \beta(x - \gamma) \right)^2 dx \right\}. \end{aligned} \tag{4.6}$$

This minimization problem can be solved numerically with an arbitrary level of accuracy, yielding

$$\boldsymbol{\theta}_{L_2} = (1.596, 0.733, 3.420)$$

Importantly, we note that $\alpha_{L_2} = 1.596$, so that the "optimal value" of the physical parameter, under the specified loss function, differs significantly from the true value $\alpha_\star = 2$.

Fixing α at its true value, we can repeat the conditional minimization problem to obtain

$$\boldsymbol{\theta}_\star = (2.000, 0.813, 1.979).$$

Figure 4.1 illustrates the difference in the simulator evaluated at the optimal value

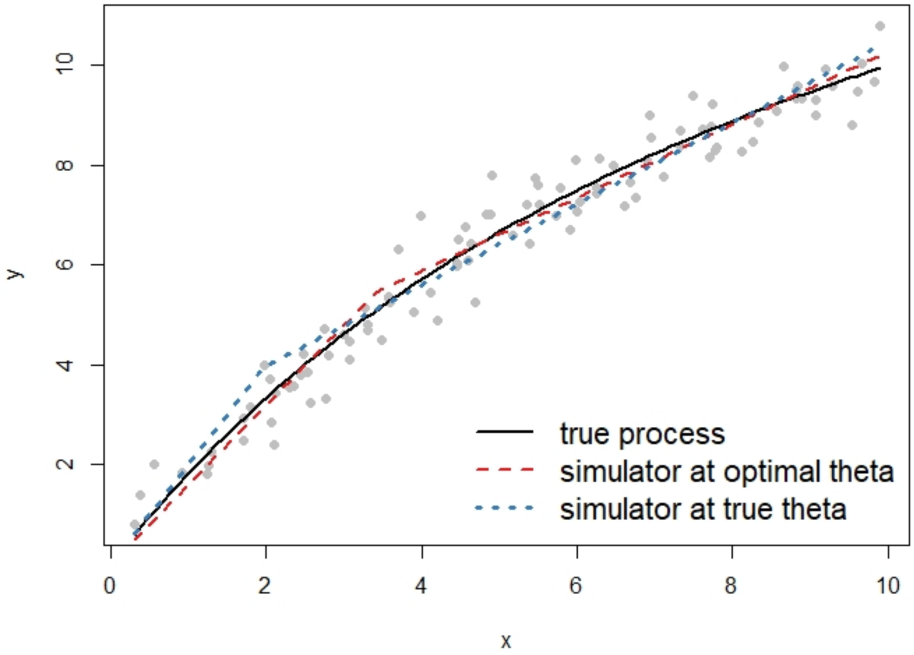


Figure 4.1: Simple machine: Data (circles), true process (solid line) and simulator evaluated at θ_{L_2} (dashed line) as well as $\theta_\star$ (dotted line).

θ_{L_2} (dashed) and at the true value $\theta_\star$ (dotted) .

Bayesian Model Calibration

Following Section 1.4.3, we attack the problem a second time using BMC. It is well known that a poor prior distribution for the model discrepancy can lead to many problems, so to simplify our discussion, the BMC hyperparameters $\boldsymbol{\psi} = (\sigma, \phi, \kappa)$ are set to reasonable starting values. In particular, we set $\sigma = 0.5$ and estimated ϕ and κ by fitting a GP to a large number of samples from the "true discrepancy function"

$$\delta_\star(\cdot) = \zeta(\cdot) - \eta(\cdot, \boldsymbol{\theta}_\star).$$

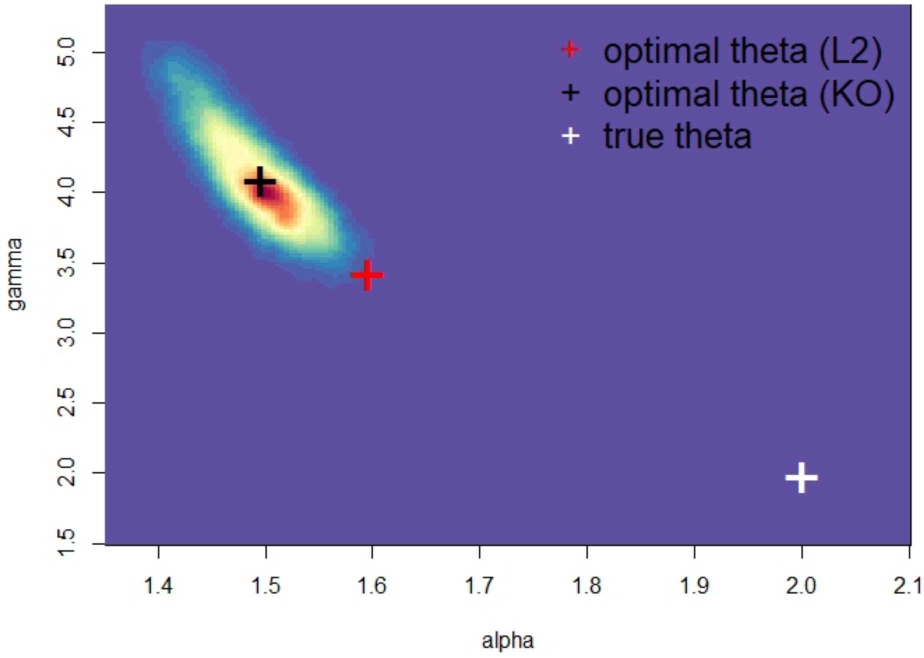


Figure 4.2: Bivariate posterior for (α, γ) under BMC in the simple machine example. The optimal and true values for $\boldsymbol{\theta}$ are also shown. Prior distribution is $\gamma \sim U(0, 10)$.

To complete the model, we specify the prior distributions

$$\begin{aligned} \alpha &\sim \text{Gamma}(6, 3) \\ \beta|\alpha &\sim \text{Unif}(0, \alpha) \\ \gamma &\sim \text{Unif}(A, B) \end{aligned} \tag{4.7}$$

The prior for α was chosen to mimic a reasonably informative prior, satisfying the prior beliefs $E(\alpha) = 2$ and $P(1 \leq \alpha \leq 3) \approx 0.8$. The prior for β describes our newfound knowledge of the physical system that friction leads to a loss in effort. The prior distribution for γ is only partially specified for the sake of discussion, but for now we will assume that $A = 0$ and $B = 10$ indicating that we have very little knowledge regarding the change point of the physical process. Figure 4.2 shows the bivariate posterior distribution of (α, γ) under BMC. It is clear from this figure that neither BMC nor L_2 calibration will be able to correctly estimate the true value $\boldsymbol{\theta}_\star$.

The Role of Prior Information

The parameter α can be interpreted, in both the true process and the simulator, as the *initial* efficiency of the simple machine. Thus the nuisance parameter γ has a physical interpretation as the effort level at which friction begins to "kick in". Note that γ remains meaningless and undefined with respect to the physical system but gains an interpretation in the computer model. Although $\gamma_{L_2} = 3.42$ provides the best fit to the data (under L_2 loss), researchers of the simple machine may be able to detect signs of work lost due to friction for much smaller effort levels. To incorporate this knowledge, they may choose to alter the prior distribution of γ shown in eq. (4.7). Setting $A = 0$ and $B = 2.5$ has a rather drastic effect on the posterior distribution for $\boldsymbol{\theta}$, as shown in Figure 4.3. For a fair comparison, we also repeat the L_2 calibration process using constrained optimization ($0 \leq \gamma \leq 2.5$). The addition of prior information has led to an improvement; the point estimators are closer

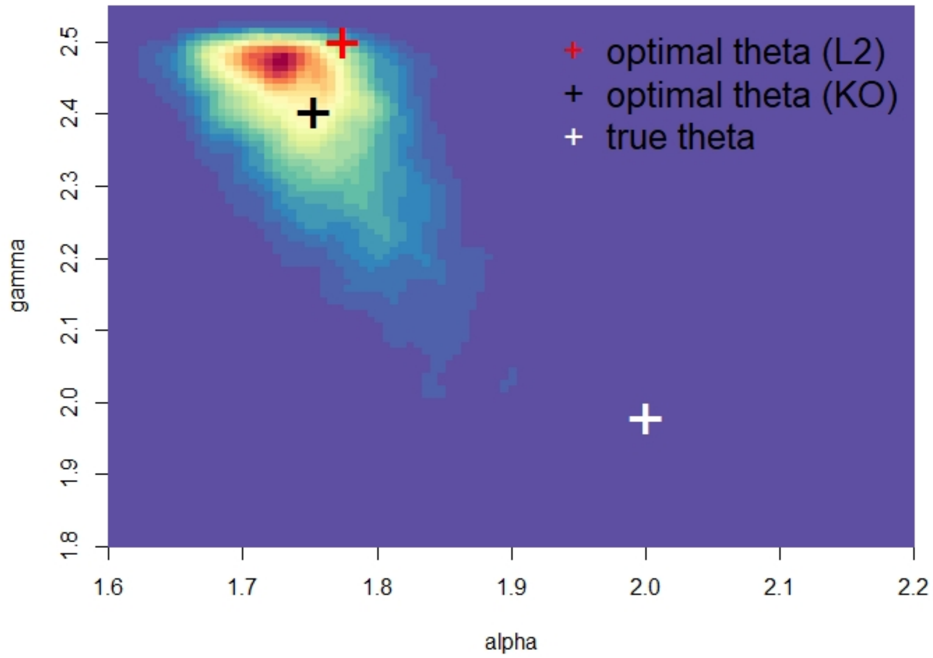


Figure 4.3: Bivariate posterior for (α, γ) under BMC in the simple machine example. The optimal and true values for $\boldsymbol{\theta}$ are also shown. Prior distribution is $\gamma \sim U(0, 2.5)$.

to the true value and the posterior distribution nearly captures the $\boldsymbol{\theta}_\star$ in its tails. Nonetheless, the true efficiency of the machine is being drastically overestimated.

The change point γ is a nuisance parameter with no inherent meaning in the physical system. Its apparent meaning is induced by the physical parameter α and can only be interpreted in the context of the inadequate computer model. Thus, it is unlikely that researchers will be able to hone in on the value $\gamma_{\mathcal{L},\star}$, but it is worth exploring the case anyways. Consider the set

$$\Gamma_\tau = \{\gamma \mid \mathcal{L}(\alpha_\star, \gamma) \leq \mathcal{L}(\boldsymbol{\theta}_\star) + \tau\}, \tag{4.8}$$

which can be interpreted as the set of γ values which lead to nearly minimal loss, conditional on $\alpha = \alpha_\star$. Under mild conditions, we have that $\Gamma_\tau \to \{\gamma_{\mathcal{L},\star}\} = \{1.979\}$

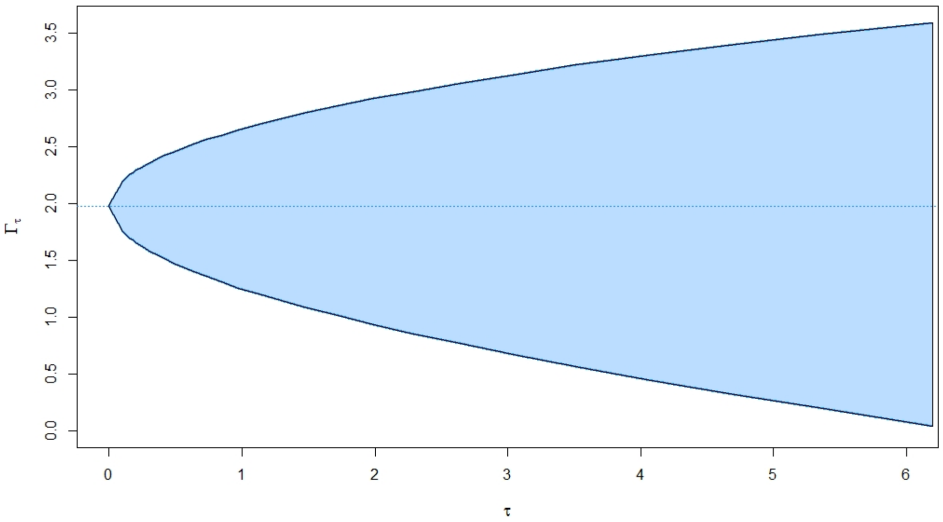


Figure 4.4: The interval Γ_τ as a function of τ.

as $\tau \to 0$ and $\Gamma_\tau \to \Gamma = [0, 10]$ as $\tau \to \infty$. For small values of τ, this set is highly concentrated around the desired value of γ, and thus τ^{-1} can be viewed as a measure of the quantity and/or strength of the researchers prior information. In this simple problem, these sets reduce to intervals $\Gamma_\tau = [A_\tau, B_\tau]$ (see Figure 4.4) and can be used to set the prior distribution for γ in eq. (4.7).

It follows that the prior distribution $\gamma \sim \text{Unif}(A_\tau, B_\tau)$ does lead to better inference for the efficiency parameter α as $\tau \to 0$. The results are summarized in Table 4.1, where the middle columns show the point estimate (posterior mean) and 95% credible interval for α corresponding to different values of τ. Even when τ is very small, the true value $\alpha_\star = 2$ is only barely captured in the credible interval. Thus a tremendous amount of prior information is needed in order to have any hope of recovering α and obtaining this information in the first place is unrealistic due the inadequacy of the computer model.

Table 4.1: Full BMC and BMC with modularization are performed using the prior distribution $\gamma \sim \text{Unif}(A_\tau, B_\tau)$ for several values of τ. For each approach, the point estimate and 95% credible interval for α are reported.

		BMC		Modularization	
τ	Γ_τ	$\hat{\alpha}$	95% *CI*	$\hat{\alpha}$	95% *CI*
0.1	$(1.75, 2.20)$	1.85	$(1.74, 2.00)$	1.89	$(1.75, 2.03)$
0.5	$(1.47, 2.46)$	1.77	$(1.67, 1.93)$	1.91	$(1.70, 2.19)$
1.0	$(1.25, 2.65)$	1.72	$(1.63, 1.83)$	1.93	$(1.66, 2.35)$
2.0	$(0.93, 2.93)$	1.66	$(1.59, 1.74)$	1.99	$(1.62, 2.73)$
6.0	$(0.08, 3.57)$	1.56	$(1.50, 1.50)$	2.77	$(1.54, 10.50)$

4.3 The Modularization Posterior

In the previous section, we saw a simple example which illustrated the challenges of solving an inverse problem when the parameter of interest has a physical interpretation. For nearly any physical system worth modeling, the computer model is destined to be inadequate, leading to the presence of parameters with no context (or altered context) in the model. When this is the case, precise learning about parameters is dangerous because we are likely to learn the wrong value. In this chapter, we would like to adopt the philosophy which can be summarized as

> *"An approximate answer to the right question is worth a great deal more than a precise answer to the wrong question" – John W. Tukey.*

Rather than trying to precisely estimate the wrong quantity, we propose focusing our efforts on estimating the physical parameters of interest while still accounting for the uncertainty of the nuisance parameters.

We assume that the data $\boldsymbol{y} = (y_1, y_2, \cdots y_n)$ is generated from a distribution with parameters $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\gamma}) \in \mathbb{R}^{p+q}$ equipped with the prior distribution $\pi(\boldsymbol{\theta}) = \pi(\boldsymbol{\alpha}, \boldsymbol{\gamma})$.

The fully Bayesian marginal posterior distribution for $\boldsymbol{\alpha}$ is given by

$$\begin{aligned}\pi(\boldsymbol{\alpha}|\boldsymbol{y}) &= \int_{\mathbb{R}^q} \pi(\boldsymbol{\alpha}, \boldsymbol{\gamma}|\boldsymbol{y})d\boldsymbol{\gamma} \\ &= \int_{\mathbb{R}^q} \pi(\boldsymbol{\alpha}|\boldsymbol{\gamma}, \boldsymbol{y})\pi(\boldsymbol{\gamma}|\boldsymbol{y})d\boldsymbol{\gamma}.\end{aligned} \tag{4.9}$$

The second equality implies that the marginal posterior distribution for the parameters of interest can be viewed as the *conditional posterior* of $\boldsymbol{\alpha}$ given $\boldsymbol{\gamma}$, averaged across the marginal posterior of $\boldsymbol{\gamma}$. In this form, we can see explicitly that to learn about the parameters of interest we must also be able to learn the posterior distribution of the nuisance parameters. In the modularization approach to inference, we consider replacing the marginal posterior of $\boldsymbol{\gamma}$ in eq. (4.9) with the marginal prior $\pi(\boldsymbol{\gamma}) = \int_{\mathbb{R}^p} \pi(\boldsymbol{\alpha}, \boldsymbol{\gamma})d\boldsymbol{\alpha}$. This distribution, given by

$$\pi_M(\boldsymbol{\alpha}|\boldsymbol{y}) = \int_{\mathbb{R}^q} \pi(\boldsymbol{\alpha}|\boldsymbol{\gamma}, \boldsymbol{y})\pi(\boldsymbol{\gamma})d\boldsymbol{\gamma}, \tag{4.10}$$

we refer to as the *modularization posterior* of $\boldsymbol{\alpha}$, and $\boldsymbol{\gamma}$ are referred to as *modularization parameters*.

In broad terms, modularization refers to a statistical procedure which is comprised of different modules [83, 94]. We view this as a form of modularization with two modules: (i) the statistical model and (ii) a prior distribution for $\boldsymbol{\gamma}$. In standard Bayesian theory these two modules would be fused as one, but we suggest treating them as separate. Treating $\boldsymbol{\gamma}$ as fixed and known, we can obtain the conditional posterior distribution of $\boldsymbol{\alpha}$ given $\boldsymbol{\gamma}$ (module 1) and then this conditional posterior is averaged across the prior distribution of $\boldsymbol{\gamma}$ (module 2). This can also be viewed as forward uncertainty propagation problem [89], where the response is the distributional solution to the inverse problem.

4.3.1 The Diamond in a Box

Before discussing modularization in the context of Bayesian model calibration, it will be useful and informative to explore the behavior of modularization using a simple tractable example. Thus we will consider a problem described in [94], presented in a new form to facilitate discussion. We call this the Diamond in a Box (DB) problem.

> *The Diamond in a Box Problem.* A valuable diamond weighing α grams is contained in a display case weighing γ grams. The curator has a scale which, when measuring an object weighing x grams, produces an output of $x + \epsilon$ where $\epsilon \sim N(0, \sigma^2)$. The curator takes the diamond out of the case and places it on this scale n_1 times before placing it back in its box. The diamond is very valuable, so it must remain its display case, but the curator can weigh the display case with the diamond inside an additional n_2 times if needed. What is the weight of the diamond?

Without any additional information, we can immediately sense that weighing the box and diamond together is a waste of time. Using the information gained from the first n_1 trials we can obtain a reasonable estimate of the diamonds weight. While the second set of n_2 trials would allow us to estimate the weight of the display case, it provides no additional information regarding the weight of the diamond. Fortunately enough, the curator is also a Bayesian, and she realizes that her situation can be improved with the use of prior information. By extensively measuring a collection of similar display cases, she is able to build an informative prior distribution for γ.

This problem can be stated mathematically as

$$y_i = \begin{cases} \alpha + \epsilon_i, & i = 1, \cdots n_1 \\ \alpha + \gamma + \epsilon_i, & i = n_1 + 1, \cdots n \end{cases} \tag{4.11}$$

$$\epsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2) \qquad \alpha \sim N(0, \sigma_\alpha^2) \qquad \gamma \sim N(0, \sigma_\gamma^2),$$

where $n = n_1 + n_2$ and typically $n_1 \ll n_2$. Although the prior mean for α and γ make little sense in the context of the diamond in a box problem, we choose these values for mathematical simplicity and without loss of generality. Defining $\bar{y}_1 = \left(\sum_{i=1}^{n_1} y_i\right)/n_1$ and $\bar{y}_2 = \left(\sum_{i=1}^{n_2} y_{n+1-i}\right)/n_2$, the marginal posterior distribution for α can be written as

$$\pi(\alpha|\boldsymbol{y}) = \\ N\left(\alpha\Big|\frac{n_1(n_2\sigma_\gamma^2 + \sigma^2)\bar{y}_1 + n_2\sigma^2\bar{y}_2}{(n_1 + \sigma^2/\sigma_\alpha^2)(n_2\sigma_\gamma^2 + \sigma^2) + n_2\sigma^2} \ , \ \frac{\sigma^2(n_2\sigma_\gamma^2 + \sigma^2)}{(n_1 + \sigma^2/\sigma_\alpha^2)(n_2\sigma_\gamma^2 + \sigma^2) + n_2\sigma^2}\right). \tag{4.12}$$

Using eq. (4.10) directly, the modularization posterior distribution for α can be written as

$$\pi_M(\alpha|\boldsymbol{y}) = N\left(\alpha\Big|\frac{n_1\bar{y}_1 + n_2\bar{y}_2}{n_1 + n_2 + \sigma^2/\sigma_\alpha^2} \ , \frac{\sigma^2(n_1 + n_2 + \sigma^2/\sigma_\alpha^2) + n_2^2\sigma_\gamma^2}{(n_1 + n_2 + \sigma^2/\sigma_\alpha^2)^2}\right). \tag{4.13}$$

Although equations 4.12 and 4.13 are a lot to unpack, some useful information can be obtained with careful study. For instance, both distributions reduce to the same conjugate posterior when $n_2 = 0$ [58]. For fixed n_2 and $n_1 \to \infty$, both posteriors converge in probability to the desired parameter α. The other extreme, where n_1 is fixed and $n_2 \to \infty$, is worth noting. Rather than converging to a constant, the posteriors instead converges in distribution to some limiting posterior. In the full Bayes case, this limiting distribution can be written as

$$\lim_{n_2\to\infty} \pi(\alpha|\boldsymbol{y}) = N\left(\alpha \mid \frac{\sigma_\gamma^2 n_1\bar{y}_1 + \sigma^2\bar{y}_2}{\sigma_\gamma^2\left(n_1 + \sigma^2/\sigma_\alpha^2\right) + \sigma^2}, \frac{\sigma_\gamma^2}{1 + (\sigma^2 + n_1)(\sigma\sigma_\gamma)^{-2}}\right). \tag{4.14}$$

The modularization posterior has the limiting posterior

$$\lim_{n_2\to\infty} \pi_M(\alpha|\boldsymbol{y}) = N\left(\alpha \mid \bar{y}_2, \sigma_\gamma^2\right). \tag{4.15}$$

In other words, the modularization posterior makes no attempt to learn about nuisance parameters and any subsequent attempt at correction will be based on the prior for γ. This implies that modularization is reasonable and useful only if the prior distribution is correct and informative. The properties of the modularization posterior, in this simple problem, will be further studied via simulation.

4.3.2 Reliance on the Prior

The investigation in Section 4.3.1 suggests that strong prior information is necessary for the modularization posterior distribution to be helpful. By saying that the prior information for the modularization parameters ($\boldsymbol{\gamma}$) should be strong, we are saying that the uncertainty surrounding these parameters should be relatively small. As seen in eq. (4.10), the modularization posterior starts with the conditional posterior and takes a weighted average across the prior distribution of $\boldsymbol{\gamma}$. If the prior variance of $\boldsymbol{\gamma}$ is large (infinite) then the variance of the modularization posterior will also be large (infinite).

Consider the DB problem again. If n_1 is zero, then α and γ are completely unidentifiable and no amount of data from the second source can help distinguish between the two quantities. When n_1 is non-zero but small, weak identifiability persists and the extent to which the two parameters can be distinguished depends primarily on n_1. The modularization alternative is intriguing here as long as strong prior information is available for the bias γ.

When $n_2 > 0$, the modularization posterior is always more conservative (less precise) than the standard marginal posterior. The level of conservatism depends heavily on the strength of the prior information available for the bias term, as well as the number of observations from the biased source. Figure 4.5 provides a comparison of the two posteriors for a single random sample with $n_1 = 10$ and $n_2 = 90$ where $\alpha = 0$, $\gamma = 0.1$ and the model uncertainty $\sigma = 1$. We set $\sigma_\alpha = \infty$ to represent a flat prior over the parameter of interest, and vary $\sigma_\gamma \in \{0.25,\ 0.5,\ 1,\ 2\}$. As the

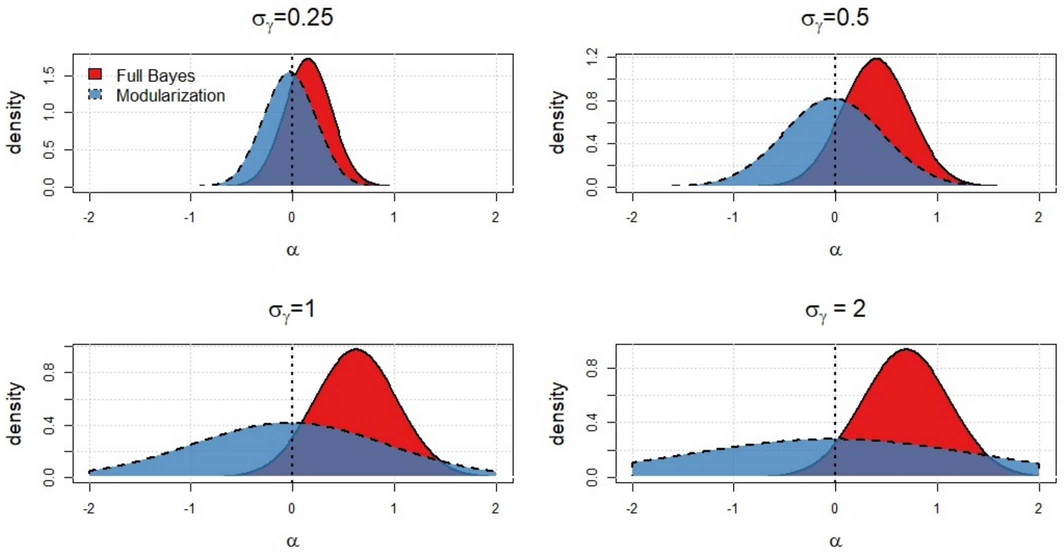


Figure 4.5: Comparison of the fully Bayesian marginal posterior (solid line) and the modularization posterior (dashed line) in the DB-problem for $\sigma_\gamma \in \{0.25,\ 0.5,\ 1,\ 2\}$. Other parameters are fixed to $\alpha = 0$, $\gamma = 0.1$, $\sigma = 1$, $\sigma_\alpha = \infty$.

amount of prior information for the γ increases, or equivalently as σ_γ decreases, the variance of the modularization posterior becomes less conservative compared to the fully Bayesian approach.

At the same time, when the prior uncertainty surrounding γ is large, the modularization posterior can become worthlessly conservative, leading to an increasingly flat posterior distribution for α. In general, if the prior variance of $\boldsymbol{\gamma}$ is large enough, the variance of the modularization posterior for α can exceed the prior variance of α. As our first rule of thumb, the modularization approach to estimation is only practical when strong prior information is available for the modularization parameters $\boldsymbol{\gamma}$.

Secondly, the success of the modularization posterior will depend on the correctness of the nuisance parameter priors. The notion of correctness implies that the nuisance parameters have a true value which, as we saw in the simple machine example at the beginning of this chapter, may not always be the case. In the DB problem however, the reliance of modularization on the prior can be directly explored via simulation. In particular, we will study the consequences of specifying the prior value $\sigma_\gamma = 0.05$ when the true uncertainty, denoted $\sigma_{\gamma,\star}$, is actually significantly larger. For the model in eq. (4.11), we specify the values $n_1 = 10$, $n_2 = 90$, $\sigma = 1$, $\alpha = 0$ and $\sigma_\alpha = \infty$. Although we will assume that $\sigma_\gamma = 0.05$, the true value of γ for each simulation will be sampled from a $N(0, \sigma^2_{\gamma,\star})$ distribution. Ten thousand simulations were performed for several values of $\sigma_{\gamma,\star}$ between 0 and 6. In each simulation, a 90% credible interval for α is constructed using (i) the marginal posterior eq. (4.12), (ii) the modularization posterior eq. (4.13) and (iii) a *partial posterior* approach. The partial posterior approach is discussed in [94], where it is shown to have better MSE properties than the Bayesian alternative whenever $\sigma^2_{\gamma,\star} > 2\sigma^2_\gamma + \frac{1}{n_1} + \frac{1}{n_2}$. This approach simply discards the n_2 measurements of the diamond inside the display case, based on the belief that there is often little to gain via inclusion of the biased data when the prior is correct, and much to lose when it is not. The results of this simulation study are summarized in Figure 4.6, which shows that the marginal posterior and (especially) the modularization posterior behave poorly when prior information is faulty.

To summarize, if the prior information on the modularization parameters is suspect, then modularization cannot be trusted to give a meaningful answer. Modularization only makes sense when the prior information is strong, and there is good reason to believe that the prior is reliable.

4.3.3 The Effect of Model Inadequacy

It is clear that the modularization approach should only be used for inference on $\boldsymbol{\alpha}$ if there is a set of parameters $\boldsymbol{\gamma}$ such that (i) we have reason to trust our prior

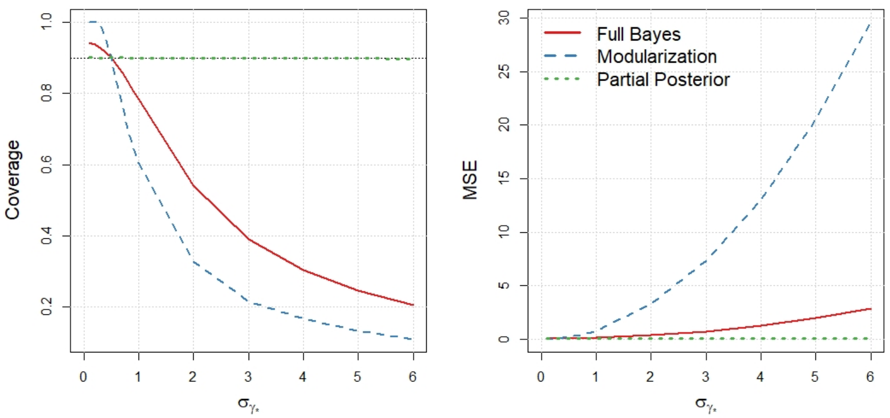


Figure 4.6: Comparison of the marginal posterior (solid), the modularization posterior (dashed) and the partial posterior (dotted) for the DB problem for $\sigma_{\gamma_\star} \in [0.1, 6]$. Other parameters are fixed to $n_1 = 10$, $n_2 = 90$, $\alpha = 0$, $\sigma = 1$, $\sigma_\gamma = 0.5$, $\sigma_\alpha = \infty$.

information and (ii) the prior uncertainty for these parameters is reasonably small. By choosing the modularization framework, we are opting to aggregate the posterior distributions corresponding to a collection of $\boldsymbol{\gamma}$ values which are deemed likely or reasonable according to the prior. With the aforementioned caveats in mind, this often leads to favorable properties such as improved estimation (in terms of MSE) and uncertainty quantification (in terms of empirical coverage). In a model calibration context, the discrepancy function is often the primary source of trouble [3, 24, 117]. To mimic the case of a misspecified model for the diamond in a box problem, we study the consequences of underestimating the variance of the observations.

We fix the values $n_1 = 10$, $n_2 = 90$, $\sigma_\gamma = 0.5$, $\alpha = 0$ and $\sigma_\alpha = \infty$. Although we will assume that $\sigma = 1$ is known, we set the true value of the observational variance to $\sigma_\star$ and perform ten thousand simulations for each of ten different observational variances between 0.1^2 and 6^2.

The results are displayed in Figure 4.7. The empirical coverage of the full Bayesian

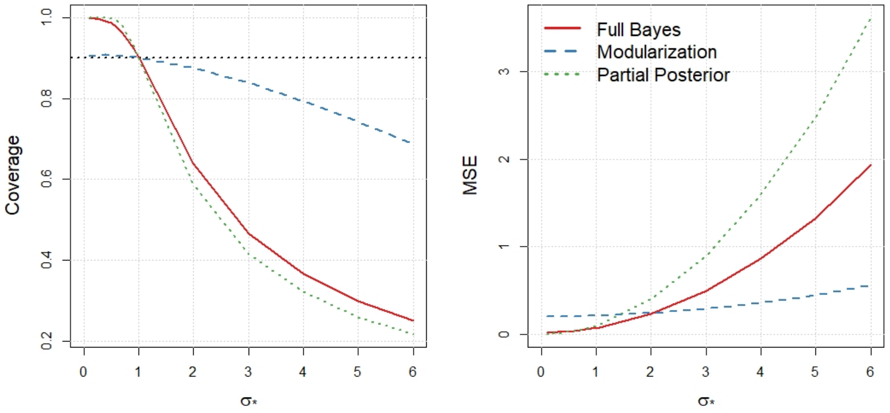


Figure 4.7: Comparison of the marginal posterior (solid), the modularization posterior (dashed) and the partial posterior (dotted) for the DB problem for $\sigma_\star \in [0.1, 6]$. Other parameters are fixed to $n_1 = 10$, $n_2 = 90$, $\alpha = 0$, $\sigma = 1$, $\sigma_\gamma = 0.5$, $\sigma_\alpha = \infty$.

and partial posterior methods drop below the nominal value as soon as $\sigma_\star > \sigma$, and they drop to less than 50% empirical coverage before $\sigma_\star$ has even reached 3σ. For the same observational variance of 3σ, the modularization posterior maintains an empirical coverage of 84%. In terms of MSE, the modularization approach surpasses the other two approaches for $\sigma_\star \gtrapprox 2\sigma$. We can also look at different forms of model inadequacy, replacing the normal measurement error terms with scaled t distributions with 2 degrees of freedom, the modularization approach outperforms the others in terms of coverage and MSE whenever the scale parameter $\sigma_\star > 0.5$.

4.4 Numerical Approximations to the Modularization Posterior

In most practical applications, especially those requiring BMC, a closed form solution for the modularization posterior (such as in eq. (4.13)) is typically unavailable.

Therefore, an algorithm will be needed to provide a numerical approximation to the distribution. In this section, we discuss several methods for approximating eq. (4.10) including our own proposal which we call *emulating the conditional posterior*, or the ECP algorithm. If we assume that MCMC is costly for the application, as will typically be the case in model calibration problems, then the ECP algorithm is a massive improvement over the simple Monte Carlo alternative.

Before we move on to discussing these approaches, we note that a similar approach was examined in [21], where the authors employed a Gibbs sampler with Metropolis steps to sample from the posterior. In Gibbs sampling, parameters are sampled iteratively from their conditional posteriors, avoiding the need to sample from the full joint distribution. The authors of [21] adopt a position of ignorance regarding the nuisance parameters by sampling each γ_i from its prior distribution, rather than from its conditional posterior. The issue with this approach, demonstrated by [118], is that there is no well-defined limiting distribution. In other words, the strategy does not generally lead to the modularization posterior distribution defined in eq. (4.10).

The Problem

Consider the set of calibration parameters $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta})$ where

i) $\boldsymbol{\alpha} \in A \subset \mathbb{R}^p$ denotes a set of physical parameters whose values are of scientific interest,

ii) $\boldsymbol{\gamma} \in \Gamma \subset \mathbb{R}^q$ denotes the set of modularization parameters, i.e. the nuisance parameters whose prior distributions we seek to modularize over,

iii) $\boldsymbol{\beta} \in B \subset \mathbb{R}^r$ denotes a set of parameters which are not included in the modularization set, but whose values are not of primary scientific interest.

The goal is to obtain M samples from the modularization posterior of $\boldsymbol{\alpha}$ with respect to $\boldsymbol{\gamma}$. For clarity, this involves first finding the modularization posterior of $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ with respect to $\boldsymbol{\gamma}$ and then integrating

$$\pi_M(\boldsymbol{\alpha}|\boldsymbol{y}) = \int_B \pi_M(\boldsymbol{\alpha}, \boldsymbol{\beta}|\boldsymbol{y})d\boldsymbol{\beta},$$

to isolate $\boldsymbol{\alpha}$. This is an important practical consideration and thus worth mentioning, but we will typically ignore the possible inclusion of $\boldsymbol{\beta}$ in this section to simplify notation.

In general, we will assume that obtaining samples from the full Bayes posterior $\pi(\boldsymbol{\alpha}|\boldsymbol{y})$ or the conditional posterior $\pi(\boldsymbol{\alpha}|\boldsymbol{\gamma}, \boldsymbol{y})$ requires MCMC and may be a time-consuming bottleneck (see Section 1.5.2, [63, 161]). Thus we will allow these distributions to be sampled from just L times and we refer to L as the *budget*.

4.4.1 A Monte Carlo Algorithm

The modularization posterior distribution in eq. (4.10) can be viewed as the conditional posterior distribution of $\boldsymbol{\alpha}$ given $\boldsymbol{\gamma}$, averaged over the prior distribution of $\boldsymbol{\gamma}$. The Monte Carlo (MC) algorithm works by sampling $\boldsymbol{\gamma}^\ell$ from the prior distribution and then, treating this value as fixed and known, performing MCMC to obtain m_ℓ samples from the conditional posterior $\pi(\boldsymbol{\alpha}\ |\boldsymbol{\gamma}^\ell, \boldsymbol{y})$. This process is then repeated L times, and the combined $M = \sum_{\ell=0}^{L} m_\ell$ samples can be treated as M approximate draws from the modularization posterior.

The samples generated by the MC algorithm can be considered exact draws from the modularization posterior when $m_\ell = 1$, but the budget L is often far smaller than the number of desired samples M, rendering this solution infeasible. For fixed $m_\ell \geq 2$, the samples can still be viewed as approximate samples from the modularization posterior but the quality of the approximation is heavily dependent on the size of the budget.

4.4.2 Emulating the Conditional Posterior

The MC implementation described above is convenient for its simplicity, but when sampling from the conditional posterior (i.e. with MCMC) is expensive, the budget required for an accurate approximation may be too large. Rather than using the budget to directly obtain samples from the modularization posterior, we propose an algorithm which builds a parametric model for the structure of the conditional posterior of $\boldsymbol{\alpha}$ given $\boldsymbol{\gamma}$. In many cases, this structure can be learned accurately and with a limited budget, facilitating fast approximate sampling from the modularization posterior. We call this method emulating the conditional posterior, or ECP.

The Univariate ECP Algorithm

We begin by assuming that $\alpha \in A \subset \mathbb{R}$. As an initial example, suppose that the conditional posterior of α given $\boldsymbol{\gamma}$ can be approximated by

$$\pi(\alpha|\boldsymbol{\gamma}, \boldsymbol{y}) = N\left(\alpha|\mu(\boldsymbol{\gamma}), \sigma(\boldsymbol{\gamma})^2\right), \tag{4.16}$$

for any $\boldsymbol{\gamma} \in \Gamma$. If this assumption holds, and if the functions $\mu(\boldsymbol{\gamma})$ and $\sigma(\boldsymbol{\gamma})$ are known, then samples can be obtained from the modularization posterior efficiently and without the need for MCMC by sampling

$$\begin{aligned} \boldsymbol{\gamma}^m &\sim \pi_\gamma(\cdot) \\ \alpha^m|(\boldsymbol{\gamma}^m, \boldsymbol{y}) &\sim N\left(\mu(\boldsymbol{\gamma}^m), \sigma(\boldsymbol{\gamma}^m)^2\right), \end{aligned} \tag{4.17}$$

for $m = 1, 2, \cdots M$. If all assumptions hold, the samples $\alpha^1, \alpha^2, \cdots \alpha^M$ can be viewed as M exact draws from the modularization posterior. For instance, the DB-problem shown in eq. (4.11) satisfies these assumptions with $\mu(\gamma) = a + b\gamma$ and $\sigma(\gamma) = c$ for appropriate constants a, b, c (these constants are tractable, but not shown here for the sake of brevity). Unfortunately, in most practical applications these functions will not be tractable and thus the appropriate structure must be learned. In the

ECP algorithm, the MCMC budget is used to learn about this hidden structure as well as possible, and then approximate samples are taken from π_M using eq. (4.17).

The assumption of normality is probably the most reasonable default parametric assumption in general, due to limit results such as the Bernstein von-Mises theorem [52], but we will not limit ourselves to this choice. We can assume that

$$\alpha|(\boldsymbol{\gamma}, \boldsymbol{y}) \sim \mathcal{F}\left(\psi_1(\boldsymbol{\gamma}), \psi_2(\boldsymbol{\gamma}), \cdots \psi_r(\boldsymbol{\gamma})\right), \tag{4.18}$$

and write

$$\pi(\alpha|\boldsymbol{\gamma}, \boldsymbol{y}) = \mathcal{F}\left(\alpha\middle|\psi_1(\boldsymbol{\gamma}), \psi_2(\boldsymbol{\gamma}), \cdots \psi_r(\boldsymbol{\gamma})\right), \tag{4.19}$$

where $\mathcal{F}$ characterizes any distribution with the properties (i) it is easy to sample from $\mathcal{F}$ (conditional on the parameters $\psi_1, \cdots \psi_r$) and (ii) given a set of iid observations from $\mathcal{F}$, the parameters $\psi_1, \cdots \psi_r$ can be consistently estimated with reasonable accuracy and efficiency.

If eq. (4.18) holds and if the structure of $\psi_j(\boldsymbol{\gamma})$ is known, then we can sample exactly from the modularization posterior using

$$\begin{aligned} \boldsymbol{\gamma}^m &\sim \pi_\gamma(\cdot) \\ \alpha^m|(\boldsymbol{\gamma}^m, \boldsymbol{y}) &\sim \mathcal{F}\left(\psi_1(\boldsymbol{\gamma}^m), \psi_2(\boldsymbol{\gamma}^m), \cdots \psi_r(\boldsymbol{\gamma}^m)\right), \end{aligned} \tag{4.20}$$

In practice, we will replace each of the unknown $\psi_j(\cdot)$ functions in eq. (4.20) with an estimator $\hat{\psi}_j(\cdot)$, with the justification that $\hat{\psi}_j(\boldsymbol{\gamma}) \to \psi_j(\boldsymbol{\gamma})$ for every $\boldsymbol{\gamma} \in \Gamma$ as $L \to \infty$. Leveraging the continuity of these functions, the structure with respect to $\boldsymbol{\gamma}$ can be modeled as a Gaussian process (see Section 1.3.2, [9,124,125]). In summary, the ECP algorithm works as follows.

1. Sample $\boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2, \cdots \boldsymbol{\gamma}^L$ from the prior distribution $\pi_\gamma(\cdot)$. *Note: For improved performance, consider instead taking a Latin hypercube sample with respect to the prior (see Section 1.5.1).*

2. For $\ell = 1, 2, \cdots L$
   - a. Perform MCMC to obtain T draws $(\alpha^1, \alpha^2, \cdots \alpha^T)$ from the conditional posterior $\pi(\alpha|\boldsymbol{\gamma}^\ell, \boldsymbol{y})$.
   - b. Use $(\alpha^1\alpha^2, \cdots \alpha^T)$ to obtain estimates $\widehat{\psi_1(\boldsymbol{\gamma}^\ell)}, \widehat{\psi_2(\boldsymbol{\gamma}^\ell)}, \cdots \widehat{\psi_r(\boldsymbol{\gamma}^\ell)}$. Write $\widehat{\psi_j^\ell} = \widehat{\psi_j(\boldsymbol{\gamma}^\ell)}$ for simplicity.
3. For $j = 1, 2 \cdots r$
   - a. Use the tuples $(\boldsymbol{\gamma}_\ell, \widehat{\psi_j^\ell})$, $\ell = 1, 2, \cdots L$ to train a Gaussian process $\hat{\psi}_j(\cdot)$.
4. For $m = 1, 2, \cdots M$
   - a. Sample $\boldsymbol{\gamma}_{\text{new}}^m$ from the prior distribution $\pi_\gamma(\cdot)$.
   - b. Set $\tilde{\psi}_j^m = E(\hat{\psi}_j(\boldsymbol{\gamma}_{\text{new}}^m))$ for $j = 1, 2, \cdots r$
   - c. Sample $\alpha^m \sim \mathcal{F}(\tilde{\psi}_1^m, \tilde{\psi}_2^m, \cdots \tilde{\psi}_r^m)$

As $L, T \to \infty$, the samples produced by this procedure can be considered exact draws from the modularization posterior, so long as eq. (4.18) holds. The convergence is a straightforward application of Slutsky's theorem [28]. For finite L and T, this is only an approximation of the desired distribution but we will shortly demonstrate that the approximation is typically much more accurate than the MC algorithm given similar computation time. For nuisance parameter sets of small to moderate dimension the ECP algorithm is quite efficient, but it may struggle in higher dimensions. We advise that at least $L = 2^{q+2}$ MCMC evaluations should be budgeted for reliable results, where q is the dimension of $\boldsymbol{\gamma}$ [43].

A Multivariate ECP Algorithm

The univariate ECP algorithm can be naturally extended to the case where $\boldsymbol{\alpha} \in A \subset \mathbb{R}^p$. The unifying feature is the parametric distribution $\mathcal{F}$ which is specified for the conditional posterior $\pi(\cdot|\boldsymbol{\gamma}, \boldsymbol{y})$. Although any distribution with the aforementioned properties can be a valid choice for $\mathcal{F}$, we will limit our discussion to the multivariate normal distribution. In particular, we assume that

$$\begin{aligned}
&\alpha_i \sim N\left(\mu_i(\boldsymbol{\gamma}), \sigma_i^2(\boldsymbol{\gamma})\right), \ i = 1, 2, \cdots p \\
&\text{Cov}(\alpha_i, \alpha_j) = \sigma_{ij}(\boldsymbol{\gamma}), \ 1 \leq i < j \leq p.
\end{aligned} \tag{4.21}$$

This implies that there are $r = 2p + \binom{p}{2}$ total $\psi_j(\cdot)$ functions which must be learned. When the dimension of $\boldsymbol{\alpha}$ is a small value, i.e. $p = 2, 3, 4$, then this may be feasible. For larger sets of physical parameters, a preposterior sensitivity study may be useful for obtaining a sparse set of parameters [4, 53].

A Sequential ECP Algorithm

A common idea for improving performance in the design of computer experiments, is to sequentialize tasks in order to maximize the use of precious and limited computational resources. Specific applications include (i) estimation of percentiles [30, 131], (ii) contour estimation and optimization [69, 121, 123], (iii) multi-fidelity modeling [62, 163] and (iv) *in situ* applications [104]. The idea is simply that certain regions of the parameter space Γ may be easier to learn about than others, and our budget can be more spent more effectively by choosing $\boldsymbol{\gamma}_\ell$ to be the location which will add the most information. This notion of equitable spending is worth discussing for ECP, although we note that the non-sequential version of our algorithm is faster (less overhead) and sufficient for every application we have considered. In higher dimensional problems however, this extension may be valuable.

In the first step of the ECP algorithm, the set of modularization *locations* $(\boldsymbol{\gamma}^1, \cdots \boldsymbol{\gamma}^L)$ are chosen simultaneously with respect to the prior. We want to transition to the case where $\boldsymbol{\gamma}^\ell$ is not chosen until the previous $\ell - 1$ locations have been assessed. We begin with a *build phase*, in which the first three steps of ECP are executed but using L_0 in place of L (with $L_0 < L$).

Once the build phase is complete, the remaining budget is $L - L_0$ and the goal is to choose the remaining locations sequentially so that the Gaussian process emulators $\hat{\psi}_1(\cdot|\mathcal{D}), \cdots \hat{\psi}_r(\cdot|\mathcal{D})$ can be learned for all reasonable values of $\boldsymbol{\gamma}$ as efficiently as possible. Following [121, 123, 131], we will define an *improvement function* $I(\boldsymbol{\gamma}|\mathcal{D})$ which is used to evaluate candidate locations, selecting the ℓ^{th} location as

$$\boldsymbol{\gamma}_\ell = \arg\max_{\boldsymbol{\gamma}\in\Gamma} I\left(\boldsymbol{\gamma}|\mathcal{D}_{\ell-1}\right), \tag{4.22}$$

where $\mathcal{D}_\ell$ represents all of the relevant information we have acquired up to time ℓ. Before defining the improvement function for sequential ECP, we should discuss a few of the associated challenges.

i) If the location space Γ is bounded, then it may be possible to search the entire space as in eq. (4.22). When the space is unbounded finding the best location becomes more difficult. The prior distribution $\pi_\gamma(\cdot)$ needs to be accounted for in some way, to ensure that we are not wasting resources exploring locations in Γ which are not "reasonable" with respect to the prior. One simple way to handle this problem is to replace the optimization space Γ with a finite set of *candidate values* Γ_c which contains N_c random draws from the prior distribution $\pi_\gamma(\cdot)$. If N_c is large enough, the chosen location can be expected to correspond to good improvement, while being constrained to a region of prior plausibility.

ii) It seems reasonable that the new modularization parameter should be placed in a location which reduces the uncertainty in the GP emulators $\hat{\psi}_j(\cdot)$. It may be the case that a value which maximizes the improvement for the j^{th} GP has little value with respect to the k^{th} GP. We will need a way of evaluating a location with respect to each parameter of the conditional posterior. Inspired by [131], we propose looking at the u-quantile of the conditional posterior $\pi(\alpha|\boldsymbol{\gamma}, \boldsymbol{y}) = \mathcal{F}(\alpha|\psi_1(\boldsymbol{\gamma}), \cdots \psi_r(\boldsymbol{\gamma})$. First of all, for most choices of u, this is guaranteed to be a meaningful combination of the r ψ_j parameters which gives us a way to evaluate the improvement with respect to each GP simultaneously. Secondly, by targeting a particular quantile of the conditional posterior, we are in fact targeting a tail of the modularization posterior. For physical parameters especially, a credible interval is typically more valuable than a point estimate, making this an enticing option.

In the univariate case, $A \subset \mathbb{R}$, we propose the improvement function

$$
\begin{aligned}
I(\boldsymbol{\gamma}|\mathcal{D}) &= \mathrm{Var}\Big(\hat{\zeta}_u(\boldsymbol{\gamma}|\mathcal{D})\Big) && (4.23)\\
&= \mathrm{Var}\Big(\mathcal{Q}\left(u\big|\hat{\psi}_1(\boldsymbol{\gamma}|\mathcal{D}), \psi_2(\boldsymbol{\gamma}|\mathcal{D}), \cdots \psi_r(\boldsymbol{\gamma}|\mathcal{D})\big)\right)\Big) && (4.24)
\end{aligned}
$$

where $\hat{\zeta}_u(\boldsymbol{\gamma})$ is the u^{th} percentile of $\mathcal{F}(\alpha|\psi_1(\boldsymbol{\gamma}), \cdots \psi_1(\boldsymbol{\gamma}))$ and $\mathcal{Q}$ is the quantile function corresponding to the distribution $\mathcal{F}$. The inclusion of $\hat{\ }$ symbols and $\mathcal{D}$ in the notation serves as a reminder that we are estimating these quantities using all available information $\mathcal{D}$.

For instance, in the case where $\mathcal{F}$ corresponds to a univariate normal distribution, the u-quantile can be written as

$$
\zeta_u(\boldsymbol{\gamma}) = \mu(\boldsymbol{\gamma}) + \Phi^{-1}(u)\sigma(\boldsymbol{\gamma}), \tag{4.25}
$$

and the improvement function in eq. (4.23) reduces to

$$
I_{\text{norm}}(\boldsymbol{\gamma}|\mathcal{D}) = \mathrm{Var}\left(\hat{\mu}(\boldsymbol{\gamma})|\mathcal{D}\right) + \left(\Phi^{-1}(u)\right)^2 \mathrm{Var}\left(\hat{\sigma}(\boldsymbol{\gamma})|\mathcal{D}\right), \tag{4.26}
$$

There may be cases where a normal distribution assumption for $\mathcal{F}$ is inappropriate, and another parametric form is desired. For instance, if expert opinion dictates that the support of α is bounded on an interval then a Beta distribution may be specified for $\mathcal{F}$. Similarly, if the parameter of interest should be strictly positive, then perhaps a Weibull or Gamma distribution should be specified. We note that

MC can always be used to approximate the improvement function. If the quantile function $\mathcal{Q}$ can be written in closed form, then analytic calculations can be used to produce an approximation to eq. (4.23). Two examples are given below.

Weibull Distribution

Suppose that

$$\mathcal{F}\left(\alpha \middle| \lambda(\boldsymbol{\gamma}), \kappa(\boldsymbol{\gamma})\right) = \frac{\alpha^{\kappa^{-1}(\boldsymbol{\gamma})-1}}{\kappa(\boldsymbol{\gamma})\lambda(\boldsymbol{\gamma})^{\kappa^{-1}(\boldsymbol{\gamma})}} \exp\left\{-\left(\alpha/\lambda(\boldsymbol{\gamma})\right)^{\kappa^{-1}(\boldsymbol{\gamma})}\right\}.$$

For notational compactness, we define $\hat{\psi} = \mathrm{E}\left(\hat{\psi}(\boldsymbol{\gamma}|\mathcal{D})\right)$ and $\tilde{\psi} = \mathrm{Var}\left(\hat{\psi}(\boldsymbol{\gamma}|\mathcal{D})\right)$ (for $\psi = \lambda, \kappa$). Using the Delta method with a second order expansion we obtain

$$I_{\text{Weibull}}(\boldsymbol{\gamma}|\mathcal{D}) = u_\star^{2\hat{\kappa}} \left\{\tilde{\lambda}\left(1 + 2\tilde{\kappa}\log^2(u_\star)\right) + \hat{\lambda}^2\tilde{\kappa}\log^2(u_\star)\left(1 - \frac{\tilde{\kappa}}{4}\log^2(u_\star)\right)\right\} \tag{4.27}$$

with $u_\star = -\log(1-u)$,

Multivariate Normal Distribution

Returning to the multivariate case ($\boldsymbol{\alpha} \in A \subset \mathbb{R}^p$), we note that eq. (4.23) is no longer well defined. Since there are multiple α_i parameters, there is no clear definition for $\zeta_u(\boldsymbol{\gamma})$. Instead, we consider a linear combination of the components

$$\alpha_0 = \sum_{i=1}^{p} t_i \alpha_i. \tag{4.28}$$

The t_i coefficients are constants, which can be fixed at one for simplicity or taken to be the square root of the marginal prior precision for α_i so that each term is weighted equally according to the prior. Now the u^{th} percentile of α_0 is a meaningful function of every parameter in $\mathcal{F}$,

$$\zeta_u(\boldsymbol{\gamma}) = \sum_{i=1}^{p} t_i\mu_i(\boldsymbol{\gamma}) + \Phi^{-1}(u)\left\{\sum_{i=1}^{p} t_i^2\sigma_i(\boldsymbol{\gamma})^2 + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} t_i t_j \sigma_{ij}(\boldsymbol{\gamma})\right\}^{1/2}. \tag{4.29}$$

The improvement function of eq. (4.23) can now be used for this quantity. Due to the square root, $\zeta_u(\boldsymbol{\gamma})$ is no longer a linear combination of independent GPs, and therefore finding $\mathrm{Var}(\hat{\zeta}_u(\boldsymbol{\gamma}|\mathcal{D}))$ is no longer straightforward. Approximation of this variance via Monte Carlo integration is still a valid option but will lead to slower evaluation of $I(\boldsymbol{\gamma})$ (especially if p is large). Alternatively, we can apply the Delta method with a second order expansion. As before, we define $\hat{\psi} = \mathrm{E}\left(\psi(\boldsymbol{\gamma}|\mathcal{D})\right)$ and $\tilde{\psi} = \mathrm{Var}\left(\psi(\boldsymbol{\gamma}|\mathcal{D})\right)$.

$$
\begin{aligned}
I_p(\boldsymbol{\gamma}|\mathcal{D}) = &\left[E(\delta_1^2) + \Phi^{-1}(u)^2 E(\delta_2) + 2\Phi^{-1}(u)E(\delta_1)E\left(\delta_2^{1/2}\right)\right] \\
&\qquad - \left[E(\delta_1) + \Phi^{-1}(u)E\left(\delta_2^{1/2}\right)\right]^2, \\
E(\delta_1) = \sum_{i=1}^{p} t_i\hat{\mu}_i, \quad & E(\delta_1^2) = E(\delta_1)^2 + \sum_{i=1}^{p} t_i^2\tilde{\mu}_i, \quad E(\delta_2) = \sum_{i=1}^{p} t_i^2\hat{\sigma}_i^2 + 2\sum_{i=1}^{p-1}\sum_{j=i+1}^{p} t_i t_j \hat{\sigma}_{ij}, \\
E\left(\delta_2^{1/2}\right) \approx E(\delta_2)^{1/2} - \frac{1}{2}&\left(\frac{1}{16E(\delta_2)}\right)^{3/2}\left(\sum_{i=1}^{p} t_i^4\tilde{\sigma_i^2} + 2\sum_{i=1}^{p-1}\sum_{j=i+1}^{p} t_i t_j \tilde{\sigma}_{ij}\right).
\end{aligned}
\tag{4.30}
$$

Despite its appearance, Equation (4.30) can be evaluated efficiently.

4.4.3 Comparison of Algorithms

To assess the convergence properties of the algorithms discussed in this section, we return to the diamond in a box problem (DB-problem) discussed in Section 4.3.1. This is convenient, because the modularization posterior is given analytically in eq. (4.13), and the numerical approximations $(\hat{\pi}_M(\cdot|\boldsymbol{y}))$ can be compared to the theoretical distribution $(\pi_M(\cdot|\boldsymbol{y}))$. To assess the convergence we will look at both Kullback-Leibler divergence (KLD) [153],

$$
\text{KL Divergence} = \int_{-\infty}^{\infty} \pi_M(\alpha|\boldsymbol{y}) \log\left(\frac{\pi_M(\alpha|\boldsymbol{y})}{\hat{\pi}_M(\alpha|\boldsymbol{y})}\right) d\alpha \tag{4.31}
$$

and Kolmogorov distance [71]

$$\text{Kolmogorov Distance} = \sup_{\alpha} \left| \int_{-\infty}^{\alpha} \left(\pi_M(a|\boldsymbol{y}) - \hat{\pi}_M(a|\boldsymbol{y})\right) da \right|. \tag{4.32}$$

Using the parameter settings described in Figure 4.5, data was simulated 500 times and the modularization posterior was approximated using variants of both the Monte Carlo (MC) and emulation of conditional posterior (ECP) algorithms. The four algorithms are (i) the ECP algorithm with a normal distribution assumed for $\mathcal{F}$, (ii) the ECP algorithm with a Weibull distribution assumed for $\mathcal{F}$, (iii) the MC algorithm using LHS and $m_\ell = 100$ and (iv) the same MC algorithm but with a normal distribution fit to the resulting samples. For each data set, the algorithms were asked to approximate eq. (4.13) five times, using budgets of $L \in \{10, 20, 30, 40, 50\}$. The results are summarized in Figure 4.8, which shows the total KL Divergence and Kolmogorov distance across the 500 samples. Even for the smallest budget ($L = 10$), both ECP implementations converged precisely to the true posterior. The MC implementations provide reasonable approximations, but a much larger budget (i.e. $L = 500$) to achieve comparable accuracy.

Complexity analysis

Let M denote the cost of running MCMC a single time. It is usually the case that running MCMC for BMC is incredibly time consuming and thus M should be viewed as large. For a fixed budget L, the complexity of the MC algorithm is simply $\mathcal{O}(LM)$. The MC implementation is also "pleasantly parallel", leading to a complexity of $\mathcal{O}\left(\frac{L}{P}M\right)$ when $P \leq L$ processors are used for parallel computation [90].

In comparison, the ECP implementation involves training a GP for each of the parameters in the specified form for the conditional posterior leading to $\mathcal{O}(LM + rL^3)$ complexity. The MCMC can be run in parallel for each location and the r GPs can

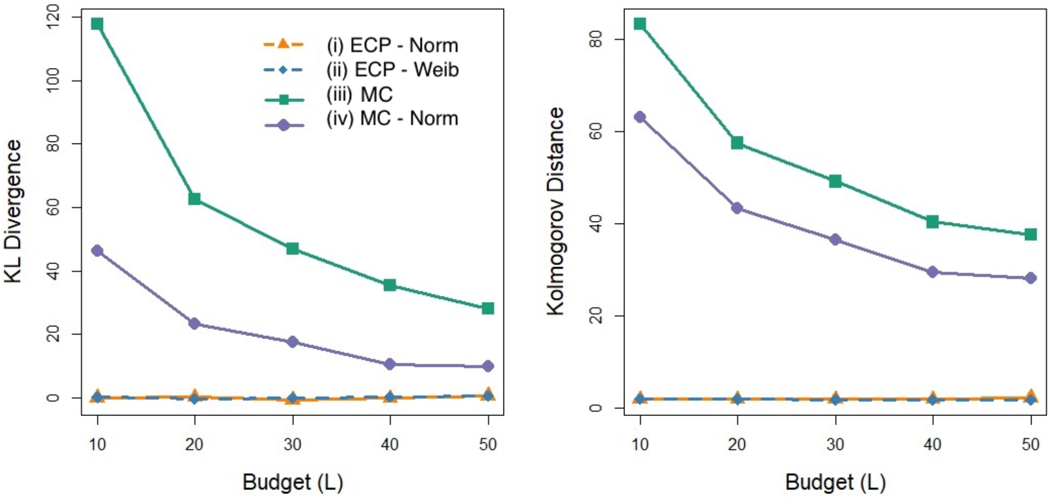


Figure 4.8: Comparison of the MC and ECP approximations to the modularization posterior for the DB-problem (eq. (4.13)). The ECP implementation matches almost perfectly even for $L = 10$. Parameters were fixed to $\alpha = 0$, $\gamma = 0.1$, $n_1 = 10$, $n_2 = 90$, $\sigma = 1$, $\sigma_\gamma = 0.5$, $\sigma_\alpha = \infty$.

be trained simultaneously, leading to a cost of $\mathcal{O}\left(\frac{L}{P}M + \frac{r}{P}L^3\right)$ for $P \leq \min\{L, r\}$ processors.

Finally, we consider the sequential ECP implementation (with $L_0 \ll L$) which has considerably more overhead. Each of the r GPs must be retrained at every time step $\ell > L_0$. Done naively this will require $\mathcal{O}(rL^4)$ time, but the use of the partition inverse equations [8, 66] can reduce this to cubic time. There is also the problem of finding the optimal location among the candidate set, which consists of prediction (with uncertainty) at N_c locations for every $\ell > L_0$. Putting everything together, the complexity of the sequential ECP algorithm becomes $\mathcal{O}\left(LM + (r + N_c)L^3\right)$. Moreover, the sequential nature of this implementation prevents us from parallelizing the conditional posterior sampling, but the training and candidate searching can still be done in parallel, giving $\mathcal{O}(LM + \frac{r+N_c}{P}L^3)$ for $P \leq r$. The inability of sequential ECP to parallelize the time-consuming MCMC steps is concerning and suggests that a block-sequential approach may be worthwhile.

Complexity analysis is important for an honest comparison of the approaches. For instance, it suggests that ECP may not be feasible for $L > 1000$, suggesting an upper bound on the number of allowable modularization parameters (somewhere around $q = 8$). We caution that complexity analysis can also be misleading here. The ECP algorithm is expected to take (slightly) longer than the MC algorithm for the same budget but is expected to make better use of its time. The example shown in this section and the examples in the rest of this Chapter illustrate that ECP makes effective use of the available resources, converging drastically faster to the target distribution than the alternative. Future work should involve theoretical results demonstrating this fact, but at present the "proof by example" is quite convincing.

4.5 Applications to Model Calibration

In this section, we explore modularization as tool for the calibration of physical parameters in BMC. First, we explore the use of modularization as a diagnostic tool for better understanding the posterior relationship between physical and nuisance parameters. As an example, we return to the simple machine example of Section 4.2. Secondly, we revisit the borehole function, first introduced in eq. (2.6), using a low fidelity simulator which induces a biased discrepancy function. Finally, we consider application of modularization to the tantalum experiments described in Section 1.2. The ECP algorithm of Section 4.4.2 is used throughout this section.

4.5.1 A Sensitivity Analysis

In this section, we describe how modularization can be useful as a tool for assessing the sensitivity of inference for $\boldsymbol{\alpha}$ on a nuisance parameter $\boldsymbol{\gamma}$. In the simple machine example of Section 4.2, we see that the quality of our inference for the efficiency of machine is heavily related to the prior distribution of γ. In the presence of model discrepancy, nuisance parameters can give a model too much flexibility leading to an overfit model and underestimation of uncertainty in the physical parameters. Modularization is shown to be heavily reliant on the prior $\pi_\gamma(\boldsymbol{\gamma})$ and somewhat robust to the effects of a misspecified model. At the same time, the modularization posterior can be equivalent to the standard Bayesian solution. We say that $\boldsymbol{\alpha}$ is a posteriori independent of $\boldsymbol{\gamma}$ if

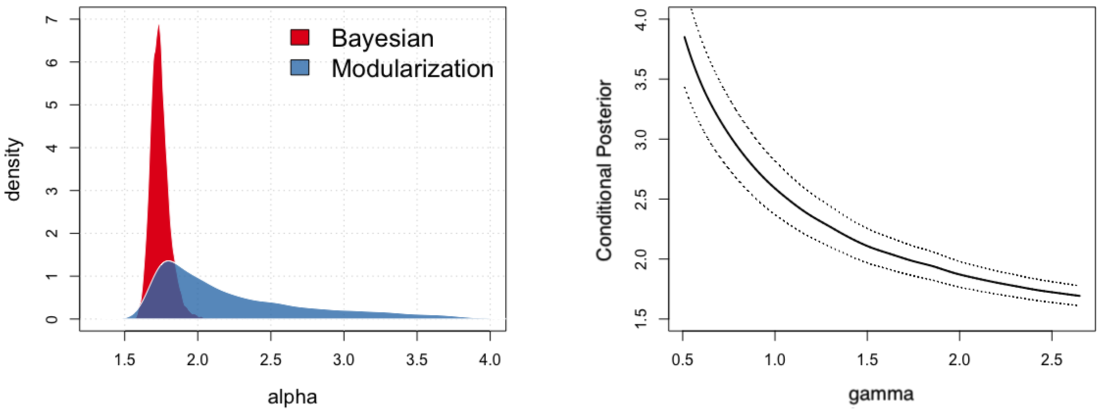
$$\pi(\boldsymbol{\alpha}|\boldsymbol{\gamma}, \boldsymbol{y}) = \pi(\boldsymbol{\alpha}|\boldsymbol{y}),$$

and this is precisely the case where the modularization posterior (eq. (4.10)) is equivalent to the marginal posterior (eq. (4.9)). If both methods are used to approach a given problem and the results are the same, we have strong evidence that the inference for $\boldsymbol{\alpha}$ is not sensitive to the treatment of $\boldsymbol{\gamma}$ (at least for the specified prior and computer model). If the modularization and fully Bayesian posteriors differ significantly, it means that (i) we are learning *something* about the nuisance parameter $\boldsymbol{\gamma}$ and (ii) this is influencing our inference for the physical parameter.

Revisiting the Simple Machine

In Section 4.2.2, we saw an example where calibration was unable to capture the true value of a physical parameter. The simulator given in eq. (4.5) represents missing physics and does not match the true process in eq. (4.3). This leads to the creation of a nuisance parameter γ, called the change point, which gains a physical interpretation only through its relationship with the physical parameter α and discrepancy function $\delta(\cdot)$.

After careful analysis researchers of the simple machine have determined that γ, which represents (in the model) the effort level at which friction starts to reduce output, must be between 0.5 and 2.65. Therefore they set $A = 0.5$ and $B = 2.65$ to complete the prior specification in eq. (4.7). Inference for the efficiency of the machine, α, is obtained using standard BMB and again using modularization (with modularization parameter γ). The posterior distributions are shown in Figure 4.9a. The BMC solution yields a narrow posterior, confidently concentrating around a value of $\alpha \approx 1.7$, and the true value $\alpha_\star = 2$ will be excluded in a 95% (or 99%) credible interval. The modularization posterior differs significantly, including for the possibility that the efficiency of the machine may be larger than the fully Bayesian posterior is suggesting. This should cause the researchers to hesitate and question the validity of the fully Bayes solution.



(a) Comparison of Bayesian and modularization posteriors for α.

(b) Conditional posterior of α as a function of γ.

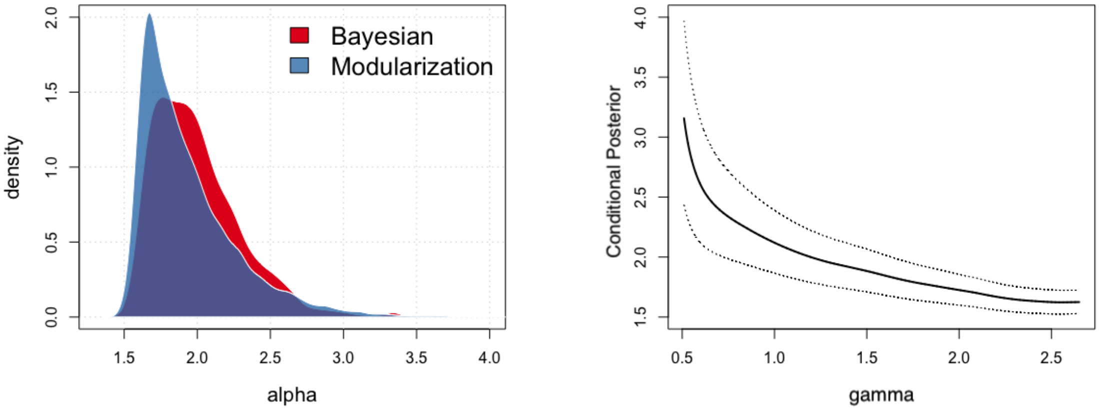
Figure 4.9: Diagnosing the simple machine (case 1).

Figure 4.9b illustrates the conditional posterior of $\boldsymbol{\alpha}$ given $\boldsymbol{\gamma}$, by plotting the mean and quantiles (0.025 and 0.975) of this distribution as a function of $\boldsymbol{\gamma}$. If α was independent of γ, we would expect this figure to consist of three parallel horizontal lines. The steep gradient implies that α is very sensitive to the treatment of γ, with respect to the current prior and model discrepancy form.

One option for a path forward is to further refine the model. Although the mechanism for friction is not yet understood, the inadequacy of the model can be reduced by increasing the order of the approximation.

$$\eta(x, \boldsymbol{\theta}) = \begin{cases} \alpha x, & x < \gamma \\ \alpha\gamma + \beta_1(x - \gamma), & \gamma \leq x < \gamma_2 \\ \alpha\gamma + \beta_1(\gamma_2 - \gamma) + \beta_2(x - \gamma_2), & \gamma_2 \leq x \end{cases} \tag{4.33}$$

$$\begin{gathered} \alpha \sim \text{Gamma}(6, 3) \qquad \beta_1|\alpha \sim \text{Unif}(0, \alpha) \qquad \beta_2|\beta_1 \sim \text{Unif}(0, \beta_1) \\ \gamma \sim \text{Unif}(0.5, 2.65) \qquad \gamma_2|\gamma \sim \text{Unif}(\gamma, 10) \end{gathered} \tag{4.34}$$



(a) Comparison of Bayesian and modularization posteriors for α.

(b) Conditional posterior of α as a function of γ.

Figure 4.10: Diagnosing the simple machine (case 2).

By improving the model, the posterior sensitivity of α on γ is reduced. The diagnostic plot in Figure 4.10b shows less evidence of a dependence between α and γ, except for when γ is very close to 0.5. The posterior distributions shown in

Figure 4.10a are now in close agreement, suggesting that α is not overly sensitive to the treatment of γ. Although the posterior distribution is significantly less narrow than before, we note that both posteriors are approximately centered around the true value $\alpha_\star = 2$.

For completeness, we note that the modularization approach was implemented for the five values of τ listed in Table 4.1. In four out of the five cases, the mean of the modularization posterior is closer to the true value of $\alpha_\star$ than the standard alternative. The modularization posterior is slightly wider than standard Bayes and the credible region formed by modularization can be viewed almost as a superset of the standard credible region. The true value of $\alpha_\star$ was captured inside the 95% CI for all five values of τ. Selected results are shown in Figure 4.11.

4.5.2 The Borehole Function

In this section, we compare the modularization and fully Bayesian model calibration approaches using synthetic data based on the well-known Borehole function with a single parameter of interest. In the presence of model discrepancy, this problem is poorly identified and the parameter of interest is confounded with one of the nuisance parameters. We demonstrate that the modularization approach to model calibration has desirable statistical properties when compared to the fully Bayesian approach, at the cost of precision in the credible intervals.

Consider the Borehole function [102, 103, 145], which models water flow through a borehole (m^3/yr) as

$$\eta(x, w, K, T, r) = \frac{600\pi T}{\ln(r/w)\left(1 + \frac{2xT}{\ln(r/w)w^2K} + \frac{T}{100}\right)}. \tag{4.35}$$

x is a known input (or design variable), which denotes the length (m) of the borehole. We assume that this input can take values $x \in [500, 4000]$ and is measured without

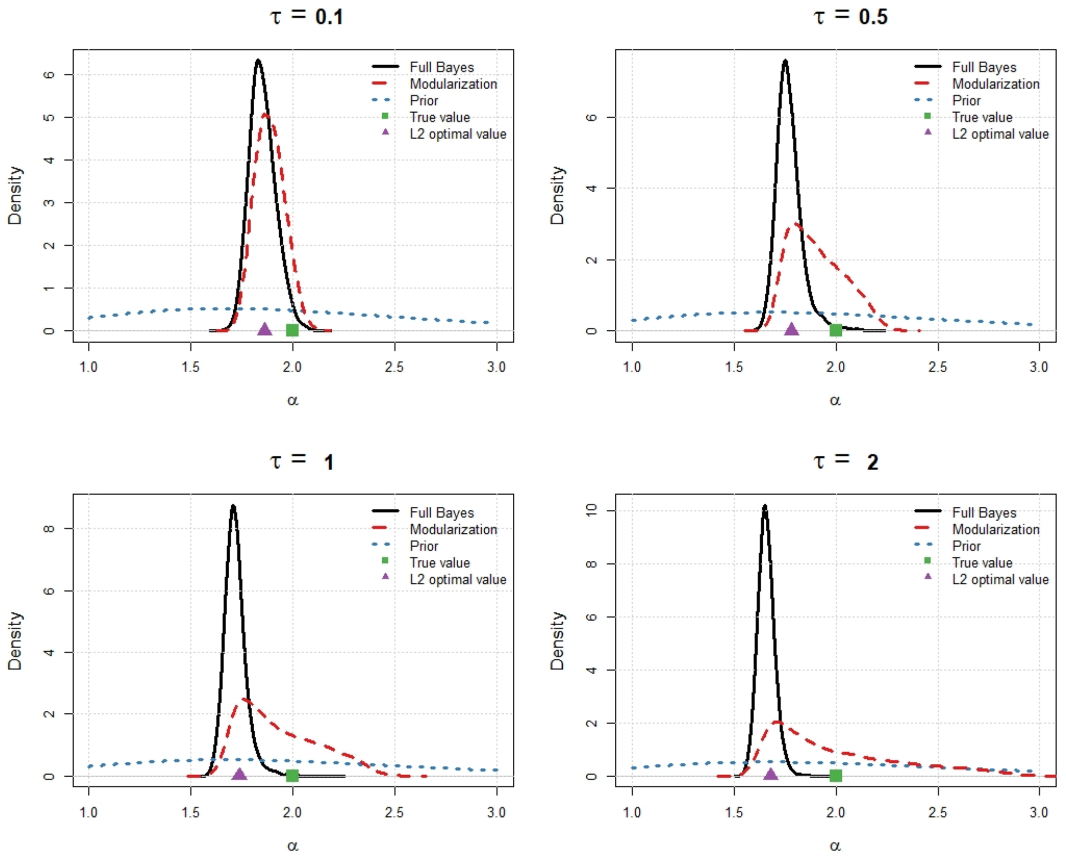


Figure 4.11: Comparison of fully Bayesian and modularization posteriors for different values of τ.

error. The remaining inputs are unknown calibration parameters. Briefly, w is the *radius of influence* (m), K is the *hydraulic conductivity* (m/yr) of the borehole, T is the *transmissivity of the upper aquifer* (m^2/yr) and r is the radius (m) of the borehole which is measured with error. The associated prior distributions are

$$\begin{aligned} &w \sim N(0.1, 0.0304^2) I_{[0.01,\infty)}(w), && K \sim U(3500, 10000), \\ &T \sim U(63070, 115600), && r \sim \log N(7.71, 1.0056). \end{aligned} \tag{4.36}$$

We assume that the single parameter of interest is $\alpha := w$ and there are three modularization parameters $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \gamma_3) := (K, T, r)$. We treat eq. (4.35) as the

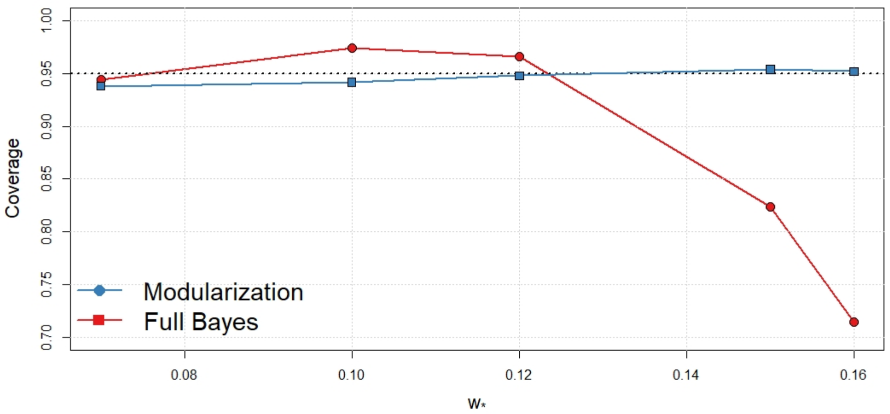


Figure 4.12: Borehole simulation study. Empirical coverages for the fully Bayesian and modularization approaches as a function of $w_\star$. Nominal coverage is 95%.

computer simulator and take the true process to equal $\zeta(x) = \eta(x, w_\star, K_\star, T_\star, r_\star) + \delta(x)$, where the $\star$ subscript denotes the true value of a parameter. Using the BMC framework of Section 1.4.3, we assume that the true data generating process is equal to

$$\begin{aligned} y_i &= \zeta(x_i) + \delta(x_i) + \epsilon_i, \;\; i = 1, 2, \cdots n \\ \epsilon_i &\overset{\text{iid}}{\sim} N(0, \sigma^2) \\ \delta(x) &\sim GP(0, \phi R). \end{aligned} \tag{4.37}$$

Depending on the generated form of the model discrepancy, the parameter of interest w is often heavily confounded with K. Specifically, an excellent fit to the data can still be obtained when both parameters are estimated to be smaller than their true values. Consequently, as the true value $w_\star$ increases, the standard BMC approach has a tendency to overfit and inference for both K and w is unreliable. In the modularization approach, by forfeiting the ability to learn about K, we can obtain robust inference for the parameter of interest. For each of five different true values of the radius of influence, $w_\star \in \{0.07, 0.10, 0.12, 0.15, 0.16\}$, we generate 500 synthetic datasets from the mechanism of eq. (4.37), setting $n = 200$, $x_i = 500 + (i-1)\frac{3500}{n-1}$, $\boldsymbol{\psi} = (\sigma = 0.2, \phi = 2, \kappa = 10^{-5})$. These parameter settings were chosen to emulate the challenges often faced in BMC applications, but we do not believe that the results of this section are qualitatively dependent on this choice. For each of the 500 data sets, we perform BMC using both modularization and full Bayes and a corresponding 95% credible interval is computed for each method. From these results, displayed in Figure 4.12, it is clear that the fully Bayesian approach is severely under-covering as $w_\star$ increases while the modularization approach maintains near nominal coverage for all values of $w_\star$.

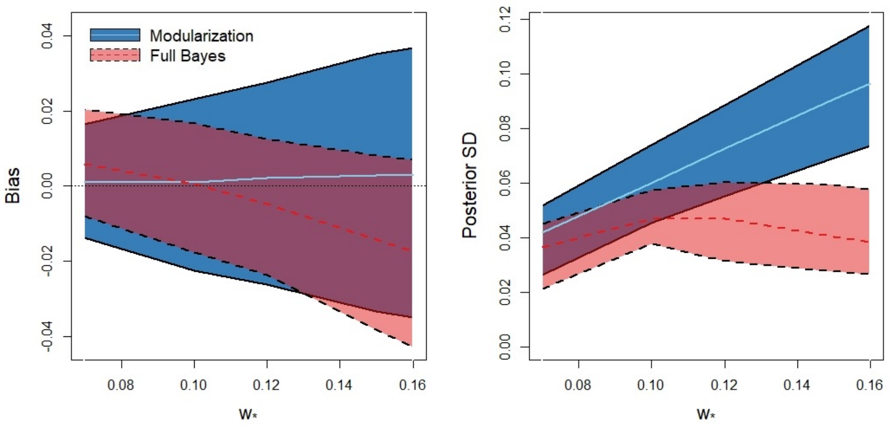


Figure 4.13: Posterior bias and standard deviation for the borehole example as a function of $w_\star$. Shaded bands represent an 80% central region across 500 simulations [106].

To better understand the differences between these two approaches, we turn to Figure 4.13. The posterior bias (left panel), defined as $E(w|\boldsymbol{y}) - w_\star$, and the posterior standard deviation (right panel) are summarized in this figure. In each panel, the shaded bands correspond to 80% central regions and the central lines represent the medians of the 500 simulations [106]. The modularization posterior (solid lines) is, on average, approximately unbiased for all values of $w_\star$. The variance of the posterior mean, which is roughly proportional to the width of the shaded band in the left panel, increases as $w_\star$ moves away from its posterior mean. To maintain good coverage of the credible interval, the posterior standard deviation (right panel) increases at a similar rate. On the other hand, the fully Bayesian posterior (dashed lines) is only unbiased on average when $w_\star$ is equal to its prior mean 0.10. Since the full Bayes approach tries to learn about w and K simultaneously, there is a tendency to underestimate w and K is used to compensate. Although the bias increases in magnitude under the fully Bayesian approach, the posterior standard deviation stops increasing, leading to deceivingly precise credible intervals and poor coverage frequencies.

In this section, the modularization posterior was approximated using the ECP algorithm with a normality assumption eq. (4.26) using $L = 50$, $\ell_0 = 20$ and $k = 5$. The results were compared to a large budget MC implementation using $L = 1000$ and $m = 10$. The two implementations produced essentially identical results, although the ECP implementation was more than 20 times faster. Although not shown here for brevity, the modularization posterior was also approximated using a Weibull assumption eq. (4.27) and again the resulting approximation was identical.

4.5.3 Compressibility of Tantalum

In this section, we analyze a set of measurements on tantalum (Ta) in which the compressibility parameters of tantalum are of interest. This example, which has been analyzed using traditional methods from physics [20], is challenging from a model calibration perspective. We will show that the calibration offered by the modularization approach is more consistent with the traditional analyses than standard BMC.

We consider a class of dynamic material experiments in which a powerful magnetic field is used to generate high pressures and induce a stress wave which propagates through a material of interest. In this particular problem, the velocity of the stress wave (Km/s), which propagates through a tantalum sample, is recorded at the interface between the tantalum and a lithium fluoride window. Details on the experimental configuration and the resulting data can be found in [20, 21, 136] The resulting data can be viewed as a function $y(t),\ t \in [0, T]$. Even with the functional structure there is a finite amount of information contained in each curve and thus it is reasonable to discretize, so that the data becomes

$$y_i = y(t_i),\ i = 1, \cdots n,$$

for fixed time points t_i [11, 161] The aim of the analysis is to estimate two material properties of tantalum, known as equation of state parameters, which describe the pressure-density relationship (or compressibility) of tantalum. The two parameters of interest are the Bulk modulus of tantalum (B_0 GPa) and its corresponding first pressure derivative (B_0'), i.e. $\boldsymbol{\alpha} = (B_0, B_0')$. We also have access to a computer simulator, which models these experiments using the ALEGRA wave propagation code [129]. Many of the inputs to the computer model are fixed at nominal values, but there are some remaining nuisance parameters whose uncertainty we wish to account for. These parameters include the initial density of tantalum (ρ_0 g $\cdot$ cm^3), the boundary condition scaling parameter (BC_{scale}) and thickness measurements for the tantalum (x_{Ta} μm) and the aluminum (x_{Al} μm), which acts as an electrode. Collectively, these 6 inputs form the set of calibration parameters $\boldsymbol{\theta}$.

In the BMC framework [76, 87] the experimental data is modeled as

$$\begin{aligned} &y_i = \eta(t_i, \boldsymbol{\theta}) + \delta(t_i) + \epsilon_i \quad i = 1, \cdots n \\ &\delta(t) \sim GP\Big(\mu, \Sigma_\delta\Big) \\ &\epsilon_i \overset{\text{ind}}{\sim} N(0, \sigma_i^2). \end{aligned} \tag{4.38}$$

Equivalently, we have that $\boldsymbol{y} \sim MVN(\boldsymbol{\eta} + \mu, \Sigma_\delta + \Sigma_\epsilon)$ where Σ_ϵ is a diagonal matrix with elements σ_i^2 which are provided by a physicist and treated as fixed and known, and $\Sigma_\delta = \phi R$ where R is a correlation matrix such that

$$R_{ij} = \exp\Big\{ - \kappa(t_i - t_j)^2 \Big\}. \tag{4.39}$$

As identifiability constraints, we assume a mean zero discrepancy function (i.e. $\mu = 0$) and fix $\kappa = 7.225 \times 10^{-2}$ (after re-scaling so that $t \in [0,1]$) based on previous work [3, 16, 21, 94].

The model in eq. (4.38) is overdetermined and the compressibility parameters of tantalum are poorly identifiable. This means that there are a large number of parameter settings $(BC_{\text{scale}}, \rho_0, B_0, B_0')$ which provide a precise fit to the data, and the addition of the model discrepancy term makes it impossible to determine which setting, if any, is the correct one. Sensitivity studies have shown ([21]) that the thickness parameters x_{Ta} and x_{Al} are not confounded with $\boldsymbol{\alpha}$, but these parameters should still be handled within the MCMC to avoid artificially precise posteriors for the parameters of interest.

To summarize the information so far, there are two parameters of scientific interest $\boldsymbol{\alpha} = (B_0, B_0')$ which describe the compressibility of tantalum, and two inputs $\boldsymbol{\gamma} = (BC_{\text{scale}}, \rho_0)$ whose values are highly confounded with the physical parameters. Equipped with strong, well-informed prior information on these inputs, these are the parameters for which it makes the most sense to modularize over. The remaining BMC parameters, we denote $\boldsymbol{\beta} = (x_{Ta}, x_{Al}, \phi)$. These parameters are not treated as modularization parameters, but they are not explicitly accounted for in the ECP algorithm in order to reduce the number of GPs which must be learned. In other words, at each iteration of the ECP algorithm samples are obtained from the conditional posterior $\pi(\boldsymbol{\alpha}, \boldsymbol{\beta} | \boldsymbol{\gamma}, \boldsymbol{y})$, but the $\boldsymbol{\beta}$ parameters are ignored after this step which is equivalent to marginalizing them out of the modularization posterior. Prior distributions for each parameter is given below.

$$\begin{array}{lll} \alpha_1 = B_0 \sim N(185, 17.3^2) & \alpha_2 = B_0' \sim U(2.9, 4.9) & \\ \gamma_1 = BC_{\text{scale}} \sim N(1, 0.004^2) & \gamma_2 = \rho_0 \sim N(16.55, .0662^2) & \\ \beta_1 = x_{Ta} \sim N(\mu_{Ta}, 1.5) & \beta_2 = x_{Al} \sim N(\mu_{Al}, 1.5) & \beta_3 = \phi \sim C_+(0, 10) \end{array} \tag{4.40}$$

First, we attempt to analyze this data using a fully Bayesian model calibration approach, using an adaptive Metropolis Hastings Algorithm [72] to obtain posterior samples. The MCMC was run for 50000 iterations using the first half as burn-in and retaining every tenth sample [63]. The modularization posterior was approximated using the multivariate normal ECP algorithm, with $L = 100$, $\ell_0 = 20$ and $k = 2$. Posterior summaries for $\boldsymbol{\alpha}$, under each approach, is shown in Figure 4.14. As expected, the modularization posterior is more conservative than the fully Bayesian approach, with significantly higher posterior variance for B_0'.

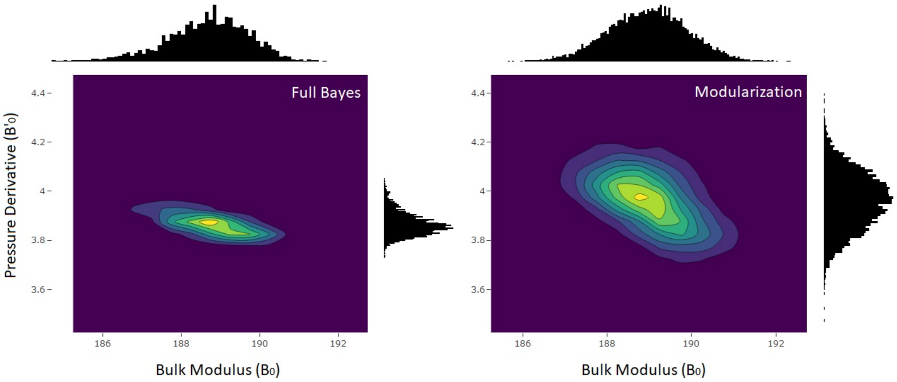


Figure 4.14: Comparison of posteriors for $\boldsymbol{\alpha}$ in the tantalum example.

The compressibility parameters B_0 and B_0' were determined to be of scientific interest, not for their own sake, but because these parameters can be used to determine the pressure-strain (P-s) relationship ([20]). This relationship is modeled using the physically inspired Vinet model ([157])

$$P(s|B_0, B_0') = 3B_0 \left(\frac{1-\xi}{\xi^2}\right) \exp\left\{\frac{3}{2}(B_0' - 1)(1-\xi)\right\}, \; \xi = (1-s)^{1/3}. \quad (4.41)$$

Previous work has shown that BMC based estimation of the compressibility parameters, coupled with the 2-parameter Vinet model, can provide reasonable calibration for low pressure behavior but commonly fails at higher pressures ($>$ 250 GPa). Thus, the BMC results here (Figure 4.14) are compared to the pressure-strain curve found via an *average Lagrangian analysis* (ALA), a state-of-the-art analytic method for modeling the pressure-strain relationship of a material ([20]; [130]). This comparison can be seen in Figure 4.15 where the '*' symbols represent the ALA estimates. The results for the full Bayes calibration and the modularization approach are shown using solid and dashed lines respectively, where the center lines indicate the pressure-strain relationship at the posterior mean (i.e. $P(s|\hat{B}_0, \hat{B}_0'))$ and the shaded bands indicate 95% credible central regions ([106]). Both calibration approaches agree with the ALA response for pressures up to about 250 GPa (left panel), but both calibrations begin to fail at higher pressures (right panel) due to the extrapolative nature of these predictions. Even at peak pressure ($P \approx 520$GPa), the modularization estimate of pressure is less biased than the fully Bayesian method, with a relative error of 8% compared to 11%. More importantly, the modularization 95% credible region captures the ALA estimates for even the highest pressures. The additional robustness offered by modularization comes at the cost of conservatism in estimation, with a 95% credible band with more than twice the width of the full BMC analogue.

Although the high uncertainty in the estimation limits the scientific usefulness of the approach in this instance, much can be learned from this analysis. The excessive width of the modularization posterior, in comparison to full BMC, indicates a lack of identifiability, which may signal anti-conservatism and bias in the fully Bayesian approach and indicates that the posterior inference should be questioned.

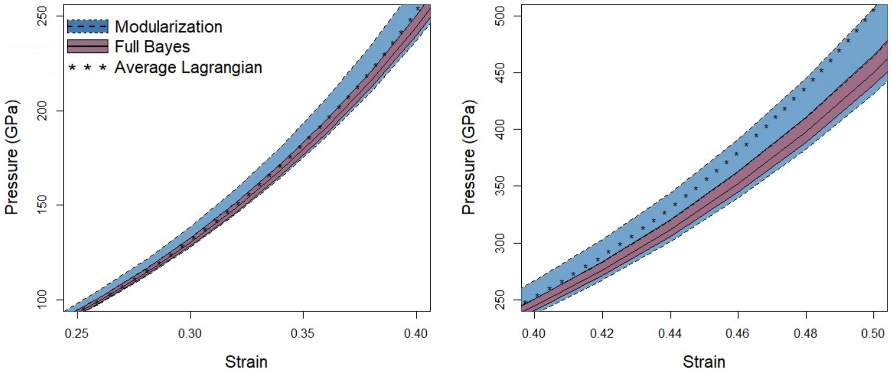


Figure 4.15: Comparison of full Bayes, modularization and ALA approaches on the Pressure-strain space.

4.6 Conclusions & Future Work

In this chapter, we discuss the challenges of model calibration when physical parameter inference is the goal and we provide a general discussion of the modularization framework. Special attention is given to the numerical approximation of this distribution, and we propose a novel algorithm (ECP) for efficient approximation. Finally, we apply the modularization framework in a variety of BMC applications, demonstrating that it has robust qualities and can be used to ascertain the sensitivity of physical parameters $\boldsymbol{\alpha}$ to the prior of the nuisance parameters. This is yet another diagnostic tool for the tool-box of ascertaining parameter identifiability, especially when parameters are viewed as inherent properties of a physical system.

Although we believe the ECP algorithm is an improvement over the previous Monte Carlo approach, there are many opportunities for improvement. When emulating the conditional posterior, a parametric form for $\mathcal{F}$ is required. In our applications, we find that the approximation of the modularization posterior is quite robust to this choice, but future work should involve theoretical and empirical analysis of the implications of a poor choice. Alternatively, we could work towards a non-parametric version of the ECP algorithm, modeling a more flexible basis representation of the posterior. This can theoretically be done without modifying the underlying framework, but computational challenges are sure to present themselves. In the sequential ECP algorithm, we could also explore a block-sequential implementation, which would restore the use of parallel predictions and improve performance. This should be possible, but the current implementation will need to be modified to avoid clustering of the k future locations at each step. Finally, we note that we have only applied the multivariate-ECP algorithm for $\boldsymbol{\alpha} \in \mathbb{R}^2$, so future work should involve application of this algorithm for a higher dimensional physical parameter vector, possibly using a sparse representation of the covariance or precision matrix [25, 39, 53] to facilitate faster performance.

Future work should also be focused on better understanding modularization as a tool for physical parameter inference, using theoretical analysis and real world applications. An immediate extension is the use of modularization for treating the model discrepancy parameters, which are presently handled using a “plug-in” approach, which is the topic of Chapter 5.

Chapter 5

Modularization, Cross Validation and Discrepancy

"To think you know when you do not is a disease." – Lao Tzu

5.1 Overview

The "plug-in" approach to model calibration refers to the process of setting certain calibration parameters to fixed values, treating them as known throughout the remainder of the analysis. Plug-in has several downsides, including hypersensitivity to the selected value and underestimation of uncertainty, but it is often the simplest way to avoid some of the identifiability issues discussed by [3, 24, 117, 148]. In this section, we write the BMC discrepancy function prior as

$$\delta(\cdot) \sim GP(\mu, \phi R(\cdot, \cdot|\kappa)),$$

where $R(\cdot, \cdot|\kappa)$ is a correlation function like the one defined in eq. (1.12) and we the discrepancy parameters are collectively denoted $\boldsymbol{\psi} = (\mu, \phi, \kappa)$. It is very common to treat these parameters using plug-in, setting $\mu = 0$ and κ equal to some fixed value as described in Section 1.4.3. We propose treating these parameters with modularization rather than plug-in because it (i) identifiability is maintained between

$\delta(\cdot)$ and $\boldsymbol{\theta}$ for the purposes of MCMC, (ii) provides a more honest assessment of uncertainty and (iii) the sensitivity of the results to the choice of discrepancy function can be evaluated.

Unfortunately, modularization is heavily reliant on prior information, which is generally lacking for the discrepancy parameters [24]. In the remainder of this chapter, we propose some options for constructing empirical priors for the discrepancy parameters, which can subsequently be used for modularization.

5.2 Cross Validation for Discrepancy Parameters

A model that fails to generalize beyond data that it has previously observed is said to be *overfit*. Cross validation (CV) [143] is a popular approach for training a balanced model. In CV, some of the observations are withheld during the training of the model, and then the model is evaluated based on its ability to predict these held out values. This process is repeated using a new set of withheld values until all of the data points have been held out and then predicted exactly one time. A model is

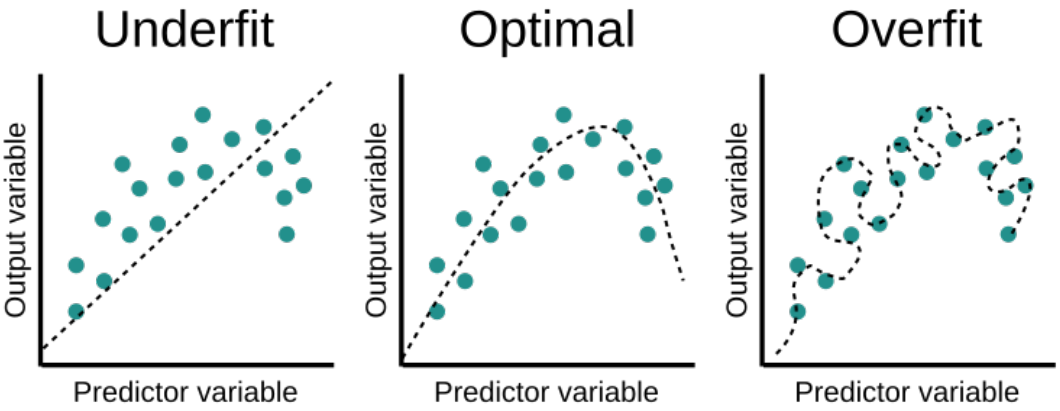


Figure 5.1: Illustration of an overfit model.

less likely to overfit using this approach since predictions occur at previously unseen locations. The term K-fold CV refers to the case where the N observations are randomly partitioned into K sets of $n = N/K$ distinct observations. This approach is designed for the case where the residuals of a model can be viewed as independent, but when the residuals exhibit spatial or temporal structure, evaluations of the hold out set may provide an overly optimistic view. When residuals are highly correlated with their spatial neighbors, held-out samples can be easily predicted for almost any value of the parameters and CV provides only limited information [128]. In this setting, we can use the modified K-*block* cross-validation procedure, in which folds are chosen to be contiguous regions of the input space. In this chapter, we focus on the temporal case where $x \in \mathcal{X} \subset \mathbb{R}$ is the sole design variable. The field data can thus be written as $\mathcal{D} = (\boldsymbol{x}, \boldsymbol{y})$ and we define a partition $\mathcal{P}$ of the field data to be any collection of sets $\{B_1, B_2, \cdots B_K\}$ such that each (x_j, y_j), $j = 1, 2, \cdots n$, belongs to exactly one set B_k. The sets in a partition are said to be temporal *blocks* if for every $x_j < x_{j'}$ we have that $x_j \in B_k$ and $x_j \in B_{k'}$ with $k \leq k'$.

Broadly, our proposal for CV in the context of model calibration is to (i) treat $\boldsymbol{\psi} = \boldsymbol{\psi}_0$ as fixed and known and (ii) partition the data $(\boldsymbol{X}, \boldsymbol{y})$ into K contiguous blocks and for $k = 1, 2, \cdots K$ (iii) use MCMC to sample from the posterior in eq. (1.27) conditional on $\boldsymbol{\psi}_0$ and based on the data with the k^{th} block withheld. The overall quality of the fixed value $\boldsymbol{\psi}_0$ can be expressed in terms of the RMSE of every observation in the full data, and the goal is to evaluate the quality of $\boldsymbol{\psi}_0$ for a large set of reasonable values, selecting the highest quality value $\hat{\boldsymbol{\psi}}_{CV}$. To summarize, the CV loss function for a fixed value $\boldsymbol{\psi}_0$ is

$$\mathcal{L}(\boldsymbol{\psi}_0|\mathcal{P}) = \frac{1}{N}\sum_{k=1}^{K}\sum_{y_i \in B_k}\left(y_i - \hat{y}_i|(\mathcal{P}_{-k}, \boldsymbol{\psi}_0)\right)^2, \tag{5.1}$$

where $\hat{y}_i|(\mathcal{P}_{-k}, \boldsymbol{\psi}_0)$ denotes the predicted response obtained from fitting the model with $\boldsymbol{\psi} = \boldsymbol{\psi}_0$ fixed and data from the k^{th} block withheld. The highest quality value can then be defined as

$$\hat{\boldsymbol{\psi}}_{CV} = \arg\min_{\boldsymbol{\psi}_0} \mathcal{L}(\boldsymbol{\psi}_0|\mathcal{P}). \tag{5.2}$$

Although MCMC is costly for BMC applications in general, it should generally be much faster in this scenario because (i) fixing $\boldsymbol{\psi}_0$ improves identifiability of the parameters (and thus improves the mixing time), (ii) the covariance matrix of the data (see Section 1.5.3) is now static and can be inverted just a single time for each instance of MCMC and (iii) some data is being withheld. Next, we discuss two empirical methods for using the results of CV to construct a prior distribution $\pi_\psi(\boldsymbol{\psi})$ which can be used for modularization

Method 1: Loss Function as a Prior

The process described above can be used to define a loss function $\mathcal{L}(\boldsymbol{\psi}|\mathcal{P})$, where $\mathcal{P}$ denotes a partition of $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$ into K contiguous blocks and is the topic of the Section 5.3. As described in Section 1.5.4, we can specify the prior

$$\pi_\psi(\boldsymbol{\psi}|\lambda, \mathcal{P}) \propto \exp\left\{-\lambda\mathcal{L}(\boldsymbol{\psi}|\mathcal{P})\right\}, \tag{5.3}$$

where λ controls the precision of the prior and can be chosen using model selection criterion such as generalized cross validation [158]. In a sensitivity analysis setting, λ can be viewed as a diagnostic parameter and ad-hoc selection methods can be used. An example of converting a CV loss function to a prior for $\psi = \mu$ is shown in Figure 5.5. Alternatively, we could define

$$\pi_\psi(\boldsymbol{\psi}|\lambda, \mathcal{P}) \propto \mathbb{1}\left(\mathcal{L}(\boldsymbol{\psi}|\mathcal{P}) \leq \lambda\mathcal{L}(\hat{\boldsymbol{\psi}}_{CV}|\mathcal{P})\right\}, \tag{5.4}$$

for some $\lambda > 1$, which defines a uniform prior over the set of $\boldsymbol{\psi}$ values which lead to almost-minimal loss. This discussion represents two possible strategies, out of many, for converting a loss function into a prior distributions.

Method 2: Empirical Prior Construction

Rather than choosing the value $\hat{\boldsymbol{\psi}}_{CV}$ which maximizes the total quality of prediction, we can focus on the values $\hat{\boldsymbol{\psi}}_k$, $k = 1, 2, \cdots K$, which minimizes the MSE for the data $\mathcal{P}_{-k}$

$$\hat{\boldsymbol{\psi}}_k = \arg\min_{\boldsymbol{\psi}_0} \left\{ \sum_{y_i \in B_k} \left(y_i - \hat{y}_i | (\mathcal{P}_{-k}, \boldsymbol{\psi}_0)\right)^2 \right\}. \tag{5.5}$$

Following [111], we can calculate the sample mean vector

$$\boldsymbol{m} = \frac{1}{K} \sum_{k=1}^{K} \hat{\psi}_k,$$

the sample covariance matrix

$$\boldsymbol{C} = \frac{1}{K-1} \left(\sum_{k=1}^{K} \hat{\boldsymbol{\psi}}_k \hat{\boldsymbol{\psi}}_k^\top - K \boldsymbol{m}\boldsymbol{m}^\top \right),$$

and specify the prior

$$\pi_\psi(\boldsymbol{\psi}|\lambda, \mathcal{P}) = N\left(\boldsymbol{\psi}|\boldsymbol{m}, \lambda \boldsymbol{C}\right). \tag{5.6}$$

5.3 Selection of Cross Validation Blocks

Selection of the block size (and the corresponding number of blocks) is an important problem for successful cross-validation and prior construction. In the standard K-fold cross-validation, it is typically the case that leave-one-out cross validation (LOOCV) is theoretically optimal, although it is computationally intensive and may not be feasible. This LOOCV approach fails when the data has temporal (or spatial) structure, since it is easy to predict a held-out observation using just its neighbors [128].

5.3.1 Effective LOOCV

The *effective sample size* (ESS), denoted n_{eff}, is a modified notion of sample size when the variables of interest are correlated, having temporal or spatial structure. As a simple example, consider sampling from a posterior with an MCMC algorithm such as Metropolis Hastings (see Section 1.5.2). Since the samples $\theta^1, \theta^2, \cdots \theta^M$ are correlated this is not equivalent to having M independent samples from the posterior, with the actual amount of information obtained being comparable to a smaller number, the effective sample size n_{eff}.

If LOOCV is viewed as the best approach for cross validation whenever (i) it is computationally feasible and (ii) the independence assumptions are met, then it is reasonable to wonder if the *effective LOOCV* approach is optimal when the data has temporal structure. Thus, we propose an ELOOCV procedure, in which $K = n_{\text{eff}}$ contiguous blocks are used, where

$$\begin{aligned} n_{\text{eff}} &= \frac{n}{\tau}, \\ \tau &= 1 + 2\sum_{\ell=1}^{\infty} \operatorname{Cor}(y_j, y_{j+\ell}) \end{aligned} \tag{5.7}$$

In practice, one can estimate τ by substituting the sample autocorrelation function (ACF) estimates (r_ℓ) into this equation and truncating the sum at some large value. A method for selecting the truncation constant L is given in [140], in which L is chosen to be the smallest value for which $L \geq C\hat{\tau}_L$, where $\hat{\tau}_L$ is the estimated autocorrelation time based on L and $C \approx 5$.

5.3.2 Gaussian Processes and Effective Range

In non-stationary cases, the sample ACF may not be an appropriate estimator of $\operatorname{Cor}(y_j, y_{j+\ell})$ leading to issues with eq. (5.7). using the sample correlation values.

Additionally, eq. (5.7) implicitly assumes that the design variables are regularly spaced across $\mathcal{X}$, i.e. that $|x_{i+1} - x_i|$ is constant for $i = 1, 2, \cdots n - 1$. Alternatively, n_{eff} can be estimated using the notion of *effective range*. First, we define the empirical discrepancy function

$$\hat{\delta}(\boldsymbol{x}) = \hat{\zeta}(\boldsymbol{x}) - \eta(\boldsymbol{x}, \hat{\boldsymbol{\theta}}), \tag{5.8}$$

where $\hat{\boldsymbol{\theta}}$ is an estimate of $\boldsymbol{\theta}$ (i.e. using LS or L_2 calibration) and $\hat{\zeta}(\cdot)$ is a smoothed response surface (for example see eq. (1.20)). We can obtain $\hat{\zeta}(\cdot)$ using a sophisticated approach (i.e. Section 1.4.2 or [148]) or using relatively simple local regression strategies [33,34]. By assuming an isotropic Gaussian correlation function (i.e. eq. (1.12)), a Gaussian process can be fit to the empirical discrepancy allowing for direct estimation of $\text{Cor}\,(\delta(\boldsymbol{x}), \delta(\boldsymbol{x}'))$. The *$\rho$-effective range* is the smallest Euclidean distance (denoted d_{eff}) such that

$$\text{Cor}\,(\delta(\boldsymbol{x}), \delta(\boldsymbol{x}')) \geq \rho \quad \Leftrightarrow \quad \|\boldsymbol{x} - \boldsymbol{x}'\|_2 \leq d_{\text{eff}} \tag{5.9}$$

If the correlation parameter of $\hat{\delta}(\cdot)$ is estimated to be $\hat{\nu}$, then we can estimate the effective range as

$$d_{\text{eff}} = \sqrt{\frac{-\log(\rho)}{\hat{\nu}}} \tag{5.10}$$

If the design variables are regularly spaced, then the effective sample size (and thus the number of ELOOCV blocks) is estimated as

$$n_{\text{eff}} = \frac{\text{Di}(\mathcal{X})}{2d_{\text{neff}}}, \tag{5.11}$$

where $\text{Di}(\mathcal{X})$ is the diameter of the input space, given by $\sup_{x,x' \in \mathcal{X}} \|\boldsymbol{x} - \boldsymbol{x}'_2\|_2^2$. In the case of unbounded input spaces, it will suffice to consider the diameter of the training set $\text{Di}(\boldsymbol{X})$.

5.3.3 Acceptable Partitions and Binary Refinement

When the design variables are irregularly spaced, or in the case of non-stationary modeling assumptions for the empirical discrepancy, more flexible solutions may be required. An example of the latter situation is a discrepancy model where the correlation parameter depends on the design variable in some way, i.e. $\nu = \nu(\boldsymbol{x})$, but we do not explicitly consider this scenario here.

We will say that a partition $\mathcal{P} = \{B_1, \cdots B_K\}$ is *acceptable*, with respect to ρ and ι, if

$$\left\{\sum_{y \in B_k} \mathbb{1}\left(\max_{y' \notin B_k}\{\text{Cor}(y, y')\} < \rho\right)\right\} \geq \iota \tag{5.12}$$

hold for every $k = 1, \cdots K$. Equation (5.12) states that every block B_k should contain at least ι response values which are sufficiently hard to predict (as determined by ρ) via the correlation structure alone. The parameters ρ and ι should be chosen to balance the notions that (i) K should be as large as possible according to the LOOCV philosophy and (ii) K must not be too large to avoid the aforementioned temporal (or spatial) difficulties of CV. If $\mathcal{P}^1$ and $\mathcal{P}^2$ are acceptable, we should prefer the one which is more *refined*, which is to say that the blocks are generally small. For given values of ρ and ι, the combinatorial nature of searching for the optimally refined acceptable partition may be infeasible. Instead, we propose a greedy recursive strategy for finding acceptable partitions.

If $\mathcal{P}^1$ and $\mathcal{P}^2$ are acceptable, we should prefer the one which is more *refined*, which is to say that the blocks are generally small. We say that a partition $\mathcal{P}^2 = \{A_1, A_2, \cdots A_{K_2}\}$ is a *refinement* of the partition $\mathcal{P}^1 = \{B_1, B_2, \cdots B_{K_1}\}$ if each B_k can be written as the disjoint union of sets in $\mathcal{P}^2$. In the present context, we will assume that the sets $B_k \in \mathcal{P}$ are contiguous blocks (defined in Section 5.2). Although this notion can be extended to higher dimensions, we remain focused on the univariate case $x \in \mathbb{R}$ and assume without loss of generality that $x_i < x_{i+1}$ for $i = 12 \cdots N-1$. This implies that a binary partition search can be conducted in linear time.

This approach, which we refer to as (greedy) binary refinement, is outlined as follows. We begin by finding (if one exists) a partition $\mathcal{P}^1 = \{B_1, B_2\}$ such that eq. (5.12) is satisfied. In each subsequent step, the partition $\mathcal{P}^t$ is obtained by finding an acceptable binary partition of each $B \in \mathcal{P}^{t-1}$ if possible. When a contiguous block B can no longer be refined in a manner that satisfies eq. (5.12), this branch of the recursion process terminates. While the final partition is unlikely to be optimally refined, it is guaranteed to be acceptable and likely to include reasonably small blocks. Although our current implementation uses a global correlation parameter ν as described in Section 5.3.2, the implementation can easily be modified to handle

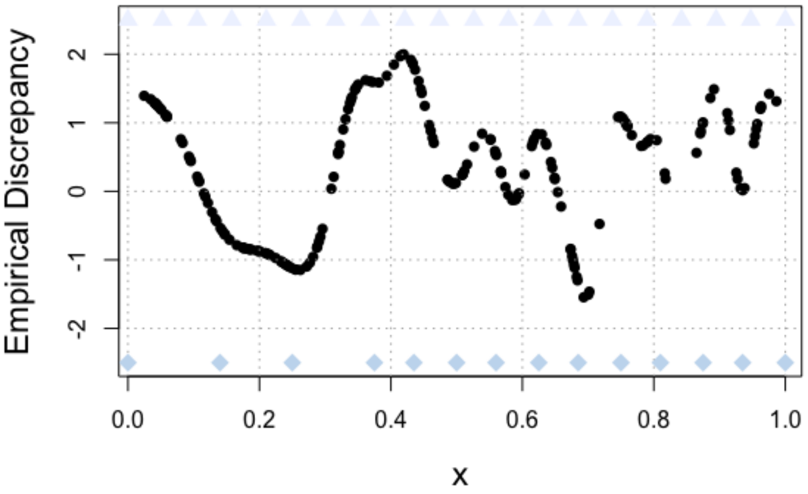
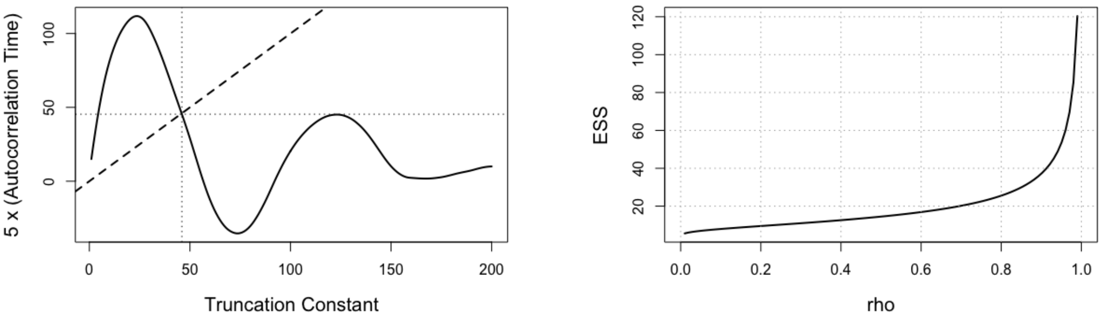


Figure 5.2: Data with non-stationary temporal structure. Triangles denote the location of the blocks using eq. (5.7) and diamonds denote the block locations using binary refinement.

non-stationarity by using local approximate Gaussian processes (as described in Section 1.3.3). If multiple partitions exist at any step, we need a criteria for choosing between them. There are at least two simple strategies: (i) choose the partition which maintains the best balance (i.e. $|B_1| \approx |B_2|$) or (ii) choose the partition which leads to the smallest block (i.e. $\min\{|B_1|, |B_2|\}$) with the hope that the larger block can be further refined. Speedup can be obtained by searching for a partition from (i) the median value and working out or (ii) from the extremes and working towards to the middle depending on which strategy is selected.

5.3.4 Illustration of Block Selection

Consider the $n = 200$ data points shown in Figure 5.2, which represent samples from an empirical discrepancy function in the form of eq. (5.8). The input locations $x_1, x_2, \cdots x_n$ are sampled uniformly at random from the unit interval and sorted (i.e. $x_i < x_{i+1}$) for simplicity. The response values were simulated using a non-stationary covariance structure, setting $\mu = 0$, $\phi = 1$, $\sigma = 0.01$ and $\kappa(x_1, x_2) = 10^3 \max\{x_1, x_2\}$.



(a) Selection of the truncation constant for estimating n_{eff} as in eq. (5.7).

(b) Correspondence of n_{eff} and ρ as in eq. (5.11)

Figure 5.3: Estimating the effective sample size n_{eff}.

Using the sample autocorrelation approach of eq. (5.7), we select a truncation value of $L = 56$ (see Figure 5.3a) leading to an estimated autocorrelation time of $\hat{\tau} = 9.07$ and effective sample size (ESS) of $n_{\text{eff}} = 22.05$. The effective range approach of eq. (5.11) suggests similar block sizes, yielding $n_{\text{eff}} = 14.5, 20.2, 37.2$ for $\rho = 0.5, 0.7, 0.9$ respectively (see Figure 5.3b for $n_{\text{eff}}(\rho)$). Based on this analysis, it seems that $K = 20$ blocks consisting of $n_k = 10$ observations is a reasonable choice. Due to the irregularity of the design variables and the non-stationarity of the temporal structure, this application may benefit from a more flexible structure. For instance, when x is small, there is a strong temporal structure and larger blocks may be needed. The greedy binary refinement algorithm provides this flexibility and constructs blocks of varying sizes as seen in Table 5.1

Table 5.1: A summary of the greedy binary refinement algorithm for block construction using parameter combinations $\rho \in \{0.5, 0.7, 0.9\}$ and $\iota \in \{5, 10, 15\}$. The number of blocks depend on the parameters and the size of the blocks vary

ρ	ι	# of Blocks	Smallest Block	Largest Block
0.9	5	27	5	13
0.9	10	13	12	25
0.9	15	8	25	25
0.7	5	19	6	13
0.7	10	10	11	25
0.7	15	8	25	25
0.5	5	16	5	25
0.5	10	8	25	25
0.5	15	8	25	25

.

5.4 Modularizing the Discrepancy Parameters

In this section, we explore the potential benefits of modularization for the discrepancy function in model calibration. We will explore and compare four general strategies for handling any combination the model discrepancy parameters $\boldsymbol{\psi} = (\mu, \phi, \kappa, \sigma)$ which are described below.

i) Plug-in approach using nominal values (i.e. prior mode) $\hat{\boldsymbol{\psi}}_0$. A common and reasonable example of this strategy is the choice $\mu = 0$, asserting that the computer model is unbiased for the true process across the input space.

ii) Plug-in approach using the cross validation estimates $\hat{\boldsymbol{\psi}}_{CV}$. This may or may not outperform the first approach, depending on the quality of the nominal values. Since the CV criterion are based on producing output which fits the data, identifiability issues may not be addressed.

iii) Use CV results to build a prior distribution $\pi_\psi(\cdot)$ and perform standard BMC. Although construction of a good prior may help to an extent, it is generally understood that BMC calibration parameters and discrepancy are not jointly identifiable [3].

iv) Use CV results to build a prior distribution $\pi_\psi(\cdot)$ and perform BMC with modularization. We view this as an extension of the plug-in approach with better uncertainty quantification properties. However, it is heavily dependent on our ability to construct a reasonable prior.

5.4.1 Modularization for the Bias

In Bayesian model calibration problems, the mean of the discrepancy function is commonly set to 0. This can be justified by noting that

$$\eta(\cdot,\cdot) + \delta_\mu(\cdot) = \eta(\cdot,\cdot) + \mu + \delta_0(\cdot),$$

where $\delta_a(\cdot)$ is a GP with mean a. In other words, including μ in the discrepancy model is equivalent to adding a calibration parameter to the computer model, which may already have a parameter (or combination of parameters) which accomplishes this. With this in mind, it makes very little sense to incorporate μ as a parameter in the standard BMC framework [94]. On the other hand, it may be worth considering and aggregating the impact of a fixed μ, across a small range of reasonable values.

To illustrate this idea, we will adapt eq. (1.10) to form a simple example. Consider the true process

$$\begin{aligned} &\zeta(x) = -w(x|1,0.8)w(c|1,0.6),\ x \in (0,1), \\ &w(x|\alpha,\gamma) = \exp\left(-\alpha(4x-3)^2\right) + \exp\left(-\gamma(4x-1)^2\right) - 0.1\sin\left(8(4x-1.99)\right). \end{aligned} \tag{5.13}$$

The computer model is taken to have the same structure, but we mimic missing physics by ignoring the oscillatory nature

$$\begin{aligned} \eta(x,\boldsymbol{\theta}) = \eta(x,\alpha,\gamma_1,\gamma_2) &= -v(x|\alpha,\gamma_1)v(c|\alpha,\gamma_2),\ x \in (0,1), \\ v(x|\alpha,\gamma) &= \exp\left(-\alpha(4x-3)^2\right) + \exp\left(-\gamma(4x-1)^2\right). \end{aligned} \tag{5.14}$$

For this problem, $x \in [0,1]$ is a design variable, $\boldsymbol{\theta} = (\alpha,\gamma_1,\gamma_2)$ are calibration

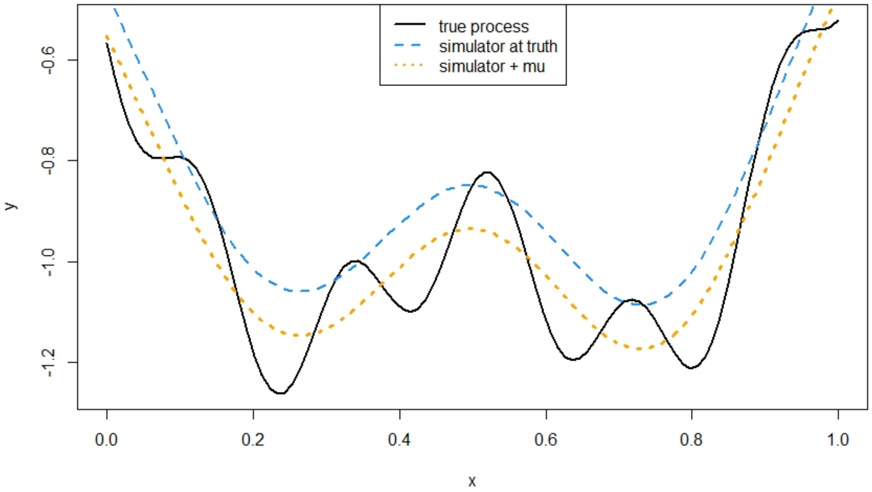


Figure 5.4: True process and simulator for a slice through the Gramacy-Lee surface.

parameters with the notion $\boldsymbol{\theta}_\star = (1, 0.8, 0.6)$ and c is a known constant which we can use to control the form of the true discrepancy function. Setting $c = 0.625$, leads to a simulator which is biased on average across the design variable space thereby violating the usual assumption of $\mu = 0$. Figure 5.4 shows the true process and the simulator $\eta(x, \boldsymbol{\theta}_\star)$ which demonstrates a clear upwards shift. This figure also shows $\eta(x, \boldsymbol{\theta}_\star) + \mu_\star$, where $\mu_\star = -0.0867$, which leads to a mean zero discrepancy function.

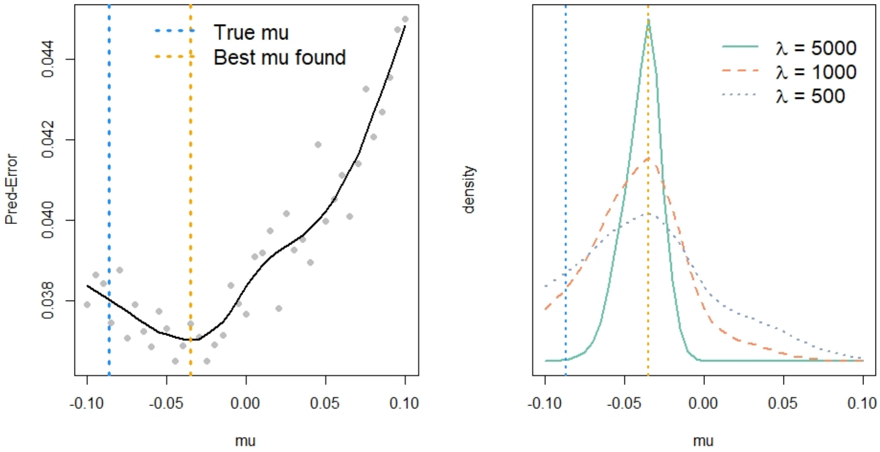


Figure 5.5: Cross validation loss and corresponding prior distribution(s) for μ.

With $\sigma = 0.1$, we simulate data from this model using $n = 100$ design points x_i equally spaced over the unit interval. Using the effective range approach from Section 5.3.2, we determine that $K = 5$ blocks with 20 points is an acceptable partition for cross validation. The left panel of Figure 5.5 shows the results of cross validation for μ leading to the CV estimate $\hat{\mu}_{CV} = -0.035$. Although CV was unable to precisely capture the true bias $\mu_\star = -0.0867$ it correctly identifies that a shift is needed. For purposes of modularization, the CV results must be turned into a prior for μ. The right panel of Figure 5.5 demonstrates the process described in Section 5.2 for λ values of $5 \cdot 10^2$, 10^3 and $5 \cdot 10^3$. Although formal criteria needs to be selected, we take a conservative ad-hoc approach here and select $\lambda = \cdot 10^3$.

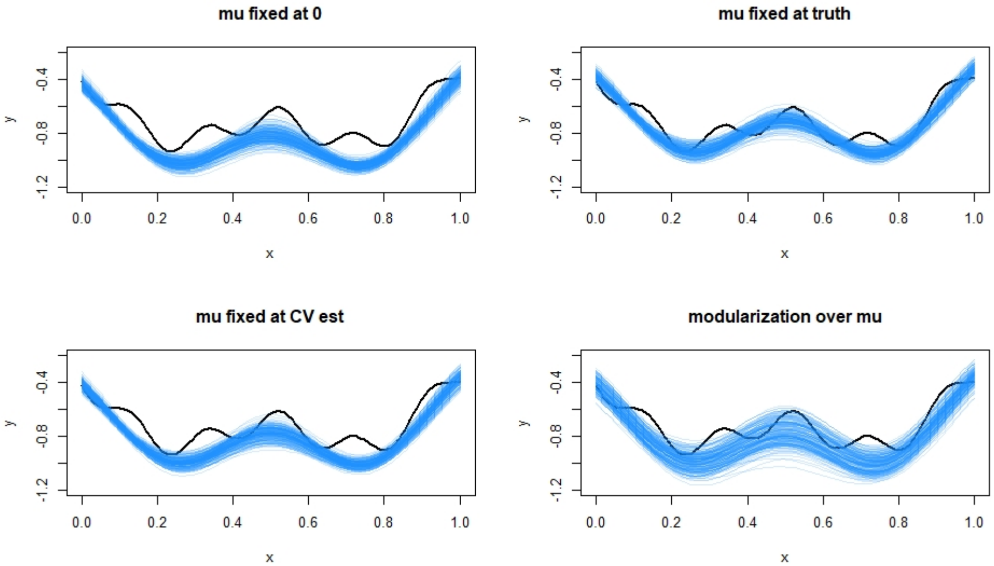


Figure 5.6: 200 posterior predictions of the perturbed true process for each approach. All models were calibrated using data from the physical system with $c = 0.625$ and the perturbed system is represented by $c' = 0.5$.

Table 5.2: MSE values for the extrapolative setting using four approaches. First row represents the prediction MSE using the posterior mean of the calibration parameters and the second row denotes the average MSE across the 200 posterior samples.

	True	Nominal	CV	Modular.
	$\mu_\star = -0.087$	$\mu = 0$	$\hat{\mu}_{CV} = -0.035$	$\lambda = 1000.$
$MSE(\hat{\boldsymbol{\theta}})$	0.006	0.021	0.013	0.012
$\int MSE(\boldsymbol{\theta})\pi(\boldsymbol{\theta}\|\mathcal{D})d\boldsymbol{\theta}$	0.72	2.31	1.503	1.755

All of the calibrated models should be able to predict the true process fairly well, due to the fact that interpolation is a well-posed problem in Bayesian model calibration. A more difficult test is to see how well the calibrated models can generalize or extrapolate to a new but related physical system. We thus perturb the system by the constant c in equations 5.13 and 5.14 to $c' = 0.5$. Figure 5.6 and Table 5.2 summarize the results of this extrapolative setting, by making 200 posterior predictions of the perturbed true process for (i) μ fixed at its "true" value $\mu_\star = -0.0867$, (ii) μ fixed at the nominal value 0, (iii) μ fixed at the CV estimate $\hat{\mu}_{CV} = -0.035$ and (iv) μ modularized over its CV prior (see Figure 5.5). Setting μ at ground truth leads to the best results but is impossible to do in practice. The modularization approach seems to best address the bias of the predictions at the cost of additional variance.

5.5 Conclusions & Future Work

This chapter is relatively exploratory in nature, outlining a strategy for using cross validation to inform the parameters of the model discrepancy prior. This step is usually skipped, for reasons explained in Section 1.4.3 and [3, 94], in favor of a plug-in approach. Recent work [117, 148] has shown that the discrepancy prior becomes a permanent fixture of the calibration parameter posterior, and thus nominal plug-in can have unintended consequences. Using CV to estimate the discrepancy parameters may not solve the underlying issue, especially in the case of the inverse-problem, so we propose combining this approach with the modularization framework described in Chapter 4.

There are many areas for future work here, including more comprehensive application of the strategy proposed here. Using a test-bed of calibration problems, we should apply the CV-modularization approach for all combinations of hyper parameters $\boldsymbol{\psi} = (\mu, \kappa, \phi, \sigma)$. An immediate area of need is the generalization of these ideas from temporal to spatial settings (i.e. $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^p$). Application of these methods also requires that the results of CV can be used to construct a prior distribution for the discrepancy parameters, and selection of λ in eq. (5.3) and eq. (5.6) needs to be discussed in far greater detail. In the binary refinement algorithm, our current implementation uses a global notion of effective range (see Section 5.3.2). The implementation can be improved by allowing for a local definition of effective range, perhaps by using local approximate GPs, which can lead to drastically better results when the temporal (or spatial) structure of the data is non-stationary.

Appendices

Appendix A

Pseudocode for SLAP-GP and LEAP-GP Emulation

A.1 SLAP-GP-PREDICT

Algorithm 1 Pseudocode for prediction with SLAP-GP as described in section 2.3.

```
Inputs:
    $\boldsymbol{x}_{\text{new}}$ - desired prediction location
    $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{\eta})$ - the training data
    $\boldsymbol{\mathcal{H}}$ - the current set of prediction hubs
    $\rho$ - a parameter between 0 and 1
    $c$ - local neighborhood size

function SLAP-GP-PREDICT($\boldsymbol{x}_{\text{new}}, \boldsymbol{\mathcal{H}}, \mathcal{D}, \rho, c$)
    if $\boldsymbol{\mathcal{H}} \neq \emptyset$ then
        $\mathcal{H}^\star \leftarrow \arg\min_{\mathcal{H} \in \boldsymbol{\mathcal{H}}} \{\mathcal{H}.\kappa \;\times\; d(\boldsymbol{x}_{\text{new}}, \mathcal{H}.\boldsymbol{x})\}$
        if $d(\boldsymbol{x}_{\text{new}}, \mathcal{H}^\star.\boldsymbol{x}) \leq \sqrt{-\log\rho \div \mathcal{H}^\star.\kappa}$ then
            for $i = 1$ to $c$ do
                $j_i \leftarrow (\mathcal{H}^\star.J)_i$
                $r_i \leftarrow R(\boldsymbol{x}_{\text{new}}, \mathcal{H}^\star.\boldsymbol{x} | \mathcal{H}^\star.\kappa)$        ▷ see eq. (1.12)
            $\hat{y} \leftarrow r^\top (\mathcal{H}^\star.\boldsymbol{\psi})$
            return $(\hat{y}, \boldsymbol{\mathcal{H}})$
    else
        $J_{\text{new}} \leftarrow$ build-neighborhood($\boldsymbol{x}_{\text{new}}, \mathcal{D}, c$)
        $(\hat{y}, \kappa_{\text{new}}, \boldsymbol{\psi}_{\text{new}}) \leftarrow$ train-GP($\boldsymbol{x}_{\text{new}}, J_{\text{new}}, \mathcal{D}$)
        $\boldsymbol{\mathcal{H}} \leftarrow \boldsymbol{\mathcal{H}} \cup$ new-hub($\boldsymbol{x}_{\text{new}}, J_{\text{new}}, \kappa_{\text{new}}, \boldsymbol{\psi}_{\text{new}}$)
    return $(\hat{y}, \boldsymbol{\mathcal{H}})$
```

A.2 LEAP-GP-BUILD, LEAP-GP-PREDICT

Algorithm 2 Pseudocode for training a LEAP-GP emulator and using it for prediction as described in section 2.4.

```
Inputs:
  $\boldsymbol{x}_{\text{new}}$ - desired prediction location
  $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{\eta})$ - the training data
  $H$ - the number of hub locations, between 1 and $d$
  $c$ - local neighborhood size
  $\boldsymbol{\mathcal{H}}$ - a of prediction hubs

function LEAP-GP-BUILD($\boldsymbol{x}_{\text{new}}, \mathcal{D}, H, c$)
    $\boldsymbol{j} \leftarrow$ PAM-Index($\mathcal{D}.\boldsymbol{X}, H$)
    for $h = 1$ to $H$ do
        $j \leftarrow \boldsymbol{k}_h$
        $J_{\text{new}} \leftarrow$ build-neighborhood($\mathcal{D}.\boldsymbol{X}_j, \mathcal{D}, c$)
        $(\kappa_{\text{new}}, \boldsymbol{\psi}_{\text{new}}) \leftarrow$ train-GP($\boldsymbol{x}_{\text{new}}, J_{\text{new}}, \mathcal{D}$)
        $\mathcal{H}_h \leftarrow$ new-hub($\mathcal{D}.\boldsymbol{X}_j, J_{\text{new}}, \kappa_{\text{new}}, \boldsymbol{\psi}_{\text{new}}$)
    $\boldsymbol{\mathcal{H}} \leftarrow \{\mathcal{H}_1, \mathcal{H}_2, \cdots \mathcal{H}_h\}$
    return $\boldsymbol{\mathcal{H}}$

function LEAP-GP-PREDICT($\boldsymbol{x}_{\text{new}}, \boldsymbol{\mathcal{H}}, \mathcal{D}$)
    $\mathcal{H}^\star \leftarrow \arg\min_{\mathcal{H} \in \boldsymbol{\mathcal{H}}} \{\mathcal{H}.\kappa \ \times \ d(\boldsymbol{x}_{\text{new}}, \mathcal{H}.\boldsymbol{x})\}$
    for $i = 1$ to $c$ do
        $j_i \leftarrow (\mathcal{H}^\star.J)_i$
        $r_i \leftarrow R(\boldsymbol{x}_{\text{new}}, \mathcal{H}^\star.\boldsymbol{x} | \mathcal{H}^\star.\kappa)$        ▷ see eq. (1.12)
    $\hat{y} \leftarrow r^\top (\mathcal{H}^\star.\boldsymbol{\psi})$
    return $\hat{y}$
```

Appendix B

Pseudocode for the ECP and Sequential ECP Algorithms

B.1 ECP-SAMPLE

Algorithm 3 Pseudocode for sampling from the modularization posterior (see eq. (4.10)) using the ECP algorithm described in section 4.4.2.

```
Inputs:
    $L$ - the budget
    $M$ - number of samples requested
    $\pi$ - prior distribution for modularization parameters
    $\mathcal{F}$ - distributional assumption for conditional posterior.
    $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$ - the field data

function ECP-SAMPLE($L, M, \pi, \mathcal{F}, \mathcal{D}$)
    $(\boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2, \cdots \boldsymbol{\gamma}^L) \overset{\text{iid}}{\sim} \pi$        ▷ Consider using LHS (section 1.5.1)
    for $\ell = 1$ to $L$ do
        $(\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \cdots \boldsymbol{\alpha}^p) \sim \pi(\boldsymbol{\alpha} | \boldsymbol{\gamma}^\ell, \mathcal{D})$        ▷ i.e. using MCMC
        $(\hat{\psi}_1^\ell, \hat{\psi}_2^\ell, \cdots \hat{\psi}_r^\ell) \leftarrow$ estimate-parameters($(\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \cdots \boldsymbol{\alpha}^p), \mathcal{F}$)
    for $j = 1$ to $r$ do
        $\widehat{\psi_j(\cdot)} \leftarrow$ train-GP($(\boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2, \cdots \boldsymbol{\gamma}^L), (\hat{\psi}_j^1, \hat{\psi}_j^2, \cdots \hat{\psi}_j^L)$)
    for $m = 1$ to $M$ do
        $\boldsymbol{\gamma}^m \sim \pi$
        $\boldsymbol{\alpha}^m \sim \mathcal{F}\left(E(\widehat{\psi_1(\boldsymbol{\gamma}^m)}), \cdots E(\widehat{\psi_r(\boldsymbol{\gamma}^m)})\right)$        ▷ Using eq. (1.13)
    return $(\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \cdots \boldsymbol{\alpha}^M)$
```

B.2 SEQUENTIAL-ECP-SAMPLE

Algorithm 4 Pseudocode for sampling from the modularization posterior (see eq. (4.10)) using the Sequential ECP algorithm.

```
Inputs:
  $L$ - the budget
  $M$ - number of samples requested
  $\pi$ - prior distribution for modularization parameters
  $\mathcal{F}$ - distributional assumption for conditional posterior.
  $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$ - the field data
  $L_0$ - the budget for the build phase
  $N_c$ - size of the candidate sets
  $I(\cdot|\cdot)$ an improvement function

function SEQUENTIAL-ECP-SAMPLE($L, M, \pi, \mathcal{F}, \mathcal{D}, L_0, N_c, I(\cdot|\cdot)$)
    $(\boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2, \cdots \boldsymbol{\gamma}^{L_0}) \overset{\text{iid}}{\sim} \pi$        ▷ Consider using LHS (section 1.5.1)
    for $\ell = 1$ to $L_0$ do
        $(\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \cdots \boldsymbol{\alpha}^p) \sim \pi(\boldsymbol{\alpha}|\boldsymbol{\gamma}^\ell, \mathcal{D})$        ▷ i.e. using MCMC
        $(\hat{\psi}_1^\ell, \hat{\psi}_2^\ell, \cdots \hat{\psi}_r^\ell) \leftarrow$ estimate-parameters$((\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \cdots \boldsymbol{\alpha}^p), \mathcal{F})$
    $\Gamma_c \leftarrow \boldsymbol{\gamma}_c^1, \cdots \boldsymbol{\gamma}_c^{N_c} \overset{\text{iid}}{\sim} \pi$
    for $\ell = L_0 + 1$ to $L$ do
        for $j = 1$ to $r$ do
            $\widehat{\psi_j^\ell(\cdot)} \leftarrow$ train-GP$((\boldsymbol{\gamma}^1, \boldsymbol{\gamma}^2, \cdots \boldsymbol{\gamma}^{\ell-1}), (\hat{\psi}_j^1, \hat{\psi}_j^2, \cdots \hat{\psi}_j^{\ell-1}))$
        $\boldsymbol{\gamma}^\ell \leftarrow \arg\max_{\boldsymbol{\gamma}_c \in \Gamma_c} I\left(\boldsymbol{\gamma}_c | \left(\widehat{\psi_r^\ell(\cdot)}, \cdots \widehat{\psi_r^\ell(\cdot)}\right)\right)$
    for $m = 1$ to $M$ do
        $\boldsymbol{\gamma}^m \sim \pi$
        $\boldsymbol{\alpha}^m \sim \mathcal{F}\left(E(\widehat{\psi_j^L(\boldsymbol{\gamma}^m)}), \cdots E(\widehat{\psi_r^L(\boldsymbol{\gamma}^m)})\right)$        ▷ Using eq. (1.13)
    return $(\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \cdots \boldsymbol{\alpha}^M)$
```

Appendix C

Pseudocode for the Greedy Binary Refinement Algorithm

C.1 IS-ACCEPTABLE

Algorithm 5 Routine to check the acceptable partition criteria of eq. (5.12).

```
Inputs:
$\mathcal{P} = (B_1, B_2, \cdots B_k)$ - a partition of the field data
$\nu$ - the global correlation parameter
$\iota$ - parameter describing the required number of hard-to-predict points
$\rho$ - correlation parameter which defines hard-to-predict

function IS-ACCEPTABLE($\mathcal{P}, \nu, \iota, \rho$)
    $K \leftarrow |\mathcal{P}|$
    $j \leftarrow 1$
    for $k = 1$ to $K$ do
        $i \leftarrow 0$
        for $x \in \mathcal{P}.B_k$ do
            $d_\star \leftarrow \min_{\boldsymbol{x}' \notin \mathcal{P}.B_k} d(\boldsymbol{x}, \boldsymbol{x}')$
            if $d_\star \leq \sqrt{-\log \rho \div \nu}$ then
                $i \leftarrow i + 1$
        if $i < \iota$ then
            return FALSE
    return TRUE
```

C.2 BINARY-PARTITION, SEARCH-FROM-MIDDLE, SEARCH-FROM-END

Algorithm 6 Routine to find an acceptable binary partition of a block B.

```
Inputs:
    $B$ - a single block containing $x_1 < x_2 < \cdots < x_n$ sorted locations
    $\nu$ - the global correlation parameter
    $\iota$ - parameter describing the required number of hard-to-predict points
    $\rho$ - correlation parameter which defines hard-to-predict

function BINARY-PARTITION($B, \nu, \iota, \rho$)
    $n \leftarrow |B|$
    for $i = 1$ to $n$ do
        $i_\star \leftarrow$ SEARCH-FROM-???($i, n$)          ▷ Use either sub-routine here
        $B_1 \leftarrow (x_1, \cdots x_{i_\star})$
        $B_2 \leftarrow (x_{i_\star+1}, \cdots x_n)$
        if IS-ACCEPTABLE($(B_1, B_2), \nu, \iota, \rho$) then
            return $(B_1, B_2)$
    return $B$

function SEARCH-FROM-MIDDLE($i, n$)
    $i_\star \leftarrow \lfloor \frac{n+1}{2} \rfloor + (-1)^i \left( \lceil \frac{i}{2} \rceil - (i \% 2) \right)$
    return $i_\star$                                        ▷ For most balanced partition

function SEARCH-FROM-ENDS($i, n$)
    $i_\star \leftarrow (i \% 2) n^{i \% 2} + (-1)^i \left( \lceil \frac{i}{2} \rceil - (i \% 2) \right)$
    return $i_\star$                                        ▷ For partition with smallest block
```

C.3 REFINE-PARTITION, BINARY-REFINEMENT

Algorithm 7 The greedy binary refinement algorithm described in section 5.3.3.

```
Inputs:
    $\mathcal{P}$ - a partition of the field data
    $\nu$ - the global correlation parameter
    $\iota$ - parameter describing the required number of hard-to-predict points
    $\rho$ - correlation parameter which defines hard-to-predict

function REFINE-PARTITION($\mathcal{P}, \nu, \iota, \rho$)
    $\mathcal{P}_{\text{new}} \leftarrow \emptyset$
    $K \leftarrow |\mathcal{P}|$
    for $k = 1$ to $K$ do
        $\mathcal{P}_{\text{new}} \leftarrow \mathcal{P}_{\text{new}} \cup$ BINARY-PARTITION($\mathcal{P}.B_k, \nu, \iota, \rho$)
    return $\mathcal{P}_{\text{new}}$

function BINARY-REFINEMENT($\mathcal{P}, \nu, \iota, \rho$)
    repeat
        $\mathcal{P}' \leftarrow \mathcal{P}$
        $\mathcal{P} \leftarrow$ REFINE-PARTITION($\mathcal{P}, \nu, \iota, \rho$)
    until $\mathcal{P} = \mathcal{P}'$
    return $\mathcal{P}$
```

References

[1] J. An and A. Owen, *Quasi-regression*, Journal of complexity **17** (2001), no. 4, 588–607.

[2] D. Anderson, L. Rueda A .and Cagigal, J. A. A. Antolinez, F. J. Mendez, and P. Ruggiero, *Time-varying emulator for short and long-term analysis of coastal flood hazard potential*, Journal of Geophysical Research: Oceans (2019).

[3] P. D. Arendt, D. W. Apley, and W. Chen, *Quantification of model uncertainty: Calibration, model discrepancy, and identifiability*, Journal of Mechanical Design **134** (2012), no. 10, 100908.

[4] ———, *A preposterior analysis to predict identifiability in the experimental calibration of computer models*, IIE Transactions **48** (2016), no. 1, 75–88.

[5] J. R. Asay, *Isentropic compression experiments on the z accelerator*, Aip conference proceedings, 2000, pp. 261–266.

[6] L. M. Barker, *High-pressure quasi-isentropic impact experiments*, Shock waves in condensed matter 1983, 1984, pp. 217–224.

[7] J. F. Barnes, P. J. Blewett, R. G. McQueen, K. A. Meyer, and D. Venable, *Taylor instability in solids*, Journal of Applied Physics **45** (1974), no. 2, 727–732.

[8] S. Barnett, *Matrix methods for engineers and scientists* (1979).

[9] L. S. Bastos and A. O'Hagan, *Diagnostics for gaussian process emulators*, Technometrics **51** (2009), no. 4, 425–438.

[10] G. E. Bates and J. Neyman, *Contributions to the theory of accident proneness. 1. an optimistic model of the correlation between light and severe accidents*, California University Berkeley, 1952.

[11] M. J. Bayarri, J. O. Berger, J. Cafeo, G. Garcia-Donato, F. Liu, J. Palomo, R. J. Parthasarathy, R. Paulo, J. Sacks, and D. Walsh, *Computer model validation with functional output*, The Annals of Statistics **35** (2007), no. 5, 1874–1906.

[12] J. L. Bentley, *Multidimensional binary search trees used for associative searching*, Communications of the ACM **18** (1975), no. 9, 509–517.

[13] J. O. Berger, V. De Oliveira, and B. Sansó, *Objective bayesian analysis of spatially correlated data*, Journal of the American Statistical Association **96** (2001), no. 456, 1361–1374.

[14] K. Beven and J. Freer, *Equifinality, data assimilation, and uncertainty estimation in mechanistic modelling of complex environmental systems using the glue methodology*, Journal of hydrology **249** (2001), no. 1-4, 11–29.

[15] A. Bhadra, J. Datta, N. G. Polson, and B. T. Willard, *Lasso meets horseshoe*, arXiv preprint arXiv:1706.10179 (2017).

[16] K. S. Bhat, M. Haran, M. Goes, and M. H. Chen, *Computer model calibration with multivariate spatial output: A case study*, Frontiers of Statistical Decision Making and Bayesian Analysis (2010), 168–184.

[17] A. Bhattacharya, D. Pati, N. S. Pillai, and D. B. Dunson, *Dirichlet–laplace priors for optimal shrinkage*, Journal of the American Statistical Association **110** (2015), no. 512, 1479–1490.

[18] P. G. Bissiri, C. C. Holmes, and S. G. Walker, *A general framework for updating belief distributions*, Journal of the Royal Statistical Society. Series B, Statistical methodology **78** (2016), no. 5, 1103.

[19] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, *Variational inference: A review for statisticians*, Journal of the American Statistical Association **112** (2017), no. 518, 859–877.

[20] J. L. Brown, C. S. Alexander, J. R. Asay, T. J. Vogler, D. H. Dolan, and J. L. Belof, *Flow strength of tantalum under ramp compression to 250 gpa*, Journal of Applied Physics **115** (2014), no. 4, 043530.

[21] J. L. Brown and L. B. Hund, *Estimating material properties under extreme conditions by using bayesian model calibration with functional outputs*, Journal of the Royal Statistical Society: Series C (Applied Statistics) **67** (2018), no. 4, 1023–1045.

[22] R. Brun, M. Kühni, H. Siegrist, W. Gujer, and P. Reichert, *Practical identifiability of asm2d parameters - systematic selection and tuning of parameter subsets*, Water research **36** (2002), no. 16, 4113–4127.

[23] R. Brun, P. Reichert, and H. R. Künsch, *Practical identifiability analysis of large environmental simulation models*, Water Resources Research **37** (2001), no. 4, 1015–1030.

[24] J. Brynjarsdottir and A. O'Hagan, *Learning about physical parameters: The importance of model discrepancy*, Inverse Problems **30** (2014), no. 11, 114007.

[25] T. Cai and W. Liu, *Adaptive thresholding for sparse covariance matrix estimation*, Journal of the American Statistical Association **106** (2011), no. 494, 672–684.

[26] R. Carnell, *lhs: Latin hypercube samples*, 2020. R package version 1.0.2.

[27] C. M. Carvalho, N. G. Polson, and J. G. Scott, *Handling sparsity via the horseshoe*, Artificial intelligence and statistics, 2009, pp. 73–80.

[28] G. Casella and R. L. Berger, *Statistical inference*, 2nd ed., Duxbury Pacific Grove, CA, 2002.

[29] G. Casella, C. P. Robert, and M. T. Wells, *Generalized accept-reject sampling schemes*, A festschrift for herman rubin, 2004, pp. 342–347.

[30] H. Chen and W. J. Welch, *Sequential computer experimental design for estimating an extreme probability or quantile*, arXiv preprint arXiv:1908.05357 (2019).

[31] S. Chib and E. Greenberg, *Understanding the metropolis-hastings algorithm*, The american statistician **49** (1995), no. 4, 327–335.

[32] C. Chivers, *Mhadaptive: General markov chain monte carlo for bayesian inference using adaptive metropolis-hastings sampling*, 2012. R package version 1.1-8.

[33] W. S. Cleveland, *Robust locally weighted regression and smoothing scatterplots*, Journal of the American statistical association **74** (1979), no. 368, 829–836.

[34] ______, *Lowess: A program for smoothing scatterplots by robust locally weighted regression*, American Statistician **35** (1981), no. 1, 54.

[35] J. Cohen, *The earth is round (p < .05)*, What if there were no significance tests?, 2016, pp. 69–82.

[36] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, *Active learning with statistical models*, Journal of artificial intelligence research **4** (1996), 129–145.

[37] P. G. Constantine, *Active subspaces: Emerging ideas for dimension reduction in parameter studies*, Vol. 2, SIAM, 2015.

[38] ______, *What is uq?*, Summer school on mathematical and statistical model uncertainty, 2018.

[39] P. G. Constantine, E. Dow, and Q. Wang, *Active subspace methods in theory and practice: applications to kriging surfaces*, SIAM Journal on Scientific Computing **36** (2014), no. 4, A1500–A1524.

[40] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, MIT press, 2009.

[41] N. Cressie, *Statistics for spatial data*, John Wiley & Sons, 1991.

[42] D. G. T. Denison, B. K. Mallick, and A. F. M. Smith, *Bayesian mars*, Statistics and Computing **8** (1998), no. 4, 337–346.

[43] P. Domingos, *A few useful things to know about machine learning*, Communications of the ACM **55** (2012), no. 10, 78–87.

[44] N. R. Draper and R. C. Van Nostrand, *Ridge regression and james-stein estimation: review and comments*, Technometrics **21** (1979), no. 4, 451–466.

[45] J. Edwards, K. T. Lorenz, B. A. Remington, S. Pollaine, J. Colvin, D. Braun, B. F. Lasinski, D. Reisman, J. M. McNaney, and J. A. Greenough, *Laser-driven plasma loader for shockless compression and acceleration of samples in the solid state*, Physical review letters **92** (2004), no. 7, 075002.

[46] B. Efron, *The jackknife, the bootstrap, and other resampling plans*, Vol. 38, Siam, 1982.

[47] X. Emery, *The kriging update equations and their application to the selection of neighboring data*, Computational Geosciences **13** (2009), no. 3, 269–280.

[48] I. K. Fodor, *A survey of dimension reduction techniques*, Lawrence Livermore National Lab., CA (US), 2002.

[49] D. Francom, *Bass: Bayesian adaptive spline surfaces*, 2020. R package version 1.2.0.

[50] D. Francom and B. Sansó, *Bass: An r package for fitting and performing sensitivity analysis of bayesian adaptive spline surfaces*, Journal of Statistical Software **2** (2019).

[51] D. Francom, B. Sansó, A. Kupresanin, and G. Johannesson, *Sensitivity analysis and emulation for functional data using bayesian adaptive splines*, Statistica Sinica (2018), 791–816.

[52] D. A. Freedman, *On the asymptotic behavior of bayes' estimates in the discrete case*, The Annals of Mathematical Statistics (1963), 1386–1403.

[53] J. Friedman, T. Hastie, and R. Tibshirani, *Sparse inverse covariance estimation with the graphical lasso*, Biostatistics **9** (2008), no. 3, 432–441.

[54] J. H. Friedman, *Multivariate adaptive regression splines*, The annals of statistics (1991), 1–67.

[55] W. J. Fu, *Penalized regressions: the bridge versus the lasso*, Journal of computational and graphical statistics **7** (1998), no. 3, 397–416.

[56] A. E. Gelfand and A. F. M. Smith, *Sampling-based approaches to calculating marginal densities*, Journal of the American statistical association **85** (1990), no. 410, 398–409.

[57] A. Gelman, *Prior distributions for variance parameters in hierarchical models (comment on article by browne and draper)*, Bayesian analysis **1** (2006), no. 3, 515–534.

[58] A. Gelman, J. B Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian data analysis*, CRC press, 2013.

[59] A. Gelman, G. O. Roberts, W. R. Gilks, et al., *Efficient metropolis jumping rules*, Bayesian statistics **5** (1996), no. 599-608, 42.

[60] D. Geman S.and Geman, *Stochastic relaxation, gibbs distributions, and the bayesian restoration of images*, IEEE Transactions on pattern analysis and machine intelligence **6** (1984), 721–741.

[61] A. Genz, *An adaptive algorithm for the approximate calculation of multiple integrals*, ACM Trans. Math. Softw **17** (1991), 437–451.

[62] M. B. Giles, *Multilevel monte carlo path simulation*, Operations research **56** (2008), no. 3, 607–617.

[63] W. R. Gilks, S. Richardson, and D. Spiegelhalter, *Markov chain monte carlo in practice*, Chapman and Hall/CRC, 1995.

[64] G. H. Golub and C. F. Van Loan, *Matrix computations*, Vol. 3, JHU press, 2012.

[65] R. B. Gramacy, *lagp: large-scale spatial modeling via local approximate gaussian processes in r*, Journal of Statistical Software **72** (2016), no. 1, 1–46.

[66] R. B. Gramacy and D. W. Apley, *Local gaussian process approximation for large computer experiments*, Journal of Computational and Graphical Statistics **24** (2015), no. 2, 561–578.

[67] R. B. Gramacy, D. Bingham, J. P. Holloway, M. J. Grosskopf, C. C. Kuranz, E. Rutter, M. Trantham, and R. P. Drake, *Calibrating a large computer experiment simulating radiative shock hydrodynamics*, The Annals of Applied Statistics **9** (2015), no. 3, 1141–1168.

[68] R. B. Gramacy and H. K. H. Lee, *Optimization under unknown constraints*, Bayesian Statistics **9** (2011), no. 9, 229. eds. Bernardo, J. and Bayarri, M. J. and Berger, J. O. and Dawid, A. P. and Heckerman, D. and Smith, A. F. M. and West, M.

[69] R. B. Gramacy and N. G. Polson, *Particle learning of gaussian process models for sequential design and optimization*, Journal of Computational and Graphical Statistics **20** (2011), no. 1, 102–118.

[70] Y. Guan, C. Sampson, J. D. Tucker, W. Chang, A. Mondal, M. Haran, and D. Sulsky, *Computer model calibration based on image warping metrics: an application for sea ice deformation*, Journal of Agricultural, Biological and Environmental Statistics **24** (2019), no. 3, 444–463.

[71] L. Györfi, I. Vajda, and E. van der Meulen, *Minimum kolmogorov distance estimates of parameters and parametrized distributions*, Metrika **43** (1996), no. 1, 237–255.

[72] H. Haario, E. Saksman, and J. Tamminen, *An adaptive metropolis algorithm*, Bernoulli **7** (2001), no. 2, 223–242.

[73] W. K. Hastings, *Monte carlo sampling methods using markov chains and their applications* (1970).

[74] A. Hernandez, *Model calibration with neural networks*, Available at SSRN 2812140 (2016).

[75] O. Heuzé, *General form of the mie–grüneisen equation of state*, Comptes Rendus Mecanique **340** (2012), no. 10, 679–687.

[76] D. Higdon, J. Gattiker, B. Williams, and M. Rightley, *Computer model calibration using high-dimensional output*, Journal of the American Statistical Association **103** (2008), no. 482, 570–583.

[77] D. Higdon, M. Kennedy, J. C. Cavendish, J. A. Cafeo, and R. D. Ryne, *Combining field data and computer simulations for calibration and prediction*, SIAM Journal on Scientific Computing **26** (2004), no. 2, 448–466.

[78] M. Hlavac, *stargazer: Well-formatted regression and summary statistics tables*, Central European Labour Studies Institute (CELSI), Bratislava, Slovakia, 2018. R package version 5.2.2.

[79] A. E. Hoerl and R. W. Kennard, *Ridge regression: Biased estimation for nonorthogonal problems*, Technometrics **12** (1970), no. 1, 55–67.

[80] M. D. Hoffman and A. Gelman, *The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo.*, Journal of Machine Learning Research **15** (2014), no. 1, 1593–1623.

[81] H. Ishwaran, J. S. Rao, et al., *Spike and slab variable selection: frequentist and bayesian strategies*, The Annals of Statistics **33** (2005), no. 2, 730–773.

[82] C. S. Jackson, M. K. Sen, G. Huerta, Y. Deng, and K. P. Bowman, *Error reduction and convergence in climate prediction*, Journal of Climate **21** (2008), no. 24, 6698–6709.

[83] P. E. Jacob, L. M. Murray, C. C. Holmes, and C. P. Robert, *Better together? statistical learning in models made of modules*, arXiv preprint arXiv:1708.08719 (2017).

[84] H. Kahn and T. E. Harris, *Estimation of particle transmission by random sampling*, National Bureau of Standards applied mathematics series **12** (1951), 27–30.

[85] C. G. Kaufman, D. Bingham, S. Habib, K. Heitmann, and J. A. Frieman, *Efficient emulators of computer experiments using compactly supported correlation functions, with an application to cosmology*, The Annals of Applied Statistics **5** (2011), no. 4, 2470–2492.

[86] L. Kaufman and P. J. Rousseeuw, *Partitioning around medoids (program pam)*, Finding groups in data: an introduction to cluster analysis **344** (1990), 68–125.

[87] M. C. Kennedy and A. O'Hagan, *Bayesian calibration of computer models*, Journal of the Royal Statistical Society: Series B (Statistical Methodology) **63** (2001), no. 3, 425–464.

[88] Y. J. Kim, *Comparative study of surrogate models for uncertainty quantification of building energy model: Gaussian process emulator vs. polynomial chaos expansion*, Energy and Buildings **133** (2016), 46–58.

[89] O. M. Knio and O. P. Le Maitre, *Uncertainty propagation in cfd using polynomial chaos decomposition*, Fluid dynamics research **38** (2006), no. 9, 616.

[90] V. Kumar, *Introduction to parallel computing*, Addison-Wesley Longman Publishing Co., Inc., 2002.

[91] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE **86** (1998), no. 11, 2278–2324.

[92] R. W. Lemke, M. D. Knudson, D. E. Bliss, K. Cochrane, J. P. Davis, A. A. Giunta, H. C. Harjes, and S. A. Slutz, *Magnetically accelerated, ultrahigh velocity flyer plates for shock wave experiments*, Journal of Applied Physics **98** (2005), no. 7, 073530.

[93] Q. Li, N. Lin, et al., *The bayesian elastic net*, Bayesian analysis **5** (2010), no. 1, 151–170.

[94] F. Liu, M. J. Bayarri, and J. O. Berger, *Modularization in bayesian analysis, with emphasis on analysis of computer models*, Bayesian Analysis **4** (2009), no. 1, 119–150.

[95] J. Loeppky, D. Bingham, and W. Welch, *Computer model calibration or tuning in practice*, University of British Columbia, Vancouver, BC, Canada, Report **221** (2006).

[96] B. MacDonald, P. Ranjan, and H. Chipman, *GPfit: An R package for fitting a gaussian process model to deterministic simulator outputs*, Journal of Statistical Software **64** (2015), no. 12, 1–23.

[97] A. Marrel, B. Looss, F. Van Dorpe, and E. Volkova, *An efficient methodology for modeling complex computer codes with gaussian processes*, Computational Statistics & Data Analysis **52** (2008), no. 10, 4731–4744.

[98] M. D. McKay, R. J. Beckman, and W. J. Conover, *Comparison of three methods for selecting values of input variables in the analysis of output from a computer code*, Technometrics **21** (1979), no. 2, 239–245.

[99] A. P. Melo, D. Cóstola, R. Lamberts, and J. L. M. Hensen, *Development of surrogate models using artificial neural network for building shell energy labelling*, Energy Policy **69** (2014), 457–466.

[100] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, *Equation of state calculations by fast computing machines*, The journal of chemical physics **21** (1953), no. 6, 1087–1092.

[101] J. W. Miller and D. B. Dunson, *Robust bayesian inference via coarsening*, Journal of the American Statistical Association **114** (2019), no. 527, 1113–1125.

[102] H. Moon, A. M. Dean, and T. J. Santner, *Two-stage sensitivity-based group screening in computer experiments*, Technometrics **54** (2012), no. 4, 376–387.

[103] M. D. Morris, T. J. Mitchell, and D. Ylvisaker, *Bayesian design and analysis of computer experiments: use of derivatives in surface prediction*, Technometrics **35** (1993), no. 3, 243–255.

[104] K. Myers, E. Lawrence, M. Fugate, C. M. Bowen, L. Ticknor, J. Woodring, J. Wendelberger, and J. Ahrens, *Partitioning a large simulation as it runs*, Technometrics **58** (2016), no. 3, 329–340.

[105] B. Narasimhan, S. G. Johnson, T. Hahn, A. Bouvier, and K. Kiau, *cubature: Adaptive multivariate integration over hypercubes*, 2018. R package version 2.0.3.

[106] N. N. Narisetty and V. N. Nair, *Extremal depth for functional data and applications*, Journal of the American Statistical Association **111** (2016), no. 516, 1705–1714.

[107] J. E. Nash and J. V. Sutcliffe, *River flow forecasting through conceptual models part i—a discussion of principles*, Journal of hydrology **10** (1970), no. 3, 282–290.

[108] R. M. Neal, *Slice sampling*, Annals of statistics (2003), 705–741.

[109] W. L. Oberkampf, S. M. DeLand, B. M. Rutherford, K. V. Diegert, and K. F. Alvin, *Error and uncertainty in modeling and simulation*, Reliability Engineering & System Safety **75** (2002), no. 3, 333–357.

[110] A. OHagan, *Polynomial chaos: A tutorial and critique from a statisticians perspective*, SIAM/ASA J. Uncertainty Quantification **20** (2013), 1–20.

[111] D. Osthus, J. Gattiker, R. Priedhorsky, and S. Y. Del Valle, *Dynamic bayesian influenza forecasting in the united states with hierarchical discrepancy (with discussion)*, Bayesian Analysis **14** (2019), no. 1, 261–312.

[112] A. B. Owen, *A central limit theorem for latin hypercube sampling*, Journal of the Royal Statistical Society: Series B (Methodological) **54** (1992), no. 2, 541–551.

[113] J.S. Park, *Optimal latin-hypercube designs for computer experiments*, Journal of statistical planning and inference **39** (1994), no. 1, 95–111.

[114] T. Park and G. Casella, *The bayesian lasso*, Journal of the American Statistical Association **103** (2008), no. 482, 681–686.

[115] P. H. Peskun, *Guidelines for choosing the transition matrix in monte carlo methods using markov chains*, Journal of Computational Physics **40** (1981), no. 2, 327–344.

[116] J. Piironen and A. Vehtari, *Sparsity information and regularization in the horseshoe and other shrinkage priors*, Electronic Journal of Statistics **11** (2017), no. 2, 5018–5051.

[117] M. Plumlee, *Bayesian calibration of inexact computer models*, Journal of the American Statistical Association **112** (2017), no. 519, 1274–1285.

[118] M. Plummer, *Cuts in bayesian graphical models*, Statistics and Computing **25** (2015), no. 1, 37–43.

[119] N. G. Polson and J. G. Scott, *Shrink globally, act locally: Sparse bayesian regularization and prediction*, Bayesian statistics **9** (2010), 501–538.

[120] N. G. Polson and V. Sokolov, *Bayesian regularization: From tikhonov to horseshoe*, Wiley Interdisciplinary Reviews: Computational Statistics **11** (2019), no. 4, e1463.

[121] M. T. Pratola, S. R. Sain, D. Bingham, M. Wiltberger, and E. J. Rigler, *Fast sequential computer model calibration of large nonstationary spatial-temporal processes*, Technometrics **55** (2013), no. 2, 232–242.

[122] Y. Qian, C. S. Jackson, F. Giorgi, B. Booth, Q. Duan, C. Forest, D. Higdon, Z. J. Hou, and G. Huerta, *Uncertainty quantification in climate modeling and projection*, Bulletin of the American Meteorological Society **97** (2016), no. 5, 821–824.

[123] P. Ranjan, D. Bingham, and G. Michailidis, *Sequential experiment design for contour estimation from complex computer codes*, Technometrics **50** (2008), no. 4, 527–541.

[124] P. Ranjan, R. Haynes, and R. Karsten, *A computationally stable approach to gaussian process interpolation of deterministic computer simulation data*, Technometrics **53** (2011), no. 4, 366–378.

[125] C. E. Rasmussen, *Gaussian processes in machine learning*, Summer school on machine learning, 2003, pp. 63–71.

[126] W. Rawat and Z. Wang, *Deep convolutional neural networks for image classification: A comprehensive review*, Neural computation **29** (2017), no. 9, 2352–2449.

[127] C. Robert and G. Casella, *A short history of markov chain monte carlo: Subjective recollections from incomplete data*, Statistical Science (2011), 102–115.

[128] D. R. Roberts, V. Bahn, S. Ciuti, M. S. Boyce, J. Elith, G. Guillera-Arroita, S. Hauenstein, J. J. Lahoz-Monfort, B. Schröder, and W. Thuiller, *Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure*, Ecography **40** (2017), no. 8, 913–929.

[129] A. Robinson, T. Brunner, S. Carroll, R. Drake, C. Garasi, T. Gardiner, T. Haill, H. Hanshaw, D. Hensinger, and D. Labreche, *Alegra: An arbitrary lagrangian-eulerian multimaterial, multiphysics code*, 46th aiaa aerospace sciences meeting and exhibit, 2008, pp. 1235.

[130] S. D. Rothman, J. P. Davis, J. Maw, C. M. Robinson, K. Parker, and J. Palmer, *Measurement of the principal isentropes of lead and lead–antimony alloy to 400 kbar by quasi-isentropic compression*, Journal of Physics D: Applied Physics **38** (2005), no. 5, 733.

[131] S. Roy, *Sequential-adaptive design of computer experiments for the estimation of percentiles*, Ph.D. Thesis, 2008.

[132] K. Rumsey, G. Huerta, J. Brown, and L. B. Hund, *Dealing with measurement uncertainties as nuisance parameters in bayesian model calibration*, Journal of Uncertainty Quantification (accepted for publication) (2020).

[133] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, *Design and analysis of computer experiments*, Statistical science (1989), 409–423.

[134] A. Saltelli, *Sensitivity analysis: Could better methods be used?*, Journal of Geophysical Research: Atmospheres **104** (1999), no. D3, 3789–3793.

[135] T. J. Santner, B. J. Williams, and W. Notz, *The design and analysis of computer experiments*, Vol. 1, Springer, 2003.

[136] M. E. Savage, L. F. Bennett, D. E. Bliss, W. T. Clark, R. S. Coats, J. M. Elizondo, K. R. LeChien, H. C. Harjes, J. M. Lehr, and J. E. Maenchen, *An overview of pulse compression and power flow in the upgraded z pulsed power driver*, 2007 16th ieee international pulsed power conference, 2007, pp. 979–984.

[137] T. Schenpper, *Location problems with k-max functions*, Ph.D. Thesis, 2017.

[138] E. Schubert and P. J. Rousseeuw, *Faster k-medoids clustering: improving the pam, clara, and clarans algorithms*, International conference on similarity search and applications, 2019, pp. 171–187.

[139] E. Snelson and Z. Ghahramani, *Sparse gaussian processes using pseudo-inputs*, Advances in neural information processing systems, 2006, pp. 1257–1264.

[140] A. Sokal, *Monte carlo methods in statistical mechanics: foundations and new algorithms*, Functional integration, 1997, pp. 131–192.

[141] J. Sreekanth and B. Datta, *Multi-objective management of saltwater intrusion in coastal aquifers using genetic programming and modular neural network based surrogate models*, Journal of Hydrology **393** (2010), no. 3-4, 245–256.

[142] M. L. Stein, Z. Chi, and L. J. Welty, *Approximating likelihoods for large spatial data sets*, Journal of the Royal Statistical Society: Series B (Statistical Methodology) **66** (2004), no. 2, 275–296.

[143] M. Stone, *Cross-validatory choice and assessment of statistical predictions*, Journal of the Royal Statistical Society: Series B (Methodological) **36** (1974), no. 2, 111–133.

[144] H. F. Stripling, R. G. McClarren, C. C. Kuranz, M. J. Grosskopf, and B. R. Rutter E .and Torralva, *A calibration and data assimilation method using the bayesian mars emulator*, Annals of Nuclear Energy **52** (2013), 103–112.

[145] S. Surjanovic and D. Bingham, *Virtual library of simulation experiments: test functions and datasets*, Simon Fraser University, Burnaby, BC, Canada, accessed May **13** (2013), 2015.

[146] R. Tibshirani, *Regression shrinkage and selection via the lasso*, Journal of the Royal Statistical Society: Series B (Methodological) **58** (1996), no. 1, 267–288.

[147] R. K. Tripathy and I. Bilionis, *Deep uq: Learning deep neural network surrogate models for high dimensional uncertainty quantification*, Journal of computational physics **375** (2018), 565–588.

[148] R. Tuo and C. F. J. Wu, *Efficient calibration for imperfect computer models*, The Annals of Statistics **43** (2015), no. 6, 2331–2352.

[149] ———, *A theoretical framework for calibration in computer models: parametrization, estimation and convergence properties*, SIAM/ASA Journal on Uncertainty Quantification **4** (2016), no. 1, 767–795.

[150] ———, *Prediction based on the kennedy-o'hagan calibration model: asymptotic consistency and other properties*, Statistica Sinica (2018), 743–759.

[151] M. Van der Laan, K. Pollard, and J. Bryan, *A new partitioning around medoids algorithm*, Journal of Statistical Computation and Simulation **73** (2003), no. 8, 575–584.

[152] S. van der Pas, B. Szabó, and A. van der Vaart, *Uncertainty quantification for the horseshoe (with discussion)*, Bayesian Analysis **12** (2017), no. 4, 1221–1274.

[153] T. Van Erven and P. Harremos, *Rényi divergence and kullback-leibler divergence*, IEEE Transactions on Information Theory **60** (2014), no. 7, 3797–3820.

[154] A. V. Vecchia, *Estimation and model identification for continuous spatial processes*, Journal of the Royal Statistical Society: Series B (Methodological) **50** (1988), no. 2, 297–312.

[155] A. Villagran, G. Huerta, C. S. Jackson, and M. K. Sen, *Computational methods for parameter estimation in climate models*, Bayesian Analysis **3** (2008), no. 4, 823–850.

[156] P. Vinet, J. Ferrante, J. H. Rose, and J. R. Smith, *Compressibility of solids*, Journal of Geophysical Research: Solid Earth **92** (1987), no. B9, 9319–9325.

[157] P. Vinet, J. H. Rose, J. Ferrante, and J. R. Smith, *Universal features of the equation of state of solids*, Journal of Physics: Condensed Matter **1** (1989), no. 11, 1941.

[158] G. Wahba, *Spline models for observational data*, Vol. 59, Siam, 1990.

[159] S. Wang, W. Chen, and K. L. Tsui, *Bayesian validation of computer models*, Technometrics **51** (2009), no. 4, 439–451.

[160] C. Wendland H.and Rieger, *Approximate interpolation with applications to selecting smoothing parameters*, Numerische Mathematik **101** (2005), no. 4, 729–748.

[161] B. Williams, D. Higdon, J. Gattiker, L. Moore, M. McKay, and S. Keller-McNulty, *Combining experimental data and computer simulations, with an application to flyer plate experiments*, Bayesian Analysis **1** (2006), no. 4, 765–792.

[162] P. Wolfson J.and Gilbert, *Statistical identifiability and the surrogate endpoint problem, with application to vaccine trials*, Biometrics **66** (2010), no. 4, 1153–1161.

[163] S. Xiong, P. Z. G. Qian, and C. F. J. Wu, *Sequential design and analysis of high-accuracy and low-accuracy computer codes*, Technometrics **55** (2013), no. 1, 37–46.

[164] A. Zellner, *On assessing prior distributions and bayesian regression analysis with g-prior distributions*, Bayesian inference and decision techniques (1986).

[165] Y. Zhang and G. Pinder, *Latin hypercube lattice sample selection strategy for correlated random hydraulic conductivity fields*, Water Resources Research **39** (2003), no. 8.

[166] H. Zou and T. Hastie, *Regularization and variable selection via the elastic net*, Journal of the royal statistical society: series B (statistical methodology) **67** (2005), no. 2, 301–320.