SEQUENTIAL LEARNING, LARGE-SCALE CALIBRATION, AND UNCERTAINTY QUANTIFICATION

Jiangeng Huang

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

STATISTICS

Robert B. Gramacy, Chair

David M. Higdon

Leanna L. House

Christopher T. Franck

June 27th, 2019

Blacksburg, Virginia

Keywords: sequential learning, computer experiments, uncertainty quantification, big data, hierarchical modeling

Sequential Learning, Large-scale Calibration, and Uncertainty Quantification

Jiangeng Huang

ABSTRACT

With remarkable advances in computing power, computer experiments continue to expand the boundaries and drive down the cost of various scientific discoveries. New challenges keep arising from designing, analyzing, modeling, calibrating, optimizing, and predicting in computer experiments. This dissertation consists of six chapters, exploring statistical methodologies in sequential learning, model calibration, and uncertainty quantification for heteroskedastic computer experiments and large-scale computer experiments. For heteroskedastic computer experiments, an optimal lookahead based sequential learning strategy is presented, balancing replication and exploration to facilitate separating signal from input-dependent noise. Motivated by challenges in both large data size and model fidelity arising from ever larger modern computer experiments, highly accurate and computationally efficient divide-and-conquer calibration methods based on on-site experimental design and surrogate modeling for large-scale computer models are developed in this dissertation. The proposed methodology is applied to calibrate a real computer experiment from the gas and oil industry. This on-site surrogate calibration method is further extended to multiple output calibration problems.

Sequential Learning, Large-scale Calibration, and Uncertainty Quantification

Jiangeng Huang

GENERAL AUDIENCE ABSTRACT

With remarkable advances in computing power, complex physical systems today can be simulated comparatively cheaply and to high accuracy through computer experiments. Computer experiments continue to expand the boundaries and drive down the cost of various scientific investigations, including biological, business, engineering, industrial, management, health-related, physical, and social sciences. This dissertation consists of six chapters, exploring statistical methodologies in sequential learning, model calibration, and uncertainty quantification for heteroskedastic computer experiments and large-scale computer experiments. For computer experiments with changing signal-to-noise ratio, an optimal lookahead based sequential learning strategy is presented, balancing replication and exploration to facilitate separating signal from complex noise structure. In order to effectively extract key information from massive amount of simulation and make better prediction for the real world, highly accurate and computationally efficient divide-and-conquer calibration methods for large-scale computer models are developed in this dissertation, addressing challenges in both large data size and model fidelity arising from ever larger modern computer experiments. The proposed methodology is applied to calibrate a real computer experiment from the gas and oil industry. This large-scale calibration method is further extended to solve multiple output calibration problems.

Acknowledgements

Along my graduate education journey, I have met many beautiful people, without whom this dissertation would not have been possible. Their insights and support made this journey a wonderful learning and growing experience for me. First and foremost, I would like to express my sincere gratitude to my advisor, Professor Robert B. Gramacy, for his devoted supervision, insightful guidance and incredible passion for research. Bobby has made a remarkable impact on my academic life by bringing me into the wonderlands of statistical computing and computer experiments. I have been very fortunate to work with him, and he is the person I will strive to emulate for many years to come.

I would like to thank my committee members, Professors David M. Higdon, Leanna L. House, and Christopher T. Franck, for their valuable suggestions and continuous support along this pleasant journey. I am especially grateful to Dave Higdon, who introduced me into the wonderful world of uncertainty quantification by the simple ball dropping example in my first year in the Ph.D. program at Virginia Tech. Dave is a great resource of both insightful advice and knowledge about uncertainty quantification. I definitely want to learn a lot more from him.

I am very grateful to all my mentors during my Ph.D. years at Virginia Tech for their instrumental instruction and supportive mentorship along my academic development. I would like to thank Professor Jennifer Van Mullekom for her patience and support in nurturing my communication and consulting skills as a professional statistician. I would like to thank Professor Geoff Vining for teaching me to have a historical perspective in doing research. I would like to thank Mirko Libraschi at BHGE for sharing his expertise in the honeycomb gas seal project and thank Andrea Panizza at BHGE for setting up the honeycomb calibration project during this collaborative research. I also would like to thank all these mentors helped at several statistical conferences during my Ph.D. studies, for both enlightening discussions and inspiring conversations to improve my work.

I would like to thank the National Science Foundation, the U.S. Department of Energy, Argonne National Laboratory, and Mary G. and Joseph Natrella for their generous support for my research.

I would like to thank all my talented colleagues and charming friends along my graduate school journey. It was such a pleasure to work, laugh, struggle, and grow together with these folks. I would like to thank my teammates at Gramacy Lab, including Mickaël Binois, Austin Cole, Adam Edwards, Furong Sun, Boya Zhang, and Nathan Wycoff, for many productive formal lab meetings and casual research discussions. Special thanks goes to Mickaël Binois, for sharing his dedication and passion in research to accelerate my growth in academic life. I also would like to thank Meng Zhao, Danni Lu, Lata Kodali, Huiying Mao, Wenyu Gao, Ruijin Lu, Shuning Huo, Byung-Jun Kim, Brandon Semel, Arindam

Fadikar, Thomas Metzger, Ana Quevedo Candela, Kyle Webb, Robert Settlage, Matthew Slifko, Sumin Shen, John Smith, Mohamed El Khouly, Christopher Grubb, Stephen Walsh, Sierra Merkes, Shane Bookhultz, Young Ho Yun, Erica Porter, Ryan Christianson, Yunnan Xu, Zhongnan Jin, Jiali Lin, Man Tang, Yafei Zhang, Zhihao Hu, Sheng-I Yang, Xinde Ji, Weizhe Weng, and Yihan Pang, for being a supportive part of my graduate school experience.

Last but definitely not least, I would like to thank my family for their endless love and consistent support for me, to let me follow my dreams and pursue a Ph.D. education. I especially thank my aunt, Mingjin Yan, for motivating me to pursure a career in statistics. Finally, I would like to thank my parents for their endless love, unlimited patience for my growing up, and unconditional support. This dissertation is dedicated to them.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background and motivation

"Human beings are curious by nature," as Greek philosopher Aristotle said over 2,000 years ago. Humankind keeps asking nature specific scientific questions through experimentation. Well-designed experimentation, careful data collection, empirical model building, uncertainty quantification, and rigorous statistical methodology in general, facilitate various scientific discovery processes throughout the entire history of humanity.

Since the pioneering work from agricultural field experiments by Sir Ronald Aylmer Fisher at the Rothamsted Experimental Station, modern statistical design and analysis of experiments has been the gold standard methodology to establish cause-and-effect relationships. Statistical methods in design and analysis of experiments have been applied extensively to a wide range of scientific, business, and industrial investigations, from randomized

controlled clinical trials for evaluating the effectiveness of new treatments to online experimentation for electronic commerce and user experience improvement. During the twentieth century, new challenges and new opportunities kept arising in the field of statistical design and analysis of experiments. Over the past century, new ideas and innovative statistical methodologies were developed to address these challenges.

Today, we live in the modern computer age. Twenty-first-century super computing power enables us to move forward from physical experiments to computer experiments, for more exciting statistical inference and better scientific discoveries. Now, we face new challenges and new opportunities of planning, designing, simulating, analyzing, modeling, optimizing, calibrating, understanding, and learning in computer experiments.

1.1.1 Computer experiments

With remarkable advances of modern computing power and technology, computer experiments play an increasingly essential role in broad scientific and industrial investigations. Scientists and engineers develop and employ sophisticated computer simulation models to represent and understand various complex physical phenomena, including biological systems, business and financial systems, engineering processes, industrial manufacturing processes, health-related systems, physical sciences based processes, and systems in social sciences.

Computer experiments, through physics-based and computationally intensive models to simulate the reality, provide a unique opportunity for scientists and engineers from

all fields to inquire about more sophisticated real-world physical phenomena, processes, and systems. Computer experiments also dramatically drive down the cost of various scientific investigations, especially where extensive physical experiments are too slow, too difficult, too expensive, or even totally infeasible.

1.1.2 Motivating example: honeycomb gas seal

A motivating example of a computer experiment considered in this dissertation is called honeycomb gas seal, developed in high pressure centrifugal compressors from Baker Hughes, a GE company (BHGE). The honeycomb gas seal is an important component widely used in BHGE's high pressure centrifugal compressors with two major areas of application:

- To enhance gas seal rotor stability in oil and gas applications
- To control leakage in aircraft gas turbines applications

The honeycomb gas seal(s) and applications at BHGE are described by input-output relationship between different variables characterizing gas seal geometry and its flow dynamics, including rotational speed, cell depth, seal diameter and length, inlet swirl, gas viscosity, gas temperature, compressibility factor, specific heat, inlet and outlet pressure, inlet and outlet clearance, static and rotoric friction factors, direct and cross stiffness coefficients, and direct and cross damping coefficients.

The honeycomb gas seal computer experiments consist of both a rotordynamic simulator called `ISOTSEAL` built upon bulk-flow theory and physical observations from field

experiments. The `ISOTSEAL` simulator, originally developed from rotordynamics experts at Texas A&M University, has been widely used in different oil and gas industrial applications. The `ISOTSEAL` simulator offers a fast evaluation (about one second) of the honeycomb seal physical process, calculating gas seal force coefficients based on seal flow physics. The field experiment, from BHGE's component-level honeycomb seal test campaign, is comprised of 292 physical runs. These runs vary a subset of those conditions, including gas seal clearance, swirl, cell depth, seal length, and seal diameter, that are believed to result in the greatest variability during turbomachinery operation.

1.2 Structure of this dissertation

This dissertation consists of six chapters, covering topics on sequential learning for heteroskadestic computer experiments, on-site surrogates for large-scale computer experiment calibration, and uncertainty quantification for large-scale computer experiments. Firstly, this chapter provides a background of computer experiments with the motivating example and overview of each projects. Secondly, Chapter 2 provides a comprehensive review of related elements in this dissertation with recent developments of relevant statistical methodology. Next, Chapter 3 proposes a new optimal look-ahead based sequential learning scheme for heteroskadestic simulation experiments. Chapter 4 proposes a new calibration strategy using on-site experimental design and surrogate modeling for large-scale computer experiments, with application to calibrate the motivating honeycomb seal problem described in Section 1.1.2. Chapter 5 further extends the on-site surrogate calibration method devel-

oped from Chapter 4 to multiple output calibration problems. Finally, Chapter 6 concludes this dissertation and summarizes directions for future work.

1.3 Overview of each project

This section provides an overview of each of the major chapters in this dissertation, including sequential learning for heteroskadestic Gaussian processes, on-site surrogates for large-scale computer experiment calibration, and multiple output calibration with uncertainty quantification for large-scale computer experiments.

1.3.1 Sequential learning for stochastic simulation experiments

Chapter 3 investigates the merits of replication, and provides methods for optimal design (including replicates), with the goal of obtaining globally accurate emulation of *noisy* computer simulation experiments. I first show that replication can be beneficial from both design and computational perspectives, in the context of Gaussian process surrogate modeling. Then I develop a lookahead based sequential design scheme that can determine if a new run should be at an existing input location (i.e., replicate) or at a new one (explore). When paired with a newly developed heteroskedastic Gaussian process model, our dynamic design scheme facilitates learning of signal and noise relationships which can vary throughout the input space. Chapter 3 shows that it does so efficiently, on both computational and statistical grounds. In addition to illustrative synthetic examples, I provide performance of two challenging real-data simulation experiments, from inventory management and epidemiology.

1.3.2 On-site surrogates for large-scale calibration

Motivated by a challenging computer model calibration problem from the oil and gas industry, involving the design of a so-called honeycomb seal, I develop a new Bayesian calibration methodology to cope with limitations in the canonical apparatus stemming from several factors. In Chapter 4, I proposes a new strategy of on-site experiment design and surrogate modeling to emulate a computer simulator acting on a high-dimensional input space that, although relatively speedy, is prone to numerical instabilities, missing data, and nonstationary dynamics. The aim is to strike a balance between data-faithful modeling and computational tractability within an overarching calibration framework—tuning the computer model to the outcome of a limited field experiment. Situating our *on-site surrogates* within the canonical calibration apparatus requires updates to that framework. In particular, Chapter 4 describes a novel yet intuitive Bayesian setup that carefully decomposes otherwise prohibitively large matrices by exploiting the sparse blockwise structure thus obtained. I illustrate empirically that this approach outperforms the canonical, stationary analog, and summarize calibration results on a toy problem and on the motivating honeycomb example.

1.3.3 Multiple outputs calibration

In Chapter 5, I extend the on-site surrogate calibration approach presented in Chapter 4 to multiple output calibration problems. The motivating honeycomb example has multiple physically meaningful outputs available. The developed on-site surrogate calibration method is focused on one output only. Combining multiple outputs for calibration can be a

more challenging task, especially when each output yields a different setting of calibration parameters. On the other hand, higher dimensional data can provide richer information in calibration, especially for learning and modeling systematic discrepancy between model and reality in high dimensional spaces when only a limited amount of field data is available. In Chapter 5, I first describe high-dimensional outputs of the honeycomb seal application, the challenges stemming from its simulation, and subsequent attempts to calibrate via on-site surrogates. Then, I apply the univariate on-site surrogate calibration method we developed in Chapter 4 individually to each of the multiple outputs in the honeycomb application. A discussion of the benefits and limitations of this univariate approach is also presented. After a careful exploratory data analysis, we extend the on-site surrogates to multiple output calibration through basis representations with new multiple output calibration results. Finally, I conclude Chapter 5 with a brief discussion with potential future research directions.

Chapter 2

Review of related work

In this chapter, we review several related key elements covered by this dissertation: (1) Gaussian processes and extensions for supervised learning; (2) design and modeling of computer experiments; and (3) model calibration for computer experiments.

2.1 Gaussian process modeling

2.1.1 Gaussian process overview

Using stochastic processes as priors for random functions to model function evaluations has a rich history in the spatial statistics literature (Matheron 1963, Cressie 1988, Cressie 1993, Stein 1999). Gaussian stochastic processes are popular priors over random functions, where any finite collection of function evaluations follow a multivariate normal distribution. More recently, Gaussian stochastic processes have also been widely used in

computer experiments and machine learning literatures. Gaussian processes have played a significant role as nonparametric Bayesian regression models for design and modeling of computer experiments (Sacks et al. 1989, Santner, Williams, and Notz 2003, Fang, Li, and Sudjianto 2006,Forrester, Sobester, and Keane 2008, Santner, Williams, and Notz 2018). In computer experiments, Gaussian processes serve as computationally cheap surrogate models or emulators with full uncertainty quantification, replacing computationally expensive numerical computations for physical systems. In the machine learning literature (Rasmussen and Williams 2006), Gaussian processes are popular supervised learning tools and tend to outperform other alternatives in out-of-sample prediction exercises in data-rich settings, where the input-output relationship contains a smooth response surface.

A Gaussian process $f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), \Sigma(\mathbf{x}, \mathbf{x}'))$ is completely described by its first two moments: mean function $\mu(\mathbf{x})$ and covariance function $\Sigma(\mathbf{x}, \mathbf{x}')$ for any pair of inputs $\mathbf{x}$ and $\mathbf{x}'$ as

$$\mu(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})], \quad \Sigma(\mathbf{x}, \mathbf{x}') = \mathbb{E}\{[f(\mathbf{x}) - \mu(\mathbf{x})][f(\mathbf{x}') - \mu(\mathbf{x}')]\} \tag{2.1}$$

where the covariance function $\Sigma(\mathbf{x}, \mathbf{x}')$ must be symmetric and positive-definite. More details are provided shortly.

As a powerful supervised learning tool for spatial statistics, computer experiments, and machine learning in general, Gaussian process regression enjoys full analytical tractability in its predictive distribution at new location $\mathcal{X}$. Given the observed data $\mathbf{D}_n = (\mathbf{X}_n, \mathbf{y}_n)$ and under the joint Gaussian process prior, the predictive distribution is analytically tractable

$\mathbf{y}(\mathcal{X}) \mid \mathbf{D}_n \sim \mathcal{GP}(\mu(\mathcal{X}), \Sigma(\mathcal{X}))$, where

$$\begin{aligned} \mu(\mathcal{X} \mid \mathbf{D}_n) &= \Sigma(\mathcal{X}, \mathbf{X}_n)\Sigma_n^{-1}\mathbf{y}_n \\ \Sigma(\mathcal{X} \mid \mathbf{D}_n) &= \Sigma(\mathcal{X}, \mathcal{X}) - \Sigma(\mathbf{X}_n, \mathcal{X})^\top \Sigma_n^{-1}\Sigma(\mathbf{X}_n, \mathcal{X}) \end{aligned} \tag{2.2}$$

using a simplifying prior assumption on mean functions $\mu(\mathcal{X}) = 0$ and $\mu(\mathbf{X}_n) = 0$. This simplifying assumption is common and reasonable in many applications, shifting all the hyperparameter learning and uncertainty quantification to its covariance function $\Sigma(\cdot)$ for a Gaussian process. The predictive equations (2.2) are also called "kriging equations", especially in geostatistics literature as interpolater for spatial data (Cressie 1993). The kriging equations enjoy the optimal property of being the Best Linear Unbiased Predictor (BLUP) from a frequentist viewpoint (Santner, Williams, and Notz 2018).

It is clear from the equations (2.2) above that the covariance function $\Sigma(\cdot, \cdot)$ plays a crucial role in Gaussian process modeling and prediction. The matrix inversion step in equations (2.2) implies the computational complexity is cubically related to the data size n, given that a standard matrix inversion algorithm is employed.

Covariance functions

Covariance functions (or *correlation functions*) of a Gaussian process measures the "similarity" between data points: the data points with close inputs $\mathbf{x}$ are generally assumed to have similar and close output value y. The covariance functions and their properties are fundamental to Gaussian processes (Abrahamsen 1997, Rasmussen and Williams 2006). A

covariance function $\Sigma(\cdot,\cdot)$ of a Gaussian process must be positive definite for establishing consistent finite-dimensional distributions,

$$\mathbf{x}^\top \Sigma(\cdot,\cdot)\mathbf{x} > 0, \quad \forall \quad \mathbf{x} \in \mathbb{R}^p \tag{2.3}$$

A covariance function is called *stationary* if it is only a function of the separation vector $r = \mathbf{x} - \mathbf{x}'$,

$$\Sigma(\mathbf{x}, \mathbf{x}') = \Sigma(r). \tag{2.4}$$

A stationary covariance function is invariant to translations in the input space. A more restricted subclass of stationary covariance functions are called *isotropic*, if the covariance function $\Sigma(\cdot,\cdot)$ is only a function of the distance measure or norm of the separation vector $d(\mathbf{x}, \mathbf{x}') = ||\mathbf{x} - \mathbf{x}'||$,

$$\Sigma(\mathbf{x}, \mathbf{x}') = \Sigma(d(\mathbf{x}, \mathbf{x}')). \tag{2.5}$$

An isotropic covariance function is invariant to rotations and translations. This more restrictive isotropic assumption on a covariance function leads to the simplicity of $\Sigma(\cdot,\cdot)$: it only depends on a single distance measure $d(\cdot)$ between $\mathbf{x}$ and $\mathbf{x}'$, which is also called a radial basis function.

The isotropy assumption imposes a strong restricted covariance function on a Gaus-

sian process, which may be unrealistic in some situations. Another more flexible class of stationary covariance functions are called *separable* covariance functions, assuming input dimensions are fully separable or partially separable in the correlation functions (see Abrahamsen 1997). One popular option is squared exponential covariance function

$$\Sigma(\mathbf{x}, \mathbf{x}') = \tau^2 \exp\left\{ -\sum_{k=1}^{p} \frac{(x_k - x_k')^2}{\theta_k} \right\}, \tag{2.6}$$

where a vectorized length-scale parameter $\theta = (\theta_1, \ldots, \theta_p)$ defines the strength of correlation to be separately determined by distance in each input direction. This separable exponential covariance function is very smooth, being infinitely differentiable (Stein 1999). When such strong smoothness assumption is impractical for every input direction, another option is to fine-tune the smoothness with additional parameters ν through a Matérn family,

$$\Sigma_\nu(||\mathbf{x} - \mathbf{x}'||) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(||\mathbf{x} - \mathbf{x}'|| \sqrt{\frac{2\nu}{\theta}} \right)^{\nu} K_\nu \left(||\mathbf{x} - \mathbf{x}'|| \sqrt{\frac{2\nu}{\theta}} \right). \tag{2.7}$$

where Γ is the Gamma function and K_ν is a modified Bessel function. When $\nu = \frac{1}{2}$, the Matérn covariance function becomes the exponential covariance function $\exp\left\{-\frac{x-x'}{\theta}\right\}$ and when $\nu \to \infty$, it converges to squared exponential covariance function in equation (2.6). In general, the Matérn family can become very flexible in various applications.

Another class of covariance functions are called *compactly supported kernels*, which creates sparsity in the covariance functions with compact support. A compactly supported kernel vanishes when the distance $||\mathbf{x} - \mathbf{x}'||$ is larger than a threshold cut-off distance d_{max},

the compact support,

$$\Sigma_{d_{max}}(||\mathbf{x}-\mathbf{x}'||) = 0, \quad \text{when} \quad ||\mathbf{x}-\mathbf{x}'|| > d_{max}. \tag{2.8}$$

Compactly supported kernels impose sparsity in the covariance function, leading to the opportunity of computational advantages, especially dealing with massive data sets. In order to guarantee the positive definiteness of a covariance function, choosing the appropriate cut-off threshold d_{max} and construction for compactly supported kernels requires some extra effort.

There are also many other commonly used covariance functions, including nonstationary classes and other novel constructions (Abrahamsen 1997, Higdon, Swall, and Kern 1999, Stein 1999, Genton 2001, Schmidt and O'Hagan 2003, Rasmussen and Williams 2006). Common choices of the covariance function $\Sigma(\cdot,\cdot)$ and construction of its hyperparameters should incorporate prior beliefs about the function spaces spanned by the predictive equations (2.2), including their smoothness and decay of correlation as a function of input distances with level of noise in the observations.

2.1.2 Recent Gaussian process developments

Gaussian processes provide a probabilistic and practical approach to many regression and classification problems. There are many recent exciting developments on Gaussian processes and related methodology in Statistics and Machine Learning literatures, including local methods to scale up Gaussian processes for large data sets, Gaussian processes with

noisy observations, and Gaussian process methods for multivariate data sets.

Scale up Gaussian processes for big data

Estimation and prediction using a Gaussian process notoriously requires cubic dense decomposition of $n \times n$ covariance matrices Σ_n. This requires extensive computational expense to train and predict, even with a modern high performance computing facility. In addition, the stationarity assumption in most covariance functions can be inappropriate and even problematic in large-scale data applications. Both the computational cost and the model assumptions limits the capacity of a Gaussian process to scale up to large-scale contexts.

A brute-force solution to large-scale Gaussian process computational bottlenecks is to use approximate methods for the big matrix computation and decomposition (see Chapter 8 of Rasmussen and Williams 2006). For example, an approximation to the eigendecomposition of the Gram matrix through Nyström method can reduce the computational cost from $\mathcal{O}(n^3)$ to $\mathcal{O}(mn^2)$ (Williams and Seeger 2001), where m is the size of a subset of training data and $m < n$. However, approximate methods for large-scale Gaussian process computation still suffer from fidelity problems: the assumptions on the covariance function, such as stationarity, fail to hold at a large global scale. Furthermore, the construction of the large-scale covariance function can be another challenging task.

A more realistic and practical idea to scale up a Gaussian process with fidelity is to use "divide and conquer". Through dividing more complicated global tasks into multiple

smaller but simpler local tasks, one can get around the otherwise prohibitively expensive computation, and at the same time, improve the fidelity of the Gaussian process modeling at a smaller local scale. Several different divide-and-conquer flavored approaches to scale up Gaussian process for large-scale situations have been proposed (Kim, Mallick, and Holmes 2005, Gramacy and Lee 2008, Gramacy and Apley 2015).

Motivated by an application in spatial statistics, Kim, Mallick, and Holmes 2005 developed a method called piecewise Gaussian processes to address the sharp changes and discontinuities in the underlying covariance function, where both the conventional stationary and nonstationary spatial models are inappropriate. A fully Bayesian fashioned partition through Voronoi tessellations was proposed, making the overall global field a piecewise stationary Gaussian process. The Voronoi tessellations partition is tractable in this work, dividing the two-dimensional input space into disjoint and independent pieces. However, piecewise Gaussian processes through Voronoi tessellations can be fraught with extreme computational burden with a large number of distinct partitions, especially when the input space dimension gets higher.

Another more computationally tractable partition method with Gaussian processes is through binary trees. Bayesian treed Gaussian process (TGP) models were developed to tackle a rocket booster simulation emulation problem with higher dimensional input space in computer experiments (Gramacy and Lee 2008). Compared to Voronoi tessellations partition, binary trees provide simpler axis-aligned partition with a larger degree of interpretability. A fully Bayesian implementation of this method called `tgp` is available for `R` (Gramacy

and Taddy 2016).

One more recent development of scaling up Gaussian processes for large-scale data set is called local approximate Gaussian process (laGP) (Gramacy and Apley 2015). Instead of partitioning the whole input space, laGP actively selects a family of local sequential designs from a local subset of the data. Local models are built on the subsets of design points $X_n(x) \subset X_N$, at local computational costs $\mathcal{O}(n^3) \ll \mathcal{O}(N^3)$ for inference, with high potential to parallelize. laGP has been directly applied to a large-scale computer experiment calibration problem arising in radiative shock hydrodynamics (Gramacy et al. 2015). More recent applications and extensions of laGP includes massive parallelization (Gramacy, Niemi, and Weiss 2014), prediction of the atmospheric drag of satellites in orbit (Sun et al. 2019a) and large-scale spatial-temporal modeling for solar irradiance (Sun et al. 2019b).

Gaussian processes with noisy observations

In many applications, it is more realistic to observe data in face of noise. A nugget effect is usually added into the covariance function of a Gaussian process,

$$\Sigma(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}, \mathbf{x}') + g\delta_{\mathbf{x},\mathbf{x}'} \tag{2.9}$$

where g is a nugget parameter and $\delta_{\mathbf{x},\mathbf{x}'}$ is the Kronecker delta. The nugget parameter g, representing the measurement error of the observed data, introduces a random noise structure into the covariance function.

Adding a nugget effect to a covariance function has several attractive properties.

Firstly, a Gaussian process with a nugget effect becomes a *smoother* across the input space while a Gaussian process without a nugget simply interpolates the input space. Secondly, adding a nugget effect can make the matrix computations better conditioned in the eigenvalues (Neal 1997). Besides the numerical stability reasons, Gramacy and Lee 2012 justifies several statistical purposes of using nuggets, arguing that even for a deterministic computer experiment, using nuggets is crucial for maintaining good statistical properties for the emulator.

The noise in equation (2.9) represents an independent noise structure: the noise between individual observations have no correlations and are assumed to be random. However, in many applications, it is more realistic to assume the noise is input-dependent. When the noise is input-dependent, the variance of the noise becomes a function of the input variable $\mathbf{x}$. Assuming the noise is a smooth function of the inputs, Goldberg, Williams, and Bishop 1998 originally proposes to use a second Gaussian process to model the noise variance, in addition to the noise-free outputs, with fully Markov chain Monte Carlo inference. Several computationally thriftier alternatives have been proposed (Kersting et al. 2007, Quan et al. 2013). Another alternative to model input-dependent noise is through stochastic kriging (Ankenman, Nelson, and Staum 2010). Without assuming a smooth noise structure, stochastic kriging estimates the level of noise through moment based estimation of variance with replication.

Combining both the strengths from stochastic kriging and latent variable process for noise, Binois, Gramacy, and Ludkovski 2018 propose a fully likelihood based inference frame-

work, called heteroskedastic Gaussian processes (hetGP), for Gaussian processes modeling with input-dependent noise. Leveraging a well-known Woodbury identity with replications, hetGP demonstrates that the computational cost for inference can be reduced from orders of full scale $\mathcal{O}(N^3)$ to the scale of unique input location number order $\mathcal{O}(n^3)$.

2.2 Design and modeling of computer experiments

Computer experiments investigate real physical systems through a computational and mathematical model M, mapping inputs $(\mathbf{X}, \mathbf{U})$ and outputs $\mathbf{y}^M$:

$$\mathbf{y}^M = M(\mathbf{X}, \mathbf{U})$$

to characterize and represent real phenomena, processes, and systems. Built from scientific understanding of the reality and realized by state-of-the-art computating ability, computer models $M(\cdot, \cdot)$ usually investigate complex input-output relationships. The input variables of computer models $M(\cdot, \cdot)$ are usually high-dimensional: some of inputs $\mathbf{X}$ are controllable and directly observable in physical experiments, while sometimes there are other tuning and calibration input variables $\mathbf{U}$ are not directly observable from physical experiments. The best setting of such input variables $\mathbf{U}$ are unknown and need to be inferred from data. In general, efficiently choosing the best settings for input variables $(\mathbf{X}, \mathbf{U})$ and building up realiable predictors for $\mathbf{y}^M$ with accurate uncertainty estimation are fundamental to computer experiments (Santner, Williams, and Notz 2003, Fang, Li, and Sudjianto 2006,

Forrester, Sobester, and Keane 2008, Santner, Williams, and Notz 2018).

2.2.1 Deterministic computer experiments

Historically, computer experiments are expensive runs of computer code using input configurations with deterministic outputs (Sacks et al. 1989, Sacks, Schiller, and Welch 1989). The deterministic nature of such computer experiments distinguishes itself from classical design of physical experiments. Classical principles including randomization and replication seem irrelevant for deterministic computer experiments. Linear regression methods and least-squares techniques become less popular due to the lack of random errors. Instead, space-filling designs and nonlinear regression models become the mainstrain techniques for design and modeling for computer experiments.

Model-independent designs

Space-filling designs have been around for several decades. One class of space-filling designs are *model-independent designs*. Latin hypercube sampling (LHS) designs (McKay, Beckman, and Conover 1979, Iman and Conover 1980, Stein 1987) and its variations are popular model-free design strategies for computer experiments. An n-run LHS design partitions each of its input coordinates into n equally sized interval, ensuring that the sample is comprised of just one point in each such intervals. LHS designs guarantee marginals are uniformly distributed. However, LHS designs alone are not unique and some LHS can be problematic, especially in high dimensional spaces. Combining LHS with other design crite-

ria, like rectangular Euclidean distance $d(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{p} |\mathbf{x}_i - \mathbf{x}'_i|$, can enhance its space-filling properties.

Designs based on distance criteria, such as minimax and maximin distance designs (Johnson, Moore, and Ylvisaker 1990), optimize any chosen distance function $d(\mathbf{x}, \mathbf{x}')$ to generate design points to cover the experimental region $\mathcal{X}$. Let $\mathbf{D} \in [0, 1]^p$ be the scaled unit design region. Both minimax and maximin designs try to fill the design space by avoiding the worst settings. Minimax distanced designs minimize the farthest point distances by

$$\arg\min_{\mathbf{D}} \max_{\mathbf{x}\in\mathcal{X}} \min_{i} d(\mathbf{x}, \mathbf{x}_i), \quad i = 1, 2, \ldots, n \tag{2.10}$$

while maximin distance designs maximize the closest point distances by

$$\arg\max_{\mathbf{D}} \min_{i,j} d(\mathbf{x}_i, \mathbf{x}_j), \quad i, j = 1, 2, \ldots, n \tag{2.11}$$

Choosing maximin distance designs within the class of LHS generates maximin Latin hypercube (maximin LHS) designs (Morris and Mitchell 1995).

$$\arg\min_{\mathbf{D}} \left\{ \sum_{i}^{n-1} \sum_{j=i+1}^{n} \frac{1}{d^k(\mathbf{x}_i, \mathbf{x}_j)} \right\}^{\frac{1}{k}} \tag{2.12}$$

In practice, maximin distance Latin hypercube designs are among the most commonly used space-filling designs because maximin LHS designs have combined optimal space-filling properties and they can be easily generated by available packages, such as Carnell 2018 and Franco

et al. 2018 in R.

Recent methodology developments for model-independent designs for computer experiments keep emerging, addressing various challenges arising in the computer experiments literature. To conduct multiple computer experiments with different levels of accuracy, statistical properties and construction methods for nested space-filling designs (Qian, Ai, and Wu 2009) and nested Latin hypercube designs (Qian 2009) have been developed. To incorporate computer experiments with both qualitative and quantitative factors, statistical properties and construction methods for sliced space-filling designs (Qian and Wu 2009) and sliced Latin hypercube designs (Qian 2012) have been developed. To ensure good projection properties in the subspaces of LHS and explore the input space in a sequential fashion, maximum projection designs (Joseph, Gul, and Ba 2015) have been proposed.

Model-dependent designs

Another class of statistical designs for computer experiments are the so-called *model-dependent designs*. Model-dependent designs take advantages of the obtained information from statistical models built upon the existing data to make subsequent design decisions. *Sequential design* of experiments and *response surface methodology* in general, as a combination of statistical techniques of linear regression, experimental design and optimization, have a rich history in the statistical literature (Myers, Montgomery, and Anderson-Cook 2016, Box and Draper 2007, Box and Draper 1987).

Common model-dependent design criteria include integrated mean squared error de-

signs (Sacks et al. 1989, Sacks, Schiller, and Welch 1989), maximum entropy designs (Shewry and Wynn 1987), and other model based criteria. Integrated mean-squared prediction error (IMSPE) criteria

$$\text{IMSPE} = \mathbb{E}\{\text{MSE}(\hat{\mathbf{y}}(\mathbf{x}))\} = \int_{\mathcal{X}} \text{MSE}[\hat{\mathbf{y}}(\mathbf{x})]\phi(\mathbf{x})d\mathbf{x} \tag{2.13}$$

can provide tractable information to generate designs by minimizing IMSPE with given weight function $\phi(\mathbf{x})$ (Sacks et al. 1989) or given the available prior information $\phi(\mathbf{x})$ on the experimental region $\mathcal{X}$ from a Bayesian perspective. This dissertation investigates and illustrates in Chapter 3 that in regular rectangular uniform situations, closed form IMSPE and its gradient information can be obtained and utilized to facilitate numerical optimization in sequential learning and Bayesian optimization.

Another common model-dependent design criterion is based on entropy, which measures the amount of "information" contained in the distribution of the data. The entropy of a distribution $p(x)$ is defined as

$$H(X) = -\int_{\mathcal{X}} p(x) \log p(x)\ dx, \tag{2.14}$$

where the lower the entropy, the distribution is more precise with less randomness while the higher the entropy, the distribution is more surprising with more randomness. Building up a design through maximizing the entropy (Shewry and Wynn 1987) introduces this notion from information theory into the experimental design, providing a measure of "amount of

information" in an experiment.

2.2.2 Stochastic computer experiments

Design and modeling for computer experiments traditionally focuses only on deterministic computer models (Sacks et al. 1989, Santner, Williams, and Notz 2018). Identical outputs from the same input configuration make the statistical notions back from Sir R.A. Fisher, such as replication and randomization, totally irrelevant in deterministic computer experiments. However, nowadays with the advances of modern computational ability, more and more computer experiments evolve with presence of *stochastic* behavior.

Stochastic computer experiments investigate more complicated dynamics of systems with more sources of uncertainty, trying to describe the known controllable signal in the face of unknown stochastic noise. As simulation gets noisier, the signal to noise ratio in the simulator decreases and learning becomes more challenging. Replication, a classical notion from design of experiments, provides a pure look at the noise through model-independent measure of pure error (Myers, Montgomery, and Anderson-Cook 2016). Generally speaking, a noisier simulator requires more replication.

The seminal work of Ankenman, Nelson, and Staum 2010 provides a framework called stochastic kriging to separate signal from noisy simulation with replication. Assuming the noise is substantial everywhere and changing dramatically across the input space, stochastic kriging uses a thrifty moment-based approach to estimate input-dependent noise. A substantial amount of replication is required everywhere across the input space, in order

to provide enough degrees of freedom for pure error estimation based on scaled Chi-Squared distributions. In addition to providing an estimation framework, useful results for optimal design location with appropriate amounts of replication have been provided in terms of IMSPE, to guide simulation decision-making. One limitation of stochastic kriging is that a heavy amount of replication is required everywhere across the entire input space to obtain moment based estimations of the noise level.

Also in the computer experiments literature, several different approaches to separate stochastic noise from signal in computer simulation experiments have been developed. A fully likelihood based inference framework called heteroskedastic Gaussian processes (hetGP) modeling has been proposed for Gaussian processes with input-dependent noise (Binois, Gramacy, and Ludkovski 2018). hetGP is more flexible about the amount of required replication and has the capability to extrapolate the noise structure to locations with no replication at all, assuming a smooth function on noise structure through a latent Gaussian process on the inputs. Leveraging a well-known Woodbury identity with replications, hetGP further demonstrates that the computational cost for inference can be reduced from orders of full scale $\mathcal{O}(N^3)$ to the scale of unique location number $\mathcal{O}(n^3)$.

Another recent nonparametric alternative for emulating stochastic simulation with a more frequentist flavor is called quantile kriging (Plumlee and Tuo 2014). Instead of imposing a Gaussian assumption on the outputs like Ankenman, Nelson, and Staum 2010 and Binois, Gramacy, and Ludkovski 2018, quantile kriging constructs an emulator through developing an emulative distribution on the quantiles of the outputs assuming no knowl-

edge of its structure. Under regularity conditions, Plumlee and Tuo 2014 also provide an asymptotically optimal convergence rate of the proposed quantile kriging framework. From a nonparametric perspective, quantile kriging is more flexible with no distributional assumption on the stochastic outputs, compared with other alternatives such as stochastic kriging and hetGP. On the other hand, quantile kriging requires more intense amount of replication in order to obtain the quantiles of a full distribution for stochastic outputs.

2.2.3 Sequential learning for computer experiments

Sequential design has been around in the statistics literature for more than half century, with a historical emphasis on building up empirical models through linear regression methods (Box and Draper 1987, Myers, Montgomery, and Anderson-Cook 2016). More recently, active learning and Bayesian optimization have become very popular tools in machine learning literature. In general, techniques called *active learning* from the machine learning community and *sequential design* in statistics literature efficiently choose subsequent input locations by leveraging the information obtained from an empirical model built upon the existing training data to facilitate sequential learning.

Active learning

Working with a feedforward neural networks framework, MacKay 1992 originally proposed to actively select data to maximize the expected information gain through two kinds of measures of information based on Shannon's entropy. Also around the same time,

Cohn 1994 explored the use of optimal experimental design as a promising tool for guiding active learning in neural networks. Extending this active learning idea from neural networks, Cohn, Ghahramani, and Jordan 1996 showed that active learning architectures can be more computationally efficient and accurate using Gaussian mixture models and locally weighted regressions. Seo et al. 2000a further extend active learning ideas to Gaussian processes, proposed two active learning criteria called Active Learning Mackay (ALM) and Active Learning Cohn (ALC), borrowing and synthesizing from the works above. More recently in the computer experiments literature, a hybrid design strategy has been developed (Gramacy and Lee 2009), combining nonstationary treed Gaussian processes modeling with active learning to adaptively build up sequential designs for nonstationary large-scale simulation experiments.

For model-based active learning and sequential design techniques, their strengths are also their weaknesses: objective and efficient active learning rely on the assumption that the underlying model is appropriate. Useful information for sequential decision-making depends on reliable and sufficient modeling. As an example of reinforcement learning, sequential learning is an iterative process, requiring continuous improvements and updates of the underlying model (Box and Draper 1987, Box, Hunter, and Hunter 2005, Box and Draper 2007, Myers, Montgomery, and Anderson-Cook 2016). Furthermore, for modern high-dimensional large-scale active learning problems, the computational cost and analytical efficiency is definitely another non-trivial venue for future investigation.

Bayesian optimization

More recently, *Bayesian optimization* techniques have become a popular tool for global optimization of black-box functions in machine learning literature (Ginsbourger and Le Riche 2010, Lam, Willcox, and Wolpert 2016, Gonzalez, Osborne, and Lawrence 2016, Picheny et al. 2016, Gramacy et al. 2016, Letham et al. 2019, Letham and Bakshy 2019). Overall, Bayesian optimization is a great example of using the tools from statistical thinking, such as experimental design and surrogate modeling, to solve complex black-box optimization problems in face of uncertainty.

The main idea of Bayesian optimization is to sequentially choose design points with the aid of training computationally cheap surrogate models, especially through Bayesian statistical models such as Gaussian processes, on existing training data, in order to optimize expensive and opaque objective functions. The design criteria in Bayesian optimization is called an *acquisition function*, a measure of the potential gain in sequential improvement on optimization. Common acquisition functions include expected improvement (Jones, Schonlau, and Welch 1998) and the knowledge gradient (Frazier, Powell, and Dayanik 2009). Bayesian optimization techniques have been developed to use sequential decision-making in face of uncertainty to solve various complex black-box optimization problems, including situations under unknown noise and unknown constraints (Gramacy and Lee 2010, Picheny et al. 2016, Gramacy et al. 2016, Letham et al. 2019).

2.3 Calibration for computer experiments

Rapid advances in computational power have enabled researchers to take advantage of *computer simulation models* (or *simulators*) to explore more complex physical systems; to test new hypothesis; and to understand situations where extensive traditional physical experimentation is too slow, too expensive, or totally infeasible. *Computer models* provide insights to push boundaries for various scientific fields, representing the state-of-the-art scientific understanding of the reality and implemented through the modern computing ability.

However, as simplified heuristics to the more complex and noisy reality, computer models usually contain different sources of uncertainty and need to be calibrated before being utilized to faithfully represent the real world. Computer models often need further specification for extra tuning inputs. In different specific applications, computer models often have extra context-specific input variables or parameters, denoted as $\mathbf{u}$, whose best settings are unknown and need to be inferred from real physical experimental data observed in field experiments. These input parameters $\mathbf{u}$, so-called *calibration parameters*, are often uncontrollable and unobservable in physical experiments. Learning the specific setting with uncertainty assessment for these calibration parameters is a key step to use computer models to faithfully simulate the reality. Furthermore, as an approximation to reality, unknown discrepancy often exists between computer models and physical field reality, due to missing physics, numerical inaccuracy, and other sources of uncertainty in computer models.

The goal of computer experiment calibration is to understand, analyze, quantify, and reduce different sources of uncertainty in computer models, especially to improve the

predictive ability with confidence for future extrapolative conditions (Higdon et al. 2004). This dissertation investigates the statistical aspects of model calibration for computer experiments. We focus on developing statistical methodology for making prediction with corresponding uncertainty evaluation, combining both sources of data from computer models with real physical experimental observations. We propose a novel and computationally efficient calibration method called *on-site surrogate* approach for large-scale computer experiments calibration problems in Chapters 4 and 5 of this dissertation.

2.3.1 Bayesian calibration framework

The *Bayesian approach* is one natural way to incorporate all sources of information into one coherent probabilistic analysis and simultaneously quantify all forms of uncertainties through the whole process. The Kennedy and O'Hagan (KOH) Bayesian calibration framework, now the mainstrain method, was originally proposed by Kennedy and O'Hagan 2001a with supplementary details (Kennedy and O'Hagan 2001b). The KOH calibration framework represents the relationship between the true process $y^F(\cdot)$ in reality, computer model $y^M(\cdot)$, model discrepancy $b(\mathbf{x})$, and true value of calibration parameter $\mathbf{u}^*$ through

$$y^F(\mathbf{x}) = y^M(\mathbf{x}, \mathbf{u}^*) + b(\mathbf{x}) + \epsilon. \tag{2.15}$$

where ϵ represents random noise in the field measurements. The KOH framework provides an explicit representation of different sources of uncertainty from unknown calibration parameters, unknown model discrepancy, and observation error through one Bayesian analysis.

Higdon et al. 2004 further outlines this comprehensive Bayesian calibration framework for fusing simulation and observed field data in order to calibrate unknown simulator input parameters, to make predictions, and to characterize predictive uncertainty using computer models. Statistical formulations and fully Markov chain Monte Carlo inference schemes are provided with three general variations of this calibration framework: using computer models directly with unlimited simulation runs when no model discrepancy is present; using Gaussian processes as surrogates with limited simulations runs when no model discrepancy is present; and using Gaussian processes as surrogates with limited simulation runs when a systematic discrepancy is present. Two real applications of this Bayesian calibration framework are further illustrated in Higdon et al. 2004 .

2.3.2 Related developments

Making prediction through the Kennedy and O'Hagan Bayesian calibration framework to quantifying, analyzing, and reducing uncertainty of computer models has attracted substantial attention and been widely used in various applications (O'Hagan 2006). There is a very rich literature on the methodology development of this highly flexible framework over the past two decades. This section provides a short survey of this highly flexible and still evolving calibration framework.

Hierarchical models and extensions

The Kennedy and O'Hagan calibration framework has been widely implemented in various real computer experiment calibration problems, yielding superior out-of-sample prediction. The superior predictive power of the KOH calibration apparatus can be attributed to the extreme flexibility offered by the hierarchical architecture of a couple of robust and well-regularized nonparametric regression models, such as using Gaussian processes for both $y^M(\mathbf{x}, \mathbf{u})$ and $b(\mathbf{x})$ in many situations. A vast variety of recent developments can be considered as natural variations and extensions of this canonical calibration apparatus.

In physical and engineering applications, multivariate outputs and functional outputs, such as space, time, and spatial-temporal outputs, are quite common. Calibration problems with multivariate and functional outputs are even more challenging than univariate calibration problems, with the additional burden of both large size and multivariate nature of the data. Higdon et al. 2008 extends the KOH framework from univariate output to high-dimensional output through basis representations. Highly multivariate simulation and experimental outputs are represented through principal components and combined into a complete Bayesian calibration framework. Fully Bayesian analysis via MCMC is still tractable under this framework. Also developed around the same time, Bayarri et al. 2007b utilizes a wavelet representation of the functional data to incorporate irregular functional outputs. To keep the structure of outputs within a Gaussian process framework, Paulo, Garcia-Donato, and Palomo 2012 further provides a KOH-style multivariate calibration framework through linear model of coregionalization. There are also empirical results showing that including multiple

outputs in model calibration can improve identifiability of calibration parameter, comparing to its univariate alternatives (Arendt et al. 2012).

Model discrepancy and identifiability

In the original work of Kennedy and O'Hagan 2001a, joint inference on the unknown calibration parameters $\mathbf{u}$ and model discrepancy $b(\mathbf{x})$ are performed through a hierarchical model consisting of a couple of Gaussian processes, as illustrated in Equation (2.15). Since then, the meaning of the "best" value or "true" value of calibration parameter and its confounding effect with the model discrepancy term has raised a large amount of attention and ensuing discussion (Kennedy and O'Hagan 2001a, Higdon et al. 2004, Bayarri et al. 2007a, Higdon et al. 2008, Gramacy et al. 2015, Tuo and Wu 2015, Wong, Storlie, and Lee 2017, Plumlee 2017).

Kennedy and O'Hagan 2001a proposes the meaning of true parameter values are the "best-fitted" in the sense of representing the data according to the error structure specified for residuals. Pointing out that it can be "dangerous" to interpret the estimated parameter values as the true physical values, Kennedy and O'Hagan 2001a further suggests "treating the model more pragmatically, as having inputs that we can 'tweak' empirically, can increase its value and predictive power." Higdon et al. 2004 and Higdon et al. 2008 point out that the interplay between the discrepancy term and calibration parameter requires further careful considerations of both their prior specifications and their underlying physical systems, especially when substantial discrepancy is present. More recently, Gramacy et al. 2015 describes

this identification issue "known to plague KOH-style calibration".

To address this notorious identifiability problem, reexamination and redefinition of the canonical KOH calibration framework is required. Several different versions of solutions have been proposed, from both frequentist and Bayesian perspectives (Tuo and Wu 2015, Tuo and Wu 2016, Wong, Storlie, and Lee 2017, Plumlee 2017). Tuo and Wu 2015 and Tuo and Wu 2016 redefine the true calibration parameter in terms of L_2 distance projection between the physical and computer observations from a frequentist perspective. Under this definition, theoretical properties, such as L_2-consistency and semiparametric efficiency, have been proven for this L_2 calibration approach to calibration. Also from a frequentist viewpoint, Wong, Storlie, and Lee 2017 proposed a simplified procedure with a new definition of identifiable parameters and corresponding theoretical guarantees, to calibrate situations where discrepancy is moderately small and only weak prior information for discrepancy is available. The model discrepancy term is estimated via smoothing methods and Bootstrap is used for uncertainty quantification. Another attempt to tackle this identifiability challenge is to add an orthogonal prior constraint from a purely Bayesian perspective. Plumlee 2017 proposes to add a prior distribution on the discrepancy term, where the discrepancy is orthogonal to the gradient of the computer model. This orthogonal constraint has been shown to mitigate this identifiability problem in some degree, at the cost of the additional orthogonal constraint and obtaining the gradient of computer model.

Modularization and optimization for large computer experiments

Situating the Kennedy and O'Hagan calibration framework to copy with increasingly larger modern computer simulation experiments requires both innovative computational and inferential updates of this canonical apparatus. In this subsection, we review several related developments on modularization and optimization for large-scale computer experiments calibration problems. We further introduce and develop on-site surrogates methodology for large-scale computer experiments calibration in Chapter 4.

Bayarri et al. 2007a provides a step-by-step validation framework based on the Kennedy and O'Hagan calibration framework. The procedure outlined contains six steps, where Bayesian methodology has been primarily demonstrated, and at the same time, several notable practical computational considerations have been raised. One practical computational consideration is to use the maximum likelihood estimates for Gaussian process's hyperparameters, instead of treating them as random in a fully Bayesian analysis, in order to allow implementation of this calibration method in vastly more complicated scenarios (Bayarri et al. 2007a). Uncertainty in calibration and tuning parameters $\mathbf{u}$, together with the uncertainty in the model discrepancy $b(\mathbf{x})$, have been pragmatically considered as the central issue during the whole validation procedure. More generally speaking, this procedure illustrates practical examples of an approximate Bayesian analysis alternative, called the *modular approach*, to build up large-scale hierarchical models under the Kennedy and O'Hagan-style calibration framework. The idea is to build up a surrogate model with all computer simulation data $y^M(\mathbf{x}, \mathbf{u})$ in the first step, then incorporate the field data $y^F(\mathbf{x})$

by a separate Bayesian analysis.

This modular approach to Kennedy and O'Hagan calibration framework has been further illustrated and discussed in Liu, Bayarri, and Berger 2009 from a more general inferential perspective. Liu, Bayarri, and Berger 2009 recommends modularization mainly for modeling uncertainties reasons, also for practical considerations. These reasons for modularization in Liu, Bayarri, and Berger 2009 include: to keep a good module separate from a suspect module and prevent contamination of information in posterior inference; to honor scientific understanding and further development of different modules; to help identifiability and avoid confounding between parameters in different modules; to improve mixing in MCMC analyses due to poor modeling; and to simplify computation to obtain an answer where otherwise ideal fully Bayesian practice is impossible.

In order to calibrate an orders of magnitude larger radiative shock hydrodynamics computer experiment, Gramacy et al. 2015 pushes the boundary of this modularization idea to an another extreme level, proposing *calibration as optimization*. Arguing modularization on purely computational grounds, Gramacy et al. 2015 combines several modern ideas together into a thrifty methodology we call the *optimization approach* to calibration in this dissertation. Working with existing simulation and field data, Gramacy et al. 2015 proposed local approximate Gaussian processes (laGP) (Gramacy and Apley 2015) for sparse and local surrogate modeling, with both potential for globally nonstationary modeling and efficient parallelizable computation. Moving one more step back to the basics, a thriftier maximum of a posterior for calibration parameter $\mathbf{u}$ is proposed, with a cascade of straightforward

optimization calculations. Putting together several modern ideas including local fast and nonstationary modeling, modularization for calibration, and derivative-free optimization, Gramacy et al. 2015 provide a thrifty alternative of the canonical Kennedy and O'Hagan calibration apparatus as *optimization* for large-scale computer experiments. While effective, Bayesian posterior uncertainty quantification ("the baby") was all but thrown out ("with the bath water"). A fully Bayesian posterior uncertainty quantification is in our wish-list for development and has been achieved in Chapter 4, through the on-site surrogates approach for large-scale computer experiments calibration.

Chapter 3

Sequential learning for stochastic simulation experiments

3.1 Introduction

Historically, design and analysis of computer experiments focused on deterministic solvers from the physical sciences via Gaussian process (GP) interpolation (Sacks et al. 1989). But nowadays computer modeling is common in the social Cioffi-Revilla 2014, Chapter 8, management (Law 2015) and biological (Johnson 2008) sciences, where stochastic simulations abound. Noisier simulations demand bigger experiments to isolate signal from noise, and more sophisticated GP models—not just adding nuggets to smooth the noise, but variance processes to track changes in noise throughout the input space in the face of heteroskedasticity (Binois et al. 2019). In this context there are not many simple tools: most

add to, rather than reduce, modeling and computational complexity.

Replication in the experiment design is an important exception, offering a pure look at noise, not obfuscated by signal. Since usually the signal is of primary interest, a natural question is: How much replication should be performed in a simulation experiment? The answer to that question depends on a number of factors. In this chapter the focus is on global surrogate model prediction accuracy and computational efficiency, and we show that replication can be a great benefit to both, especially for heteroskedastic systems.

There is evidence to support this in the literature. Ankenman, Nelson, and Staum 2010 demonstrated how replicates could facilitate signal isolation, via stochastic kriging (SK), and that accuracy could be improved without much extra computation by augmenting existing degrees of replication in stages (also see Liu and Staum 2010; Quan et al. 2013; Mehdad and Kleijnen 2018). Wang and Haaland 2017 showed that replicates have an important role in characterizing sources of inaccuracy in SK. Boukouvalas, Cornford, and Stehlík 2014 demonstrated the value of replication in (Fisher) information metrics, and Plumlee and Tuo 2014 provided asymptotic results favoring replication in quantile regression. Finally, replication has proved helpful in the surrogate-assisted (i.e., Bayesian) optimization of noisy blackbox functions (Horn et al. 2017; Jalali, Nieuwenhuyse, and Picheny 2017).

However, none of these studies address what we see as the main decision problem for design of GP surrogates in the face of noisy simulations. That is: how to allocate a set of unique locations, and the degree of replication thereon, to obtain the best overall fit to the data. That sentiment has been echoed independently in several recent publications (Kleijnen

2015; Weaver et al. 2016; Jalali, Nieuwenhuyse, and Picheny 2017; Horn et al. 2017). The standard approach of allocating a uniform number of replicates leaves plenty of room for improvement. One exception is Chen and Zhou 2014; Chen and Zhou 2017 who proposed several criteria to explore the replication/exploration trade-off, but only for a finite set of candidate designs.

This chapter tackles the issue sequentially, one new design element at a time. We study the conditions under which the new element should be a replicate, or rather explore a new location, under an integrated mean-square prediction error (IMSPE) criterion. We also highlight how replicates offer computational savings in surrogate model fitting and prediction with GPs, augmenting results of Binois et al. 2019 with fast updates as new data arrives. Inspired by those findings, we develop a new IMSPE-based criterion that offers lookahead over future replicates. This criterion is the first to acknowledge that exploring now offers a new site for replication later, and conversely that replicating first offers the potential to learn a little more (cheaply, in terms of surrogate modeling computation) before committing to a new design location. A key component in solving this sequential decision problem in an efficient manner is a closed form expression for IMSPE, and its derivatives, allowing for fast numerical optimization.

While our IMSPE criterion corrects for myopia in replication, it is important to note that it is not a full lookahead scheme. Rather, we illustrate that it is biased toward replication: longer lookahead horizons tend to tilt toward more replication in the design. In our experience, full lookahead, even when approximated, is impractical for all but the

most expensive simulations. Even the cleverest dynamic programming-like schemes (e.g., Ginsbourger and Le Riche 2010; Gonzalez, Osborne, and Lawrence 2016; Lam, Willcox, and Wolpert 2016; Huan and Marzouk 2016) require approximation to remain tractable or otherwise only manage to glimpse a few steps into the future despite enormous computational cost. Our more thrifty scheme can search dozens of iterations ahead. That flexibility allows us to treat the horizon as a tuning parameter that can be adjusted, online, to meet design and/or surrogate modeling goals. When simulations are cheap and noisy, we provide an adaptive horizon scheme that favors replicates to keep surrogate modeling costs down; when surrogate modeling costs are less of a concern, we provide a scheme that optimizes out-of-sample RMSE, which might or might not favor longer horizons (i.e., higher replication).

The structure of the remainder of this chapter is as follows. First we summarize relevant elements of GPs, sequential design and the computational savings enjoyed through replication in Section 3.2. Then in Section 3.3 we detail IMSPE, with emphasis on sequential applications and computational enhancements (e.g., fast GP updating) essential for the tractability of our framework. Section 3.4 discusses our lookahead scheme, while Section 3.5 provides practical elements for the implementation, including tuning the horizon of the lookahead scheme. Finally, in Section 3.6 results are presented from several simulation experiments, including illustrative test problems, and real simulations from epidemiology and inventory management, which benefit from disparate design strategies.

3.2 Computer experiments with replication

This section introduces relevant surrogate modeling and design elements while at the same time illustrating proof-of-concept for the main methodological contributions for Chapter 3. Namely that replication can be valuable computationally, as well as for accuracy in surrogate modeling.

3.2.1 Gaussian process regression with replication

We consider Gaussian process (GP) surrogate models for an unknown function over a fixed domain $f : D \subset \mathbb{R}^d \to \mathbb{R}$ based on noisy observations $\mathbf{Y} = (y_1, \ldots, y_N)^\top$ at design locations $\mathbf{X} = (\mathbf{x}_1^\top, \ldots, \mathbf{x}_N^\top)$. For simplicity, we assume a zero-mean GP prior, completely specified by covariance kernel $k(\cdot, \cdot)$, a positive definite function. Many different choices of kernel are possible, while in the computer experiments literature the power exponential and Matérn families are the most common. Often the families are parametererized by unknown quantities such as lengthscales, scales, etc., which are inferred from data see, e.g., Rasmussen and Williams 2006; Santner, Williams, and Notz 2003. The noise is presumed to be zero-mean i.i.d. Gaussian, with variance $r(\mathbf{x}) = \mathbb{V}\mathrm{ar}[Y(\mathbf{x})|f(\mathbf{x})]$. While we discuss our preferred modeling and inference apparatus in Section 3.2.3, for now we make the (unrealistic) assumption that kernel hyperparameters, along with the potentially non-constant $r(\mathbf{x})$, are known. Altogether, the data-generating mechanism follows a multivariate normal distribution, $\mathbf{Y} \sim \mathcal{N}_N(0, \mathbf{K}_N)$, where $\mathbf{K}_N$ is an $N \times N$ matrix comprised of $k(\mathbf{x}_i, \mathbf{x}_j) + \delta_{ij} r(\mathbf{x}_i)$, for $1 \leq i, j \leq N$ and with δ_{ij} being the Kronecker delta function.

Conditional properties of multivariate normal (MVN) distributions yield that the predictive distribution $Y(\mathbf{x})|\mathbf{Y}$ is Gaussian with

$$\mu_N(\mathbf{x}) = \mathbb{E}(Y(\mathbf{x})|\mathbf{Y}) = \mathbf{k}_N(\mathbf{x})^\top \mathbf{K}_N^{-1}\mathbf{Y}, \quad \text{with } \mathbf{k}_N(\mathbf{x}) = (k(\mathbf{x},\mathbf{x}_1),\ldots,k(\mathbf{x},\mathbf{x}_N))^\top;$$
$$\sigma_N^2(\mathbf{x}) = \mathbb{V}\text{ar}(Y(\mathbf{x})|\mathbf{Y}) = k(\mathbf{x},\mathbf{x}) + r(\mathbf{x}) - \mathbf{k}_N^\top(\mathbf{x})\mathbf{K}_N^{-1}\mathbf{k}_N(\mathbf{x}). \tag{3.1}$$

It can be shown that $\mu(\mathbf{x})$ is a best linear unbiased predictor (BLUP) for $Y(\mathbf{x})$ (and $f(\mathbf{x})$). Although testaments to the high accuracy and attractive uncertainty quantification features abound in the literature, one notable drawback is that when N is large the computational expense of $\mathcal{O}(N^3)$ due to decomposing $\mathbf{K}_N$ (e.g., to solve for $\mathbf{K}_N^{-1}$) can be prohibitive.

When the observations $y(\mathbf{x})$ are deterministic (i.e., $r(\mathbf{x}) = 0$), often N can be kept to a manageable size. When data are noisy, with potentially varying noise level, many samples may be needed to separate signal from noise. Indeed in our motivating applications, the signal-to-noise ratios can be very low, so even for a relatively small input space, thousands of training observations are necessary. In that context replication can offer significant computational gains. To illustrate, let $\bar{\mathbf{x}}_i$, $i = 1,\ldots,n$ denote the $n \leq N$ unique input locations, and $y_i^{(j)}$ be the j^{th} out of $a_i \geq 1$ replicates observed at $\bar{\mathbf{x}}_i$, i.e., $j = 1,\ldots,a_i$, where $\sum_{i=1}^n a_i = N$. Also, let $\bar{\mathbf{Y}}_{(N,n)} = (\bar{y}_1,\ldots,\bar{y}_n)^\top$ store averages over replicates, $\bar{y}_i = \frac{1}{a_i}\sum_{j=1}^{a_i} y_i^{(j)}$. Then Binois, Gramacy, and Ludkovski 2018 show that predictive equations based on this "unique-n" formulation, i.e., following Eq. (3.1) except with $\bar{\mathbf{Y}}_{(N,n)}$ and $\mathbf{K}_{(N,n)} = \left(k(\bar{\mathbf{x}}_i,\bar{\mathbf{x}}_j) + \delta_{ij}\frac{r(\bar{\mathbf{x}}_i)}{a_i}\right)_{1\leq i,j\leq n}$, are identical. Compared to the "full-N" formulation, the respective costs are reduced from $\mathcal{O}(N^3)$ to just $\mathcal{O}(n^3)$, without any ap-

proximations.

3.2.2 Sequential design for GPs

Although there are many criteria dedicated to design for GP regression see, e.g., Pronzato and Müller 2012, our focus here is on global predictive accuracy defined via integrated mean-squared prediction error (IMSPE). Fixing $\mathbf{X}$, the IMSPE integrates the "denoised" posterior variance $\check{\sigma}_N^2(\mathbf{x}) = \sigma_N^2(\mathbf{x}) - r(\mathbf{x})$ over D,

$$\text{IMSPE}(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \int_{\mathbf{x} \in D} \check{\sigma}_N^2(\mathbf{x})\, d\mathbf{x} =: I_N. \tag{3.2}$$

Note that although this definition removes $r(\mathbf{x})$, it is still present in $\mathbf{K}_N$ and therefore affects $\check{\sigma}_N^2(\mathbf{x})$. Removing $r(\mathbf{x})$ is not required, but since $\int r(\mathbf{x})\, d\mathbf{x}$ is constant over $\mathbf{x}_1, \ldots, \mathbf{x}_N$, it simplifies future expressions.

Even in the highly idealized case were all covariance $k(\cdot, \cdot)$ and noise $r(\cdot)$ relationships are presumed known, one-shot design—i.e., choosing all N locations $\mathbf{X}$ at once to minimize (3.2)—is an extraordinarily difficult task owing to the $(N \times d)$-dimensional search space. Only in very specific cases, such as $d = 1$ and a exponential kernel (Antognini and Zagoraiou 2010), or with the simpler task of allocating N replicates to a fixed set of n unique sites $\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_n$ (Ankenman, Nelson, and Staum 2010), is a computationally tractable solution known.

Therefore, we consider here the simpler case of a purely sequential design, building up a big design greedily, one simulation at a time. Note that this means that N grows by

1 after each iteration. While n is also evolving, the precise change is dependent on whether a replicate or a new location is selected. In the generic step, we condition on existing $\mathbf{x}_1, \ldots, \mathbf{x}_N$ locations and optimize IMSPE$(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{x}_{N+1})$ over $\mathbf{x}_{N+1}$. Recall that the posterior variance $\check{\sigma}^2_N$ only depends on the geometry of $\mathbf{X}$, i.e., it is independent of the outputs $\mathbf{Y}$ and hence we can view the above as minimizing $I_{N+1}(\mathbf{x}_{N+1}) :=$ IMSPE$(\mathbf{x}_{N+1}|\mathbf{x}_1, \ldots, \mathbf{x}_N)$. Later we establish specific closed-form expressions both for I_{N+1} and its gradient which enable fast optimization via library-based numerical schemes. Foreshadowing these developments, and utilizing the calculations detailed therein, we illustrate here the possibility that $\mathbf{x}_{N+1} = \text{argmin}_{\mathbf{x}} I_{N+1}(\mathbf{x})$ is a replicate. The conditions under which replication is advantageous, which we describe shortly in Section 3.3.1, have to our knowledge only been illustrated empirically (Boukouvalas 2010), or conceptually (e.g., Wang and Haaland 2017 highlight that replication is more beneficial as the signal-to-noise ratio decreases, via upper bounds on the MSPE), or to bolster technical results (e.g., Plumlee and Tuo 2014 demand a sufficient degree of replication to ensure asymptotic efficiency).

The *left* panel of Figure 3.1 shows two different noise levels, $r(\mathbf{x})$, for a stylized heteroskedastic GP predictor trained at $N = 5$ locations whose $\mathbf{x}_1, \ldots, \mathbf{x}_5$ values are shown as red dots. The fact that the two $r(\mathbf{x})$ curves coincide at these locations is not material to this discussion. Later in Section 3.3.1 this feature and a description of the gray-dotted curve will be provided. The right panel in the figure shows the predicted IMSPE, $I_{N+1}(\mathbf{x}) =$ IMSPE$(\mathbf{x}_1, \ldots, \mathbf{x}_5, \mathbf{x})$ derived from $\check{\sigma}^2_N$ calculations using those $\mathbf{x}_1, \ldots, \mathbf{x}_5$ values combined with the two $r(\mathbf{x})$ settings. With smaller IMSPE being better, we see that the solid blue

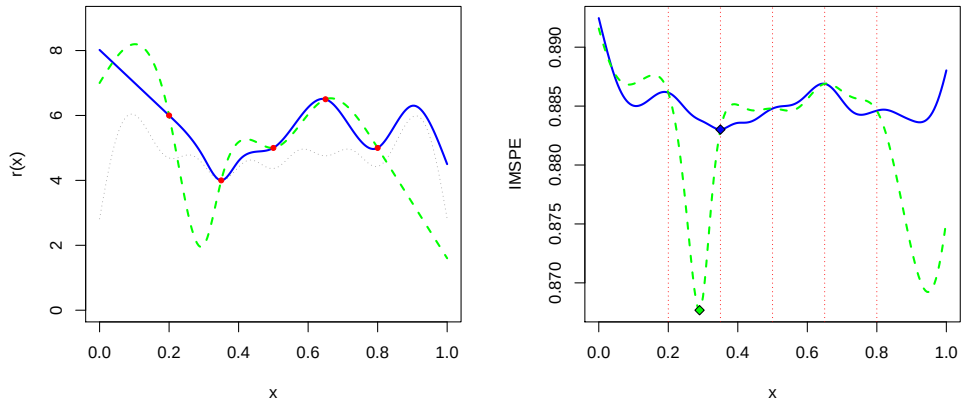


Figure 3.1: Noise Variance on IMSPE optimization. Illustration of the effect of noise variance on IMSPE optimization. Left: two examples of noise variance functions $r(\cdot)$ (blue solid and green dashed lines), with observations at $\mathbf{X}$ (five red points). The grey dotted line represents the minimum $r(\mathbf{x})$ that guarantees that replicating is optimal. Right: $I_{N+1}(\mathbf{x})$ for the two respective $r(\cdot)$. Diamonds highlight minimum values, and red dotted lines the existing designs $\mathbf{x}_1, \ldots, \mathbf{x}_5$.

regime calls for $\mathbf{x}_6$ being a replicate ($\operatorname{argmin} I_{N+1}(\mathbf{x}) = \mathbf{x}_2$), whereas the dashed green regime wants to explore at a new unique location ($\operatorname{argmin} I_{N+1}(\mathbf{x}) \simeq 0.32$). Also note that the IMSPE surfaces are multi-modal, which may pose a challenge to numerical optimizers, and that even for the dashed green curve there are replicates (e.g., at $\mathbf{x}_2$) with lower IMSPE than some local minima, meaning that augmenting with a cheap discrete search over replicates may be more effective than deploying a multi-start optimization scheme.

Ultimately, we entertain the far more realistic setup of unknown kernel hyperparameters and noise processes. In this context, sequential design to "learn-as-you-go" is essential. We take this approach not simply to avoid pathologies in hyperparameter mis-specification, as discussed in homoskedastic setups (e.g., Seo et al. 2000b; Krause and Guestrin 2007), but explicitly to gain the flexibility to sample non-uniformly in a manner that can only be adapted after a degree of initial sampling allows a fit of the noise process $\hat{r}(\mathbf{x})$ to be ob-

tained, and further refined. Our empirical results illustrate that reasonable, yet inaccurate, *a priori* simplifications such as constant $r(\mathbf{x})$ may—even if just for the purposes of design, not subsequent fitting—lead to inferior prediction. Previously such adaptive behavior and non-uniform sampling was only available via more cumbersome fully nonstationary methods, say involving treed partitioning (Gramacy and Lee 2008).

3.2.3 Heteroskedastic modeling

One way of dealing with heteroskedasticity in GP regression is to use empirical estimates of the variance as in SK (Ankenman, Nelson, and Staum 2010), described briefly in Section 3.2. Although this has the downside of requiring a minimum amount of replication, the calculations are straightforward and computations are thrifty. However, sequential design requires predicting the variance at new locations, and to accommodate that within SK Ankenman, Nelson, and Staum, recommend fitting a second, independent, GP for $\hat{r}(\mathbf{x})$ to smooth the empirical variances.

An alternative is to model the (log) variance as a latent process, jointly with the original "mean process" (Goldberg, Williams, and Bishop 1998; Kersting et al. 2007). However these methods can be computationally cumbersome, and are not tailored to leverage the computational savings that come with replication. Here we rely on the hybrid approach detailed by Binois, Gramacy, and Ludkovski 2018, leveraging replication and learning the latent log-variance GP based on a joint log-likelihood with the mean GP. We offer the following by way of a brief review.

For common choices of stationary kernel $k(\mathbf{x}, \mathbf{x}') = \nu c(\mathbf{x}-\mathbf{x}')$, the covariance matrix for the "mean GP" may be characterized as $\mathbf{K}_n = \nu(\mathbf{C}_n + \mathbf{\Lambda}_n)$ with $\mathbf{C}_n = (c(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j))_{1\leq i,j\leq n}$; and for the "noise GP" we take the analog $\log \mathbf{\Lambda}_n = \mathbf{C}_{(g)}(\mathbf{C}_{(g)} + g\mathbf{A}_n^{-1})^{-1}\mathbf{\Delta}_n$ where $\mathbf{C}_{(g)}$ is the equivalent of $\mathbf{C}_n$ for the second GP with kernel $k_{(g)}$. That is, $\log \mathbf{\Lambda}_n$ is the prediction given by a GP based on latent variables $\mathbf{\Delta}_n = (\delta_1, \ldots, \delta_n)$ that can be learned as additional parameters, alongside hyperparameters of $k_{(g)}$ and nugget g.

Based on this representation, the MLE of ν is

$$\hat{\nu}_N := N^{-1}\left(N^{-1}\sum_{i=1}^{n}\frac{a_i}{\lambda_i}s_i^2 + \bar{\mathbf{Y}}^\top(\mathbf{C}_n + \mathbf{A}_n^{-1}\mathbf{\Lambda}_n)^{-1}\bar{\mathbf{Y}}\right)$$

with $s_i^2 = \frac{1}{a_i}\sum_{j=1}^{a_i}(y_i^{(j)} - \bar{y}_i)^2$ whereas the rest of the parameters and hyperparameters can be optimized based on the concentrated joint log-likelihood:

$$\begin{aligned}\log \tilde{L} = & -\frac{N}{2}\log\hat{\nu}_N - \frac{1}{2}\sum_{i=1}^{n}\left[(a_i-1)\log\lambda_i + \log a_i\right] - \frac{1}{2}\log|\mathbf{C}_n + \mathbf{A}_n^{-1}\mathbf{\Lambda}_n| \\ & - \frac{n}{2}\log\hat{\nu}_{(g)} - \frac{1}{2}\log|\mathbf{C}_{(g)} + g\mathbf{A}_n^{-1}| + \text{Const},\end{aligned}$$

with $\hat{\nu}_{(g)} = n^{-1}\mathbf{\Delta}_n^\top(\mathbf{C}_n + g\mathbf{A}_n^{-1})^{-1}\mathbf{\Delta}_n$. Closed form derivatives are given in Binois et al. 2019, while an R (R Core Team 2017) package with embedded C++ subroutines is available as `hetGP` on CRAN.

Notice that for stationary kernels, the Eq. (3.3) reduces to $\text{IMSPE}(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \nu(1 - \text{tr}(\mathbf{C}_n^{-1}\mathbf{W}_n))$. The look-ahead IMSPE over replicates (3.7) becomes $I_{N+1}(\bar{\mathbf{x}}_k) = \nu(1 -$

$\mathrm{tr}(\mathbf{B}_k'\mathbf{W}_n))$ with $\mathbf{B}_k' = \frac{\left(\left(\mathbf{C}_n+\mathbf{A}_n^{-1}\mathbf{\Lambda}_n\right)^{-1}\right)_{\cdot,k}\left(\left(\mathbf{C}_n+\mathbf{A}_n^{-1}\mathbf{\Lambda}_n\right)^{-1}\right)_{k,\cdot}}{a_k(a_k+1)/\lambda_k-\left(\mathbf{C}_n+\mathbf{A}_n^{-1}\mathbf{\Lambda}_n\right)^{-1}_{k,k}}$. Also, the gradient of $I_{N+1}(\tilde{\mathbf{x}})$ from (3.5) involves $\partial r(\tilde{\mathbf{x}})/\partial\tilde{\mathbf{x}}_{(p)}$, which for `hetGP` reduces to

$$\frac{\partial \mathbf{k}_{(g)}(\tilde{\mathbf{x}})(\mathbf{C}_{(g)}+g\mathbf{A}_n^{-1})^{-1}\mathbf{\Delta}_n}{\partial\tilde{\mathbf{x}}_{(p)}} = \frac{\partial \mathbf{k}_{(g)}(\tilde{\mathbf{x}})}{\partial\tilde{\mathbf{x}}_{(p)}}(\mathbf{C}_{(g)}+g\mathbf{A}_n^{-1})^{-1}\mathbf{\Delta}_n.$$

3.3 IMSPE through the lens of replication

Over the years several authors (e.g., Ankenman, Nelson, and Staum 2010; Anagnostopoulos and Gramacy 2013; Burnaev and Panov 2015; Leatherman, Santner, and Dean 2017) have provided closed form expressions for IMSPE (i.e., for the integral in (3.2)) via variations on the criterion's definition (i.e., versions somewhat different than our preferred version in (3.2)), or via simplifications to the GP specification or to the argument $\mathbf{x}_1,\ldots,\mathbf{x}_N$, obtained by constraining the search set. Others have argued in more general contexts that d-dimensional numerical integration, usually via sums over a (potentially random) reference set, is the only viable option in their setting (Seo et al. 2000b; Gramacy and Lee 2009; Gauthier and Pronzato 2014; Gorodetsky and Marzouk 2016; Pratola et al. 2017).

Here this chapter provides a new closed-form expression for the IMSPE which, despite being intimately connected to earlier versions, is quite general and, we think, could replace many of the prevailing numerical schemes. This development uses the "unique-n" representation for efficient calculation under replication, however the analogue "full-N" version is immediate. We then consider an "add one" variation, $I_{N+1}(\tilde{\mathbf{x}}) = \mathrm{IMSPE}(\tilde{\mathbf{x}}|\mathbf{x}_1,\ldots,\mathbf{x}_N)$,

for efficient calculation in the sequential design setting and derive a condition under which replication is preferred for the next sample. Here we use $\tilde{\mathbf{x}}$ for a potential new location, while $\mathbf{x}_{N+1}$ will ultimately be chosen as the best candidate (i.e., minimizing IMSPE over $\tilde{\mathbf{x}}$). Note that if $\mathbf{x}_{N+1}$ turns out to be a replicate, n would not increase.

One important reason to have a closed-form IMSPE is the calculation of gradients, also in closed form, to aid in optimization. This dissertation provides the first such derivative expressions of which we are aware. Finally, acknowledging the dual role of replication (to speed calculations and separate signal from noise) this dissertation describes two new lookahead IMSPE heuristics for tuning the lookahead horizon in an online fashion, depending on whether speed or accuracy is more important.

3.3.1 IMSPE closed-formed expressions

This section starts by writing the IMSPE, shorthanded as I_N in Eq. (3.2), as an expectation:

$$I_N = \int_{\mathbf{x}\in D} \check{\sigma}_n^2(\mathbf{x})\, d\mathbf{x} = \mathbb{E}[\check{\sigma}_n^2(X)] = \mathbb{E}[k(X,X)] - \mathbb{E}[\mathbf{k}_n(X)^\top \mathbf{K}_n^{-1}\mathbf{k}_n(X)]$$

with X uniformly sampled in D, and using the linearity of the expectation. Notice that $\mathbf{K}_n$ depends on the number of replicates per unique design, so this representation includes a tacit dependence on the noise and replication counts $a_1, \ldots, a_n$. Then, as shown in Lemma 3.3.1, the integration of $\check{\sigma}_n^2$ over D may be reduced to integrations of the covariance function.

Lemma 3.3.1. *Let* $\mathbf{W}_n$ *be an* $n \times n$ *matrix with entries comprising integrals of kernel products* $w(\mathbf{x}_i, \mathbf{x}_j) = \int_{\mathbf{x}\in D} k(\mathbf{x}_i, \mathbf{x})k(\mathbf{x}_j, \mathbf{x})\, d\mathbf{x}$ *for* $1 \leq i, j \leq n$, *and let* $E = \int_{\mathbf{x}\in D} k(\mathbf{x}, \mathbf{x})\, d\mathbf{x}$. *Then*

$$I_N = E - \mathrm{tr}(\mathbf{K}_n^{-1}\mathbf{W}_n). \tag{3.3}$$

Proof. The first part, involving E, follows simply by definition. For the second part, let $\mathbf{z}$ be a random vector of size n with mean $\mathbf{m}$ and covariance $\mathbf{M}$. **Petersen2008** provides that $\mathbb{E}\left[\mathbf{z}^\top\mathbf{K}_n^{-1}\mathbf{z}\right] = \mathrm{tr}\left(\mathbf{K}_n^{-1}\mathbf{M}\right) + \mathbf{m}^\top\mathbf{K}_n^{-1}\mathbf{m}$. Therefore using $\mathbf{m}^\top\mathbf{K}_n^{-1}\mathbf{m} = \mathrm{tr}(\mathbf{K}_n^{-1}\mathbf{m}\mathbf{m}^\top)$, we have $\mathbb{E}[\mathbf{k}_n(X)^\top\mathbf{K}_n^{-1}\mathbf{k}_n(X)] = \mathrm{tr}(\mathbf{K}_n^{-1}(\mathbf{M} + \mathbf{m}\mathbf{m}^\top))$ where $\mathbf{m} = \mathbb{E}[\mathbf{k}_n(X)]$ and $\mathbf{M} = \mathbb{C}ov(\mathbf{k}_n(X)^\top, \mathbf{k}_n(X))$. Observing that $\mathbf{W}_n = \mathbf{M} + \mathbf{m}\mathbf{m}^\top$ gives the desired result. □

Our interest in the re-characterization in (3.3) is three-fold. First and foremost, some of the most commonly used kernels enjoy closed form expressions of E and $w(\cdot, \cdot)$. In Appendix **??** we provide $w(\cdot, \cdot)$ for (i) Gaussian, (ii) Matérn-5/2, (iii) Matérn-3/2, and (iv) Matérn-1/2 families. For those families, E further reduces to their scale hyperparameter. Section 3.2.3 offers specific forms for the generic expression (3.3) under our `hetGP` model. Second, note that even when closed forms are not available, as may arise when the kernel $k(X, X)$ cannot be analytically integrated over D, this formulation may still be advantageous. Numerically integrating $k(\mathbf{x}, \cdot)$ inside $\mathbf{W}_n$ will likely be far easier than the alternative of integrating $\check{\sigma}_n^2$, which can be highly multi-modal. Third, we remark that $\mathrm{tr}(\mathbf{K}_n^{-1}\mathbf{W}_n) = \mathbf{1}^\top(\mathbf{K}_n^{-1} \circ \mathbf{W}_n)\mathbf{1}$ where $\circ$ stands for the Hadamard (i.e., element-wise) product. Once $\mathbf{K}_n^{-1}$ and $\mathbf{W}_n$ are computed, the cost is in $\mathcal{O}(n^2)$, whereas the naïve alternative is $\mathcal{O}(n^3)$.

Now, in sequential application the goal is to choose a new $\mathbf{x}_{N+1}$ by optimizing

$I_{N+1}(\tilde{\mathbf{x}})$ over candidates $\tilde{\mathbf{x}}$. Fixing the first n unique design elements simplifies calculations substantially if we assume that $\mathbf{K}_n^{-1}$ and $\mathbf{W}_n$ are previously available. In that case, write

$$\mathbf{K}_{n+1} = \begin{bmatrix} \mathbf{K}_n & \mathbf{k}_n(\tilde{\mathbf{x}}) \\ \mathbf{k}_n(\tilde{\mathbf{x}})^\top & k(\tilde{\mathbf{x}},\tilde{\mathbf{x}}) + r(\tilde{\mathbf{x}}) \end{bmatrix}, \quad \mathbf{W}_{n+1} = \begin{bmatrix} \mathbf{W}_n & \mathbf{w}(\tilde{\mathbf{x}}) \\ \mathbf{w}(\tilde{\mathbf{x}})^\top & w(\tilde{\mathbf{x}},\tilde{\mathbf{x}}) \end{bmatrix}$$

with $\mathbf{w}(\tilde{\mathbf{x}}) = (w(\tilde{\mathbf{x}}, \bar{\mathbf{x}}_i))_{1\le i\le n}$. The partition inverse equations (Barnett 1979) give

$$\mathbf{K}_{n+1}^{-1} = \begin{bmatrix} \mathbf{K}_n^{-1} + \mathbf{g}(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top \sigma_n^2(\tilde{\mathbf{x}}) & \mathbf{g}(\tilde{\mathbf{x}}) \\ \mathbf{g}(\tilde{\mathbf{x}})^\top & \sigma_n^2(\tilde{\mathbf{x}})^{-1} \end{bmatrix}, \tag{3.4}$$

where $\mathbf{g}(\tilde{\mathbf{x}}) = -\sigma_n^2(\tilde{\mathbf{x}})^{-1}\mathbf{K}_n^{-1}\mathbf{k}_n(\tilde{\mathbf{x}})$ and $\sigma_n^2(\tilde{\mathbf{x}}) = \check{\sigma}_n^2(\tilde{\mathbf{x}}) + r(\tilde{\mathbf{x}})$ as in (3.1). Combining those two results together leads to

$$\begin{aligned} I_{N+1}(\tilde{\mathbf{x}}) &= E - \mathbf{1}^\top[\mathbf{K}_{n+1}^{-1} \circ \mathbf{W}_{n+1}]\mathbf{1} \\ &= E - \left(\mathbf{1}^\top[\mathbf{K}_n^{-1} \circ \mathbf{W}_n]\mathbf{1} + \sigma_n^2(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top\mathbf{W}_n\mathbf{g}(\tilde{\mathbf{x}}) + 2\mathbf{w}(\tilde{\mathbf{x}})^\top\mathbf{g}(\tilde{\mathbf{x}}) + \sigma_n^2(\tilde{\mathbf{x}})^{-1}w(\tilde{\mathbf{x}},\tilde{\mathbf{x}})\right) \\ &= I_N - \left(\sigma_n^2(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top\mathbf{W}_n\mathbf{g}(\tilde{\mathbf{x}}) + 2\mathbf{w}(\tilde{\mathbf{x}})^\top\mathbf{g}(\tilde{\mathbf{x}}) + \sigma_n^2(\tilde{\mathbf{x}})^{-1}w(\tilde{\mathbf{x}},\tilde{\mathbf{x}})\right). \end{aligned} \tag{3.5}$$

Both (3.4–3.5) only require $\mathcal{O}(n^2)$ computation.

After optimizing the latter part of (3.5) over $\tilde{\mathbf{x}}$ and choosing the best new design $\mathbf{x}_{N+1}$, one may utilize those inverse equations again to update the GP fit. Although similar identities have been provided in the literature (e.g., Gramacy and Polson 2011; Chevalier, Ginsbourger, and Emery 2014), the ones we provide here are the first to exploit the thrifty "unique-n" representation, and to tailor to the setting where $\mathbf{x}_{N+1}$ is a replicate, i.e., an $\bar{\mathbf{x}}_k$,

for $k \in \{1, \ldots, n\}$, versus a new distinct $\bar{\mathbf{x}}_{n+1}$ location.

Lemma 3.3.2. *Suppose $\mathbf{x}_{N+1} = \bar{\mathbf{x}}_k$. Then the updated predictive mean and variance (increasing N but not n) are given by*

$$\mu_{(N+1,n)}(\mathbf{x}) := \mu_{(N,n)}(\mathbf{x}) + \mathbf{k}_n(\mathbf{x})^\top (\mathbf{K}_{(N,n)}^{-1}(\bar{\mathbf{Y}}_{(N+1,n)} - \bar{\mathbf{Y}}_{(N,n)}) - \mathbf{B}_k \bar{\mathbf{Y}}_{(N+1,n)}),$$

$$\sigma^2_{(N+1,n)}(\mathbf{x}) = \sigma^2_{(N,n)}(\mathbf{x}) - \mathbf{k}_n(\mathbf{x})^\top \mathbf{B}_k \mathbf{k}_n(\mathbf{x}),$$

with $\mathbf{B}_k = \frac{\left(\mathbf{K}_{(N,n)}^{-1}\right)_{\cdot,k}\left(\mathbf{K}_{(N,n)}^{-1}\right)_{k,\cdot}}{a_k(a_k+1)/r(\bar{\mathbf{x}}_k) - \left(\mathbf{K}_{(N,n)}\right)^{-1}_{k,k}}$, *a rank-one matrix.*

Proof. By adding a replicate at $\bar{\mathbf{x}}_k$, the only change is to augment a_k by one in $\mathbf{K}_{(N+1,n)}$, namely $\mathbf{K}_{(N+1,n)} - \mathbf{K}_{(N,n)} = -\operatorname{Diag}\left(0, \ldots, 0, \frac{r(\bar{\mathbf{x}}_k)}{a_k(a_k+1)}, 0, \ldots, 0\right) =: -r(\bar{\mathbf{x}}_k)\mathbf{u}\mathbf{u}^\top = r(\bar{\mathbf{x}}_k)\mathbf{u}'\mathbf{u}^\top$ with $\mathbf{u}' = -\mathbf{u}$. Similarly, $\bar{\mathbf{Y}}_{(N+1,n)} - \bar{\mathbf{Y}}_{(N,n)} = \left(0, \ldots, 0, \frac{1}{a_k+1}(y_k^{(a_{k+1})} - \bar{y}_k^{(N)}), \ldots, 0\right)$ has only one non-zero element, residing in position k.

The Sherman-Morrison (i.e., rank-one Woodbury) formula gives

$$\mathbf{K}^{-1}_{(N+1,n)} = (\mathbf{K}_{(N,n)} + r(\bar{\mathbf{x}}_k)\mathbf{u}'\mathbf{u}^\top)^{-1} = \mathbf{K}^{-1}_{(N,n)} + \frac{\left(\mathbf{K}^{-1}_{(N,n)}\right)_{\cdot,k}\left(\mathbf{K}^{-1}_{(N,n)}\right)_{k,\cdot}}{(r(\bar{\mathbf{x}}_k)u_k^2)^{-1} - \left(\mathbf{K}^{-1}_{(N,n)}\right)_{k,k}} = \mathbf{K}^{-1}_{(N,n)} + \mathbf{B}_k. \tag{3.6}$$

This enables us to write $\mu_{(N+1,n)}(\mathbf{x}) - \mu_{(N,n)}(\mathbf{x}) = \mathbf{k}_n(\mathbf{x})^\top \left(\mathbf{K}^{-1}_{(N+1,n)}\bar{\mathbf{Y}}_{n+1} - \mathbf{K}^{-1}_{(N,n)}\bar{\mathbf{Y}}_{(N,n)}\right)$ and $\sigma^2_{(N+1,n)}(\mathbf{x}) - \sigma^2_{(N,n)}(\mathbf{x}) = \mathbf{k}_n(\mathbf{x})^\top \left(\mathbf{K}^{-1}_{(N+1,n)} - \mathbf{K}^{-1}_{(N,n)}\right)\mathbf{k}_n(\mathbf{x})$ and substitute $\mathbf{K}^{-1}_{(N+1,n)} - \mathbf{K}^{-1}_{(N,n)} = \mathbf{B}_k$ from (3.6). From the proof we also see that adding a replicate $\mathbf{x}_{N+1}$ incurs $\mathcal{O}(n)$ rather than the usual $\mathcal{O}(n^2)$ cost. □

As a corollary we obtain the following formula for one-step-ahead IMSPE at existing

designs $I_{N+1}(\bar{\mathbf{x}}_k)$ (relying on the fact that $\mathbf{W}_n$ is unchanged when replicating):

$$I_{N+1}(\bar{\mathbf{x}}_k) = E - \text{tr}(\mathbf{K}_{(N+1,n)}^{-1}\mathbf{W}_n) = E - \text{tr}((\mathbf{K}_{(N,n)}^{-1} + \mathbf{B}_k)\mathbf{W}_n) = I_N - \text{tr}(\mathbf{B}_k\mathbf{W}_n). \quad (3.7)$$

Besides enabling a "quick check" (with cost $\mathcal{O}(n^2)$) for finding the best replicate, perhaps a more important application of this result is that (3.7) yields an explicit condition under which replication is optimal.

Proposition 3.3.1. *Given n unique design locations $\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_n$, replicating is optimal (with respect to I_{N+1}) if*

$$r(\tilde{\mathbf{x}}) \geq \frac{\mathbf{k}(\tilde{\mathbf{x}})^\top \mathbf{K}_n^{-1}\mathbf{W}_n\mathbf{K}_n^{-1}\mathbf{k}(\tilde{\mathbf{x}}) - 2\mathbf{w}(\tilde{\mathbf{x}})^\top \mathbf{K}_n^{-1}\mathbf{k}(\tilde{\mathbf{x}}) + w(\tilde{\mathbf{x}}, \tilde{\mathbf{x}})}{\text{tr}(\mathbf{B}_{k^*}\mathbf{W}_n)} - \check{\sigma}_n^2(\tilde{\mathbf{x}}), \quad \forall \tilde{\mathbf{x}} \in D, \quad (3.8)$$

where $k^* \in \text{argmin}_{1\leq k\leq n} I_{N+1}(\bar{\mathbf{x}}_k)$.

Proof. We proceed by comparing $I_{N+1}(\tilde{\mathbf{x}})$ values when $\tilde{\mathbf{x}}$ is a replicate vis-à-vis a new design. Summarizing our results from above, we have $I_{N+1}(\bar{\mathbf{x}}_k^*) = I_N - \text{tr}(\mathbf{B}_{k^*}\mathbf{W}_n)$ for the (best) replicate and $I_{N+1}(\tilde{\mathbf{x}}) = I_N - \left(\sigma_n^2(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top \mathbf{W}_n\mathbf{g}(\tilde{\mathbf{x}}) + 2\mathbf{w}(\tilde{\mathbf{x}})^\top \mathbf{g}(\tilde{\mathbf{x}}) + \sigma_n^2(\tilde{\mathbf{x}})^{-1}w(\tilde{\mathbf{x}}, \tilde{\mathbf{x}})\right)$ for a new design. Replicating is better if $I_{N+1}(\bar{\mathbf{x}}_k^*) \leq I_{N+1}(\tilde{\mathbf{x}})$ for all $\tilde{\mathbf{x}}$, or when

$$\text{tr}(\mathbf{B}_{k^*}\mathbf{W}_n) \geq \sigma_n^2(\tilde{\mathbf{x}})^{-1}\left(\mathbf{k}(\tilde{\mathbf{x}})^\top \mathbf{K}_n^{-1}\mathbf{W}_n\mathbf{K}_n^{-1}\mathbf{k}(\tilde{\mathbf{x}}) - 2\mathbf{w}(\tilde{\mathbf{x}})^\top \mathbf{K}_n^{-1}\mathbf{k}(\tilde{\mathbf{x}}) + w(\tilde{\mathbf{x}}, \tilde{\mathbf{x}})\right).$$

Using the fact that $\sigma_n^2(\tilde{\mathbf{x}}) = \check{\sigma}_n^2(\tilde{\mathbf{x}}) + r(\tilde{\mathbf{x}})$ establishes the desired result. □

Referring back to Figure 3.1, the gray-dotted line in the *left* panel represents the

right hand side of Eq. (3.8). Thus, any noise surfaces with $r(\mathbf{x})$ above this line will lead to the I_{N+1} minimizer being a replicate, cf. the solid blue $r(\mathbf{x})$ case in the figure. Although this illustration involves a heteroskedastic example, the inequality in (3.8) can also hold in the homoskedastic case. In practice, replication in homoskedastic processes is most often at the edges of the input space, however particular behavior is highly sensitive to the settings of the n design locations, and their degrees of replication, a_i.

3.3.2 Gradient expressions

To facilitate the optimization of $I_{N+1}(\tilde{\mathbf{x}})$ with respect to $\tilde{\mathbf{x}}$, we provide closed-form expressions for its gradient, via partial derivatives. Below the subscript (p) denotes the p-th coordinate of the d-dimensional design $\tilde{\mathbf{x}} \in D$. As a starting point, the chain rule gives

$$\frac{\partial I_{N+1}(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}_{(p)}} = -\frac{\partial \operatorname{tr}(\mathbf{K}_{n+1}^{-1}\mathbf{W}_{n+1})}{\partial \tilde{\mathbf{x}}_{(p)}} = -\operatorname{tr}\left(\mathbf{K}_{n+1}^{-1}\frac{\partial \mathbf{W}_{n+1}}{\partial \tilde{\mathbf{x}}_{(p)}}\right) - \operatorname{tr}\left(\frac{\partial \mathbf{K}_{n+1}^{-1}}{\partial \tilde{\mathbf{x}}_{(p)}}\mathbf{W}_{n+1}\right). \tag{3.9}$$

To manage the computational costs, we notate below how the partial derivatives are distributed in another application of the partition inverse equations:

$$\frac{\partial \mathbf{K}_{n+1}^{-1}}{\partial \tilde{\mathbf{x}}} = \begin{bmatrix} \mathbf{H}(\tilde{\mathbf{x}}) & \mathbf{h}(\tilde{\mathbf{x}}) \\ \mathbf{h}(\tilde{\mathbf{x}})^\top & v_1(\tilde{\mathbf{x}}) \end{bmatrix} \tag{3.10}$$

$$\frac{\partial \mathbf{W}_{n+1}}{\partial \tilde{\mathbf{x}}_{(p)}} = \begin{bmatrix} \mathbf{0}_{n\times n} & \mathbf{c}_1(\tilde{\mathbf{x}}) \\ \mathbf{c}_1(\tilde{\mathbf{x}})^\top & c_2(\tilde{\mathbf{x}}) \end{bmatrix} \tag{3.11}$$

where the detailed expressions and derivations are given in Section 3.3.3.

The expressions above are collected into the following lemma.

Lemma 3.3.3. *The p^{th} component of the gradient for sequential ISMPE is*

$$-\frac{\partial I_{N+1}}{\partial \tilde{\mathbf{x}}_{(p)}} = 2\mathbf{c}_1(\tilde{\mathbf{x}})^\top \mathbf{g}(\tilde{\mathbf{x}}) + \mathbf{c}_2 \sigma_n^2(\tilde{\mathbf{x}})^{-1} + \mathbf{1}_n^\top [\mathbf{H}(\tilde{\mathbf{x}})\sigma_n^2(\tilde{\mathbf{x}}) \circ \mathbf{W}_n]\mathbf{1}_n \tag{3.12}$$
$$+ 2\mathbf{w}(\tilde{\mathbf{x}})^\top \mathbf{h}(\tilde{\mathbf{x}}) + v_1(\tilde{\mathbf{x}}) w(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}).$$

Proof. Beginning with Eq. (3.9), substitute (3.10) for the partial derivative of $\mathbf{K}_{n+1}^{-1}$, and (3.11) for that of $\mathbf{W}_{n+1}$. Then, note that $\mathbf{1}_n^\top [\mathbf{H}(\tilde{\mathbf{x}})\sigma_n^2(\tilde{\mathbf{x}}) \circ \mathbf{W}_n]\mathbf{1}_n$ can be rewritten as $v_2(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top \mathbf{W}_n \mathbf{g}(\tilde{\mathbf{x}}) + 2\sigma_n^2(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top \mathbf{W}_n \mathbf{h}(\tilde{\mathbf{x}})$. □

Since no further matrix decompositions are required, note that calculating the gradient of $I_{N+1}(\tilde{\mathbf{x}})$ in this way incurs computational costs in $\mathcal{O}(n^2)$.

3.3.3 Detailed gradient expressions

This subsection provides expressions for the Section 3.3.3 discussion on the gradient of the IMSPE.

$$\frac{\partial \mathbf{K}_{n+1}^{-1}}{\partial \tilde{\mathbf{x}}} = \frac{\partial}{\partial \tilde{\mathbf{x}}} \begin{bmatrix} \mathbf{K}_n^{-1} + \mathbf{g}(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top \sigma_n^2(\tilde{\mathbf{x}}) & \mathbf{g}(\tilde{\mathbf{x}}) \\ \mathbf{g}(\tilde{\mathbf{x}})^\top & \sigma_n^2(\tilde{\mathbf{x}})^{-1} \end{bmatrix} = \begin{bmatrix} \mathbf{H}(\tilde{\mathbf{x}}) & \mathbf{h}(\tilde{\mathbf{x}}) \\ \mathbf{h}(\tilde{\mathbf{x}})^\top & v_1(\tilde{\mathbf{x}}) \end{bmatrix} \quad \text{as in Eq. (3.10),}$$

where $v_1(\tilde{\mathbf{x}}) := \dfrac{\partial \sigma_n^2(\tilde{\mathbf{x}})^{-1}}{\partial \tilde{\mathbf{x}}_{(p)}} = \dfrac{2\mathbf{d}(\tilde{\mathbf{x}})^\top \mathbf{K}_n^{-1}\mathbf{k}(\tilde{\mathbf{x}})}{\left(k(\tilde{\mathbf{x}},\tilde{\mathbf{x}}) - \mathbf{k}(\tilde{\mathbf{x}})^\top \mathbf{K}_n^{-1}\mathbf{k}(\tilde{\mathbf{x}})\right)^2} + \dfrac{\partial r(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}_{(p)}}, \quad \mathbf{d}(\tilde{\mathbf{x}}) := \dfrac{\partial \mathbf{k}(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}_{(p)}}$

$$\mathbf{h}(\tilde{\mathbf{x}}) := \frac{\partial \mathbf{g}(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}_{(p)}} = -\mathbf{K}_n^{-1}\left(v_1(\tilde{\mathbf{x}})\mathbf{k}(\tilde{\mathbf{x}}) + \sigma_n^2(\tilde{\mathbf{x}})^{-1}\mathbf{d}(\tilde{\mathbf{x}})\right)$$

$$\mathbf{H}(\tilde{\mathbf{x}}) := \frac{\partial \mathbf{g}(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top \sigma_n^2(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}_{(p)}} = \frac{\partial \sigma_n^2(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}_{(p)}}\mathbf{g}(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top + \sigma_n^2(\tilde{\mathbf{x}})\mathbf{h}(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top + \sigma_n^2(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})\mathbf{h}(\tilde{\mathbf{x}})^\top$$

$$= v_2(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top + \sigma_n^2(\tilde{\mathbf{x}})\left(\mathbf{h}(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top + (\mathbf{h}(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top)^\top\right),$$

and $v_2(\tilde{\mathbf{x}}) = -2\mathbf{d}(\tilde{\mathbf{x}})^\top \mathbf{K}_n^{-1}\mathbf{k}(\tilde{\mathbf{x}})$. Similarly, since $\mathbf{W}_n$ does not depend on $\tilde{\mathbf{x}}$:

$$\frac{\partial \mathbf{W}_{n+1}}{\partial \tilde{\mathbf{x}}_{(p)}} = \begin{bmatrix} \mathbf{0}_{n\times n} & \mathbf{c}_1(\tilde{\mathbf{x}}) \\ \mathbf{c}_1(\tilde{\mathbf{x}})^\top & c_2(\tilde{\mathbf{x}}) \end{bmatrix}, \qquad \text{as presented in Eq. (3.11).}$$

Expressions for $\mathbf{c}_1(\cdot)$ and $c_2(\cdot)$ for particular kernels may be found in Appendix **??**.

3.3.4 Expressions for common kernels

This subsection considers four kernels common in practice: Gaussian (or squared exponential) and Matérn with parameter $\alpha = 5/2,\ 3/2,\ 1/2$ (the last one being the exponen-

tial kernel) and gives the corresponding expressions for E, w, $\mathbf{d}$, $\mathbf{c}_1$ and c_2 as introduced in Section 3.3. Notice that all these kernels are stationary, i.e., $k(\mathbf{x}, \mathbf{x}') = \nu c(\mathbf{x} - \mathbf{x}')$ with ν the process variance and c the correlation function. As a consequence, $E = \int_{\mathbf{x}\in D} k(\mathbf{x}, \mathbf{x})\ d\mathbf{x} = \int_{\mathbf{x}\in D} \nu c(\mathbf{0}) d\mathbf{x} = \nu$.

In their separable form, over $D = [0, 1]^d$, these kernel write $k(\mathbf{x}, \mathbf{x}') = \nu \prod_{p=1}^{d} k_i(x_p, x'_p)$ with k_i one of the aforementioned kernel. By using the separability we get:

$$\nu w(\mathbf{x}_i, \mathbf{x}_j) = \int_{\mathbf{x}\in D} k(\mathbf{x}_i, \mathbf{x}) k(\mathbf{x}_j, \mathbf{x}) d\mathbf{x} = \nu \prod_{p=1}^{d} \int_{x\in[0,1]} k_i(x_{i,p}, x) k_i(x_{j,p}, x) dx = \nu \prod_{p=1}^{d} w_p(x_{i,p}, x_{j,p}).$$

Below subsections provide parameterization of these kernels in univariate form along with the corresponding expressions for w, $\mathbf{d}$, $\mathbf{c}_1$ and c_2.

Gaussian kernel

The univariate Gaussian kernel is $k_G(x, x') = \exp\left(-\frac{(x-x')^2}{\theta}\right)$. Therefore:

$$d_i = \frac{\partial k_G(x_i, x)}{\partial x} = \frac{2(x_i - x)}{\theta} \exp\left(-\frac{(x_i - x)^2}{\theta}\right)$$

$$w(x_i, x_j) = \frac{\sqrt{2\pi\theta}}{4} \exp\left(-\frac{(x_i - x_j)^2}{2\theta}\right) \left(\operatorname{erf}\left(\frac{2 - (x_i + x_j)}{\sqrt{2\theta}}\right) + \operatorname{erf}\left(\frac{x_i + x_j}{\sqrt{2\theta}}\right)\right),\ 1 \le i, j \le n$$

with erf the error function. In addition:

$$c_2 = \frac{\partial w(x_i, x_i)}{\partial x_i} = \exp\left(-\frac{2x_i^2}{\theta}\right) - \exp\left(-\frac{(1 - 2x_i)^2}{\theta}\right)$$

and, for the vector $\mathbf{c}_1$, $1 \leq i \leq n$:

$$\frac{\partial w(x, x_i)}{\partial x} = \sqrt{\frac{\pi}{8\theta}} \exp\left(-\frac{(x-x_i)^2}{2\theta}\right) \left[(x-x_i)\left(\operatorname{erf}\left(\frac{x+x_i-2}{\sqrt{2\theta}}\right) - \operatorname{erf}\left(\frac{x+x_i}{\sqrt{2\theta}}\right)\right) + \sqrt{\frac{2\theta}{\pi}}\left(\exp\left(-\frac{(x+x_i)^2}{2\theta}\right) - \exp\left(\frac{-(x+x_i-2)^2}{2\theta}\right)\right)\right].$$

Remark: this is the kernel used in Figure 3.1, with hyperparameters $\nu = 1$, $\theta = 0.01$.

Matérn kernels with $\alpha = \{1, 3, 5\}/2$

This section uses the following parameterization of the Matérn kernel for specific values of α:

$$k_{M,1/2}(x, x') = \exp\left(-\frac{|x-x'|}{\theta}\right)$$

$$k_{M,3/2}(x, x') = \left(1 + \frac{\sqrt{3}|x-x'|}{\theta}\right) \exp\left(-\frac{\sqrt{3}|x-x'|}{\theta}\right)$$

$$k_{M,5/2}(x, x') = \left(1 + \frac{\sqrt{5}|x-x'|}{\theta} + \frac{5(x-x')^2}{2\theta^2}\right) \exp\left(-\frac{\sqrt{5}|x-x'|}{\theta}\right)$$

The derivatives with respect to x, i.e., in $\mathbf{d}$ are:

$$\frac{\partial k_{M,1/2}(x, x')}{\partial x} = \frac{(-1)^{\delta_{x<x'}}}{\theta} \exp\left(-\frac{|x-x'|}{\theta}\right)$$

$$\frac{\partial k_{M,3/2}(x, x')}{\partial x} = \frac{(-1)^{\delta_{x<x'}} \times 3|x-x'|}{\theta^2} \exp\left(-\frac{\sqrt{3}|x-x'|}{\theta}\right)$$

$$\frac{\partial k_{M,5/2}(x, x')}{\partial x} = (-1)^{\delta_{x<x'}} \frac{\left(\frac{10}{3} - 5\right)|x-x'| - \frac{5\sqrt{5}}{3\theta}(x-x')^2}{\theta^2} \exp\left(-\frac{\sqrt{5}|x-x'|}{\theta}\right)$$

To get closed form derivatives of $w(x_i, x_j)$ in Lemma 3.3.1, first consider $x_i \leq x_j$ to drop absolute values, then divide integration into components p_1 $(0 \to x_i)$, p_2 $(x_i \to x_j)$, p_3 $(x_j \to 1)$. We rely on symbolic solvers for the most tedious components, see e.g., `https://www.integral-calculator.com/`. To reduce expression, define $\beta = \exp\left(\frac{2\sqrt{3}}{\theta}\right)$ and $\gamma = \exp\left(\frac{2\sqrt{5}}{\theta}\right)$.

The first term is given by:

$$p_{1,1/2} = \int_0^{x_i} \exp\left(-\frac{(x_i - x)}{\theta}\right) \exp\left(-\frac{(x_j - x)}{\theta}\right) dx = \frac{\theta}{2}\left(\exp\left(\frac{2x_i}{\theta}\right) - 1\right) \exp\left(\frac{-x_j - x_i}{\theta}\right),$$

and similarly

$$p_{1,3/2} = \frac{1}{12\theta}\left[\left(\theta\left(5\sqrt{3}\theta + 9x_j - 9x_j\right)\exp\left(\frac{2\sqrt{3}x_i}{\theta}\right) - 5\sqrt{3}\theta^2 - 9\left(x_j + x_i\right)\theta \right.\right.$$
$$\left.\left. -2{\cdot}3^{\frac{3}{2}}x_i x_j\right)\exp\left(-\frac{\sqrt{3}\left(x_i + x_j\right)}{\theta}\right)\right]$$

$$p_{1,5/2}\cdot t_1 = \theta^2\left(63\theta^2 + 9{\cdot}5^{\frac{3}{2}}x_j\theta - 9{\cdot}5^{\frac{3}{2}}x_i\theta + 50x_j^2 - 100x_i x_j + 50x_i^2\right)\exp\left(\frac{2\sqrt{5}x_i}{\theta}\right)$$
$$- 63\theta^4 - 9{\cdot}5^{\frac{3}{2}}\left(x_j + x_i\right)\theta^3 - 10\left(5x_j^2 + 17x_i x_j + 5x_i^2\right)\theta^2 - 8{\cdot}5^{\frac{3}{2}}x_i x_j\left(x_j + x_i\right)\theta - 50x_i^2 x_j^2,$$

with $t_1 = 36\sqrt{5}\theta^3\exp\left(\frac{\sqrt{5}(x_j + x_i)}{\theta}\right)$.

The second term:

$$p_{2,1/2} = \int_{x_i}^{x_j} \exp\left(-\frac{(x-x_i)}{\theta}\right) \exp\left(-\frac{(x_j-x)}{\theta}\right) = (x_j - x_i) \exp\left(-\frac{x_j - x_i}{\theta}\right)$$

$$p_{2,3/2} = \frac{(x_j - x_i)\left(2\theta^2 + 2\sqrt{3}\,(x_j - x_i)\,\theta + x_j^2 - 2x_i x_j + x_i^2\right) \exp\left(-\frac{\sqrt{3}(x_j - x_i)}{\theta}\right)}{2\theta^2}$$

$$p_{2,5/2} \cdot t_2 = (x_j - x_i)\,54\theta^4 + \left(54\sqrt{5}x_j - 54\sqrt{5}x_i\right)\theta^3 + \left(105x_j^2 - 210x_i x_j + 105x_i^2\right)\theta^2$$
$$+ \left(3{\cdot}5^{\frac{3}{2}}x_j^3 - 9{\cdot}5^{\frac{3}{2}}x_i x_j^2 + 9{\cdot}5^{\frac{3}{2}}x_i^2 x_j - 3{\cdot}5^{\frac{3}{2}}x_i^3\right)\theta + 5x_j^4 - 20x_i x_j^3 + 30x_i^2 x_j^2 - 20x_i^3 x_j + 5x_i^4$$

with $t_2 = 54\theta^4 \exp\left(\frac{\sqrt{5}(x_i - x_j)}{\theta}\right)$.

The third term:

$$p_{3,1/2} = \int_{x_j}^{1} \exp\left(-\frac{(x-x_i)}{\theta}\right) \exp\left(-\frac{(x-x_j)}{\theta}\right)$$
$$= \frac{\theta}{2}\left(\exp\left(\frac{x_i - x_j}{\theta}\right) - \exp\left(\frac{x_j + x_i - 2}{\theta}\right)\right)$$

$$p_{3,3/2} \cdot t_3 = \theta\left(5\theta + 3^{\frac{3}{2}}\,(x_j - x_i)\right)\beta$$
$$- \left(\theta\left(5\theta - 3^{\frac{3}{2}}\,(x_j + x_i - 2)\right) + 6\,(x_i - 1)\,x_j - 6x_i + 6\right)\exp\left(\frac{2\sqrt{3}x_j}{\theta}\right)$$

$$p_{3,5/2}\cdot t_4 = \exp\left(\frac{2\sqrt{5}x_j}{\theta}\right)\cdot\Big[\theta\left(\theta\left(9\theta\left(7\theta-5^{\frac{3}{2}}\left(x_j+x_i-2\right)\right)+10x_j\left(5x_j+17x_i-27\right)\right.\right.$$
$$\left.+10\left(5x_i^2-27x_i+27\right)\right)-8\cdot5^{\frac{3}{2}}\left(x_i-1\right)\left(x_j-1\right)\left(x_j+x_i-2\right)\Big)+50\left(x_i-1\right)^2\left(x_j-2\right)x_j$$
$$+50\left(x_i-1\right)^2\Big]-\theta^2\left(63\theta^2+9\cdot5^{\frac{3}{2}}x_j\theta-9\cdot5^{\frac{3}{2}}x_i\theta+50x_j^2-100x_ix_j+50x_i^2\right)\gamma$$

with $t_3 = 4\theta\sqrt{3}\exp\left(\frac{\sqrt{3}(x_j-x_i+2)}{\theta}\right)$, $t_4 = -36\sqrt{5}\theta^3\exp\left(\frac{\sqrt{5}(x_j-x_i+2)}{\theta}\right)$.

The case when $x_i > x_j$ is obtained by swapping x_i and x_j above. Derivatives with respect to x_i and x_j, to account for both of these cases, are provided as follows:

$$\frac{\partial w_{1/2}(x_i,x_j)}{\partial x_i} = -\frac{\left(2\left(x_i+\theta-x_j\right)\exp\left(\frac{2x_i}{\theta}\right)+\theta\exp\left(\frac{2x_j}{\theta}\right)-\theta\right)\exp\left(-\frac{x_i+x_j}{\theta}\right)}{2\theta}$$

$$\frac{\partial w_{1/2}(x_i,x_j)}{\partial x_j} = \frac{\left(\theta\exp\left(\frac{2x_j}{\theta}\right)-2\exp\left(\frac{2x_i}{\theta}\right)x_j+2\left(\theta+x_i\right)\exp\left(\frac{2x_i}{\theta}\right)+\theta\right)\exp\left(-\frac{x_j+x_i}{\theta}\right)}{2\theta}$$

$$\frac{\partial w_{3/2}(x_i,x_j)}{\partial x_i}t_5 = \exp\left(\frac{2\sqrt{3}x_i}{\theta}\right)\Big[2\sqrt{3}\beta x_i^3+\left(-6\theta-2\cdot3^{\frac{3}{2}}x_j\right)\beta x_i^2+$$
$$+\left(\left(\left(6x_j-6\right)\theta-3^{\frac{3}{2}}\theta^2\right)\exp\left(\frac{2\sqrt{3}x_j}{\theta}\right)+\left(2\sqrt{3}\theta^2+12x_j\theta+2\cdot3^{\frac{3}{2}}x_j^2\right)\beta\right)x_i+$$
$$\left(2\theta^3+\left(4\sqrt{3}-\sqrt{3}x_j\right)\theta^2+\left(6-6x_j\right)\theta\right)\exp\left(\frac{2\sqrt{3}x_j}{\theta}\right)+\left(-2\sqrt{3}x_j\theta^2-6x_j^2\theta-2\sqrt{3}x_j^3\right)\beta\Big]$$
$$+\left(-3^{\frac{3}{2}}\theta^2-6x_jx_i\right)\beta x_i+\left(-2s^3-\sqrt{3}x_j\theta^2\right)\beta$$

$$\frac{\partial w_{3/2}(x_i,x_j)}{\partial x_j}t_6 = \theta\left[\left(3^{\frac{3}{2}}\theta - 6x_i + 6\right)x_j - \theta\left(2\theta - \sqrt{3}\,(x_i-4)\right) + 6x_i - 6\right]\exp\left(\frac{2\sqrt{3}\,(x_j+x_i)}{\theta}\right)$$
$$- 2\sqrt{3}\exp\left(\frac{2\sqrt{3}\,(x_i+1)}{\theta}\right)x_j^3 - 2\left(3\theta - 3^{\frac{3}{2}}x_i\right)\exp\left(\frac{2\sqrt{3}\,(x_i+1)}{\theta}\right)x_j^2 -$$
$$\beta\left(2\sqrt{3}\theta^2\exp\left(\frac{2\sqrt{3}x_i}{\theta}\right) - 12x_i\theta\exp\left(\frac{2\sqrt{3}x_i}{\theta}\right) + 2\cdot 3^{\frac{3}{2}}x_i^2\exp\left(\frac{2\sqrt{3}x_i}{\theta}\right) - 3^{\frac{3}{2}}\theta^2 - 6x_i\theta\right)x_j$$
$$+ 2x_i\left(\sqrt{3}\theta^2 - 3x_i\theta + \sqrt{3}x_i^2\right)\exp\left(\frac{2\sqrt{3}\,(x_i+1)}{\theta}\right) + \theta^2\left(2\theta + \sqrt{3}x_i\right)\beta$$

with $t_5 = -4\theta^3\exp\left(\frac{\sqrt{3}(x_i+x_j+2)}{\theta}\right)$, $t_6 = -t_5$.

$$\frac{\partial w_{5/2}(x_i,x_j)}{\partial x_i}t_7 = \Big[2\cdot 5^{\frac{3}{2}}\gamma x_i^5 + \left(-100\theta - 2\cdot 5^{\frac{5}{2}}x_j\right)\gamma x_i^4 + \left(18\cdot 5^{\frac{3}{2}}\theta^2 + 400x_j\theta + 4\cdot 5^{\frac{5}{2}}x_j^2\right)\gamma x_i^3 +$$
$$\left(\left(150\theta^3 + \left(24\cdot 5^{\frac{3}{2}} - 24\cdot 5^{\frac{3}{2}}x_j\right)\theta^2 + \left(150x_j^2 - 300x_j + 150\right)\theta\right)\exp\left(\frac{2\sqrt{5}x_j}{\theta}\right) +$$
$$\left(-210\theta^3 - 54\cdot 5^{\frac{3}{2}}x_j\theta^2 - 600x_j^2\theta - 4\cdot 5^{\frac{5}{2}}x_j^3\right)\gamma\right)x_i^2 + \Big(\Big(-3\cdot 5^{\frac{5}{2}}\theta^4 + (270x_j - 570)\,\theta^3 +$$
$$\left(-12\cdot 5^{\frac{3}{2}}x_j^2 + 72\cdot 5^{\frac{3}{2}}x_j - 12\cdot 5^{\frac{5}{2}}\right)\theta^2 + \left(-300x_j^2 + 600x_j - 300\right)\theta\Big)\exp\left(\frac{2\sqrt{5}x_j}{\theta}\right)$$
$$+ \left(42\sqrt{5}\theta^4 + 420x_j\theta^3 + 54\cdot 5^{\frac{3}{2}}x_j^2\theta^2 + 400x_j^3\theta + 2\cdot 5^{\frac{5}{2}}x_j^4\right)\gamma\Big)x_i +$$
$$\Big(54\theta^5 + \left(108\sqrt{5} - 33\sqrt{5}x_j\right)\theta^4 + \left(30x_j^2 - 330x_j + 450\right)\theta^3 + \left(12\cdot 5^{\frac{3}{2}}x_j^2 - 48\cdot 5^{\frac{3}{2}}x_j + 36\cdot 5^{\frac{3}{2}}\right)\theta^2$$
$$+ \left(150x_j^2 - 300x_j + 150\right)\theta\Big)\exp\left(\frac{2\sqrt{5}x_j}{\theta}\right) +$$
$$\left(-42\sqrt{5}x_j\theta^4 - 210x_j^2\theta^3 - 18\cdot 5^{\frac{3}{2}}x_j^3\theta^2 - 100x_j^4\theta - 2\cdot 5^{\frac{3}{2}}x_j^5\right)\gamma\Big]\exp\left(\frac{2\sqrt{5}x_i}{\theta}\right) +$$
$$\left(-150\theta^3 - 24\cdot 5^{\frac{3}{2}}x_j\theta^2 - 150x_j^2\theta\right)\gamma x_i^2 + \left(-3\cdot 5^{\frac{5}{2}}\theta^4 - 270x_j\theta^3 - 12\cdot 5^{\frac{3}{2}}x_j^2\theta^2\right)\gamma x_i$$
$$+ \left(-54\theta^5 - 33\sqrt{5}x_j\theta^4 - 30x_j^2\theta^3\right)\gamma$$

with $t_7 = -108\theta^5\exp\left(\frac{\sqrt{5}(x_j+x_i+2)}{\theta}\right)$.

3.3. IMSPE THROUGH THE LENS OF REPLICATION

$$\frac{\partial w_{5/2}(x_i,x_j)}{\partial x_j}t_7 = \Bigg(\Big(150\theta^3 + 24\cdot 5^{\frac{3}{2}}\left(1-x_i\right)\theta^2 + \left(150x_i^2 - 300x_i + 150\right)\theta\Big)\exp\left(\frac{2\sqrt{5}x_i}{\theta}\right)x_j^2 +$$

$$\Big(-3\cdot 5^{\frac{5}{2}}\theta^4 + \left(270x_i - 570\right)\theta^3 - 12\cdot 5^{\frac{3}{2}}\left(x_i^2 - 6x_i + 1\right)\theta^2 - 300\left(x_i^2 - 2x_i + 1\right)\theta\Big)\exp\left(\frac{2\sqrt{5}x_i}{\theta}\right)x_j$$

$$+\Big(54\theta^5 + \left(108\sqrt{5} - 33\sqrt{5}x_i\right)\theta^4 + \left(30x_i^2 - 330x_i + 450\right)\theta^3 + \left(12\cdot 5^{\frac{3}{2}}x_i^2 - 48\cdot 5^{\frac{3}{2}}x_i + 36\cdot 5^{\frac{3}{2}}\right)\theta^2 +$$

$$\left(150x_i^2 - 300x_i + 150\right)\theta\Big)\exp\left(\frac{2\sqrt{5}x_i}{\theta}\right)\Bigg)\exp\left(\frac{2\sqrt{5}x_j}{\theta}\right) + 2\cdot 5^{\frac{3}{2}}\exp\left(2\sqrt{5}x_i/\theta + 2\sqrt{5}/\theta\right)x_j^5 +$$

$$\left(100\theta - 2\cdot 5^{\frac{5}{2}}x_i\right)\exp\left(\frac{2\sqrt{5}(x_i+1)}{\theta}\right)x_j^4 + \left(18\cdot 5^{\frac{3}{2}}\theta^2 - 400x_i\theta + 4\cdot 5^{\frac{5}{2}}x_i^2\right)\exp\left(\frac{2\sqrt{5}\left(x_i+1\right)}{\theta}\right)x_j^3$$

$$+\Bigg(\Big(210\theta^3 - 54\cdot 5^{\frac{3}{2}}x_i\theta^2 + 600x_i^2\theta - 4\cdot 5^{\frac{5}{2}}x_i^3\Big)\exp\left(\frac{2\sqrt{5}\left(x_i+1\right)}{\theta}\right) +$$

$$\Big(-150\theta^3 - 24\cdot 5^{\frac{3}{2}}x_i\theta^2 - 150x_i^2\theta\Big)\gamma\Bigg)x_j^2 + \Bigg(\Big(42\sqrt{5}\theta^4 - 420x_i\theta^3 + 54\cdot 5^{\frac{3}{2}}x_i^2\theta^2 - 400x_i^3\theta +$$

$$2\cdot 5^{\frac{5}{2}}x_i^4\Big)\exp\left(\frac{2\sqrt{5}\left(x_i+1\right)}{\theta}\right) + \Big(-3\cdot 5^{\frac{5}{2}}\theta^4 - 270x_i\theta^3 - 12\cdot 5^{\frac{3}{2}}x_i^2\theta^2\Big)\gamma\Bigg)x_j +$$

$$\Big(-42\sqrt{5}x_i\theta^4 + 210x_i^2\theta^3 - 18\cdot 5^{\frac{3}{2}}x_i^3\theta^2 + 100x_i^4\theta - 2\cdot 5^{\frac{3}{2}}x_i^5\Big)\exp\left(\frac{2\sqrt{5}\left(x_i+1\right)}{\theta}\right) +$$

$$\Big(-54\theta^5 - 33\sqrt{5}x_i\theta^4 - 30x_i^2\theta^3\Big)\gamma$$

Finally, the expressions for c_2 from (3.11) are provided:

$$c_{2,1/2} = \exp\left(-\frac{2x_i}{\theta}\right);$$

$$c_{2,3/2}\cdot t_8 = \Big(3x_i^2 - 2\left(\sqrt{3}\theta + 3\right)x_i + \theta^2 + 2\sqrt{3}\theta + 3\Big)\exp\left(\frac{4\sqrt{3}x_i}{\theta}\right) - 3\beta x_i^2$$

$$- 2\sqrt{3}\theta\beta x_i - \theta^2\beta;$$

$$c_{2,5/2} \cdot t_9 = \exp\left(\frac{4\sqrt{5}x_i}{\theta}\right) \cdot \Big[25\theta^4 - 2\left(3{\cdot}5^{\frac{3}{2}}\theta + 50\right)x_i^3 + 3\left(\theta\left(25\theta + 6{\cdot}5^{\frac{3}{2}}\right) + 50\right)x_i^2 -$$
$$2\left(3\theta\left(\theta\left(3\sqrt{5}\theta + 25\right) + 3{\cdot}5^{\frac{3}{2}}\right) + 50\right)x_i + 9\theta^4 + 18\sqrt{5}\theta^3 + 75\theta^2 + 6{\cdot}5^{\frac{3}{2}}\theta + 25\Big] -$$
$$25\gamma x_i^4 - 6{\cdot}5^{\frac{3}{2}}\theta\gamma x_i^3 - 75\theta^2\gamma x_i^2 - 18\sqrt{5}\theta^3\gamma x_i - 9\theta^4\gamma$$

with $t_8 = -\theta^2 \exp\left(\frac{2\sqrt{3}(x_i+1)}{\theta}\right)$, $t_9 = -9\theta^4 \exp\left(\frac{2\sqrt{5}(x_i+1)}{\theta}\right)$.

3.4 Looking ahead over replication

Under certain conditions, sequential design via IMSPE, i.e., greedily minimizing I_{N+1} to choose $\mathbf{x}_{N+1}$, can well-approximate a one-shot batch design of size $N_{\max}$ because the criterion is monotone supermodular (Das and Kempe 2008; Krause, Singh, and Guestrin 2008). However, these results assume a known kernel hyperparameterization $k(\cdot,\cdot)$ and constant noise level $r(\cdot)$. In the more realistic case where those quantities must be estimated from data, and potentially with non-constant variance, there is ample evidence in the literature suggesting that sequential design can be *much better* than a batch design, e.g., based on a poorly-chosen parameterization, and no worse than an idealistic one (Seo et al. 2000b; Gramacy and Lee 2008). However, that does not mean that greedy, myopic, selection is optimal. By accounting for potential future selections in choosing the very next one, it is possible to obtain substantially improved final designs. However, the calculations involved, especially to "look ahead" from early sequential decisions to a far-away horizon $N_{\max}$, require expensive dynamic programming techniques to search an enormous decision space.

Approximating that full search, by limiting the lookahead horizon or otherwise reducing the scope of the decision space, has become an active area in Bayesian optimization via expected improvement (Ginsbourger and Le Riche 2010; Gonzalez, Osborne, and Lawrence 2016; Lam, Willcox, and Wolpert 2016). Targeting overall accuracy has seen rather less development, the work by Huan and Marzouk 2016 being an important exception. Here we aim to port many of these ideas to our setting of IMSPE optimization, where the nature of our approximation involves a weak bias towards replication which we have shown can be doubly beneficial in design.

The essential decision boils down to either choosing an $\mathbf{x}_{N+1}$ to explore, i.e., a new design element $\bar{\mathbf{x}}_{n+1}$, or choosing to replicate with $\mathbf{x}_{N+1}$ taken to be some $\bar{\mathbf{x}}_k$, for $k \in \{1, \ldots, n\}$. However, rather than directly minimizing (3.5) or (3.7), respectively, we perform a "rollout" lookahead procedure similar to Lam, Willcox, and Wolpert 2016 in order to explore the impact of those choices on a limited space of future design decisions. The updating equations in the previous subsections make this tractable.

In particular this chapter considers a horizon $h \in \{0, 1, 2, \ldots\}$ determining the number of design iterations to look ahead, with $h = 0$ representing ordinary (myopic) IMSPE search. Although larger values of h entertain future sequential design decisions, the goal (for any h) is to determine what to do *now*. Toward that end, we evaluate $h+1$ "decision paths" spanning alternatives between exploring sooner and replicating later, or vice versa. During each iteration along a given path, either (3.5) or (3.7) (but not simultaneously) is taken up as the hypothetical action. On the first iteration, if a new $\bar{\mathbf{x}}_{n+1}$ is chosen by optimizing

Eq. (3.5), that location (along with the existing $\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_n$) are considered as candidates for future replication over the remaining h lookahead iterations (when $h \geq 1$). If instead a replicate is chosen in the first iteration, the lookahead recursively searches over the choice of which of the remaining h iterations will pick a new $\bar{\mathbf{x}}_{n+1}$, with the others optimizing over replicates. This recursion is resolved by moving to the second iteration and again splitting the decision path into the choice between replicate-explore-replicate-... and replicate-replicate-..., etc. After recursively optimizing up to horizon h along the $h+1$ paths, the ultimate IMSPE for the respective hypothetical design with size $N+1+h$ is computed, and the decision path yielding the smallest IMSPE is noted. Finally, the next $\bar{\mathbf{x}}_{N+1}$ is a new location if the explore-first path was optimal, and is a replicate otherwise.

A diagram depicting the decision space is shown in Figure 3.2. In this example we attempt to augment the design from Figure 3.3 using $h = 3$. The path yielding the lowest IMSPE at the horizon involves here replicating three times (adding copies of design elements 6, 5, and 7 respectively), with exploration at $x_{n+4} = 0.175$ in the final stage. Consequently, replication is preferred over exploration and the next design element will be a replicate, duplicating $\bar{\mathbf{x}}_6$. The figure also illustrates that the cost of searching for the best replicate over adding a new design element involves at most $h+1$ global optimizations of Eq. (3.5), using the gradient. Although $(h+1)(h+2)/2 - 1$ discrete searches over (3.7) are required, the diagram indicates that $(h+1)$ searches of mixed continuous and discrete type may be performed in parallel. In practice, global optimization with a budget of at least the same order as n is an order of magnitude more expensive than looking for the best replicate.

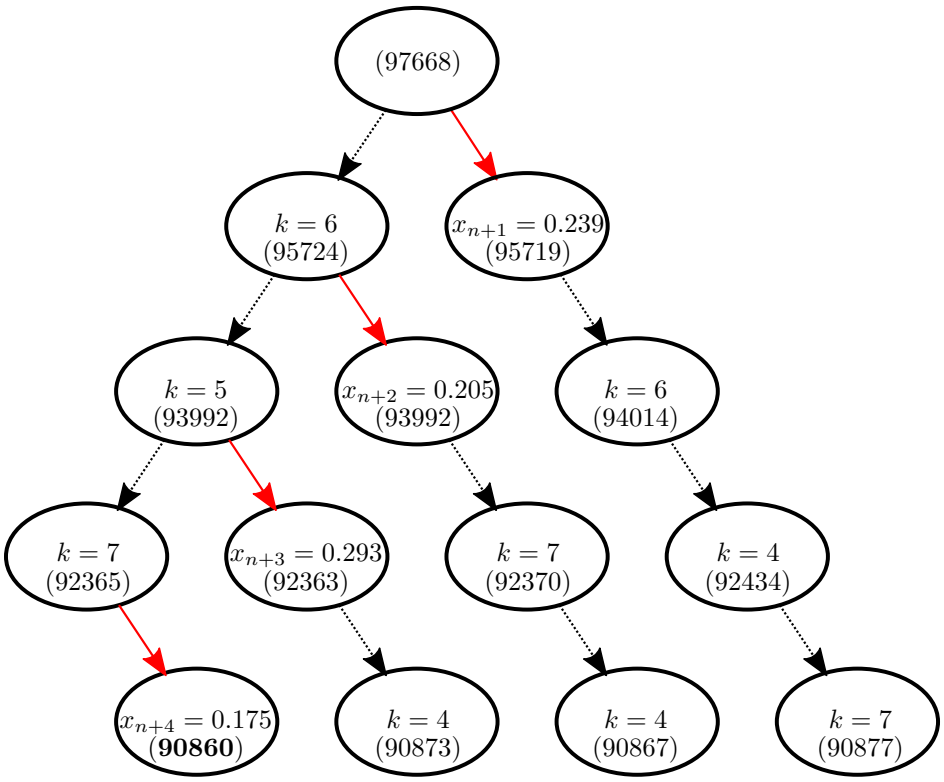


Figure 3.2: Lookahead strategy for $h = 3$. Starting with n unique designs, each white ellipse is a state, with a specific training set. Black dashed arrows represent the action of adding the best replicate, red solid arrows represent adding the best new design. This example considers augmenting the design for the example shown in Figure 3.3. Numbers in parenthesis indicates the IMSPE at each stage (values have been multiplied by 10^8).

In this scheme the horizon, h, determines the extent to which replicates are entertained in the lookahead, and therefore larger h somewhat inflates the likelihood of replication. Indeed, as h grows, there are more and more decision paths that delay exploration to a later iteration; if any of them yield a smaller IMSPE than the explore-first path, the immediate action is to replicate. However, note that although larger h allows more replication before committing to a new, unique $\bar{\mathbf{x}}_{n+1}$, it also magnifies the value of an $\bar{\mathbf{x}}_{n+1}$ chosen in the first iteration, as it could potentially accrue its own replicates in subsequent rollout iterations. Therefore, although we do find in practice that larger h leads to more replication in

the final design, this association is weak. Indeed, we frequently encounter situations where exploration is (temporarily) preferred for arbitrarily large horizons.

3.5 Modeling, inference and implementation

This section considers inference and implementation details, in particular for learning the noise process $r(\cdot)$. Our presumption is that little is known about the noise, however it is worth noting that this assumption may not be well aligned to some data-generating mechanisms, e.g., as arising from Monte Carlo simulations with known convergence rates (Picheny and Ginsbourger 2013). After reviewing a promising new framework called `hetGP`, for heteroskedastic GP surrogate modeling (Binois, Gramacy, and Ludkovski 2018), we provide extensions facilitating fast sequential updating of that predictor along with its (hyper-) parameterization. We conclude with schemes for adjusting the lookahead horizon introduced in Section 3.4.

3.5.1 Sequential heteroskedastic modeling

Optimizing IMSPE with lookahead over replication [Section 3.4] is only practical if the `hetGP` model can be updated efficiently when new simulations are performed. Two different update schemes are necessary: one for potential new designs, considered during the process of evaluating alternatives under the criteria [Eqs. (3.5–3.7)]; and another for the actual update with new simulation $y(\mathbf{x}_{N+1})$.

When looking-ahead, no new y-value is entertained, so hyperparameters of both

GPs stay fixed and only the latents may need to be augmented. Updating $\mathbf{K}_n$ follows (3.4) or (3.6), depending on whether the candidate $\tilde{\mathbf{x}}$ is new or a replicate. In the latter case, only $\mathbf{A}_n$ is updated for the "noise GP". Conversely, if a new location is added, an estimate of $r(\bar{\mathbf{x}}_{n+1})$ is required, which can come from the noise GP via exponentiating the usual GP predictive equations. That is, the new latent δ_{n+1} is taken as the predicted value by the noise GP.

The second update scheme—using the $y(\mathbf{x}_{N+1})$ observation—will require updating all the GPs' hyperparameters (including latents). Optimizing all hyperparameters of our heteroskedastic GP model is a potentially costly $\mathcal{O}(n^3)$ procedure. Instead of starting from scratch, a warm start of the MLE optimization is performed. Where they exist, previous values can be re-used as starting values, leaving only the latent $\tilde{\delta}$ at the newest design point, that is $\tilde{\delta} = \delta_{n+1}$ for a new location or $\tilde{\delta} = \delta_k$ for a replicate, requiring special attention.

As in the first case, $\tilde{\delta}$ may be initialized at its predicted value. But taking into account the new $y(\mathbf{x}_{N+1})$ makes it possible to combine information from the latent noise GP with results from empirical estimation of the log-variances. Kamiński 2015 explores this for updating SK models when new observations are added—a special case of the typical GP update formulas. The resulting combination of two predictions is via the geometric mean and can be summarized by the Gaussian $\mathcal{N}(\tilde{\delta}, V_{\tilde{\delta}})$ with

$$\tilde{\delta} = \left(\frac{\mu_{(g)}(\mathbf{x}_{N+1})}{\check{\sigma}^2_{(g)}(\mathbf{x}_{N+1})} + \frac{\hat{\delta}}{V_{\hat{\delta}}}\right)\left(\frac{1}{\check{\sigma}^2_{(g)}(\mathbf{x}_{N+1})} + \frac{1}{V_{\hat{\delta}}}\right)^{-1}, \quad V_{\tilde{\delta}} = \left(\frac{1}{\check{\sigma}^2_{(g)}(\mathbf{x}_{N+1})} + \frac{1}{V_{\hat{\delta}}}\right)^{-1},$$

where $\mu_{(g)}(\mathbf{x}_{N+1})$ and $\check{\sigma}^2_{(g)}(\mathbf{x}_{N+1})$ are the prediction from the noise GP[1] while $\hat{\delta}$ is the empirical estimate of the log variance at $\mathbf{x}_{N+1}$, itself with variance $V_{\hat{\delta}}$. We take $\hat{\sigma}^2 = \hat{\nu}_N^{-1} \frac{1}{\tilde{a}} \sum_{j=1}^{\tilde{a}} \left(y^{(j)}(\mathbf{x}_{N+1}) - \mu_n(\mathbf{x}_{N+1})\right)^2$, i.e., the uncorrected sample variance estimator that exists even for $\tilde{a} = 1$, i.e., the number of observations at $\mathbf{x}_{N+1}$. Supposing that the $y^{(j)}(\mathbf{x}_{N+1})$'s are i.i.d. Gaussian, we have $\tilde{a}\hat{\sigma}^2/\sigma^2 \sim \chi^2_{\tilde{a}}$. Accounting for the log-transformation, as in Boukouvalas 2010, we get $\hat{\delta} = \log(\hat{\sigma}^2) - \Psi((\tilde{a})/2) - \log(2) + \log(\tilde{a})$ and $V_{\hat{\delta}} = \Psi_2(\tilde{a}/2)$ with Ψ and Ψ_2 the digamma and trigamma functions.

Finally, the quick updates described above are predicated on improving local searches, and are thus not guaranteed to globally optimize the likelihood, which is always a challenge in MLE settings. The risk of becoming trapped in an inferior local mode is greater at earlier stages in the sequential design, i.e., when n is small. In practice, we find it beneficial to periodically restart the optimization with conservative (potentially random) initializations, which is cheap in that (small n) setting. As n increases, and the likelihood becomes more peaked, we find that costly restarts are of limited practical value. Local refinements, as described above, are fast and reliable.

3.5.2 Defining the horizon

Although the horizon h in the lookahead criteria in Section 3.4 could be fixed arbitrarily, or chosen based on computational considerations (smaller being faster), here we

[1]To avoid predictive variances close to zero for replicates, i.e., $\tilde{\delta} = \delta_k$, such that $\sigma^2_{(g)}(\mathbf{x}_{N+1}) \approx 0$ (g should be small), the variance is given by the "downdated" GP instead (i.e., the predicted variance if removing the replicated design), that are usually used for Leave-One-Out estimations and can be found, e.g., in Bachoc 2013, giving $\sigma^2_{(g)}(\mathbf{x}_{N+1}) = \left(\left(\mathbf{C}_{(g)} + g\mathbf{A}_n^{-1}\right)^{-1}_{k,k}\right)^{-1}$.

propose two heuristics to set it adaptively based on design goals. The adaptiveness means that $h \equiv h_N$ is now indexed by the current design size.

The first heuristic involves managing surrogate modeling costs, targeting a fixed ratio $\rho = n/N$ of unique to full design size. The goal is to ensure that each new unique location is, "worth its weight" in replicates from a computational perspective. The choice of n/N is arbitrary —other targets will do—but we focus on this particular one because its magnitude is easy to intuit. The *Target* heuristic we use to "maintain ρ" as sequential design steps progress is as simple as it is effective:

$$h_{N+1} \leftarrow \begin{cases} h_N + 1 & \text{if } n/N > \rho \quad \text{and a new point } \bar{\mathbf{x}}_{n+1} \text{ is chosen;} \\ \max\{h_N - 1, -1\} & \text{if } n/N < \rho \quad \text{and a replicate is chosen;} \\ h_N & \text{otherwise.} \end{cases} \tag{3.13}$$

If the current ratio is too high and a new point $\bar{\mathbf{x}}_{n+1}$ was recently added, making the ratio even higher, the horizon is increased to encourage future replication. If rather a replicate has been added while the current ratio was too low, then the horizon is decreased, encouraging exploration. Otherwise the evolution is on the right trajectory and the horizon is unchanged. Observe that (3.13) allows a horizon of -1, which is described shortly.

To implement the continuous search (3.5), we deploy a limited multistart scheme over `optim` searches in R with `method="lbfgsb"` and closed form gradients (3.12). In parallel, a discrete search over $\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_n$ is carried out via (3.7). The two solutions thus obtained are compared against thresholds, and if the $\tilde{\mathbf{x}}$ found via continuous search (or its relative

objective value) is within (say) $\varepsilon = 10^{-6}$ of that of the best $\bar{\mathbf{x}}_{k^*}$, the replicate is preferred on computational grounds. The horizon $h_N = -1$ is an exception, adding a new $\tilde{\mathbf{x}}$ no matter how close it is to the replicate candidate. Thus, $h \equiv -1$ can be roughly thought of as incrementing n by 1 along with N at each iteration; in practice it still occasionally generates replicates, primarily at the corners of the input space, if the corresponding multistart scheme determines that the I_{N+1}-minimizer lies at the boundary of D. On the other hand, $h \equiv 0$ obtains many replicates due to thresholding, which yields a "soft" clustering mechanism for $(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_n)$. Indeed, every iteration where we have a situation resembling Figure 3.1, the $h = 0$ rule will select a replicate and not increment n. In contrast, $h = -1$ will only "stumble" into a replicate if the optimizer finds a global minimum at the edge of the domain. It does not explicitly entertain replicates via (3.7).

Our empirical work [Section 3.6] illustrates how horizon targeting effectively manages computational costs. Although at times the horizon h_N can reach quite high values (upwards of $h_N = 20$), the computational cost of search is negligible compared to updating the GP fits. Meanwhile high horizons represent a "light touch" preference for replication: they do not preclude exploration, rather they somewhat discourage it. Thus, while the ultimate number of unique locations n is dependent on the entire history of the simulations, and hence comes with a sampling distribution, the corresponding search heuristic is much simpler than one that would impose a hard constraint on the final n.

When accuracy is the ultimate goal we prefer a different adaptation of h, making a more explicit link between ρ and the signal-to-noise ratio in the data. In *linear* regression

contexts, one way to deal with heterogeneity is to allocate replications on unique designs such that the ratio of the empirical variance over number of replicates are close to each other, i.e., to enforce homogeneity of $\hat{\sigma}_i^2/a_i$ (Kleijnen 2015). This approach captures the basic idea that more replicates are needed where $r(\mathbf{x})$ is high, but applicability to our setup is not direct because such a scheme does not factor in correlations estimated by GPs. Ankenman, Nelson, and Staum 2010 address this within SK by considering the allocation of the remaining budget of evaluations over existing designs, i.e., to determine where to augment with additional replicates. In particular, they show that the optimal allocation of the N simulations across n unique designs is summarized by $\mathbf{A}_n^*$, a diagonal matrix with components

$$a_i^* \approx N \frac{\sqrt{r(\bar{\mathbf{x}}_i)K_i}}{\sum_{j=1}^{n} \sqrt{r(\bar{\mathbf{x}}_j)K_j}}, \quad \text{where} \quad K_i = (\mathbf{K}_n^{-1}\mathbf{W}_n\mathbf{K}_n^{-1})_{i,i}. \tag{3.14}$$

We emphasize that (3.14) only addresses the replication aspect—the designs $\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_n$ must be entered *a priori* by the user. Thus, this recipe is not directly implementable in a sequential design setting. One solution could be to generate (e.g., by space-filling) a candidate design of pre-determined size n and $\underline{\mathrm{r}}$ replicates per design and then, after learning $\mathbf{K}_n$ and $r(\bar{\mathbf{x}}_i)$'s, apply (3.14). However, in that case one may end up with $a_i^* < \underline{\mathrm{r}}$, as is the case in Figure 3.3. This illustrative 1d example highlights that in areas with low noise, a lower number of replicates would have been better, while in more noisy areas, more points are necessary. The right panel shows the a_i^* at this stage (referred to as *batch*) compared to the greedy sequential allocation of 105 replicates. The latter is more realistic because it acknowledges

that design decisions cannot be undone[2].

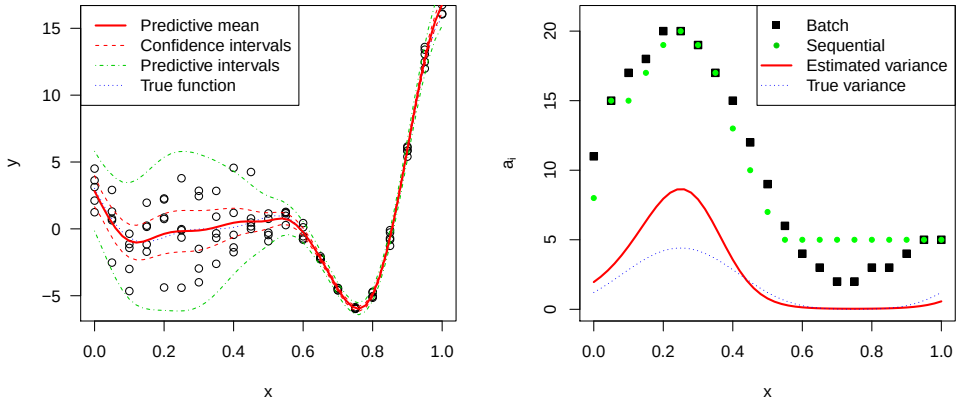


Figure 3.3: Illustrative heteroskedastic Gaussian process example. Left: toy example with 5 replicates at each of the 21 uniformly spaced unique design points. Right: proposed allocation of new 105 replicates (total 210 observations) based on (3.14) and a greedy sequential approach.

Instead of such two-stage design, we utilize (3.14) in a sequential fashion, by making a comparison between the allocation a_i^* via (3.14) (employing the current estimates of the noise $r(\bar{\mathbf{x}})$ at that particular stage) and the actual a_i's collected so far from the sequential design. The existing number of replicates a_i is then either too high, in which case no more replicates should be added, or too low, and could benefit from more replication. We use this information in the *Adapt* scheme to adjust the horizon by sampling

$$h_{N+1} \sim \text{Unif}\{a'_1, \ldots, a'_n\} \quad \text{with} \quad a'_i := \max(0, a_i^* - a_i). \tag{3.15}$$

Hence, if there are locations that require many more replicates according to (3.14), h_{N+1} could be large to encourage replication.

[2]The batch scheme recommends fewer than five replicates after five replicates where already used.

3.6 Experiments

This section illustrates our methods and simpler variants on a suite of examples spanning synthetic and real data from computer simulation experiments. The main metric is out-of-sample root mean-square (prediction) error (RMSE) over the sequential design iterations, and in particular after the final iteration. Since accurate estimation of variances over the input space is also an important consideration (especially in the heteroskedastic context)—even though our IMSPE criteria does not explicitly target learning variances—we consider RMSE to the true log variance, when it is known, and when it is not we use a proper scoring rule (Gneiting and Raftery 2007, Eq. (27)) combining mean and variance forecasts out-of-sample. Our main comparators are non-sequential (space-filling) designs, homoskedastic GP predictors, and combinations thereof.

3.6.1 Illustrative one-dimensional example

We start by reusing the 1d toy example from above [surrounding Figure 3.2 and 3.3] to show qualitatively the effect of the horizon choice on the resulting designs. The underlying function is $f(x) = (6x-2)^2 \sin(12x-4)$, from Forrester, Sobester, and Keane 2008, and the noise function is $r(x) = 1.1 + \sin(2\pi x)$. The experiment starts with an initial maximin LHS with 10 points, no replicates, and the GPs use a Gaussian kernel.

Results are presented in Figure 3.4 for a total budget of $N_{\max} = 500$. Each panel in the figure corresponds to a different look-ahead horizon h, with the final two involving Adaptive and Target schemes. There are several noteworthy observations. Notice that as

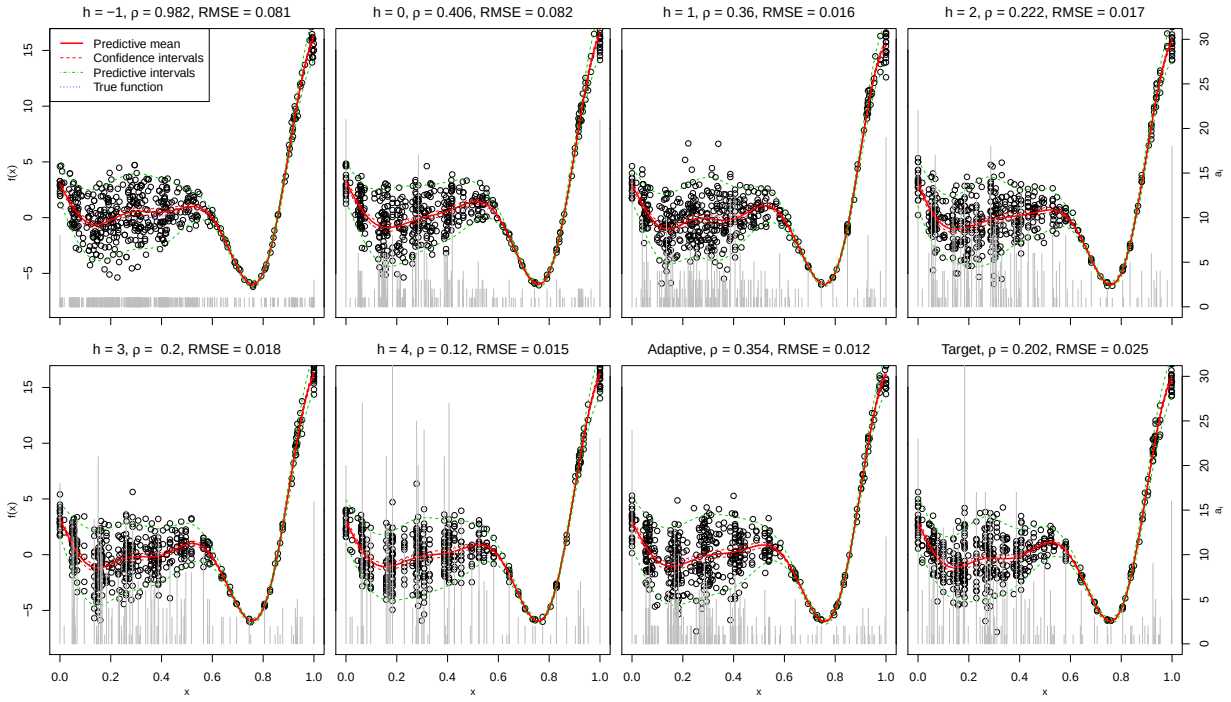


Figure 3.4: Illustrative one-dimensional example. Results on the 1d example by varying the horizon. Grey vertical segments indicate the number of replicates at a given location.

the horizon is increased, more replicates are added. See $\rho = n/N$ reported in the main title of each panel. The design density is greatest in the high variance parts of the space, and that density is increasingly replaced by replication when the horizon is increased. The effect is most drastic from $h = -1$ to $h = 0$, with the ratio of unique designs over total designs dropping by more than half without impact on performance. Notice that replicates are added even with $h = -1$, at the extremities of the space. Results with high horizons and Adapt and Target schemes end up having both fewer unique designs and a higher accuracy. The very best RMSE results are provided by $h = 4$ and the Adapt (3.15) scheme. In the latter case just 60 unique locations are used ($\rho = 0.12$), with some design points replicated as many as thirty times.

3.6.2 Synthetic simulation experiment

Here we expand previous 1-d illustrative example by exploring variation over data-generating mechanisms via Monte Carlo (MC) with input space $\mathbf{x} \in [0, 1]$. Using the hyper-parameter setting outlined in Section 3.2.3, we consider a process with noise structure $\mathbf{\Lambda}_n$ sampled as $\log \mathbf{\Lambda}_n \sim \mathrm{GP}(0, \nu_g \mathbf{C}_{(g)})$, where $\mathbf{C}_{(g)}$ is stationary with Matérn 5/2 kernel $k_{(g)}$. Then observations are drawn via $Y|\mathbf{\Lambda}_n \sim \mathrm{GP}(0, \mathbf{K}_n)$, where $\mathbf{K}_n = \nu(\mathbf{C}_n + \mathbf{\Lambda}_n)$ and $\mathbf{C}_n$ is again Matérn 5/2. We set $\theta = 0.1$, and $\nu = 1$ for the mean GP, and $\theta_{(g)} = 0.5$ and $\nu_{(g)} = 7^2$ for the noise GP. To manage the MC variance between runs we normalized the $\mathbf{\Lambda}_n$-values thus obtained so that the average signal-to-noise ratio was one.

We considered a budget of $N = 200$ and studied various strategies for design—comparing one-shot space-filling designs without or with replication to sequential designs with a lookahead horizon of $h = 0$—and for modeling, testing both homoskedastic and heteroskedastic GPs. These are enumerated as follows: (i) homoskedastic GP without replication using an $n = 200$ grid design; (ii) `hetGP` without replication, again with an $n = 200$ grid; (iii) `hetGP` with one-shot space-filling design with random replication on an $n = 40$ grid with random $a_i \in \{1, \ldots, 10\}$; (iv) sequential learning and design using a homoskedastic GP initialized with a single-replicate $n = 40$ grid, iterating until $N = 200$; and (v) sequential learning and design using `hetGP` initialized with a single-replicate $n = 40$ grid, iterating until $N = 200$.

Figure 3.5 summarizes the results from 1000 MC replicates, illustrating the subtle balance between replication and exploration. As can be seen in the left panel, our pro-

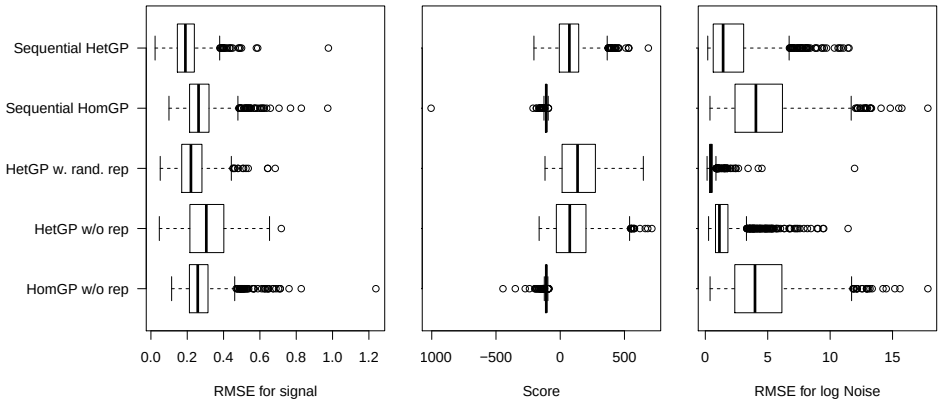


Figure 3.5: Synthetic simulation experiment. Result from on-dimensional synthetic Monte Carlo experiment in terms of RMSE to the true mean (left), proper scores (middle) and RMSE to true log noise (right).

posed sequential `hetGP` performs the best in terms of out-of-sample RMSE. To investigate the statistical significance of RMSE differences we conducted one-sided matched Wilcoxon signed-rank tests of adjacent performers (better v. next-best) with the order based on median RMSE. The corresponding p-values are 4.80×10^{-24}, 9.66×10^{-26}, 0.0956 and 2.66×10^{-12}. For example, the test involving our best method, `hetGP` with sequential design, versus the second best, `hetGP` with random replication, suggests that the former significantly out-performs the latter. Since our IMSPE design criteria emphasized mean-squared prediction error, it is refreshing that our proposed method wins (significantly) on that metric. The only such comparison which did not "reject the null at the 5% level" involved pitting sequential versus uniform design with a homoskedastic GP. The value of proceeding sequentially is much diminished without the capability to learn a differential noise level.

The center and right panels of the figure show that other design variations may be preferred for other performance metrics. Observe that one-shot space-filling design with ran-

dom replication using `hetGP` wins when using proper scores. Apparently, random replication yields better estimates of predictive variance when comparing to the truth. See the right panel. Space-filling and uniform replication are easily achieved in this one-dimensional case, but may not port well to higher dimension as our later, more realistic, examples show. Our sequential `hetGP`, coming in second here on score and log noise RMSE, offers more robustness as the input dimension increases.

3.6.3 Susceptible-Infected-Recovered (SIR) epidemic model

Our first real example deals with estimating the future number of infecteds in a stochastic Susceptible-Infected-Recovered (SIR) epidemic model. This is a standard model for cost-benefit analysis of public health interventions related to communicable diseases, such as influenza or dengue. For our purposes we treat it as a 2d input space indexed by the count $I_0 \in \mathbb{N}$ of initial infecteds and $S_0 \in \mathbb{N}$ of initial susceptibles (the total population size $M \geq I_0 + S_0$ is pre-fixed; the rest of the population is viewed as immune to the disease). The pair $(I_t, S_t) \in \{S + I \leq M\}$ evolves as a continuous-time Markov chain (easily simulated) following certain non-linear (hence analytically intractable) transition rates, until eventually $I_t = 0$ and the epidemic dies out. The response $f(S, I)$ is the expected aggregate number of infected-days, $\int_0^\infty I_t \, dt$ averaged across the Markov chain trajectories; determining $f(S, I)$ is a first step towards constructing adaptive epidemic response policies. It is important to note that the signal-to-noise ratio is varying drastically over D, with a zero variance at $I = 0$ (where $Y \equiv 0$) and up to $r(\mathbf{x}) \approx 90^2$ on the left part of the domain, in the critical region

where the stochasticity in infections leads to either a quick burn-out in infecteds or a rapid infection of a significant population fraction.

Whereas Binois, Gramacy, and Ludkovski 2018 considered static space-filling designs with random numbers of replicates, with a favorable comparison to SK, here we focus on aspects of sequential design, in particular the effect of horizon h in the IMSPE with lookahead over replication. We perform a Monte Carlo experiment wherein designs are initialized with $n = N = 10$ unique design locations (just one observation each), and grown to size $N = 500$ over sequential design iterations, disregarding how many unique locations n are chosen along the way. A Matérn kernel with $\nu = 5/2$ is used. The experiment is repeated 30 times and averages of various statistics are reported in Figures 3.6, 3.7 and Table 3.1 based on a testing set placed on a dense grid with a thousand replications each.

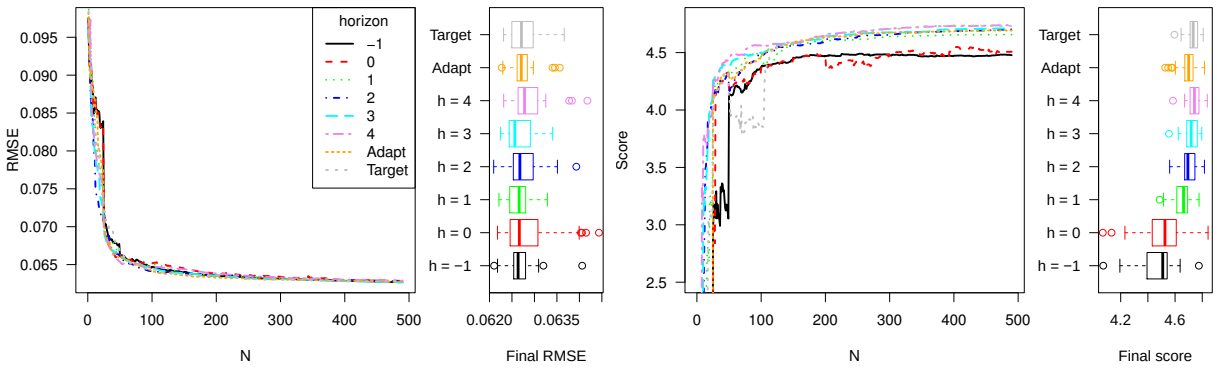


Figure 3.6: Sequential design for SIR epidemic model. RMSE and score results on the SIR test case over sequential design iterations, via summaries 30 MC repetitions. The Target scheme aims for $\rho = n/N = 0.2$.

In Figure 3.6, the results in terms of RMSEs and score are presented. While the RMSEs are barely distinguishable, the scores exhibit more spread, and the best results are obtained by the methods leaning the most toward replication (i.e., $h = 4$ and Target scheme).

Since the signal-to-noise ratio is low in some parts of the input space, replication is beneficial in terms of RMSE *and score*. One reason for the RMSEs not to be very different between the alternatives is that the underlying function is very smooth. However, the variance surface is more challenging, such that having more replicates is helpful in this case, as highlighted by the differences in score, shown in the final panel of Figure 3.6.

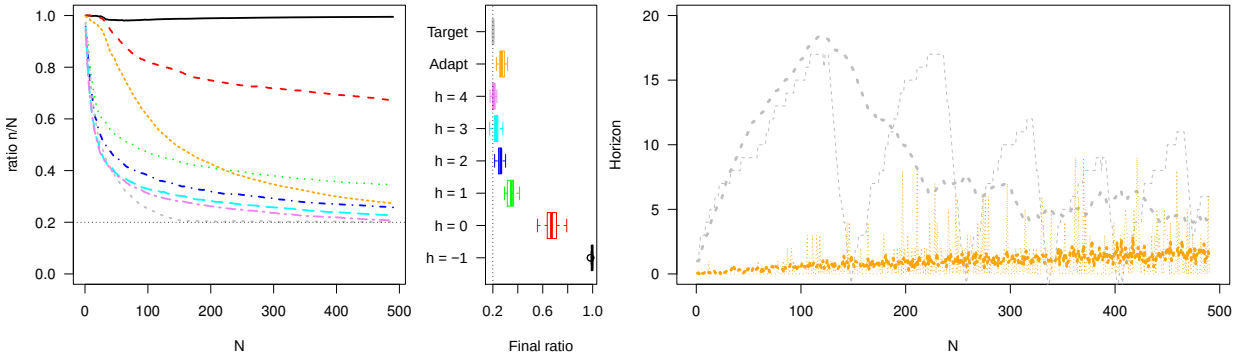


Figure 3.7: Ratio n/N and horizon evolution on SIR. Ratio n/N and horizon evolution on the SIR test case over sequential design iterations, via summaries over 30 MC repetitions. The thin dotted line indicates the Target ratio of 0.2. Right: thin dotted (resp. thick) lines represent one iteration (resp. average) of the horizons for the Adapt and Target schemes. Colors are the same as in Figure 3.6.

Table 3.1: Average percentage of designs points with no replicates, with more than five, and running time on the SIR test problem.

Horizon	-1	0	1	2	3	4	Adapt	Target
Percentage of 1s	99.2	49.6	13.6	6.9	4.8	3.5	8.8	4.1
Percentage of 5s and more	0.04	1.6	5.7	6.9	7.7	7.9	6.9	7.6
Time (s)	812	473	278	257	259	271	306	288

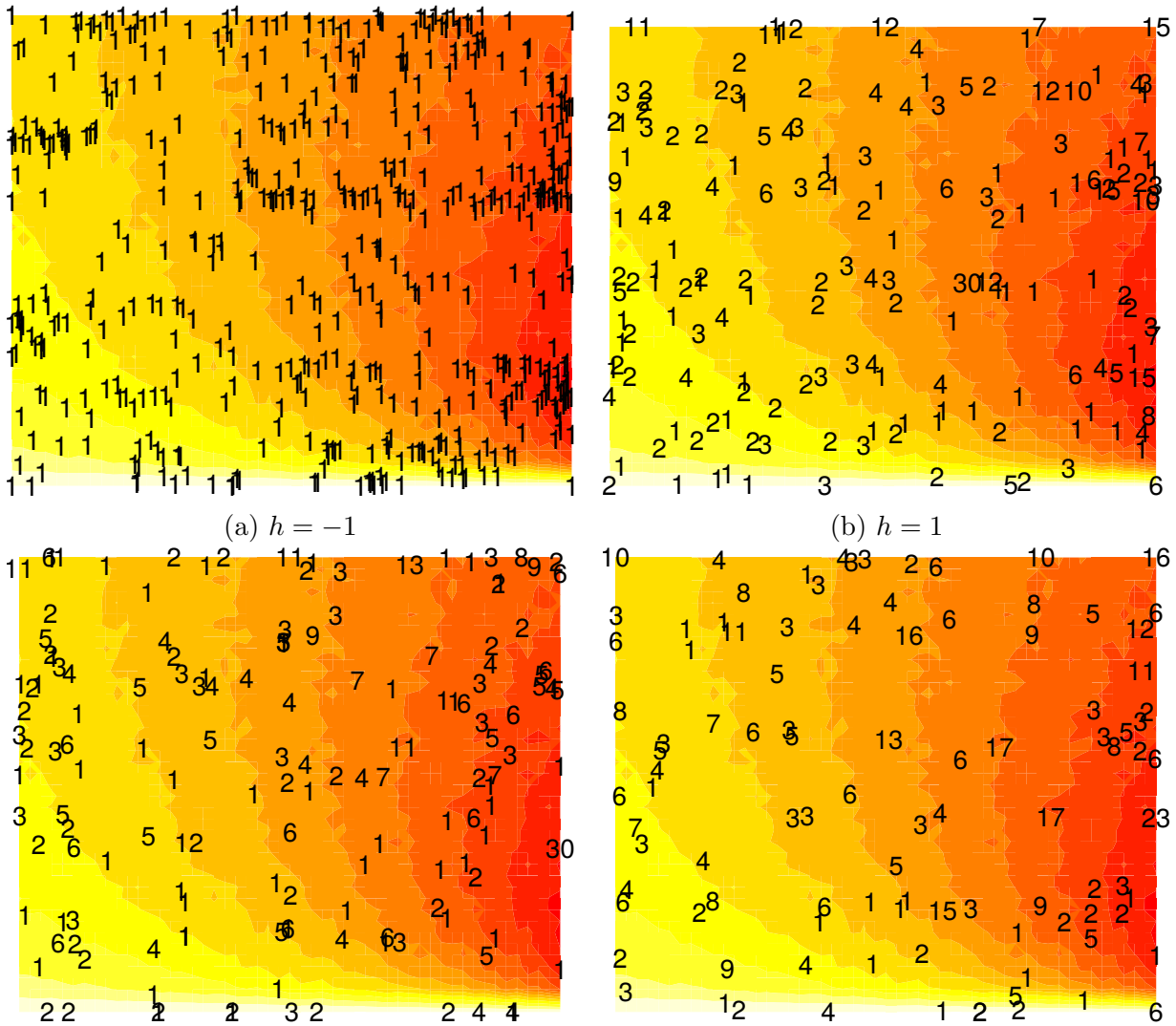
Moving on to Figure 3.7, the left and center panels show the ratio of unique locations over the total design size: n/N. As expected, as the horizon h increases, more replicates are selected. In turn, this lowers the computation time, as reported in Table 3.1. In particular, observe that the computational cost of looking ahead is negligible next to the cost saved

by having smaller n relative to N. The final panel in Figure 3.7 shows how the horizon h evolves when fixing a Target ratio of $\rho = 0.2$ in (3.13), i.e., an average of 5 replicates per unique design location) or learning it with the adaptive scheme (3.15). Notice that the Target scheme with $\rho = 0.2$ sometimes utilizes horizons higher than $h \geq 15$, yet the computational cost is never higher than the high-fixed-horizon results, which offer the best performance for this problem. Due to its random nature, the Adapt scheme changes abruptly between algorithm runs, but its horizon h_N is increasing on average in N.

Figure 3.8 provides a visual indication of the density of design throughout the input space for fixed and tuned horizons. As expected, in all panels the density of inputs in the design is higher in high variance parts of the input space. The numbers in the plot indicate the numbers of replicates a_i. Observe that low-horizon heuristics result in mostly $a_i = 1$, whereas for the longer horizons clusters of tightly grouped unique locations are replaced with replicates. Table 3.1 demonstrates that this feature is consistent over MC repetitions. Thus, our heuristic is adept at capturing the basic logic of amalgamating singleton design locations into replicates, which apparently maintains essentially the same statistical efficiency while reducing computational overhead by a factor of more than 3.

3.6.4 Inventory management

The assemble to order (ATO) simulation, first introduced by Hong and Nelson 2006 with implementation in MATLAB later provided by Xie, Frazier, and Chick 2012, comes from inventory management. The inputs determine stocks and replenishment schedules for key



(a) $h = -1$ (b) $h = 1$

(c) Adapt (d) Target

Figure 3.8: Designs from different horizons for SIR. Designs obtained with different strategies for the horizon, where numbers indicate how many replicates a_i are performed at a given location $\bar{\mathbf{x}}_i$. Darker colors indicate higher variance. The x-axis is the number of susceptibles, from 1200 to 2000 while the y-axis is the initial number of infecteds, from, 0 to 200.

items in assembled products, and the simulator estimates revenue by combining inventory costs with profits obtained from orders which come in following a compound Poisson random process. Binois, Gramacy, and Ludkovski 2018 showed the benefit of heteroskedastic modeling, versus several homoskedastic alternatives, on random space-filling designs with $n = 1000$

unique locations with a random number of replications (uniform in $1, \ldots, 10$) so that the average full data size was $N = 5000$. Here, one of our aims is to illustrate that by building a better design (sequentially), a much lower N is possible without sacrificing accuracy. Binois et al. used a proper scoring rule Gneiting and Raftery 2007, Eq. (27) as their main metric. Since our IMSPE criterion targets accuracy via squared-error loss we report RMSEs, but include scores to facilitate comparison to those space-filling designs. The best average score reported in Figure 2 of that paper was 3.3, with a min and max of 2.8 and 3.6 respectively.

Similar to the SIR experiment, we perform the following variations on sequential IMSPE design, varying the horizon, h, of lookahead and offering the two adaptive horizon schemes outlined in Section 3.4. We initialize with $n = 100$ unique space-filling locations and a random number of replicates, uniform in $\{1, \ldots, 10\}$ so that the starting size is $N = 500$ on average. Subsequently, sequential design iterations are performed until $N = 2000$ total samples are gathered, irregardless of how many unique locations, n, result. The experiment is repeated in a Monte Carlo fashion, with thirty repeats.

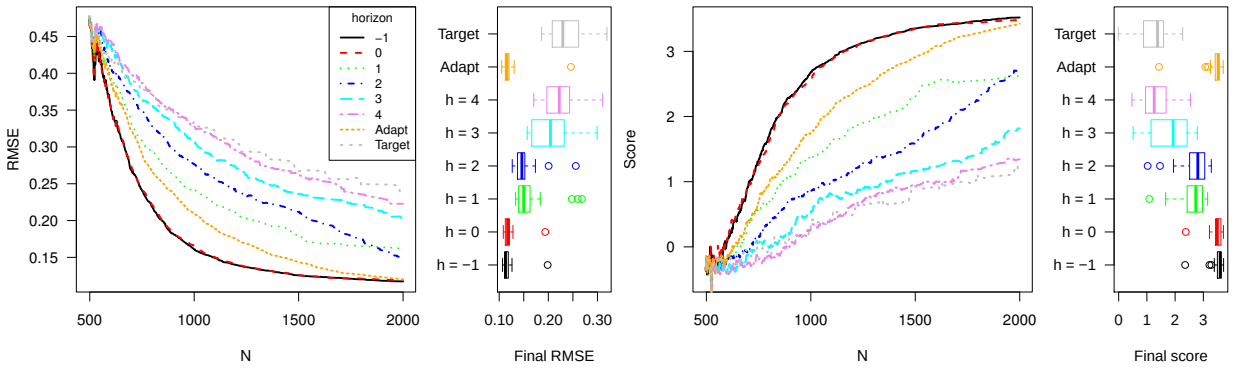


Figure 3.9: RMSE and score results on the ATO problem. Implemented by thirty Monte Carlo iterations, in a format similar to Figure 3.6.

Figure 3.9 summarizes the results of the experiment in a format similar to Figure

3.6. The take-home message is fairly evident: in contrast to the SIR example, shorter lookahead horizon is better, owing to the relatively higher signal-to-noise ratio. Observe that our average score is 3.5, and the min and max are 2.4 and 3.7 respectively. So scores based on just $N = 2000$ samples are higher than in the space-filling $N = 5000$ experiment, however the spread is a little wider. Finally, note that the Adapt heuristic (3.15) eventually performs as well as the best horizon ($h = -1$). Targeting $\rho = 0.2$ via (3.13), by contrast, leads to far too little exploration. The Adapt scheme required an average of 682 minutes to build up to a design of size $N = 2000$ with an average of $n = 1086$ unique sites (min and max of 465 and 1211 respectively), whereas Target took only 183 minutes thanks to using $n = 400$ locations on average (399 to 405).

3.7 Conclusion and perspectives

This chapter addresses a design question which has been around the surrogate modeling literature, and informally in the community, for many years. There is general agreement that the "fully batched" version of the problem, of finding n locations and numbers of replicates $a_1, \ldots, a_n$ on each, is not computationally tractable, although there are some attempts in the recent literature. We therefore consider the simpler task of deciding whether the *next* sample should explore or replicate in a sequential design context. The condition we derive is simple to express, and leads to an intuitive suite of visualizations. Proceeding sequentially has merits, not only computationally but also facilitating "as you go" adjustments to help avoid pathologies arising from feedbacks between design and inference. However the

procedure is still myopic. To help correct this we introduced a computationally tractable lookahead scheme that emphasizes the role of replication in design. Tuning the horizon of that scheme allows the user to trade off the dual roles of replication in surrogate modeling design: computational thriftiness of inference against out-of-sample accuracy, although as we show these are not always at odds.

Our presentation focused on the integrated mean-squared prediction error (IMSPE) criteria. We chose IMSPE because it is popular, but also because it leads to closed form derivatives for optimization, updating equations for search over look-ahead horizons, and simplifications for entertaining replicates. There is, of course, a vast literature on model-based design criteria (see, e.g., Chen and Zhou 2017; Kleijnen 2015) targeting alternative quantities of interest, such as entropy or information for unknown model parameters. Although designs for prediction and estimation sometimes coincide, like for linear regression, the correlation structure for GPs can be a game-changer (Müller, Pronzato, and Waldl 2012). It may well be that other criteria lead to strategies similar to ours, which may be an interesting avenue for future research.

Our implementation and empirical work leveraged a new heteroskedastic Gaussian process modeling library called `hetGP`, available for `R` on CRAN (Binois, Gramacy, and Ludkovski 2018). Our IMSPE, updates, lookahead procedures, and more are provided in a recently updated version of the package. To aid in reproducibility, our supplementary material contains codes using that library to reproduce the smaller examples from the paper [Figures 3.2–3.4]. The other examples require rather more computing, and/or linking be-

tween R and MATLAB for simulation [ATO], which somewhat challenges ease of replication. However, we are happy to provide those codes upon request.

Processes (i.e., data generating mechanisms) benefiting from a heteroskedastic feature bring out the best in our sequential design schemes, demanding a greater degree of replication in high-noise regions relative to low-noise ones, confirming the intuition that replication becomes more valuable for separating signal from noise as the data get noisier (e.g., Wang and Haaland 2017). However, the results we provide are just as valid in the homoskedastic setting, albeit with somewhat less flair. In that context, inferring the right level of replication is a global affair, except perhaps at the edges of the input space which tend to prefer a slightly higher degree.

Our three sets of examples illustrated that the method both does what it is designed to do, and that designs with the right trade-off between exploration and replication perform better than ones which are designed more naïvely. These examples span a range of features, from low to high noise (and slow to rapid change in noise), low to moderate input dimension, and synthetic to real simulation experiments. The behavior is diverse but the results are consistent: sequential design with lookahead-based IMSPE leads to accurate prediction, and the slight bias toward replication yields computationally more thrifty predictors without a compromise on accuracy. However, sequential design might not always be appropriate. Sometimes batching, at least to a small degree, cannot be avoided. Addressing this situation represents an exciting avenue for further research.

Chapter 4

On-site surrogates for large-scale calibration

4.1 Introduction

With remarkable advances in computing power, complex physical systems today can be simulated comparatively cheaply and to high accuracy by using mature libraries. The ability to simulate has dramatically driven down the cost of scientific inquiry in engineering settings, at least at initial proof-of-concept stages. Even so, computer models often idealize the system—they are biased—or require the setting of tuning parameters: inputs unknown or uncontrollable in actual physical processes in the field.

An excellent example is the simulation of a free-falling object, which is a potentially involved if well-understood enterprise from a modeling perspective. The acceleration due to

gravity might be known, but possibly not precisely. The coefficient of drag may be completely unknown. A model incorporating both factors (in the ideal case where drag is known) but not other factors such as ambient air disturbance or rotational velocity could be biased in consistent but potentially unpredictable ways.

Researchers are interested in calibrating such models to experimental data. With a flexible yet sturdy apparatus, a limited number of field observations from physical experiments can provide valuable information to fine tune, improve fidelity, understand uncertainty, and correct bias between computer simulations and the physical phenomena they model. When done right, tuned and bias-corrected simulations are more realistic, forecasts more reliable, and the entire apparatus can serve as a guide to subsequent simulation redevelopment, if necessary. Such is the enterprise of statistical computer model *calibration*.

Here we are motivated by a calibration and uncertainty quantification goal in the development of a so-called *honeycomb seal*, a component in high-pressure centrifugal compressors, with collaborators at Baker Hughes, a General Electric company (BHGE). Several studies in the literature treat similar components from a mechanical engineering perspective e.g., D'Souza and Childs 2002. To our knowledge, however, no one has yet coupled mathematical models and field experimentation in this setting. With access to a commercial simulator and having performed a limited field experiment, our BHGE colleagues performed a nonlinear least-squares (NLS) calibration as a proof of concept. The results left much to be desired.

Although we were initially optimistic that we could readily improve on this method-

ology, a careful exploratory analysis on computer model and field data revealed challenges hidden just below the surface. These included data size (simultaneously too little and too much), dimensionality, computer model reliability, and the nonstationary nature of the dynamics under study. Taken separately, each stretches the limits of the canonical computer model calibration setup, especially in our favored Bayesian setting. Taken all at once, these challenges demanded a fresh perspective.

Contributions by Kennedy and O'Hagan 2001a, KOH and Higdon et al. 2004 lay the foundation for flexible Bayesian calibration of computer experiments, tailored to situations where the simulations are computationally expensive and cheap, respectively. Our situation is somewhere in between, as we describe in more detail in Section 4.2. To set the stage and establish some notation, we offer the following brief introduction. Denote by $\mathbf{x} \in \mathbb{R}^{p_x}$ the controllable inputs in a physical experiment and by $\mathbf{u} \in \mathbb{R}^{p_u}$ any additional (tuning) parameters to the computer model that are unobservable or uncontrollable (or even meaningless, such as mesh size) in the field. In the KOH framework, the physical field observations $y^F(\mathbf{x})$ are connected with computer model simulations $y^M(\mathbf{x}, \mathbf{u}^*)$ through a discrepancy term, or bias correction $b(\mathbf{x})$, between simulation and field as follows:

$$y^F(\mathbf{x}) = y^M(\mathbf{x}, \mathbf{u}^*) + b(\mathbf{x}) + \epsilon. \tag{4.1}$$

Here, $\mathbf{u}^*$ is the unknown "true" or "best" setting for the calibration input parameters, and $\epsilon \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\epsilon^2)$ represents random noise in the field measurements.

The main distinguishing feature between KOH and the work of Higdon et al. is the

treatment of $y^M(\cdot,\cdot)$. If simulation is fast, then Higdon et al. describe how evaluations may be collected on-demand within the inferential procedure, for each choice of $\mathbf{u}$ entertained, with bias $b(\cdot)$ trained directly on residuals $y^F(\mathbf{X}^F) - y^M(\mathbf{X}^F, \mathbf{u})$ observed at a small number of N_F field data input sites, $\mathbf{X}^F$. When simulations are slow or if the computer model is not readily available for on-demand evaluation, then KOH prescribe surrogate modeling to obtain a fitted $\hat{y}^M(\cdot,\cdot)$ from N_M training evaluations $[(\mathbf{X}^M, \mathbf{U}^M), \mathbf{Y}^M]$, with inference being joint for the bias $b(\cdot)$ and tuning parameter settings, $\mathbf{u}$, via a Bayesian posterior. If Gaussian processes (GPs) are used both for the surrogate model and bias, a canonical choice in the computer experiments literature Sacks et al. 1989; Santner, Williams, and Notz 2003, then that posterior enjoys a large degree of analytical tractability. Numerical methods such as Markov chain Monte Carlo (MCMC) facilitate learning in $\mathbf{u}$-space, potentially averaging over any GP hyperparameters, such as characteristic lengthscale or nugget. For GP details, see Rasmussen and Williams 2006.

The KOH framework has been successfully implemented in many applications and has demonstrated empirically superior predictive power for new untried physical observations, that is, out-of-sample, compared with many other techniques. The method is at the same time highly flexible and well regularized. Its main ingredients, coupled GPs ($\hat{y}^M$ and $\hat{b}$) and a latent input space ($\mathbf{u}$), have separately been proposed as tactics for adding fidelity to fitted GP surfaces, in particular as a thrifty means of relaxing stringent stationarity assumptions Ba and Joseph 2012; Bornn, Shaddick, and Zidek 2012. However, KOH is not without its drawbacks. One is identifiability, which is not a primary focus of this paper see,

e.g., Plumlee 2017; Tuo and Wu 2015. Of more pressing concern for us are computational demands, especially in the face of the rapidly growing size of modern computer experiments, both in the number of runs N_M and in the input dimension p_x or, to a lesser extent, p_u. GPs require calculations cubic in N_M to decompose large $N_M \times N_M$ covariance matrices, limiting experiment sizes to the small thousands in practice. KOH exacerbates this situation with $(N_M + N_F) \times (N_M + N_F)$ matrices. Moreover, a Bayesian analysis in input high dimension (p_x, p_u, or $p_x + p_u$), coupled with the large N_M that could be required to adequately cover such a big computer simulation space, is all but impossible without modification.

Inroads have recently been made in order to effectively and tractably calibrate in settings where the computer experiment is orders of magnitude larger than typical. For example, Gramacy et al. 2015 simplified KOH with three modern ideas: modularization Liu, Bayarri, and Berger 2009 to simplify joint inference, local GP approximation Gramacy and Apley 2015 for fast nonstationary modeling, and derivative-free optimization Abramson et al. 2013; Le Digabel 2011 for point estimation. While effective, Bayesian posterior uncertainty quantification ("the baby") was all but thrown out ("with the bath water").

Here we propose a setup that borrows some of these themes, while at the same time backing off on others. We develop a flavor of local GP approximation that we call an *on-site surrogate, or OSS* that does not require modularization in order to fit within the KOH framework. As a result, we are able to stay within a Bayesian joint inferential setup, although we find it effective to perform a preanalysis via optimization, in part to prime the MCMC. We show how our on-site surrogates accommodate a degree of nonstationarity while

imposing a convenient sparsity structure on otherwise huge $(N_M+N_F)\times(N_M+N_F)$ coupled-GP covariance matrices, leading to fast decomposition under partitioned inverse identities. The result is a tractable framework that leads to calibrated fits that are both more accurate out-of-sample and more descriptive about uncertainties than is the NLS approach previously developed at BHGE for the honeycomb seal.

The reminder of this chapter is outlined as follows. Section 4.2 describes the honeycomb seal application, the challenges stemming from its simulation, and subsequent attempts to calibrate via a limited field data. Section 4.3 introduces our novel on-site surrogate strategy for emulation within a calibration framework and elucidates its merits including its blazingly fast application within an optimization/point-estimate calibration setting. Section 4.4 expands this setup in Bayesian KOH-style. Returning to our motivating example, Section 4.5 demonstrates calibration results from both optimization and fully Bayesian approaches, including comparison with the simpler NLS strategy at BHGE. Section 4.7 concludes this chapter with a brief discussion.

4.2 Honeycomb seal

The honeycomb seal is an important component widely used in BHGE's high-pressure centrifugal compressors to enhance rotor stability in oil and gas applications or to control leakage in aircraft gas turbines. The seal(s) and applications at BHGE are described by $p_x = 13$ design variables $\mathbf{x}$ characterizing geometry and flow dynamics: rotational speed, cell depth, seal diameter and length, inlet swirl, gas viscosity, gas temperature, compress-

ibility factor, specific heat, inlet/outlet pressure, and clearance. The field experiment, from BHGE's component-level honeycomb seal test campaign, comprises $N_F = 292$ runs varying a subset of those conditions, $\mathbf{X}^F$, believed to have greatest variability during turbomachinery operation: clearance, swirl, cell depth, seal length, and seal diameter. Measured outputs include direct/cross stiffness and damping, at multiple frequencies. Here our focus is on the direct stiffness output $y \equiv k_{\text{dir}}$ at 28 Hz.

A few hundred runs in thirteen input dimensions is hardly sufficient to understand honeycomb seal dynamics to any reasonable degree in this highly nonlinear setting. Fortunately, the rotordynamics of seals like the honeycomb are relatively well understood, at least from a mathematical and computational modeling standpoint. Although the input dimension is somewhat high by computer model calibration standards, library-based numerical routines provide ready access to calculations for direct/cross stiffness and damping for inputs like the ones listed above. In what follows, we provide some insight into one such solver and the advantages as well as challenges to using it (along with the field data) to better understand and predict the dynamics of our honeycomb seal.

4.2.1 `ISOTSEAL` simulator

A simulator called `ISOTSEAL`, developed at Texas A&M University (Kleynhans and Childs 1997), offers a relatively speedy evaluation (about one second) of the response(s) of interest for the honeycomb seal under study at BHGE. `ISOTSEAL` is built on bulk-flow theory, calculating gas seal force coefficients based on seal flow physics. Our BHGE colleagues have

developed an R interface mapping seventeen scalar inputs for the honeycomb seal experiment into the format required for `ISOTSEAL`. Thirteen of those inputs match up with the columns of $\mathbf{X}^F$ (i.e., they are $\mathbf{x}$'s); four are tuning parameters $\mathbf{u}$, which could not be controlled in the field.

These comprise statoric and rotoric friction coefficients n_s, n_r and exponents m_s, m_r. They are the *friction factors* of the honeycomb seal. In the turbulent-lubrication model from bulk-flow theory, the shear stress f is a function of the friction coefficient n and exponent m through the Blasius model $f = n\mathrm{Re}^m$, where Re is the Reynolds number Hirs 1973. Applied separately for the stator (s) and rotor (r), friction factors n and m must be determined empirically from experimental data. To protect BHGE's intellectual property, but also for practical considerations, we work with friction factors coded to the unit cube.

$$(n_s, m_s, n_r, m_r)^\top \to (u_1, u_2, u_3, u_4)^\top \in [0,1]^4$$

These are treated as calibration parameters $\mathbf{u}$, with the goal of learning their setting via field data and `ISOTSEAL` simulations.

Although `ISOTSEAL` is fast and has a reputation for delivering outputs faithful to the underlying physics, we identified several drawbacks in our application. For some input settings `ISOTSEAL` fails to terminate, especially with friction factors ($\mathbf{u}$) near the boundary of their physically meaningful ranges.[1] Figures 4.1 and 4.2 demonstrate the missing pattern

[1] At least this is the case for the commercial version of the simulator in use at BHGE, paired with their input–mapping front-end. The R wrapper aborts the simulation and returns `NA` after seven seconds of execution.

of the ISOTSEAL simulator on two of these physical input sites with highest missing rate, site 41 at $\mathbf{x}_{41}$ and site 135 at $\mathbf{x}_{135}$. The pairwise plot shown are the converged runs out of total 50,000-run random space-filling design on $\mathbf{u}$. For site $\mathbf{x}_{41}$, only 13,445 (26.89%) out of 50,000 runs converged successfully and their locations are shown in figure 4.1. For site $\mathbf{x}_{135}$, only 15,655 (31.31%) out of 50,000 runs converged successfully and their locations are shown in figure 4.2. Noticing the missing patterns are nonlinear and discontinuous, both of these plots indicate complex and unknown constraints on the parameter space of ISOTSEAL simulator.

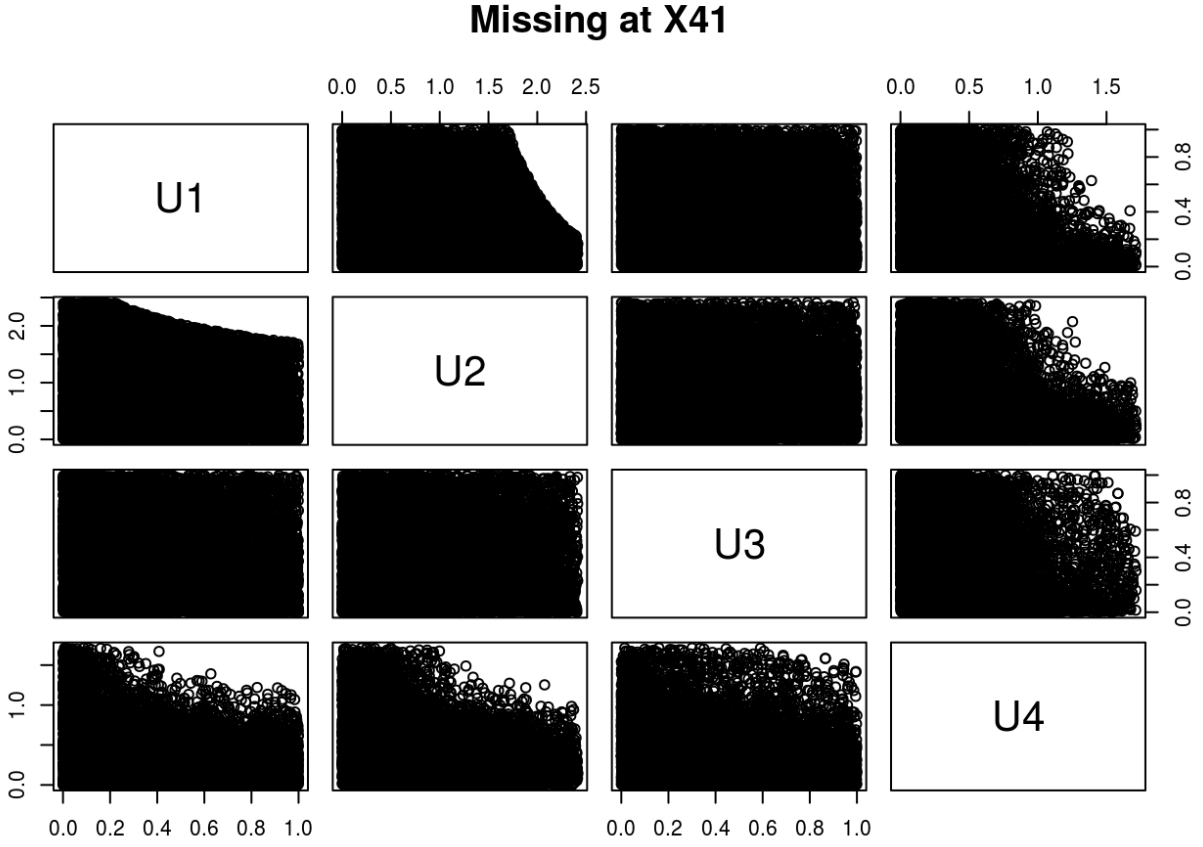


Figure 4.1: Missing pattern on physical site $\mathbf{x}_{41}$. 13,445 (26.89%) runs converge out of total 50,000 on-site ISOTSEAL simulation at site 41 $\mathbf{x}_{41}$ from a random space-filling design on $\mathbf{u}$

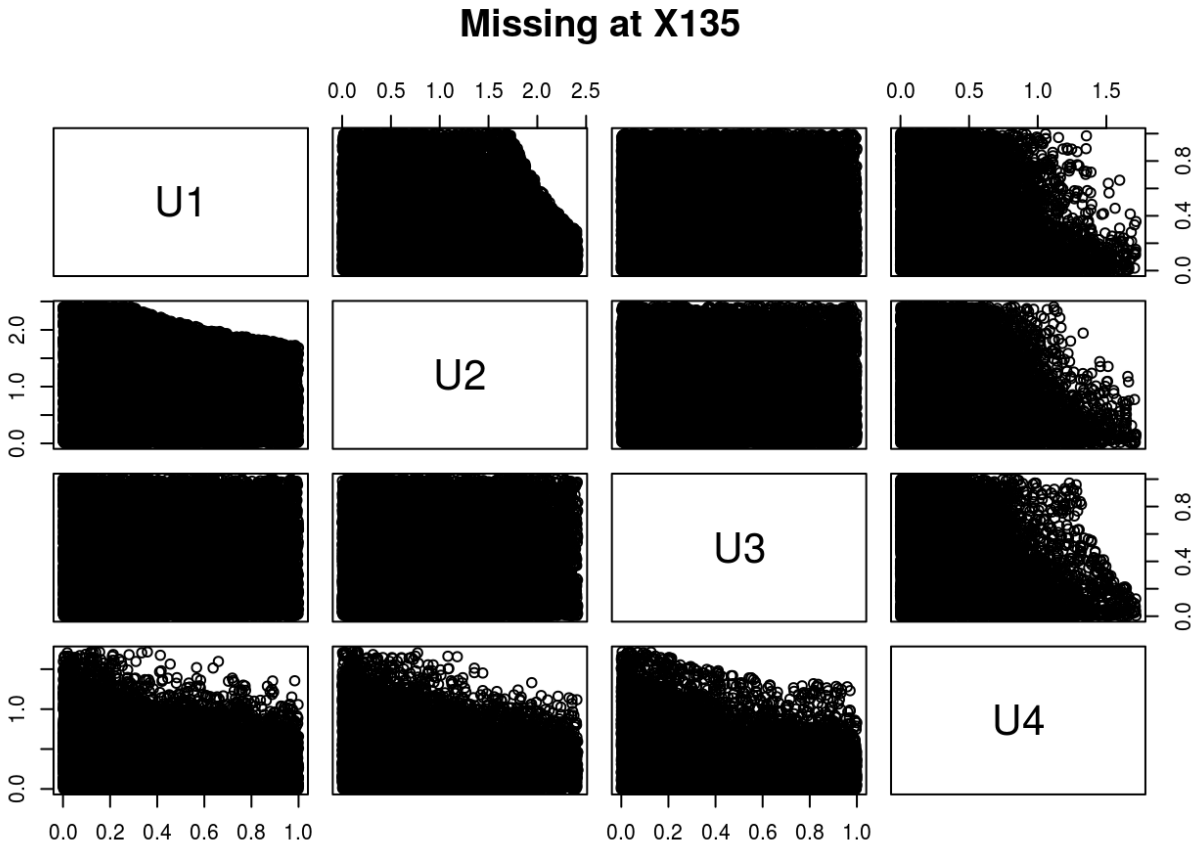


Figure 4.2: Missing pattern on physical site $\mathbf{x}_{135}$. 15,655 (31.31%) runs converged out of total 50,000 on-site `ISOTSEAL` simulation at site 135 $\mathbf{x}_{135}$ from a random space-filling design on $\mathbf{u}$

For others, where a response is provided, numerical instabilities and diverging approximation are evident. Although evaluations are operationally deterministic, in the sense that providing the same input always yields the same output, the behavior can seem otherwise unpredictable in certain regimes. Even subtle numerical "jitters" of this sort can thwart conventional GP interpolation Gramacy and Lee 2012. As we show below, `ISOTSEAL`'s jitters can be extreme in some regimes. Others have commented on similar drawbacks Vannarsdall 2011; modern applications of `ISOTSEAL` may be pushing the boundaries of its engineering.

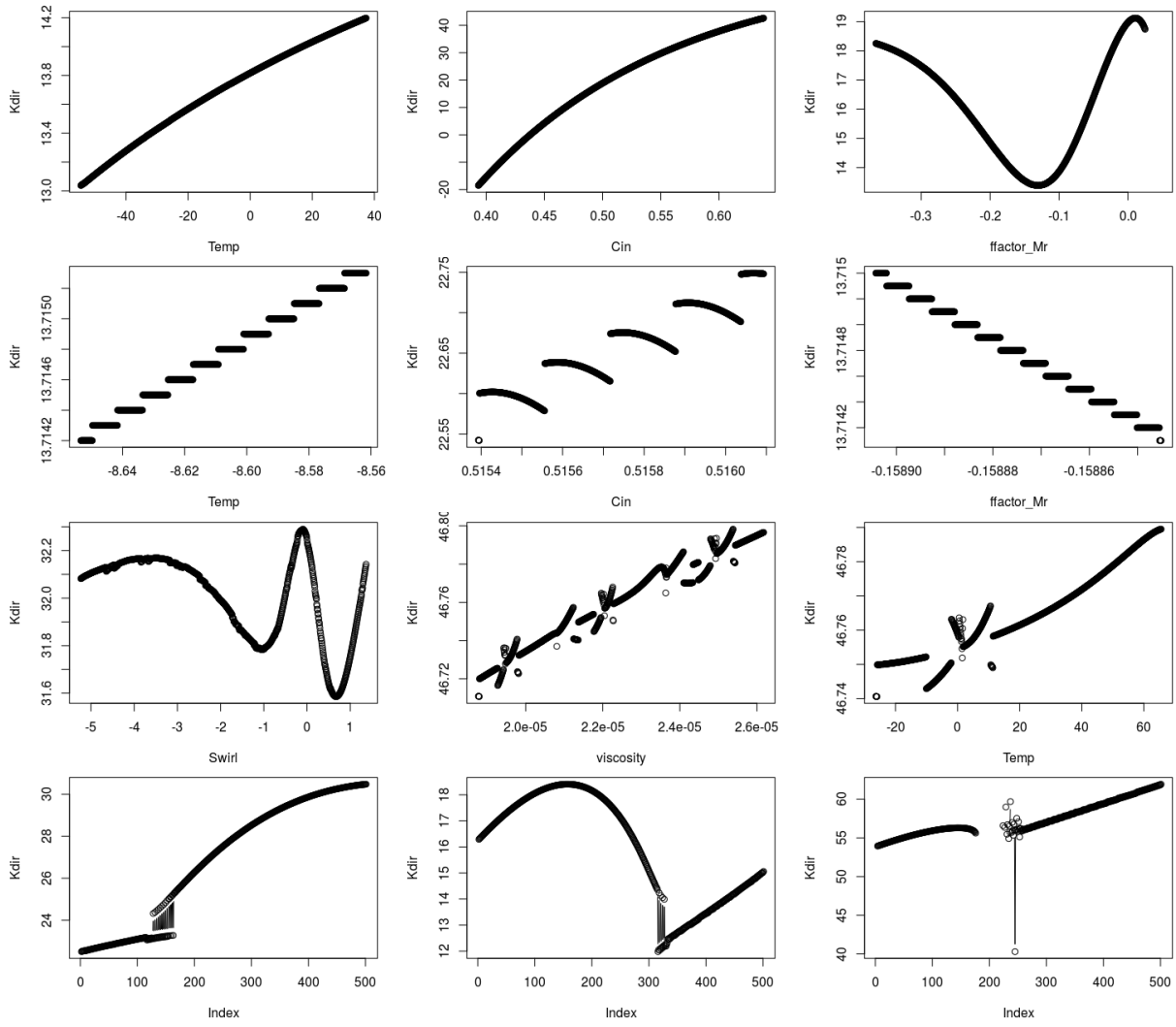


Figure 4.3: Local plots of ISOTSEAL response surface for direct stiffness (K_{direct}). Row 1: change of one input in grid in wide input ranges. Row 2: zoomed-in versions of row 1, changing one input in a much denser grid. Row 3: inexact simulations, changing one input in a grid in input space. Row 4: input trajectory between two arbitrary points from the input space, varying all inputs in grids.

Figure 4.3 shows some example outputs $\mathbf{y}^M$ obtained by varying one input at a time in a narrow range, while fixing the others at sensible values (first three rows); and varying all inputs in grids between two arbitrary points (fourth row). The first row in the figure shows three ideal settings: the response is a smooth function of the input over the

range(s) entertained. The second row, however, which shows zoomed-in versions of the same input–response scenario, exemplifies a "staircase" or "striation" effect sometimes present at small scales. The third row shows more concerning macro-level behavior over both narrow and wide input ranges. According to BHGE's rotordynamics experts, these "staircase" and discontinuity features could be related to tolerances imposed on first-order equilibrium and flow equations implemented in `ISOTSEAL`. As our BHGE rotor-dynamic engineers point out, the coefficients in the first order equation (small perturbation around equilibrium position) in `ISOTSEAL` could be affected by the same discretized tolerances, making the iteration converges at the same point with certain small variation of input variable.

The last row illustrates unpredictable regime-changing behavior and gaps due to termination failure in the simulation. Particular challenges exhibited by the bottom row notwithstanding, dynamics are clearly nonstationary from a global perspective. A great example of this is in the first column of the third row, where the response is at first slowly changing and then more rapidly oscillating. In that example, the regime change is smooth. In other cases, however, as in the middle column of the bottom row, a "noisy" discontinuity separates a hill-like feature from a steadier slope. An ordinary GP model, even with a nugget deployed to smooth over noiselike features by treating them as genuine noise Gramacy and Lee 2012, could not accommodate such regime changes, smooth or otherwise.

Consequently, initial attempts to emulate `ISOTSEAL`-generated response surfaces via the canonical GP in the full (17-dimensional) input space of interest were not successful. Even with space-filling designs sized in the several thousands, pushing the limits of the $\mathcal{O}(N^3)$

bottleneck of large matrix decompositions, we were unable to adequately capture the distinct features we saw in smaller, more localized experiments. Modest reductions in the input dimensions—holding some inputs fixed—and, similarly, reductions in the width of the input domain for the remaining coordinates led to unremarkable improvement in terms of accuracy in out-of-sample predictions. Global nonstationarity, local features, numerical artifacts, and high input dimension proved to be a perfect storm. Section 4.3 uses those unsuccessful proof-of-concept fits as a benchmark, showing how our proposed on-site surrogate offers a far more accurate alternative, at least from a purely out-of-sample emulation perspective.

4.2.2 Nonlinear least-squares calibration

To obtain a crude calibration to the small amount of field data they had, our BHGE colleagues performed a nonlinear least-squares analysis. Starting in a stable part of the input space, from the perspective of `ISOTSEAL` behavior, they used a numerical optimizer—a Nash variant of Marquardt NLS via QR linear solver, `nlfb` Nash 2016—to tune $\mathbf{u}$-values, that is, the four friction factors, based on a quadratic loss between simulated $y_i^M(\mathbf{x}_i, \mathbf{u})$ and observed output $y_i^F(\mathbf{x}_i)$ at the input training data sites $\mathbf{X}^F$.

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \left\{ \frac{1}{N_F} \sum_{i=1}^{N_F} \left[y_i^F(\mathbf{x}_i) - y_i^M(\mathbf{x}_i, \mathbf{u}) \right]^2 \right\}, \tag{4.2}$$

In search for $\hat{\mathbf{u}}$, each new $\mathbf{u}$-value tried by the `nlfb` optimizer triggered N_F calls to `ISOTSEAL`, one for each row of the design parameters $\mathbf{X}^F$, much in the style of Higdon et al. 2004 but without estimating a bias correction. To cope with failed `ISOTSEAL` runs,

`nlfb` monitors the rate of missing values in evaluations. When the missingness rate is below a threshold (e.g., 10%), a predetermined large residual value (100 on the original scale) is imputed for the missed residual to discourage convergence toward solutions nearby. Once above the threshold, `nlfb` reports an error message and is started afresh.

We repeated this experiment, starting instead from 100 random space-filling $\mathbf{u}$ values in hopes of improving on our BHGE colleagues' results with a best value at RMSE = 8.567 and having a strong straw man for later comparison. Because this NLS setup does not model a discrepancy between $\mathbf{y}^M$ and $\mathbf{y}^F$, converged solutions have large quadratic loss, even in-sample. Among 100 restarts, two failed; and the other losses, mapped to the scale of $\mathbf{y}^F$ by taking the square root, had the following distribution.

min	25%	med	mean	75%	max
6.605	8.161	8.401	10.117	9.099	25.787

The blue/circle marks in Figure 4.4 show the observed residuals between field data and NLS calibrated `ISOTSEAL` with the best solution we obtained, $\hat{u} = (0.000, 0.000, 0.821, 0.996)^\top$, which had three out of four friction factors set at or near their limit values. This restart benefited from a serendipitous initialization, having initial RMSE of 9.219 converging to 6.605. However, Figure 4.4 shows that many large residuals still remain (blue/circles). The red/crosses comparator is based on our proposed methodology and is described in subsequent sections. For comparison, and to whet the reader's appetite, we note that the in-sample RMSE we obtained was 1.125. Out-of-sample results are provided in Section 4.6. We attribute NLS's relatively poor performance to two features. One is its inability to compensate for biases in `ISOTSEAL` runs, relative to the outcome of field experiments. Another is that

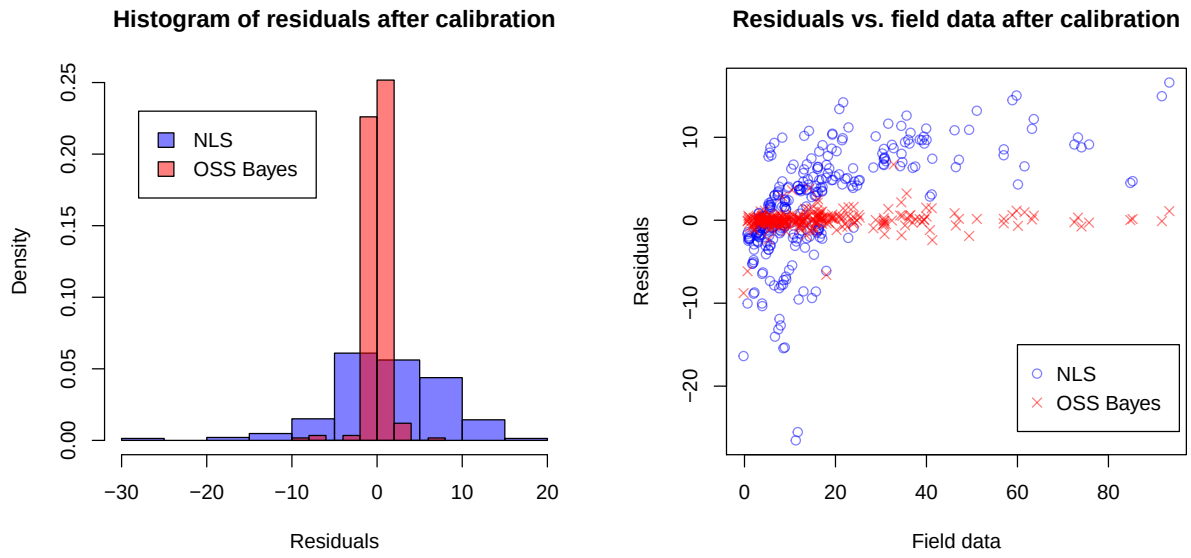


Figure 4.4: In-sample residuals between NLS and OSS Bayes calibration. Observed in-sample residuals between NLS calibrated `ISOTSEAL` and OSS Bayes from field data. The left panel shows histograms of the residuals; the right panel shows them versus the true response. The NLS has in-sample RMSE = 6.605. The OSS Bayes has in-sample RMSE = 1.125, which is further discussed in Section 4.6.

the solutions found were highly localized to the neighborhood of the starting configuration. A post mortem analysis revealed that this was due primarily to large missingness rates.

Although we were confident that we could improve on this methodology and obtain more accurate predictions by correcting for systematic bias between field and simulation in a Bayesian framework, it quickly became apparent that a standard, KOH-style analysis would be fraught with difficulty. In a test run, we used a space-filling design $\mathbf{X}^M$ and fit a global GP emulator in the 17-dimensional space of `ISOTSEAL` runs $\mathbf{y}^M$ thus obtained. That surrogate offered nice-looking predictive surfaces and provided posterior surfaces for calibrated friction factors substantially different from those obtained from NLS (e.g., away from the boundary), but unfortunately the surrogates were highly inaccurate out of sample,

as we illustrate empirically below.

4.3 Local design and emulation for calibration

Failed attempts at surrogate modeling `ISOTSEAL`, either generally or for the specific purpose of calibration to field data [see Section 4.2.1], motivate our search for a new perspective. Local emulation has been proposed in the recent literature Gramacy et al. 2015 as a means of circumventing large-data GP surrogate modeling for calibration, leveraging the important insight that surrogate evaluation is required only at field data locations $\mathbf{X}^F$, of which we have relatively few ($N_F = 292$). But in that context the input dimension was small, and here we are faced with the added challenges of numerical instability, nonstationary dynamics, and missing data. In this section we port that idea to our setting of on-site surrogates, leveraging relatively cheap `ISOTSEAL` simulation, while mitigating problems of big N_M, big $p_x + p_u$, and challenging simulator dynamics.

4.3.1 On-site surrogates

On-site surrogates reduce a $p = p_x + p_u = 17$-dimensional problem into a $p_u = 4$-dimensional problem by building as many surrogates as there are field data observations, $N_F = 292$. Let $\mathbf{x}$ denote a generic design variable setting and $\mathbf{u}$ a generic tuning vector (e.g., friction factor in `ISOTSEAL`). Then the mapping from one big surrogate to many smaller ones

may be conceptualized by the following chart:

$$\hat{y}^M(\mathbf{x}, \mathbf{u}) \longrightarrow \hat{y}^M(\mathbf{x}_i, \mathbf{u}) \longrightarrow \hat{y}_i^M(\mathbf{u}), \quad \text{for } \; i = 1, 2, \ldots, N_F. \tag{4.3}$$

That is, rather than building one big emulator for the entire p-dimensional input space $\hat{y}^M(\mathbf{x}, \mathbf{u})$, we instead train separate emulators $\hat{y}_i^M(\mathbf{u})$ focused on each site $\mathbf{x}_i$ where field data has been collected. In this way, OSSs are a divide-and-conquer scheme that swap joint modeling in a large $(\mathbf{x}, \mathbf{u})$-space, where design coverage and modeling fidelity could at best be thin, for many smaller models in which, separately, ample coverage is attainable with modestly sized design in $\mathbf{u}$-space only. Fitting and simulation can be performed in parallel, since the calculations for each field data site $\mathbf{x}_i$, $i = 1, \ldots, N_F$ are both operationally and statistically independent. Nonstationary modeling is implicit, since each surrogate focuses on a different part of the input space. If simulations are erratic for some $(\mathbf{x}_i, \mathbf{u})$, say, the OSS indexed by i can compensate by smoothing over with nonzero nuggets Gramacy and Lee 2012. If dynamics are well behaved for other sites j, those OSSs can interpolate after the typical fashion.

In some ways, OSSs are akin to an in situ emulator Gul et al. 2018. Whereas the in situ emulator is tailored to uncertainty quantification around a nominal input setting, however, the OSS is applied in multitude for each element of $\mathbf{X}^F$ in the calibration setting. Another key distinction is the role of design in building the elements of the OSS. Here we propose separate designs at each $\mathbf{x}_i$ to learn each $\hat{\mathbf{y}}_i^M(\mathbf{u})$, rather than working with design subsets. We use maximin Latin hypercube sample (hereafter maximin LHS) designs because

of their space-filling and uniform margin properties see, e.g., Morris and Mitchell 1995, via `maximinLHS` in the R package `lhs` Carnell 2018.

Specifically, at each of the $N_F = 292$ field data sites, we create novel 1000-run maximin LHS designs for friction factors in $p_u = 4$-dimensional $\mathbf{u}$-space. In this way, we separately design a total of $N_M = 292000$ `ISOTSEAL` simulation runs. With about one second for evaluation (for successfully terminating runs and about seven seconds waiting to terminate a failed run), this is a manageable workload requiring about 81 core-hours, or about one day on a modern hyperthreaded multicore workstation.

Let $\mathbf{y}_i^M = y^M(\mathbf{U}_i)$ be a vector holding the n_i converged `ISOTSEAL` runs (out of the 1,000) at the i^{th} site, for $i = 1, \ldots, N_F$. $\mathbf{U}_i$ is the corresponding $n_i \times p_u$ on-site design matrix. In our `ISOTSEAL` experiment, where $N_F = 292$, a total of $N_M = \sum_{i=1}^{N_F} n_i = 286,282$ runs terminated successfully. Most sites (241) had $n_i = 1,000$ successful runs from a complete on-site maximin LHS. Of the 51 with missing responses of varying multitudes, the smallest was $n_{238} = 574$.

On each site, the OSS comprises a fitted GP regression between successful on-site `ISOTSEAL` run outputs $\mathbf{y}_i^M$ and $\mathbf{U}_i$. Specifically, $\hat{y}_i^M(\mathbf{U}_i)$ is built by fitting a stationary zero-mean GP using a scaled and nugget-augmented separable Gaussian power exponential kernel

$$V_i(\mathbf{u}, \mathbf{u}') = \tau_i^2 \exp\left\{ - \sum_{k=1}^{p_u} \frac{||\mathbf{u}_{ik} - \mathbf{u}'_{ik}||^2}{\theta_{ik}} + \delta_{u,u'} \eta_i \right\}, \tag{4.4}$$

where τ_i^2 is a site-specific scale parameter, $\boldsymbol{\theta}_i = (\theta_{i1}, \theta_{i2}, \ldots, \theta_{ip_u})^\top$ is vector of site-specific

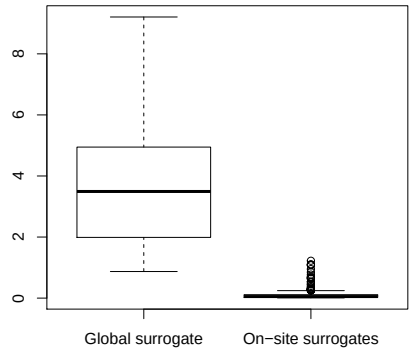
lengthscales, η_i is a nugget parameter, and $\delta_{u,u'}$ is the Kronecker delta[2] . Denote the set of hyperparameters of the i^{th} OSS as $\boldsymbol{\phi}_i = \{\tau_i^2, \boldsymbol{\theta}_i, \eta_i\}$, for $i = 1, 2, \ldots, N_F$. Although nuggets η_i are usually fit to smooth over noise, here we are including them to smooth over any deterministic numerical "jitters." Other mean and covariance structures may be reasonable, so in what follows let $\boldsymbol{\phi}_i$ stand in generically for the estimable quantities of each OSS. Although numerous options for inference exist, we prefer plug-in maximum likelihood estimates (MLEs) $\hat{\boldsymbol{\phi}}_i$, calculated in parallel for each $i = 1, \ldots, N_F = 292$ via `L-BFGS-B` Byrd et al. 1995 using analytic derivatives via `mleGPsep` in the `laGP` package (Gramacy and Sun 2018; Gramacy 2016 for `R`). As we illustrate momentarily, this simple OSS strategy provides far more accurate emulation out-of-sample than does the best global alternative we could muster with a commensurate computational effort.

4.3.2 Merits of on-site surrogates

To build a suitable global GP competitor, we created an $N_M = 8000$-run maximin LHS in $p = 17$ input dimensions, fit a zero-mean GP based on a separable covariance structure (4.4), and estimated the 19-dimensional hyperparameters $\hat{\boldsymbol{\phi}}_g = \{\tau_g^2, \boldsymbol{\theta}_g, \eta_g\}$ via maximum likelihood, identical to the procedure for each $\hat{\boldsymbol{\phi}}_i$ described for the OSSs above. We chose 8,000 runs because that demanded a comparable computational effort to the OSS setup described in Section 4.3.1. Although the `ISOTSEAL` simulation effort for 8,000 runs is far less than the 292K for the OSSs, the hyperparameter inference effort and subsequent

[2]The nugget η_i augmentation is applied only when $\mathbf{u}'$ and $\mathbf{u}$ are identically indexed, i.e., on the diagonal of a symmetric covariance matrix; not simply when their values happen to coincide.

prediction for an $N_M = 8000$-sized design is commensurate with that required for our 292 size $n_i \approx 1000$ OSS calculations. Repeated matrix decompositions in likelihood and derivative calculations in search of the MLE, requiring $\mathcal{O}(N_M^3)$ flops for the global surrogate, represented a heavy burden even when parallelized by multi-threaded linear algebra libraries such as the Intel Math Kernel Library. Similarly threaded calculations of $O(n_i^3)$ flops were faster even in 292 copies, in part because fewer evaluations were needed to learn hyperparameters $\hat{\boldsymbol{\phi}}_i$ in the lower-dimensional $\mathbf{u}$-space.[3]



	global	OSS
min	0.871	0.008
25%	1.991	0.023
med	3.492	0.050
mean	3.619	0.120
75%	4.928	0.112
max	9.207	1.223

Figure 4.5: Boxplots of 292 out-of-sample on-site RMSEs for `ISOTSEAL`. Each RMSE is computed by using novel $n_i' \leq 1,000$ on-site data from both global surrogate and OSSs.

Since the OSSs were trained on a much larger corpus of simulations, it is perhaps not surprising that they provide more accurate predictions out of sample. To demonstrate that empirically, Figure 4.5 summarizes the results of emulation accuracy from both global surrogate and OSSs. For our calibration goal, we need accurate emulation only at locations where we have field data $\mathbf{X}^F$. Therefore we entertain out-of-sample prediction accuracy only for those $\mathbf{X}^F$ sites. At each of the 292 field input sites $\mathbf{x}_i$, we design $\mathbf{U}_i'$ with 1,000

[3]The OSSs learn $|\boldsymbol{\phi}_i| = 6$ compared to $|\boldsymbol{\phi}_g| = 19$ for the global analog. The latter thus demands more expensive gradient calculations. Moreover, the former generally converges to the same local optima when reinitialized, whereas the latter have many local minima due to nonstationary and locally "jittery" responses. Multiple restarts are required to mitigate the chance of finding vastly inferior local optima.

runs each, the same amount as the training set, via maximin LHS. In total we collected $N'_M = 286,224$ testing `ISOTSEAL` runs, which is fewer than we ran since some came back missing. A pair of RMSEs, based on the OSSs and global surrogates, were calculated at each site $i = 1, 2, \ldots, N_F = 292$ based on the $n'_i \approx 1,000$ testing runs located there. The distribution of these values is summarized in Figure 4.5. From those boxplots, one can easily see that the OSSs yield far more accurate predictions.

Figure 4.6 supplements those results with a window into the behavior of the OSSs, in three glimpses. The first row shows three relatively well-behaved input settings by varying two $\mathbf{u}$-coordinates at $\mathbf{x}^F_{17}$, and one at $\mathbf{x}^F_{243}$. In all three cases, the three dashed-red lines describing the predictive distribution (via mean, and 95% interval) completely cover the `ISOTSEAL` simulations in that space. Both flat (middle panel) and wavier dynamics (outer panels) are exhibited, demonstrating a degree of nonstationary flexibility. The horizontal line indicates the field data y^F_i value, and in two of those cases there is a substantial discrepancy between $y^M(\mathbf{x}_i, \mathbf{u})$, and y^F_i for the range of $\mathbf{u}$-values on display. The middle row in the figure shows what happens when `ISOTSEAL` runs fail to converge, again via two $\mathbf{u}$-coordinates for one OSS, at $\mathbf{x}^F_{41}$, and one for another $\mathbf{x}^F_{249}$. Notice that failures happen more often toward the edges of $\mathbf{u}$-space, but not exclusively. In all three cases the extrapolations are sensible and reflect diversity in waviness (first two flatter, third one wavier) that could not be accommodated by a globally stationary model. All three have the corresponding y^F_i-value within range, but only in the extrapolated regime. The last row of the figure shows how a nugget is used to smooth over bifurcating regime changes in the output from `ISOTSEAL`,

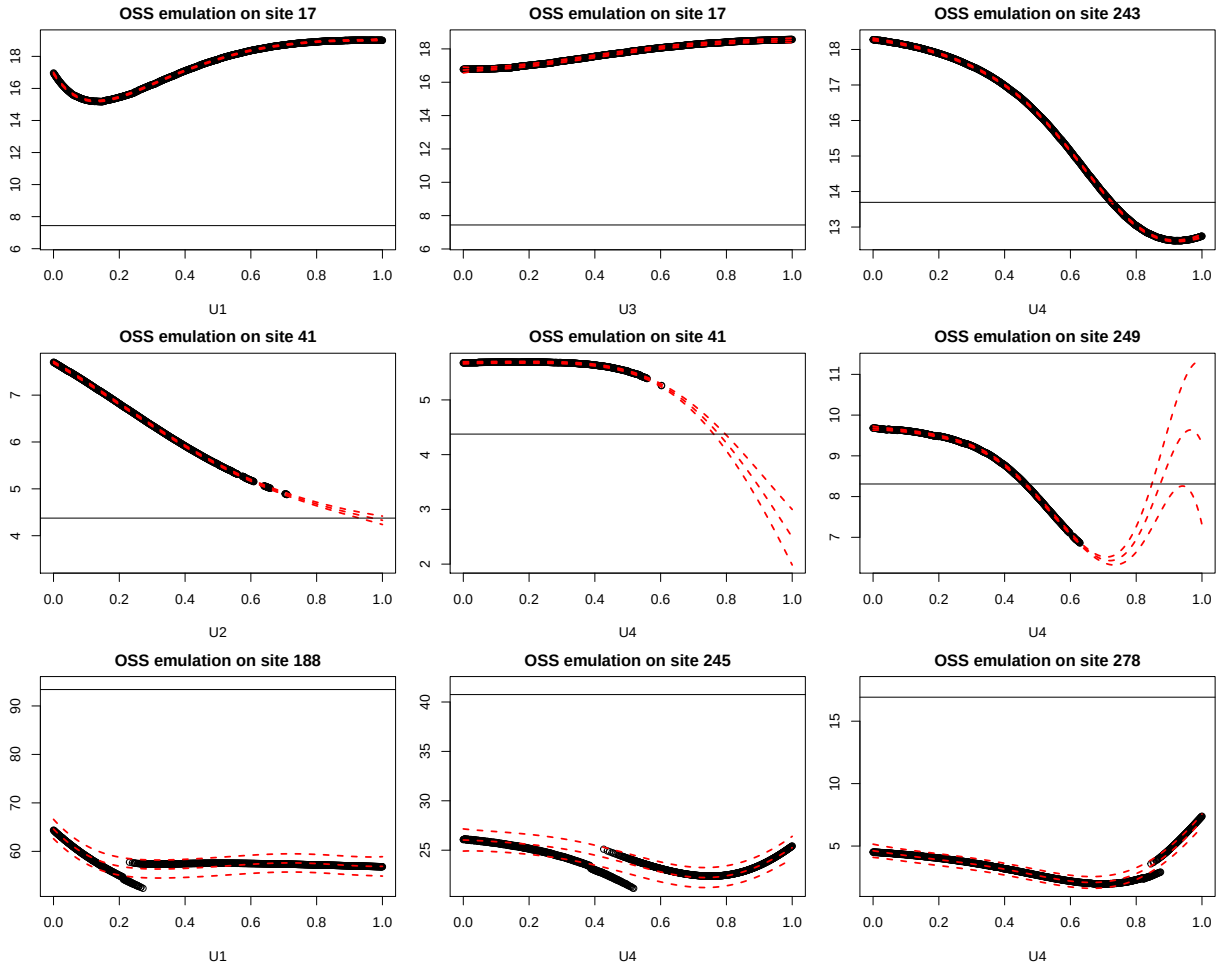


Figure 4.6: Profile plots of OSSs via predictive means and 95% predictive intervals. First row shows three well-behaved cases; middle row illustrates extrapolations to partially missing regimes; last row shows three cases where smoothing is required in order to cope with discontinuities. Red lines are the predicted mean (solid) and 95% predictive intervals (dashed-red). Black horizontal lines show the field response y_i^F at that location, $\mathbf{x}_i$, with i provided in the main title.

offering a sensible compromise and commensurately inflated uncertainty in order to cope with both regimes. All three cases map to outlying RMSE values (open circles beyond the whiskers OSS boxplot) from Figure 4.5. Although they are among the hardest to predict out of sample, the overall magnitude of the error is small. Since the corresponding y_i^F-values (horizontal lines) are far from $y^M(\mathbf{x}_i, \mathbf{u})$, and $\hat{y}_i^M(\mathbf{u})$ in the $\mathbf{u}$-range under study, a substantial

degree of bias correction is needed to effectively calibrate in this part of the input space.

4.3.3 Calibration as optimization with on-site surrogates

Even with accurate OSSs at all field data locations, Bayesian calibration can still be computationally challenging in large-scale computer experiments. In the KOH framework (4.1), both $\mathbf{u}^*$ and a bias correcting GP $b(\mathbf{x})$, via hyperparameters $\boldsymbol{\phi}_b$, are unknown and must jointly be estimated. The size of that parameter space, using a separable Gaussian kernel (4.4) for $b(\cdot)$, is large (19d) in our motivating honeycomb seal application. MCMC in such a high-dimensional space is fraught with computational challenges.

As an alternative to the fully Bayesian method, presented shortly in Section 4.4 taking advantage of a sparse matrix structure, and to serve as a smart initialization of the resulting MCMC scheme, we propose here an adaptation of Gramacy et al. 2015's modularized (Liu, Bayarri, and Berger 2009) calibration as optimization. Instead of sampling a full posterior distribution, $\hat{b}(\cdot)$ and $\hat{\mathbf{u}}$ are calculated as

$$\hat{\mathbf{u}} = \arg\max_{\mathbf{u}} \left\{ p(\mathbf{u}) \left[\max_{\boldsymbol{\phi}_b} p_b(\boldsymbol{\phi}_b \mid \mathbf{D}^B_{N_F}(\mathbf{u})) \right] \right\}, \tag{4.5}$$

which explores different values of $\hat{\mathbf{u}}$ via the resulting posterior probability of discrepancy hyperparameters $p_b(\phi_b \mid \mathbf{D}^B_{N_F}(\mathbf{u}))$ applied to a data set of residuals $\mathbf{D}^B_{N_F}(\mathbf{u})$. Specifically, $\mathbf{D}^B_{N_F}(\mathbf{u}) = (\mathbf{X}^F_{N_F}, \hat{\mathbf{y}}^{B|u}_{N_F})$ is the observed field inputs $\mathbf{X}^F_{N_F}$ and discrepancies $\hat{\mathbf{y}}^{B|\mathbf{u}}_{N_F} = \mathbf{y}^F_{N_F} - \hat{\mathbf{y}}^{M|\mathbf{u}}_{N_F}$ given a particular $\mathbf{u}$. The probability $p_b(\cdot \mid \cdot)$ refers to the marginal likelihood of the GP with parameters $\hat{\boldsymbol{\phi}}_b$ fit to those residuals via their own "inner" derivative-based optimization

routine. The object in Eq. (5.6) basically encodes the idea that **u**-settings leading to better-fitting GP bias corrections are preferred. A uniform prior $p(\mathbf{u})$ is a sensible default; however, we prefer independent $u_j \sim \text{Beta}(2, 2)$ in each coordinate as a means of regularizing the search by mildly penalizing boundary solutions, in part because we know that frictions factors at the boundaries of **u**-space lean heavily on the surrogate as runs of `ISOTSEAL` fail to converge there. Of course, any genuine prior information on **u** could be used here to further guide the calibration.

Actually, this approach is not unlike the NLS one described in Section 4.2.2, augmented with OSSs (rather than raw `ISOTSEAL` runs) and with bias correction. Instead of optimizing a least-squares criterion, our GP marginal likelihood-based loss is akin to a spatial Mahalanobis criterion Bastos and O'Hagan 2009. In practice, the log of the criteria in Eq. (5.6) can be optimized numerically with robust library methods such as "L-BFGS-B", via `optim` (R Core Team 2018), or `nloptr` (Ypma, Borchers, and Eddelbuettel 2017). Since the optimizations are fast but local and since the surface being optimized can have many local optima, we entertain a large set of random initializations—in parallel—in search for the best (most global) solution for $\hat{\mathbf{u}}$ and $\hat{b}(\cdot)$.

To economize on space, we summarize here the outcome of this approach on the honeycomb seal, alongside its fully Bayesian KOH analog. A more detailed discussion of the full Bayesian calibration is provided in Section 4.4 and a discussion of results in Section 4.5. As mentioned above, our main use of this procedure is to prime the fully Bayesian KOH MCMC. Foreshadowing somewhat, we can see from Figure 4.12 that the point estimates $\hat{\mathbf{u}}$

so-obtained are not much different from the maximum *a posteriori* (MAP) found via KOH, yet at a fraction of the computational cost. Since the MCMC is inherently serial and since our randomly initialized optimizations may proceed in parallel, we can get a good $\hat{\mathbf{u}}$ in about an hour, whereas getting a good (effective) sample size from the posterior takes about a day.

4.4 Fully Bayesian calibration via on-site surrogates

The approach in Section 4.3.3 is Bayesian in the sense that marginal likelihoods are used to estimate hyperparameters to the GP-based on-site surrogates and discrepancy $b(\cdot)$, and priors are entertained for the friction factors $\mathbf{u}$. However, the modularized approach to joint modeling, via residuals from (posterior) predictive quantities paired with optimization-based point inference, makes the setup a poor man's Bayes at best. In the face of big data—large N_M, N_F and p_u—such a setup may represent the only computationally tractable alternative. However, in our setting with moderate N_F and $N_M = \sum_{i=1}^{N_F} n_i$ composed of independently modeled computer experiments of moderate size ($n_i \leq 1000$), fully Bayesian KOH-style calibration is within reach. As we show below, a careful application of partition inverse identities allows the implicit decomposition of a huge matrix via its sparse structure.

4.4.1 KOH setup using OSS

Our OSSs for Section 4.3.1 are trained via p_u-dimensional on-site designs $\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_{N_F}$. Their row dimension, $n_i \leq 1000$, depends on the proportion of `ISOTSEAL` runs that successfully completed. Collect these $N_M = \sum_{i=1}^{N_F} n_i$ outputs of those simulations, each tacitly

paired with inputs $\mathbf{x}_i$, as $\mathbf{y}^M = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{N_F})^\top$. The KOH framework compensates for surrogate biased computer model predictions under an unknown setting $\mathbf{u}$ by estimating a discrepancy $b(\cdot)$ via N_F field data runs $\mathbf{y}^F$ observed at $N_F \times p_x$ inputs $\mathbf{X}^F$:

$$\mathbf{y}^F = y^M(\mathbf{U}) + b(\mathbf{X}^F), \quad \text{where} \quad \mathbf{U} = [\mathbf{u}^\top; \cdots; \mathbf{u}^\top]^\top$$

stacks N_F identical p_u-dimensional row vectors $\mathbf{u}^\top$. Under joint GP priors, for each of N_F OSSs and $b(\cdot)$, the sampling model can be characterized by the following multivariate normal (MVN) distribution.

$$\begin{bmatrix} \mathbf{y}^M \\ \mathbf{y}^F \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_{N_F} \\ \mathbf{y}^F \end{bmatrix} = \begin{bmatrix} y_1(\mathbf{U}_1) \\ y_2(\mathbf{U}_2) \\ \vdots \\ y_{N_F}(\mathbf{U}_{N_F}) \\ y^M(\mathbf{U}) + b(\mathbf{X}^F) \end{bmatrix} \sim \mathcal{N}_{N_M+N_F}(\mathbf{0}, \mathbb{V}(\mathbf{u})) \tag{4.6}$$

Generally speaking, $\mathbb{V}(\mathbf{u})$ would be derived by hyperparameterized pairwise inverse distances between inputs on $(\mathbf{x}, \mathbf{u})$-space. In our OSS setup, however, it has a special structure owing to the independent surrogates fit at each $\mathbf{x}_i$, for $i = 1, \ldots, N_F$.

Let $\mathbf{V}_i \equiv V_i(\mathbf{U}_i, \mathbf{U}_i)$ denote the $n_i \times n_i$ covariance matrix for the i^{th} OSS, for example, following Eq. (4.4). This notation deliberately suppresses dependence on hyperparameters $\boldsymbol{\phi}_i$, which is a topic we table momentarily to streamline the discussion here.

Similarly, $V_b \equiv V_b(\mathbf{X}^F)$. Let $V_i(\mathbf{U}) \equiv V_i(\mathbf{U}, \mathbf{U}_i)$ be the $n_i \times N_F$ matrix of the i^{th} OSS's cross-covariances between field data locations, paired with $\mathbf{u}$-values, and $(\mathbf{x}_i, \mathbf{U}_i)$ design locations. Since the i^{th} OSS is tailored to $\mathbf{x}_i$ only, independent of the other $\mathbf{X}^F$, this matrix is zero except in the i^{th} row. Let $\mathbf{v}\mathbb{I}_{N_F}$ be a $N_F \times N_F$ diagonal matrix holding $V_i(\mathbf{u}', \mathbf{u}')$ values. Although expressed as a function of $\mathbf{u}'$ it is not actually a function of $\mathbf{u}'$ because the distance between $\mathbf{u}'$ and itself is zero. Using Eq. (4.4) would yield $\mathbf{v}\mathbb{I}_{N_F} = \text{Diag}[\tau_i^2(1+\eta_i)]$. With those definitions, we have the following:

$$\mathbb{V}(\mathbf{u}) = \begin{bmatrix} \mathbf{V}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & V_1(\mathbf{U})^\top \\ \mathbf{0} & \mathbf{V}_2 & \mathbf{0} & \mathbf{0} & V_2(\mathbf{U})^\top \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{V}_{N_F} & V_{N_F}(\mathbf{U})^\top \\ V_1(\mathbf{U}) & V_2(\mathbf{U}) & \dots & V_{N_F}(\mathbf{U}) & \mathbf{v}\mathbb{I}_{N_F} + V_b(\mathbf{X}^F) \end{bmatrix} \equiv \begin{bmatrix} \mathbb{V}_o & \mathbb{V}_{ob}^\top(\mathbf{u}) \\ \mathbb{V}_{ob}(\mathbf{u}) & \mathbb{V}_b \end{bmatrix}. \quad (4.7)$$

Although $\mathbb{V}(\mathbf{u})$ is huge, being $(N_M+N_F)\times(N_M+N_F)$ or roughly $292292 \times 292292 >$ 85 billion entries in our honeycomb setup, it is sparse, having several orders of magnitude fewer nonzero entries—about 292 million in our setup. That is still very big, too big even for sparse matrix storage and computation on most machines. Fortunately, the block diagonal structure makes it possible to work with, via more conventional libraries. Toward that end, denote by $\mathbb{V}_o = \text{Diag}[\mathbf{V}_i(\mathbf{U}_i, \mathbf{U}_i)]$ the huge $N_F \cdot (n_i \times n_i)$ upper-left block diagonal submatrix from the OSSs. Let $\mathbb{V}_b = \mathbf{v}\mathbb{I}_{N_F} + V_b(\mathbf{X}^F)$ represent the remaining (dense) lower-right block, corresponding to the bias. Abstract by $\mathbb{V}_{ob}(\mathbf{u})$ and $V_{ob}^\top(\mathbf{u})$ the remaining, symmetric, rows

and columns on the edges. Recall that the $V_i(\mathbf{U})$ therein are themselves sparse, comprising a single row of nonzero entries.

Before detailing in Section 4.4.2 how we use these blocks, we focus on the specific operations required. A fully Bayesian approach to inference for the calibration parameters $\mathbf{u}$ via posterior $p(\mathbf{u} \mid \mathbf{y}^M, \mathbf{y}^F) \propto p(\mathbf{y}^M, \mathbf{y}^F \mid \mathbf{u}) \cdot p(\mathbf{u})$ involves evaluating a likelihood, whose MVN structure implies

$$p(\mathbf{y}^M, \mathbf{y}^F \mid \mathbf{u}) \propto |\mathbb{V}(\mathbf{u})|^{-\frac{1}{2}} \times \exp\left\{ -\frac{1}{2} \begin{bmatrix} \mathbf{y}^M \\ \mathbf{y}^F \end{bmatrix}^\top \mathbb{V}^{-1}(\mathbf{u}) \begin{bmatrix} \mathbf{y}^M \\ \mathbf{y}^F \end{bmatrix} \right\}. \tag{4.8}$$

The main computational challenges are manifest in the inverse $\mathbb{V}^{-1}(\mathbf{u})$ and determinant $|\mathbb{V}(\mathbf{u})|$ calculations, both involving $\mathcal{O}((N_M + N_F)^3)$ flops in addition to $\mathcal{O}((N_M + N_F)^2)$ storage, assuming a dense representation. However, substantial savings comes not only from the sparse structure (4.7) of $\mathbb{V}(\mathbf{u})$ but also from the fact that only a portion—the edges—involves $\mathbf{u}$.

4.4.2 On-site surrogate decomposition

Partition inverse and determinant equations (e.g., Petersen and Pedersen 2008) provide convenient forms for the requisite decompositions of $\mathbb{V}(\mathbf{u})$:

$$\begin{aligned}\mathbb{V}^{-1}(\mathbf{u}) &= \begin{bmatrix} \mathbb{V}_o & \mathbb{V}_{ob}^\top(\mathbf{u}) \\ \mathbb{V}_{ob}(\mathbf{u}) & \mathbb{V}_b \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \mathbb{V}_o^{-1} + \mathbb{V}_o^{-1}\mathbb{V}_{ob}^\top(\mathbf{u})\mathbb{C}^{-1}(\mathbf{u})\mathbb{V}_{ob}(\mathbf{u})\mathbb{V}_o^{-1} & -\mathbb{V}_o^{-1}\mathbb{V}_{ob}^\top(\mathbf{u})\mathbb{C}^{-1}(\mathbf{u}) \\ -\mathbb{C}^{-1}(\mathbf{u})\mathbb{V}_{ob}(\mathbf{u})\mathbb{V}_o^{-1} & \mathbb{C}^{-1}(\mathbf{u}) \end{bmatrix} \end{aligned} \tag{4.9}$$

$$|\mathbb{V}(\mathbf{u})| = \det \begin{bmatrix} \mathbb{V}_o & \mathbb{V}_{ob}^\top(\mathbf{u}) \\ \mathbb{V}_{ob}(\mathbf{u}) & \mathbb{V}_b \end{bmatrix} = \det(\mathbb{V}_o) \times \det(\mathbb{C}(\mathbf{u})), \tag{4.10}$$

where $\mathbb{C}(\mathbf{u}) = \mathbb{V}_b - \mathbb{V}_{ob}(\mathbf{u})\mathbb{V}_o^{-1}\mathbb{V}_{ob}^\top(\mathbf{u})$. Eqs. (4.9–4.10) both involve a potentially huge $N_M \times N_M$ component $\mathbb{V}_o$, with $N_M = 286,282$ in the honeycomb example. Since it is block diagonal, thanks to the OSS structure, we have

$$\mathbb{V}_o^{-1} = \text{Diag}[\mathbf{V}_i^{-1}] \quad \text{and} \quad \det(\mathbb{V}_o) = \prod_{i=1}^{N_F} \det[\mathbf{V}_i]. \tag{4.11}$$

In this way, an otherwise $\mathcal{O}(N_M^3)$ operation may instead by calculated via $N_F \times \mathcal{O}(n_i^3)$ calculations, potentially in parallel. If some n_i are big, then the burden could still be substantial. However, both are constant with respect to $\mathbf{u}$, so only one such decomposition is required, even when entertaining thousands of potential $\mathbf{u}$. With $n_i \leq 1000$ in our honeycomb application, these calculations require mere seconds, even in serial.

Similar tricks extend to other quantities involved in Eq. (4.9). Consider $\mathbb{V}_o^{-1}\mathbb{V}_{ob}^{\top}(\mathbf{u})$, which appears multiple times in original and transposed forms. We have

$$\mathbb{V}_o^{-1}\mathbb{V}_{ob}^{\top}(\mathbf{u}) = \text{Diag}[\mathbf{V}_i^{-1}V_i(\mathbf{U})] = \text{Diag}[\mathbf{h}_i(\mathbf{u})] \quad \text{where} \quad \mathbf{h}_i(\mathbf{u}) = \mathbf{V}_i^{-1}V_i(\mathbf{u}), \tag{4.12}$$

and $V_i(\mathbf{u})$ is a vector holding the nonzero part of $V_i(\mathbf{U})$. In other words, $\mathbb{V}_o^{-1}\mathbb{V}_{ob}^{\top}(\mathbf{u})$ is a $N_M \times N_F$ matrix comprising N_F column vectors, whose n_i nonzero entries $\mathbf{h}_i$, are arranged in a block structure for columns $i = 1, \dots N_F$. If required, each $\mathbf{h}_i(\mathbf{u})$ can be updated in parallel for new $\mathbf{u}$.

Next consider $\mathbb{C}(\mathbf{u}) = \mathbb{V}_b - \mathbb{V}_{ob}(\mathbf{u})\mathbb{V}_o^{-1}\mathbb{V}_{ob}^{\top}(\mathbf{u})$, which appears in each block of Eq. (4.9). $\mathbb{C}(\mathbf{u})$ is dense but is easy to compute because it is just $N_F \times N_F$. Recall from Eq .4.7 that $\mathbb{V}_b = \mathbf{v}\mathbb{I}_{N_F} + V_b(\mathbf{X}^F)$, which requires inversion only once because it is constant in $\mathbf{u}$. The next part $\mathbb{V}_{ob}(\mathbf{u})\mathbb{V}_o^{-1}\mathbb{V}_{ob}^{\top}(\mathbf{u})$ extends nicely from $\mathbb{V}_o^{-1}\mathbb{V}_{ob}^{\top}(\mathbf{u}) = \text{Diag}[V_i(\mathbf{u})^{\top}\mathbf{h}_i(\mathbf{u})]$ following Eq. (4.12), an $N_F \times N_F$ diagonal matrix whose entries can be calculated alongside the $\mathbf{h}_i(\mathbf{u})$, similarly parallelized over $i = 1, \dots, N_F$.

Combining $\mathbb{V}_o^{-1}\mathbb{V}_{ob}^{\top}(\mathbf{u})$ and $\mathbb{C}(\mathbf{u})$ results gives $\mathbb{V}_o^{-1}\mathbb{V}_{ob}^{\top}(\mathbf{u})\mathbb{C}^{-1}(\mathbf{u}) = \mathbf{H}(\mathbf{u}) \circ \mathbb{C}^{-1}(\mathbf{u})$, where "$\circ$" is the Hadamard product applied columnwise to $\mathbb{C}^{-1}(\mathbf{u})$ and where $\mathbf{H}(\mathbf{u}) =$

$[\mathbf{h}_1(\mathbf{u}); \ldots; \mathbf{h}_{N_F}(\mathbf{u})]$. More concretely,

$$\mathbb{V}_o^{-1}\mathbb{V}_{ob}^{\top}(\mathbf{u})\mathbb{C}^{-1}(\mathbf{u}) = \begin{bmatrix} c_{1,1}\mathbf{h}_1(\mathbf{u}) & c_{1,2}\mathbf{h}_1(\mathbf{u}) & \ldots & c_{1,N_F}\mathbf{h}_1(\mathbf{u}) \\ c_{2,1}\mathbf{h}_2(\mathbf{u}) & c_{2,2}\mathbf{h}_2(\mathbf{u}) & \ldots & c_{2,N_F}\mathbf{h}_2(\mathbf{u}) \\ \vdots & \vdots & \ddots & \vdots \\ c_{N_F,1}\mathbf{h}_{N_F}(\mathbf{u}) & c_{N_F,2}\mathbf{h}_{N_F}(\mathbf{u}) & \ldots & c_{N_F,N_F}\mathbf{h}_{N_F}(\mathbf{u}) \end{bmatrix},$$

where $c_{i,j}$ are scalar elements of $\mathbb{C}^{-1}(\mathbf{u})$.

That takes care of $\mathbb{V}^{-1}(\mathbf{u})$. Returning to Eq. (4.10), and combining with Eq. (4.11), we have the following analog for the determinant.

$$|\mathbb{V}(\mathbf{u})| = \det(\mathbb{V}_o) \times \det(\mathbb{C}(\mathbf{u})) = \prod_{i=1}^{N_F} \det[\mathbf{V}_i] \times \det(\mathbb{C}(\mathbf{u})) \tag{4.13}$$

The first component, $\prod_{i=1}^{N_F} \det[\mathbf{V}_i(\mathbf{U}_i, \mathbf{U}_i)]$, is composed of $\mathcal{O}(n_i^3)$ computations, constant in $\mathbf{u}$. Only the second component, $\det(\mathbb{C}(\mathbf{u}))$ needs to be updated with new $\mathbf{u}$.

In summary, OSSs can be exploited to circumvent huge matrix computations involved in likelihood evaluation (4.8), yielding a structure benefiting from a degree of precalculation, and from parallelization if desired. These features come on top of largely improved emulation accuracy demonstrated in Section 4.3.2, compared with the global alternative.

4.4.3 Priors and computation

As briefly described in Section 4.3.3, we consider two priors on $\mathbf{u}$, the friction factors in our motivating honeycomb example. The first is independent uniform, $u_j \overset{\text{iid}}{\sim} \text{Unif}(0,1)$. The second is $u_j \overset{\text{iid}}{\sim} \text{Beta}(2,2)$ as a means of regularizing posterior inference. The marginal posterior for $\mathbf{u}$ is known to sometimes concentrate on the boundaries $\mathbf{u}$-space, because of identifiability challenges in the KOH framework see, e.g., Gramacy et al. 2015. Furthermore, we know that `ISOTSEAL` is least stable in that region. $\text{Beta}(2,2)$ slightly discourages that boundary and commensurately elevates the posterior density of central values. This choice has the added benefit of providing better mixing in the MCMC described momentarily.

The coupled GPs involved in the KOH setup are hyperparameterized by scales, lengthscales, and nuggets as in Eq. (4.4). A fully Bayesian analysis would include these in the parameter space for posterior sampling, augmenting the dimension by an order of magnitude in many cases. In other words, the posterior becomes $p(\mathbf{u}, \boldsymbol{\Phi} \mid \mathbf{y}^M, \mathbf{y}^F)$, where $|\Phi| \in \mathcal{O}(p{+}p_x)$, $p = p_x + p_u$ for surrogate and p_x for discrepancy, which would work out to more than thirty parameters in our honeycomb example. Because of that high dimensionality, a common simplifying tactic is to fix those $\boldsymbol{\Phi}$ at their MLE or MAP setting $\hat{\Phi}$, found via numerical optimization. In our OSS setup, with $N_F = 292$ independent surrogates, the burden of hyperparameterization is exacerbated, with $|\Phi| \in \mathcal{O}(N_F p_u + p_x)$ being several orders of magnitude higher in dimension, over one thousand for honeycomb. This all but demands a setup where point estimates are first obtained via maximization, following the scheme outlined in Section 4.3.3. That leaves only $\mathbf{u}$ for posterior sampling via $p(\mathbf{u} \mid \mathbf{y}^M, \mathbf{y}^F, \hat{\boldsymbol{\Phi}})$.

Additionally, we initialize our Monte Carlo search of the posterior with $\hat{\mathbf{u}}$ values found via Section 4.3.3.

Following KOH, we employ MCMC (Hastings 1970; Gelfand and Smith 1990) to sample from the posterior in a Metropolis-within-Gibbs fashion see, e.g., Hoff 2009. Each Gibbs step utilizes a marginal random-walk Gaussian proposal $u'_j = u_j + s_j$, $s_j \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_j^2)$, $j = 1, \ldots, p_u$. A pilot tuning stage was used to tune the σ_j, leading to $\sigma = (0.02, 0.01, 0.2, 0.1)^\top$ in the honeycomb example. Figures 4.11–4.12 in Section 4.5.2 indicate good mixing and adequate posterior exploration of the four-dimensional space of friction factors.

4.5 Empirical results

Before detailing the outcome of this setup on our motivating honeycomb example, we illustrate the methodology in a more controlled setting.

4.5.1 Illustrative example

Consider a mathematical model $y^{M^*}(\cdot)$ with three inputs (x, u_1, u_2), following

$$y^{M^*}(x, u_1, u_2) = \cos\left(\frac{25\sin(x) \times x \times u_1}{x + u_2}\right), \tag{4.14}$$

where $x \in [0,1]$ is a one-dimensional field input and $\mathbf{u} = (u_1, u_2) \in [0,1]^2$ are two-dimensional calibration parameters. Suppose the real process follows

$$y^R(x) = y^{M^*}(x, 0.8, 0.2) + b(x) \quad \text{where} \quad b(x) = \sin(4x).$$

Mimicking the features of `ISOTSEAL`, suppose the computer model y^M is unreliable in its evaluation of the mathematical model y^{M^*}, sometimes returning `NA` values. Specifically, suppose the response is missing when the $\mathbf{u}$ input is in its upper quartile, $u1 \times u2 > 0.5$, and $[5y^{M^*}] \mod 2 \equiv 0$, where $[\cdot]$ rounds to the nearest integer. Figure 4.7 provides an illustration. Each panel in the figure shows the response as a function of (u_1, u_2) for a

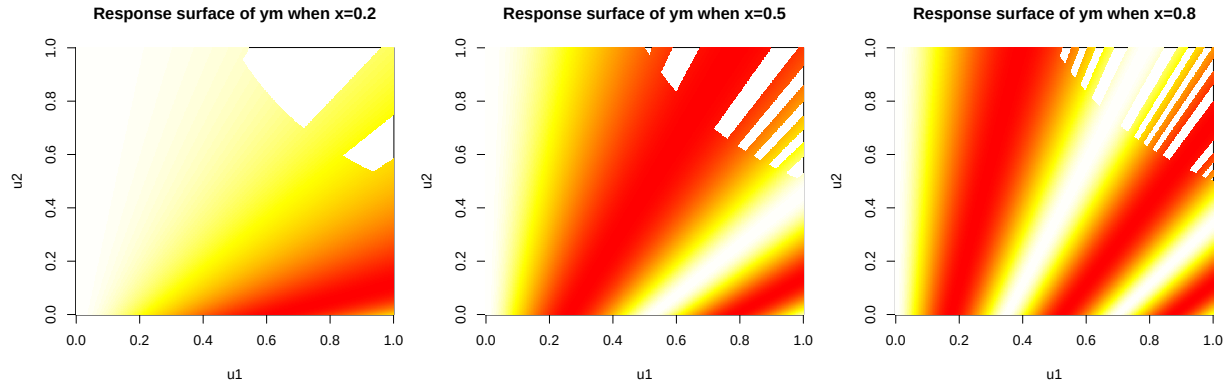
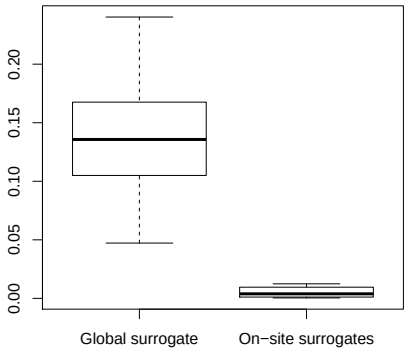


Figure 4.7: Response surfaces of illustrating example. Response surfaces illustrating computer model specified in Eq. (4.14) with missing values under three different settings of $x = .2, \quad x = .5, \quad \text{and} \quad x = .8$.

different setting of x. Observe the nonstationary dynamics manifest in increasing waviness of the surface as x increases. Similarly, the pattern of missingness becomes more complex for increasing x. Therefore, a global surrogate would struggle on two fronts: with stationarity as well as with (nonmissing) coverage of the design in $\mathbf{u}$-space.

Now consider observing N_F field realizations of as $y^R(x)+\varepsilon$, where $\varepsilon \sim \mathcal{N}(0, 0.02^2)$, under a maximin LHS in x-space, and two variations on a computer experiment toward a calibrated model. The first involves a global GP surrogate fit to $N_M = 500$ computer model evaluations via a maximin LHS in $(x, \mathbf{u})$-space, where 33 (6.6%) came back `NA`. The second uses OSSs trained on $n_i = 200$ maximin LHSs in $\mathbf{u}$-space, paired with x_i^F for $i = 1, \ldots, N_F$. Of the $N_M = 2,000$ such simulations, 95 came back missing (4.75%). The sizes of these computer experiment designs were chosen so that the computing demands required for the global and OSS surrogates were commensurate. Counting flops, the global approach requires about $500^3 = 1.25 \times 10^8$, whereas the OSSs need $10 \times 200^3 = 8 \times 10^7$, which can be 10-fold parallelized if desired.

Before turning to calibration, consider first the accuracy of the two surrogates. Mirroring Figure 4.5 for `ISOTSEAL` in our honeycomb example, Figure 4.8 shows the result of an out-of-sample comparison of otherwise identical design. The story here is similar to



	global	OSS
min	0.0472	0.0004
25%	0.1090	0.0012
med	0.1357	0.0039
mean	0.1398	0.0052
75%	0.1673	0.0090
max	0.2403	0.0124

Figure 4.8: Boxplots of 10 out-of-sample on-site RMSEs for illustrative example. Boxplots of 10 out-of-sample RMSEs, where each RMSE is computed by using novel $n_i' \leq 200$, for $i = 1, \ldots, N_F$.

the one for `ISOTSEAL`. Clearly, the OSSs are more accurate. They are better able to capture

the nonstationarity nature of computer model $y^M(\cdot,\cdot)$ nearby to the field sites.

Here, we provide a visualization of the sparsity in the covariance matrix $\mathbb{V}(\mathbf{u})$ in 4.7 for combined data $(\mathbf{y}^M, \mathbf{y}^F)$ in this illustrative example. Five physical data sites with ten on-site simulation data for each are demonstrated in Figure 4.9. Both the block-diagonal $N_M \times N_M$ matrix $\mathbb{V}_o$ representing the on-site simulation and the small dense $N_F \times N_F$ matrix $\mathbb{V}_b$ representing the physical sites are constant to different settings of calibration parameter $\mathbf{u}$. Only the edges $\mathbb{V}_{ob}$, which also has a block-diagonal structure, are updated with different settings of calibration parameter $\mathbf{u}$.

Next, we compare calibration results from global surrogate optimization, OSSs via modularization/optimization [Section 4.3.3, and OSSs via full Bayes [Section 4.4]. In this simple toy example, uniform priors $u_i \overset{\text{iid}}{\sim} \text{Unif}(0,1)$ are sufficient for good performance. The first row of Figure 4.10 shows the distributions of converged $\hat{\mathbf{u}}$ via Eq. (5.6) from the optimization approach described in Section 4.3.3. The left panel corresponds to the lower-fidelity global surrogate and the right panel to the higher-fidelity OSSs. Converged solutions from 500 random initializations are shown. Terrain colors on the ranked log posteriors are provided to aid in visualization. The best single coordinate $\hat{\mathbf{u}}$ is indicated by the black dot. For comparison, the true $\mathbf{u}^*$ value is shown as red-dashed crosshairs. Although the best $\hat{\mathbf{u}}$ values found cluster near the truth, both are sometimes fooled by a posterior ridge in another quadrant of the space. The second row of Figure 4.10 shows the posterior distribution of $\mathbf{u}$ in full (left) and zoomed-in ranges (right). Compared with the OSS-based optimization approach, the KOH analog found $\mathbf{u}$'s tightly coupled around the truth.

In this simple example, posterior uncertainty is low, in part because a relatively large computer experiment could be entertained in a small input dimension. In fact all three methods worked reasonably well. However, as we entertain more realistic settings, such as the honeycomb in 17 dimensions, only the methods based on OSSs are viable computationally (assuming a relatively dense sampling of the computer model is viable).

4.5.2 KOH versus modularized optimization: on honeycomb

Here we return to our motivating honeycomb seal example, first providing a qualitative comparison between our two approaches based on OSSs, via modularized optimization [Section 4.3.3] and KOH [Section 4.4]. We then turn to an out-of-sample comparison, pitting the KOH framework against the initial NLS analysis. Throughout, we use a regularizing independent Beta$(2, 2)$ prior on the components of $\mathbf{u}$. Appendix 4.5.2 provides an analog presentation under a uniform prior, accompanied by a brief discussion.

Figure 4.11 shows traces of the samples obtained via our Metropolis-within-Gibbs scheme, described in Section 4.4.3. The figure indicates clear convergence to the stationary distribution with mixing that is qualitatively quite good. The effective sample sizes (ESS, Kass et al. 1998), marginally for all four friction factors, are sufficiently high at $\text{ESS}_{u_1} = 1026$, $\text{ESS}_{u_2} = 684$, $\text{ESS}_{u_3} = 2062$, $\text{ESS}_{u_4} = 1462$, respectively.

Figure 4.11 clearly shows that the posterior is, at least marginally, far more concentrated for the first two friction factors (first row) than for the last two. For a better joint glimpse at the four-dimensional posterior distribution of $\mathbf{u}$, the bottom-left panels

of Figure 4.12 show these samples via pairs of coordinates. The points are colored by a rank-transformed log-scaled posterior evaluation as a means of better visualizing the high concentrations in a cramped space. Histograms along the diagonal panels show individual margins; panels on the top-right mirror those on the bottom-left but instead show solutions found by the modular/optimal approach [Section 4.3.3] in 500 random restarts.

Several notable observations can be drawn from the plots in that figure. For one, consistency is high between the two approaches: KOH and modular/opt. Although the values of log posteriors evaluations are not directly comparable across the models, both agree on most probable values (black dots in the off-diagonal panels). A diversity of solutions from the optimization-based approach indicates that the solver struggles to navigate the log posterior surface but usually finds estimates that are in the right ballpark. The full posterior distribution via KOH indicates that the first two friction factors are well pinned-down by the posterior. However, posterior concentration is more diffuse for the latter two. A complicated correlation structure is evident in (u_3, u_4).

A similar suite of results under an independent uniform prior is also provided in Section 4.5.2. The story there is more or less similar, except that the posterior sampling concentrates more heavily on the boundary of $\mathbf{u}$-space for all four parameters. Considering that we know our `ISOTSEAL` simulator is less reliable in those regimes, leading to far more missing values and thus requiring greater degree of extrapolation from our OSSs, we prefer the more stable regime (better emulation and MCMC mixing) offered by light penalization under a Beta$(2, 2)$ prior.

Calibration under uniform prior

For completeness, we provide calibration results under a uniform prior in Figure 4.13, complementing those from Figure 4.12 under Beta$(2, 2)$. Compared with those results, the ones shown here more heavily concentrate on the boundaries of the study region. Also, somewhat more inconsistency exists between the modular/opt results and the fully Bayesian analog. The regularization effect of the Beta$(2, 2)$ leads to better numerics.

4.5.3 Calibration without discrepancy correction

To investigate the role of model discrepancy and to generate a baseline for later prediction comparison in this honeycomb probelm, we provide three versions of calibration results with no discrepancy correction under the Kennedy and O'Hagan calibration framework: the non-linear least square solutions; least-square calibration without discrepancy correction; and optimization calibration without discrepancy correction.

Nonlinear least-square calibration

First version in comparison is a global search for $\mathbf{u}$ using the nonlinear least-square calibration described in Section 4.2.2. Initializing from 100 random maximinLHS design on $\mathbf{u}$ values, we search for improved NLS solutions. Since this NLS setup does not model a discrepancy between $\mathbf{y}^M$ and $\mathbf{y}^F$, converged solutions still contain substantial discrepancy indicated by quadratic loss. Among 100 restarts, two failed; and the other losses, mapped to the scale of $\mathbf{y}^F$ by taking the square root, had the following distribution.

min	25%	med	mean	75%	max
6.605	8.161	8.401	10.117	9.099	25.787

Figure 4.14 demonstrates the pair-wise distribution of these 98 converged solutions of $\hat{\mathbf{u}}^{\text{NLS}}$. Notice there is a slight vague pattern in the distribution of $\hat{\mathbf{u}}^{\text{NLS}}$ and the so-far best NLS solution (marked with "+") has three out of four parameter values at the boundary of parameter space. The pattern in figure 4.14 also indicates there is a complex interplay between the calibration parameter and model discrepancy in the honeycomb problem. The NLS global search results with the minimal RMSE = 6.605 suggests that for this honeycomb problem, a model discrepancy is not negligible and always exists substantially, no matter how we tune the calibration parameter values from the entire parameter space. Another speculation of this noisy and unclear pattern in figure 4.14 can be related with the nonstationarity and local numerical instabilities behaviors of the `ISOTSEAL` simulator. As the NLS approach employs the temperamental `ISOTSEAL` simulator directly, we can also employ the on-site surrogates with more stable and complete emulation to further confirm what we found out about the model discrepancy in this honeycomb problem.

OSS least-square no-bias calibration

We can naturally extend the NLS calibration to OSS least-square no-bias calibration by replacing the `ISOTSEAL` simulator with the on-site surrogates. Without a discrepancy term, the KOH framework becomes

$$y^F(\mathbf{x}) = \hat{y}^M(\mathbf{x}, \hat{\mathbf{u}}^{\text{OSS}}_{\text{LS-nobias}}) + \epsilon. \tag{4.15}$$

and the objective function becomes

$$\hat{\mathbf{u}}^{\text{OSS}}_{\text{LS-nobias}} = \arg\min_{\mathbf{u}} \left\{ \frac{1}{N_F} \sum_{i=1}^{N_F} \left[y_i^F(\mathbf{x}_i) - \hat{y}_i^M(\mathbf{x}_i, \mathbf{u}) \right]^2 \right\}, \tag{4.16}$$

We implemented a global OSS least-square no-bias search from 500 space-filling initializations on $\mathbf{u}$. The following table summaries the RMSEs of these 500 converged solutions. Figure 4.15 shows the pair-wise distribution of these 500 $\hat{\mathbf{u}}^{\text{OSS}}_{\text{LS-nobias}}$.

min	25%	med	mean	75%	max
6.773	7.764	7.792	7.677	8.087	10.016

Compared to the NLS solutions in figure 4.14 using `ISOTSEAL` simulator directly, the distribution of $\hat{\mathbf{u}}^{\text{OSS}}_{\text{LS-nobias}}$ has a different yet much clearer pattern. The best fitted values are concentrated around the green area, with the minimal RMSE = 6.773. With reliable and accurate on-site emulation through OSS, as illustrated in Section 4.3, these OSS least-square no-bias results further confirm that a substantial model discrepancy (with minimal RMSE = 6.773) always exists for the honeycomb problem, no matter how we tune the calibration parameters.

OSS optimization no-bias calibration

Another variation of KOH calibration using OSS without discrepancy correction is a likelihood-based optimization approach similar to the one described in Section 4.3.3. The

model without bias correction becomes

$$y^F(\mathbf{x}) = \hat{y}^M(\mathbf{x}, \hat{\mathbf{u}}^{\text{OSS}}_{\text{nobias}}) + \epsilon. \tag{4.17}$$

The only difference from the previous method is the objective function becomes a likelihood/posterior on the Gaussian noise, instead of the sum of squared errors. To make the analysis consistent with other results from discrepancy correction, a Beta$(2, 2)$ prior on the parameter $\mathbf{u}^{\text{OSS}}_{\text{nobias}}$ is used here. The optimization solution of the posterior distribution of $\mathbf{u}^{\text{OSS}}_{\text{nobias}}$ is shown in figure 4.16. The results in figure 4.16 demonstrate a very similar pattern as in figure 4.15, since for Gaussian noise the least square estimates are equivalent to the maximum of likelihood estimates. The slight different between figure 4.16 and figure 4.15 can be attributed to the impact of regularizing Beta$(2, 2)$ prior on the parameter $\mathbf{u}^{\text{OSS}}_{\text{nobias}}$, making the maximum of posterior optimization slightly closer to the center of the parameter space. Another observation is that, although both methods lead to consistent distribution pattern of the parameter, this likelihood based method has more local optima than the results from the least-square method.

In this section, we investigate several variations of KOH calibration with no discrepancy correction, using surrogate or direct simulation. From all of these three no bias exercise in this section, it is clear that for this honeycomb problem, a substantial model discrepancy exists and bias correction is indeed necessary. After discussing all the other alternatives with discrepancy correction in the coming seciton, we further summarize all the fitted calibration parameters and corresponding RMSEs in table 4.1.

4.6 Out-of-sample prediction through OSS

In this section, we provide both methodological illustrations and empirical results for out-of-sample prediction on honeycomb problem using on-site surrogates, including several variations of point-wise prediction and ultimately extend to fully Bayesian. We firstly start with several piece-wise leave-one-out prediction exercise using OSS for both hypotheses with and without discrepancy correction. Next, we derivative the tractable expressions of fully Bayesian prediction using on-site surrogates for out-of-sample prediction. We summarize all empirical in/out-of-sample prediction results in table 4.1.

4.6.1 Pointwise leave-one-out prediction comparison

To close the loop on our NLS comparison from Section 4.2.2, particularly Figure 4.4 highlighting in-sample prediction, we conclude our empirical work on the honeycomb with an exercise measuring out-of-sample predictive accuracy. Pointwise comparators based on several variations are entertained, e.g., with/without OSSs, with/without estimated discrepancies compensating for bias. Finally, we complete the Bayesian KOH OSS setup (Section 4.4) with a predictor that tractably propagates uncertainty through the sparse covariance structure.

The NLS baseline from Section 4.2.2 involves direct application of `ISOTSEAL` for new physical (testing) site $\mathbf{x}_{\text{new}}$, paired with plug-in $\hat{\mathbf{u}}$ furnished by our BHGE colleagues:

$$\hat{y}^F(\mathbf{x}_{\text{new}}) = y^M(\mathbf{x}_{\text{new}}, \hat{\mathbf{u}}^{\text{NLS}}) \tag{4.18}$$

No bias correction is applied. Figure 4.17, augmenting Figure 4.4, provides a view into residuals under this comparator, and others explained momentarily. We clarify that these NLS results are "in-sample" as they use the same data our BHGE colleagues trained on. Precise root mean-squared errors (RMSEs) and $\hat{\mathbf{u}}$-values are summarized in Table 4.1.

Feeding $\hat{\mathbf{u}}$ directly into `ISOTSEAL` is problematic because simulation dynamics are nonstationary, unstable, and unreliable. We had trouble getting an implementation of this variation to behave reliably enough in order to report meaningful out-of-sample results. As demonstrated in Section 4.2.1, and the second row in Figure 4.6, `ISOTSEAL` can fail to converge and instead return $y^M(\mathbf{x}_{\text{new}}, \hat{\mathbf{u}}) = \texttt{NA}$ especially at $\mathbf{u}$ around the upper limit of their range(s). Our BHGE colleagues carefully engineered their NLS search to avoid problematic $\mathbf{u}$-settings. OSSs were proposed in order to more gracefully cope with `NA`s and to correct for other idiosyncrasies. When predicting out of sample, a new OSS $\hat{y}^M(\mathbf{x}_{\text{new}}, \cdot)$ must be fit via new on-site design $\mathbf{U}_{\text{new}}$ paired with $\mathbf{x}_{\text{new}}$. As with OSS training described in Section 4.3.1, we shall utilize a size $n = 1000$ maximin LHS.

Surrogate $\hat{y}^M(.,.)$ enables a full search of the entire $\mathbf{u}$-space, offering the potential of finding a better $\hat{\mathbf{u}}$ especially nearby regions where direct `ISOTSEAL` runs may fail. Acting on OSSs without discrepancy correction, we find $\hat{\mathbf{u}}^{\text{OSS}}_{\text{nobias}}$ slightly different from the $\hat{\mathbf{u}}^{\text{NLS}}$ using direct `ISOTSEAL` runs. See Table 4.1. To compare the predictive performance directly to the in-sample NLS, we plug-in $\hat{\mathbf{u}}^{\text{OSS}}_{\text{nobias}}$ for new site $\mathbf{x}_{\text{new}}$ though the new OSS:

$$\hat{y}^F(\mathbf{x}_{\text{new}}) = \hat{y}^M(\mathbf{x}_{\text{new}}, \hat{\mathbf{u}}^{\text{OSS}}_{\text{nobias}}). \tag{4.19}$$

Figure 4.17 indicates similar residual behavior for these two comparators. OSSs without bias correction fares slightly worse than the NLS analog, however note that the latter is truly out-of-sample and the former was technically in-sample. The OSS version $\hat{y}^M(\mathbf{x}_{\text{new}}, \hat{\mathbf{u}}^{\text{OSS}}_{\text{nobias}})$ offers fuller uncertainty quantification in predictions, via local GP predictive variances.

Now consider variations which correct for potential bias between OSS and field data measurements. Feed $\hat{\mathbf{u}}$ through the OSS and obtain

$$\hat{y}^F(\mathbf{x}_{\text{new}}) = \hat{y}^M(\mathbf{x}_{\text{new}}, \hat{\mathbf{u}}) + \hat{b}(\mathbf{x}_{\text{new}}) \tag{4.20}$$

To benchmark these predictions out of sample we designed the following leave-one-out (LOO) cross-validation (CV) experiment. Alternately excluding each field data location $i = 1, \ldots, N_F = 292$, we fit 292 LOO discrepancy terms $\hat{b}^{(-i)}(\cdot)$ via residuals $\mathbf{y}^F_{(-i)} - \hat{\mathbf{y}}^M_{(-i)}$ and $\mathbf{X}^F_{(-i)}$. We could build a new OSS for $\mathbf{x}_i$, treating it as a $\mathbf{x}_{\text{new}}$ as described above, but instead it is equivalent (and computationally more thrifty) to use the $\mathbf{U}_i$ we already have. Based on those calculations, point predictions are composed of

$$\hat{y}^F(\mathbf{x}_i) = \hat{y}^M(\mathbf{x}_i, \hat{\mathbf{u}}) + \hat{b}^{(-i)}(\mathbf{x}_i), \quad \text{for} \quad i = 1, 2, \ldots, N_F. \tag{4.21}$$

Predictions thus obtained are compared with true outputs $\mathbf{y}^F$ and residuals for RMSE calculations. We note that this experiment focuses primarily on bias correction. New $\hat{\mathbf{u}}_{(-i)}$ are not calculated for each of $i = 1, \ldots, N_F$ due to the prohibitive computational cost.

Figure 4.17 shows those LOO residuals graphically alongside our other compara-

tors. Only results for $\hat{\mathbf{u}}$ via modular/opt framework are shown here since $\hat{\mathbf{u}}$ from the fully Bayes KOH setup are similar. The panels in the figure indicate that bias correction offers substantial improvement over NLS: in-sample NLS residuals are worse than LOO OSS Bayes results. Summarizing those residuals, modular/opt calibration with discrepancy has an leave-one-out RMSE of 2.126, being even smaller than both the in-sample NLS value of 6.605 reported in Section 4.2.2 and the in-sample OSS no-bias value of 6.818. Furthermore, LOO OSS modular/opt RMSE is comparable to its in-sample analog of 1.125.

Method	$\hat{u}_1$	$\hat{u}_2$	$\hat{u}_3$	$\hat{u}_4$	RMSE
In-sample NLS	0.00000	0.00000	0.82123	0.99615	6.605
OSS LS No Bias	0.00000	0.20536	1.00000	0.93874	6.773
OSS No Bias	0.00877	0.17352	0.94893	0.94474	6.818
In-sample OSS Bayes	0.93659	0.98348	0.28441	0.25975	1.125
LOO OSS modular/opt	0.93659	0.98348	0.28441	0.25975	2.126
LOO OSS KOH full Bayes	0.93659	0.98348	0.28441	0.25975	1.957

Table 4.1: Estimated $\hat{\mathbf{u}}$ and RMSEs from in-sample and LOO comparisons.

4.6.2 Fully Bayesian prediction through OSS

Next we develop fully Bayesian prediction for $y^F(\mathbf{X}^F_{\text{new}})$ at N'_F new physical locations $\mathbf{X}^F_{\text{new}} = (\mathbf{x}^{\text{new}}_1, \mathbf{x}^{\text{new}}_2, \ldots, \mathbf{x}^{\text{new}}_{N'_F})^\top$. As in the pointwise case, N'_F new OSSs must be built

on N'_M new on-site simulations $\mathbf{y}^M_{\text{new}} = (\mathbf{y}_{N_F+1}, \ldots, \mathbf{y}_{N_F+N'_F})^\top$. Following from Eq. (4.8),

$$\begin{bmatrix} \mathbf{y}^M \\ \mathbf{y}^F \\ \mathbf{y}^M_{\text{new}} \\ \mathbf{y}^F_{\text{new}} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_{N_F} \\ \mathbf{y}^F \\ \mathbf{y}_{N_F+1} \\ \vdots \\ \mathbf{y}_{N_F+N'_F} \\ \mathbf{y}^F_{\text{new}} \end{bmatrix} = \begin{bmatrix} y_1(\mathbf{U}_1) \\ \vdots \\ y_{N_F}(\mathbf{U}_{N_F}) \\ y^M(\mathbf{U}) + b(\mathbf{X}^F) \\ y_{N_F+1}(\mathbf{U}_{N_F+1}) \\ \vdots \\ y_{N_F+N'_F}(\mathbf{U}_{N_F+N'_F}) \\ y^M_{\text{new}}(\mathbf{U}_{N'_F}) + b(\mathbf{X}^F_{\text{new}}) \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbb{V}^P(\mathbf{u})) \tag{4.22}$$

where $\mathbf{U}_{N'_F} = [\mathbf{u}^\top; \cdots; \mathbf{u}^\top]^\top$ stacks N'_F identical p_u-dimensional row vectors $\mathbf{u}^\top$. The $(N_M + N_F + N'_M + N'_F) \times (N_M + N_F + N'_M + N'_F)$ covariance matrix $\mathbb{V}^P(\mathbf{u})$, combining OSS training data and out-of-sample data elements, may be built as follows

$$\mathbb{V}^P(\mathbf{u}) = \begin{bmatrix} \mathbb{V}_o & \mathbb{V}^\top_{ob}(\mathbf{u}) & \mathbf{0} & \mathbf{0} \\ \mathbb{V}_{ob}(\mathbf{u}) & \mathbb{V}_b & \mathbf{0} & \mathbb{V}^\top_b(\mathbf{X}^F_{\text{new}}, \mathbf{X}^F) \\ \mathbf{0} & \mathbf{0} & \mathbb{V}^{\text{new}}_o & \mathbb{V}^{\text{new}}_{ob}(\mathbf{u})^\top \\ \mathbf{0} & \mathbb{V}_b(\mathbf{X}^F_{\text{new}}, \mathbf{X}^F) & \mathbb{V}^{\text{new}}_{ob}(\mathbf{u}) & \mathbb{V}^{\text{new}}_b \end{bmatrix}, \tag{4.23}$$

borrowing notation for $\mathbb{V}(\mathbf{u})$ from Eq. (4.7).

Like $\mathbb{V}(\mathbf{u})$, $\mathbb{V}^P(\mathbf{u})$ emits sparse block-wise structure due to the OSSs. Extending from Eq. (4.7), we have $\mathbb{V}^{\text{new}}_o = \text{Diag}[\mathbf{V}_i(\mathbf{U}_i, \mathbf{U}_i)]$, for $i = N_F + 1, \ldots, N_F + N'_F$, an upper-left

block diagonal submatrix. Similarly $\mathbb{V}_b^{\text{new}} = \mathbf{v}_{\text{new}}\mathbb{I}_{N'_F} + V_b(\mathbf{X}^F_{\text{new}})$, where $\mathbf{v}_{\text{new}}\mathbb{I}_{N'_F}$ is a diagonal of nugget effects from the new OSSs, and $V_b(\mathbf{X}^F_{\text{new}})$ is the covariance matrix on $\mathbf{X}^F_{\text{new}}$ from the bias correction. $\mathbb{V}_{ob}^{\text{new}}(\mathbf{u})$ and $\mathbb{V}_{ob}^{\text{new}}(\mathbf{u})^\top$ are similar to $\mathbb{V}_{ob}(\mathbf{u})$ and $\mathbb{V}_{ob}^\top(\mathbf{u})$, composed of $V_i(\mathbf{U}_{N'_F})$ with $i = N_F + 1, \ldots, N_F + N'_F$. Each $V_i(\mathbf{U}_{N'_F})$ is sparse with single row of non-zero entries. In Eq. (4.23), the new OSS on $\mathbf{X}^F_{\text{new}}$ is sparse between training data $(\mathbf{y}^M, \mathbf{y}^F)$ and new data $(\mathbf{y}^M_{\text{new}}, \mathbf{y}^F_{\text{new}})$, involving only the small $N'_F \times N_F$ bias covariance $\mathbb{V}_b(\mathbf{X}^F_{\text{new}}, \mathbf{X}^F)$.

Using those definitions, the predictive distribution of $\mathbf{y}^F_{\text{new}}$ conditioning on both data sources, $(\mathbf{y}^M, \mathbf{y}^M_{\text{new}})$ from simulation and $\mathbf{y}^F$ from physical experiments, hyperparameters $\mathbf{\Phi}$ and calibration parameter $\mathbf{u}$, is MVN with mean $\mathbf{m}_{\text{new}}$ and covariance $\mathbf{V}_{\text{new}}$ following

$$\mathbf{m}_{\text{new}} = \mathbb{V}_b(\mathbf{X}^F_{\text{new}}, \mathbf{X}^F)\mathbb{C}^{-1}(\mathbf{u})[\mathbf{y}^F - \mathbb{V}_{ob}(\mathbf{u})\mathbb{V}_o^{-1}\mathbf{y}^M] + \mathbb{V}_{ob}^{\text{new}}(\mathbf{u})(\mathbb{V}_o^{\text{new}})^{-1}\mathbf{y}^M_{\text{new}} \tag{4.24}$$

$$\mathbf{V}_{\text{new}} = \mathbb{V}_b^{\text{new}} - \mathbb{V}_b(\mathbf{X}^F_{\text{new}}, \mathbf{X}^F)\mathbb{C}^{-1}(\mathbf{u})\mathbb{V}_b(\mathbf{X}^F_{\text{new}}, \mathbf{X}^F)^\top - \mathbb{V}_{ob}^{\text{new}}(\mathbf{u})(\mathbb{V}_o^{\text{new}})^{-1}\mathbb{V}_{ob}^{\text{new}}(\mathbf{u})^\top. \tag{4.25}$$

Fully Bayesian uncertainty quantification using Eqs. (4.24–4.25) is tractable. Sparse-matrix decompositions can be applied in a manner similar to likelihood evaluation Section 4.4.2.

Consider deploying these equations in our out-of-sample setup, re-using the new OSSs trained for the pointwise comparisons. Following a similar LOO setup, we derive $(\mathbf{y}_i^F \mid \mathbf{y}_{-i}^M, \mathbf{y}_{-i}^F, \mathbf{y}_i^M, \mathbf{\Phi}, \mathbf{u}^{(t)}) \sim \mathcal{N}_i(\mathbf{m}_i, \mathbf{V}_i)$ via Eqs. (4.24–4.25) integrating over $\mathbf{u}$ by aggregating over Monte Carlo samples for $\{\mathbf{u}^{(t)}\}_{t=1}^T$ shown in bottom-left panels of Figure 4.12. When aggregating covariances, covariances of sample means are incorporated respecting the law of total variance. Figure 4.18 shows this fully Bayesian predicted mean with 95% credible interval over each observed $\mathbf{y}^F$. In contrast to to the previous leave-one-out experiments

described in Eq. 4.21, which involved 292 LOO discrepancy terms $\hat{b}^{(-i)}(\cdot)$ via residuals $\mathbf{y}^F_{(-i)} - \hat{\mathbf{y}}^M_{(-i)}$ and $\mathbf{X}^F_{(-i)}$, results in Figure 4.18 provide full out-of-sample posterior predictive uncertainty for both the simulation and the discrepancy correction.

4.7 Discussion

Motivated by a computer model calibration problem the design of a seal used in turbines, we developed a thrifty new method to address several challenging features. Those challenges include a high-dimensional input space, local instability in computer model simulations, nonstationary simulator dynamics, and modeling for large computer experiments. Taken alone, each of these challenges has solutions that are, at least in some cases, well established in the literature. Taken together, a more deliberate and custom development was warranted. To meet those challenges, we developed the method of on-site surrogates. The construction of OSSs is motivated by the unique structure of the posterior distribution under study in the canonical Kennedy and O'Hagan calibration framework, where predictions are needed only at a limited number of field data sites, no matter how big the computer experiment is. This unique structure allowed us to map a single, potentially high-dimensional problem, into a multitude of low-dimensional ones where computation can be performed in parallel. Two OSS-based calibration settings were entertained, one based on simple bias-corrected maximization and the other akin to the original KOH framework. Both were shown to empirically outperform simpler, yet high-powered, alternatives.

Despite its many attractive features, there is clearly much potential to refine this

approach, in particular the design and modeling behind the OSSs. While simple Latin hypercube samples and GPs with exponential kernels and nuggets work well, several simple extensions could be quite powerful. The need for such extensions, along at least one avenue, is perhaps revealed by the final row of Figure 4.6. Those plots show bifurcating `ISOTSEAL` runs due to numerical instabilities. Although inflated nuggets enable smoothing over those regimes, the result is uniformly high uncertainty for all inputs rather than just near the trouble spot. The reason is that the GP formulation being used is still (locally) stationary. Specifically, the error structure is homoskedastic. Using a heteroskedastic GP instead (Binois, Gramacy, and Ludkovski 2018), say via `hetGP` on CRAN (Binois and Gramacy 2018), could offer a potential remedy. In a follow-in paper Binois et al. 2019 showed how designs for effective `hetGP` modeling could be built up sequentially, balancing an appropriate amount of exploration and replication in order to effectively learn signal-to-noise relationships in the data. Such an approach could represent an attractive alternative to simple LHSs in $\mathbf{u}$-space.

Here we only entertained a single output k_{dir}, at a single frequency, among a multitude of others and at other frequencies. In future work we plan to investigate a multiple output approach to calibration. Much work remains to assess the potential for such an approach, say via simple *co-kriging* (Ver Hoef and Barry 1998) or a linear model of coregionalization (e.g., Wackernagel 1998). Our BHGE collaborators' pilot study also indicated that there could potentially be input-dependent variations in the best setting of the friction factors. That is, we could be looking at a $\hat{\mathbf{u}}(\mathbf{x})$, perhaps for a subset of the coordinates of the 13-dimensional $\mathbf{x}$ input. Whether a simple partition-based or linear scheme might

be appropriate, or if something more nonparametric like Brown and Atamturktur 2018 is required, remains an open question.

We'd like to close with a thought on confounding and identifiability, an ever-present concern in the KOH setting. OSSs are no help here, essentially chopping up the design space, limiting information sharing and reducing the (Bayesian) learning that could transpire about calibration parameters compared to the usual (global) setup. Although we have seen no evidence of concern, it is possible that OSSs would exacerbate the problem. However, we note that the underlying framework – linking a latent $\mathbf{u}$-variable to a nonparametric discrepancy – is identical whether or not OSSs are deployed. Accordingly, simplifications (Tuo and Wu 2015) or extensions (Plumlee 2017) are similarly viable as a means of limiting sources of confounding that challenges identifiability.

There are many reasons to calibrate, with KOH or otherwise. One is simply predictive; another is to get a sense of how the apparatus could be tuned, or to quantify how much information is in the data (and prior) about promising $\mathbf{u}$ settings. Both are very doable, and worth doing, even in the face of confounding. Our posterior summaries for $\mathbf{u}$ are a testament in this regard. In our toy example, which has many features in common with the motivating honeycomb seal, the posterior is quite peaked. Does this mean our $\hat{\mathbf{u}}$ or Bayesian samples $\mathbf{u}^{(t)}$ have identified the right $\mathbf{u}^*$? Possibly not in general, except that we know the truth in this case and identification can be confirmed. Our posterior for $\mathbf{u}$ in the honeycomb example shows sharp concentration for some inputs, less for others, and interpretable correlation in one pair (u_3, u_4). Our colleagues at BHGE were not surprised by these results, and found

them to be helpful in designing new field experiments. Although we cannot be confident about identification in this example, KOH has been a useful exercise.

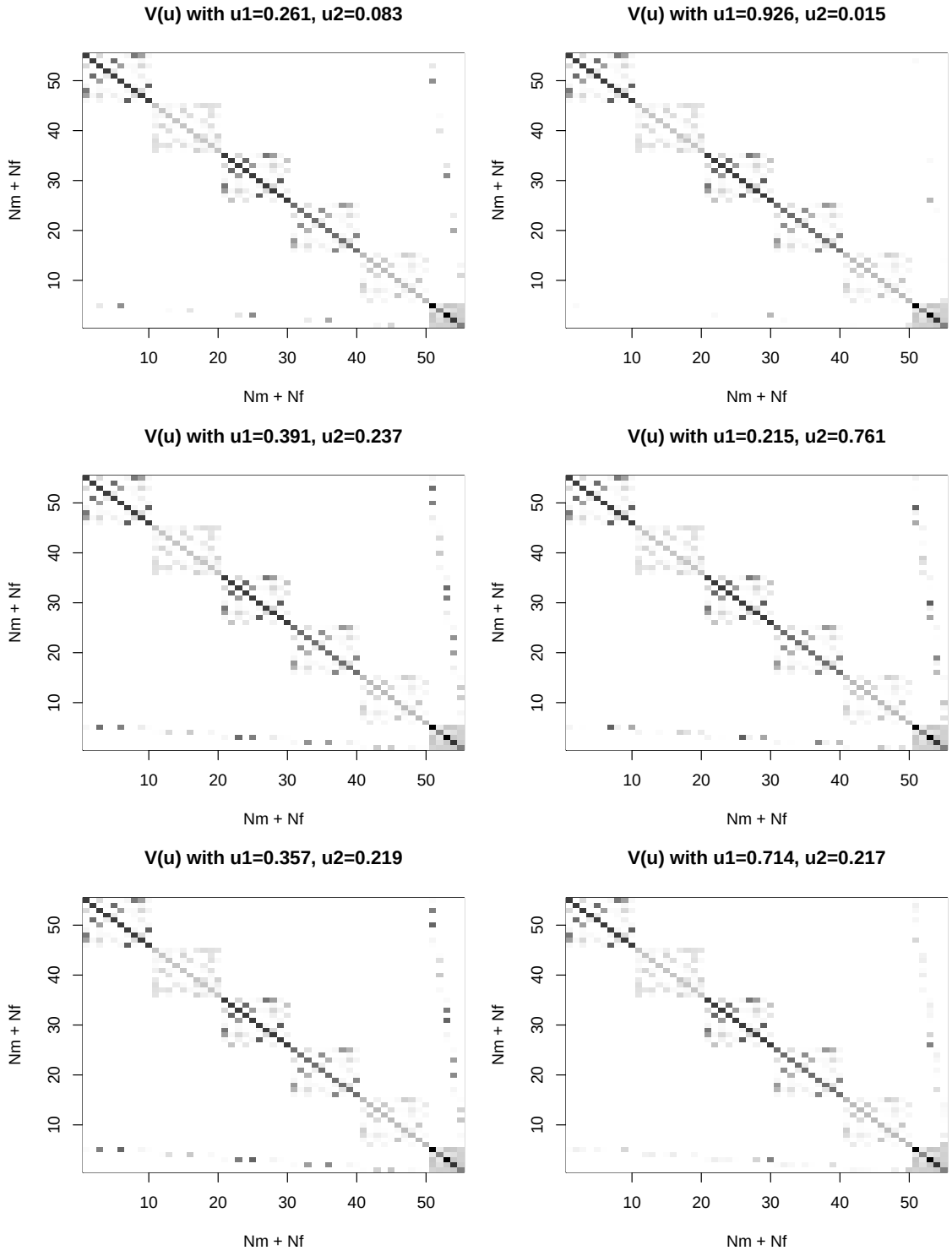


Figure 4.9: Visualization of sparse $\mathbb{V}(\mathbf{u})$ in illustrative example. In total $N_F = 5$ physical sites and 10 on-site simulations for each. Different $\mathbf{u}$ settings are implemented to illustrate the impact of calibration parameter on the full sparse covariance matrix.

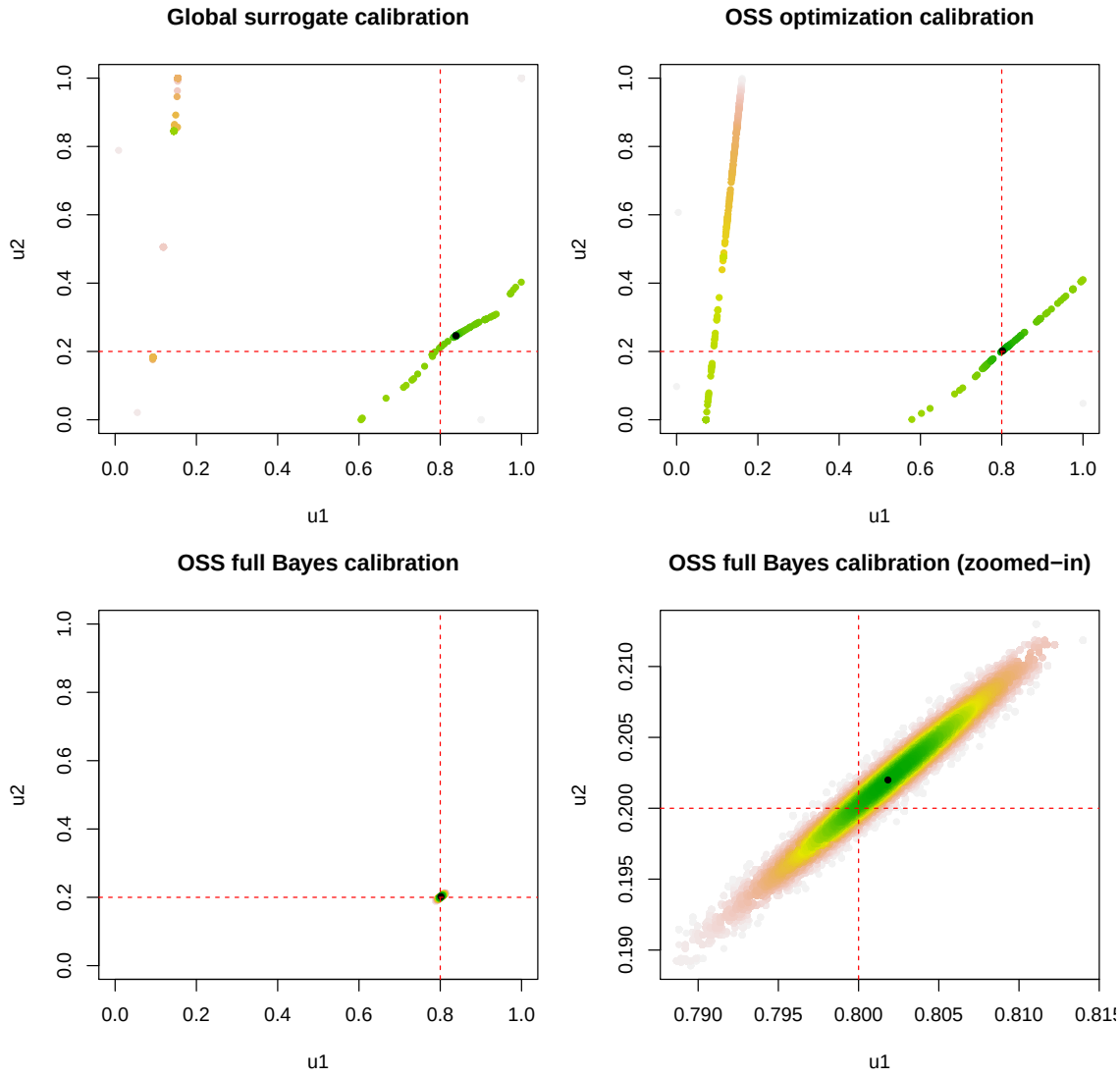


Figure 4.10: Calibration results for illustrative example. Calibration results from both optimization and full Bayes for the illustrative example. Terrain colors are derived from ranks of log-scaled posteriors as a visual aid; black dots indicate the MAP setting; red dashed lines are the true values of calibration parameters, $\hat{\mathbf{u}}$.

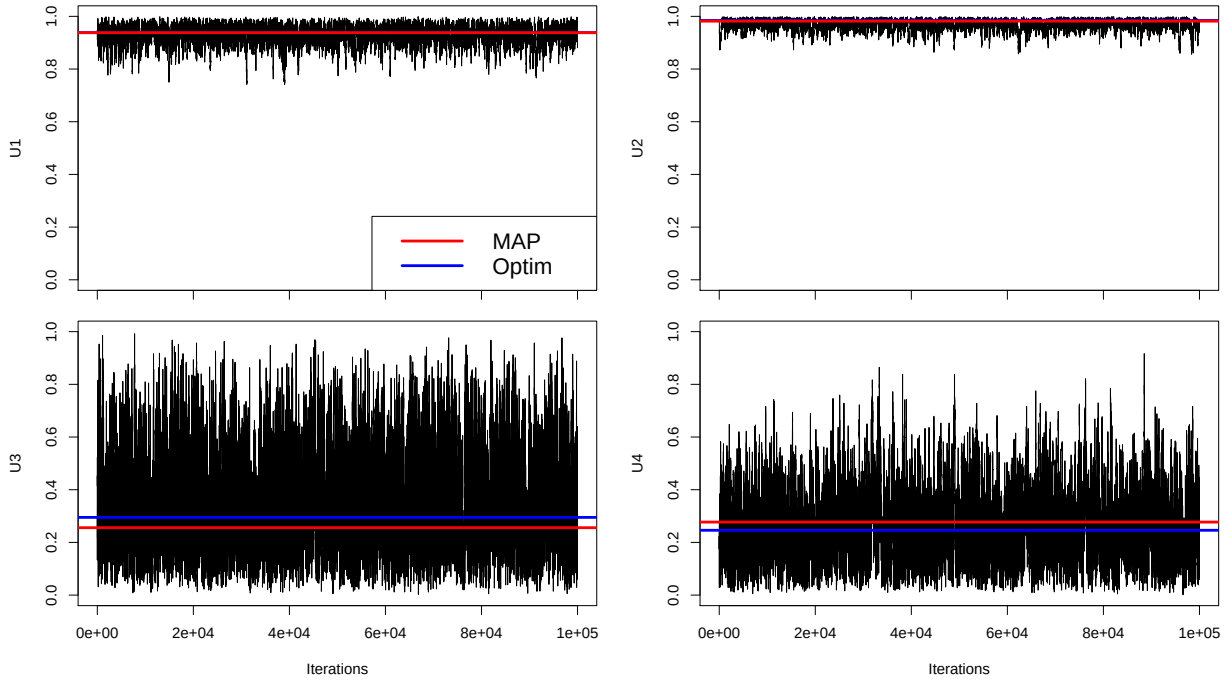


Figure 4.11: Trace plots of MCMC samples of calibration parameters **u** for honeycomb. Trace plots of MCMC samples for all calibration parameters **u** after burn-in. Blue line indicates the best setting for **u** from optimization approach. Red line indicates the maximum of a posterior (MAP) for **u** extracted from the samples, with the modular/opt shown for comparison.

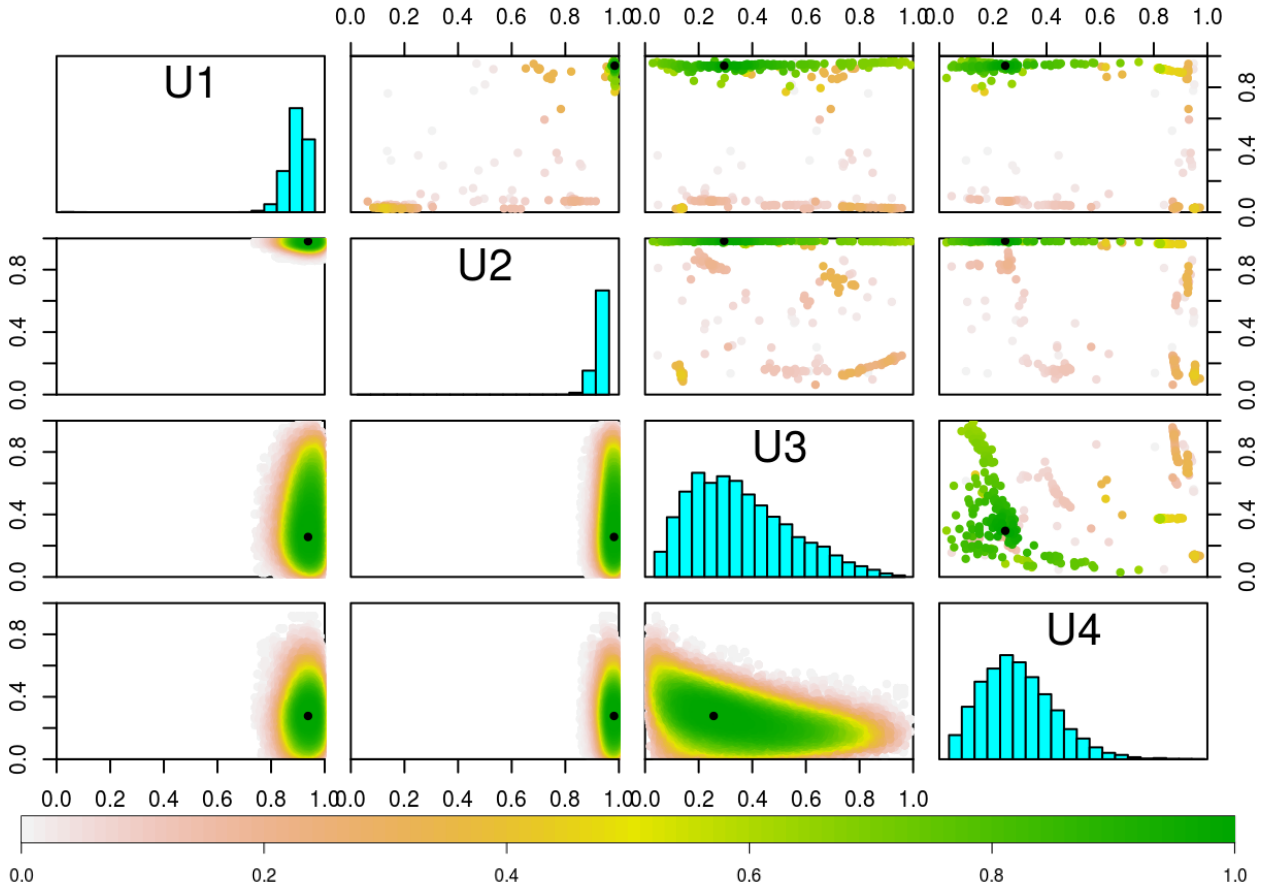


Figure 4.12: Calibration results for honeycomb with Beta prior on K_{direct}. Bayesian KOH (lower and diagonal) posterior for $\mathbf{u}$ vs. modularlized optimization (upper) analog, both under an independent Beta$(2, 2)$ prior for $\mathbf{u}$. Colors are derived from ranks of posterior probabilities to aid in visualization. Modularized results are from 500 converged optimization under random initialization. Black dots indicate MAP values.

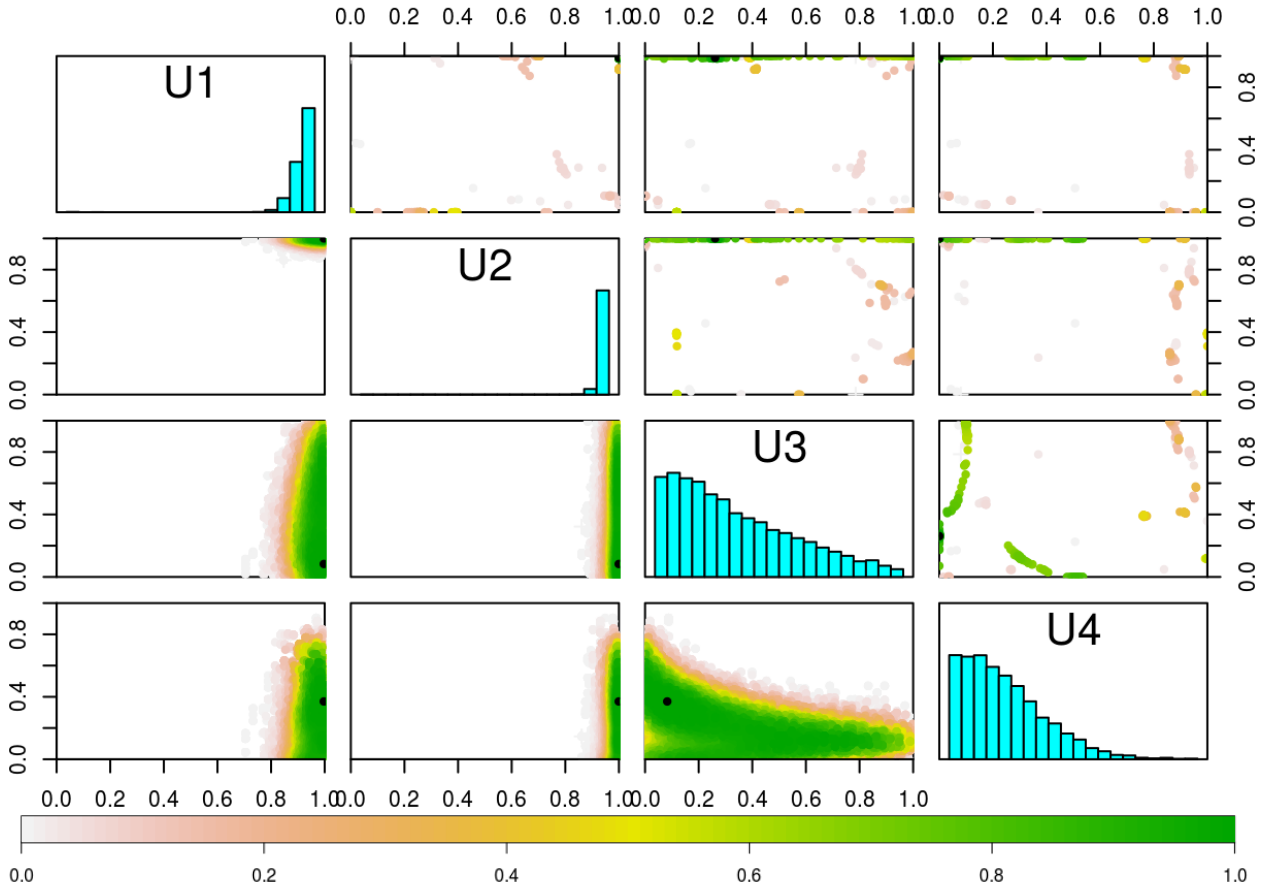


Figure 4.13: Calibration results for honeycomb with Uniform prior on K_{direct}. Bayesian KOH (lower and diagonal) posterior for $\mathbf{u}$ vs. modularized optimization (upper) analog, both under an independent uniform prior for $\mathbf{u}$. Colors are derived from ranks of posterior probabilities to aid in visualization. Modularized results are from 500 converged optimization under random initialization. Black dots indicate MAP values.

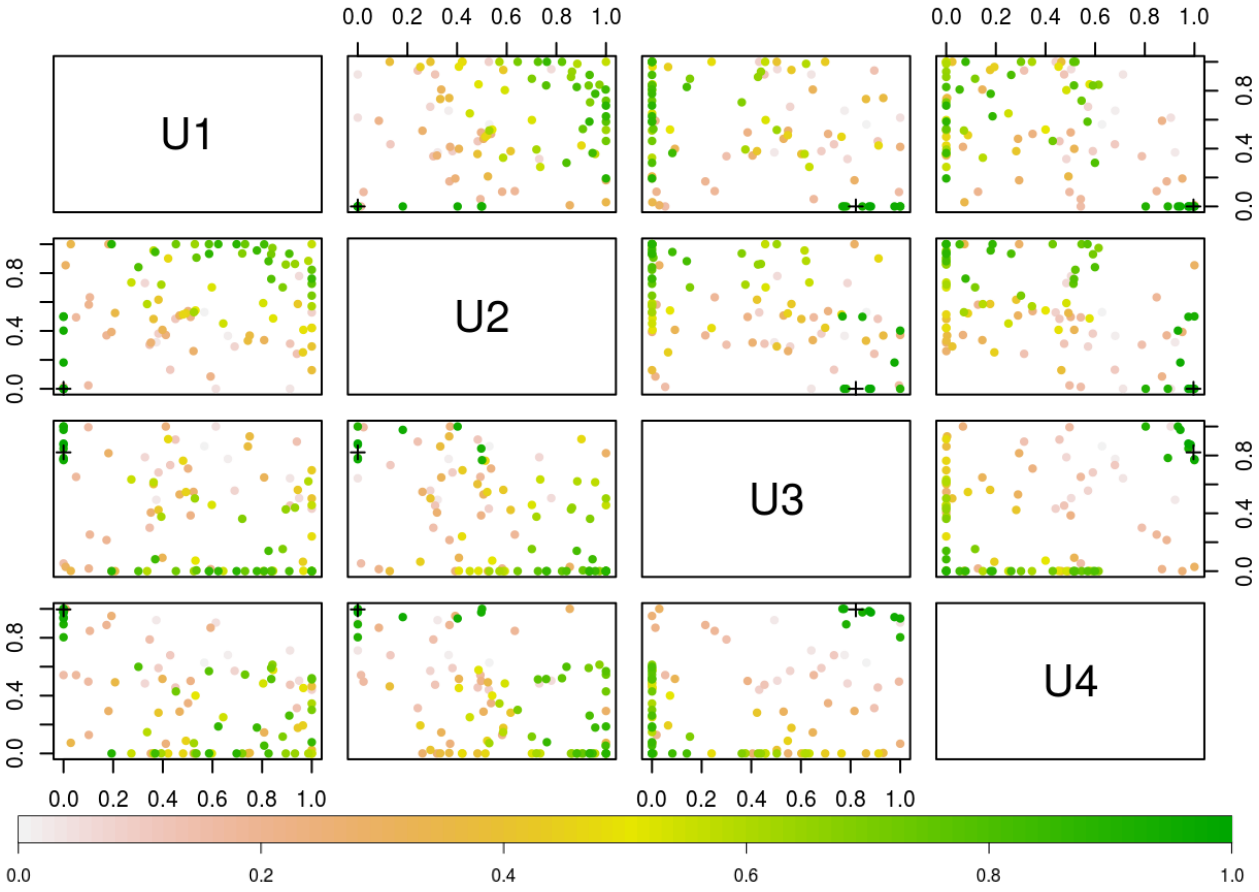


Figure 4.14: NLS calibration results for K_{direct}. Nonlinear least-square solutions of $\hat{\mathbf{u}}^{\mathrm{NLS}}$. Colors are derived from ranks of sum of squared errors to aid in visualization. These results are from 100 random space-filling initialization, with 98 of 100 converged and 2 failed to converge. Black "+" sign indicate the best values.

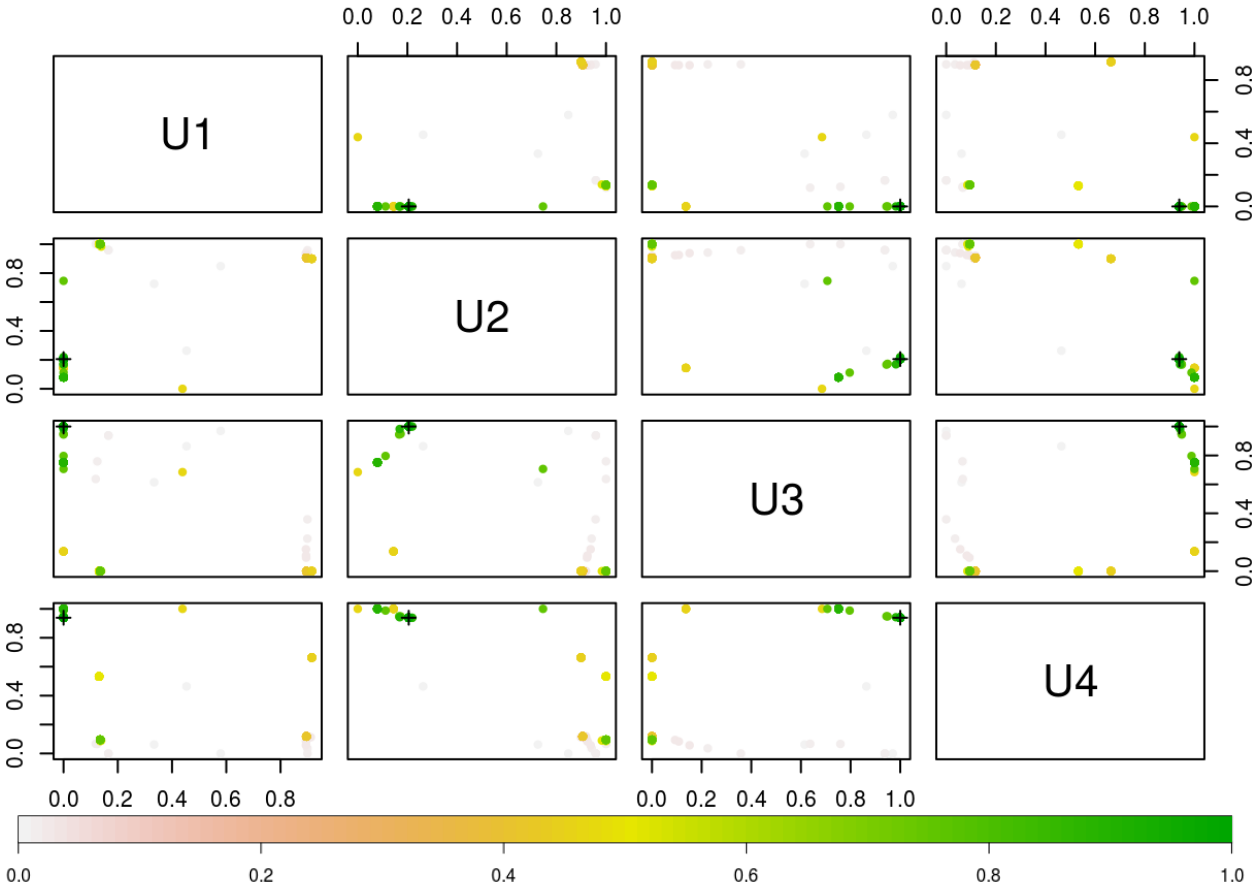


Figure 4.15: OSS LS no-bias calibration results for K_{direct}. OSS least-square no-bias solutions of $\hat{\mathbf{u}}^{\text{OSS}}_{\text{LS-nobias}}$. Colors are derived from ranks of RMSEs to aid in visualization. These converged results are from 500 random space-filling initialization. Black "+" sign indicate the best values.

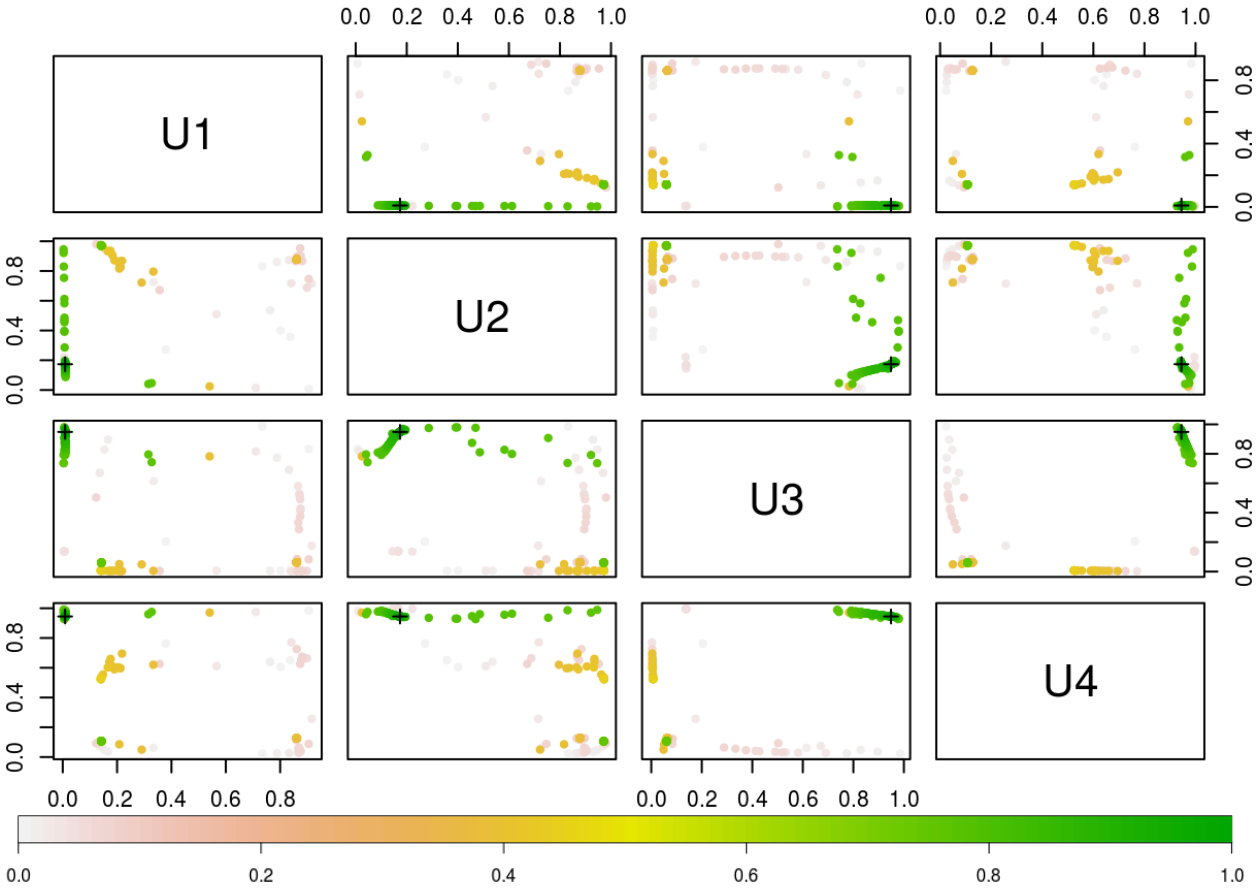


Figure 4.16: OSS optimization no-bias calibration results for K_{direct}. OSS no-bias solutions of $\hat{\mathbf{u}}^{\text{OSS}}_{\text{nobias}}$. Colors are derived from ranks of log-posterior to aid in visualization. These converged results are from 500 random space-filling initialization. Black "+" sign indicate the best values.

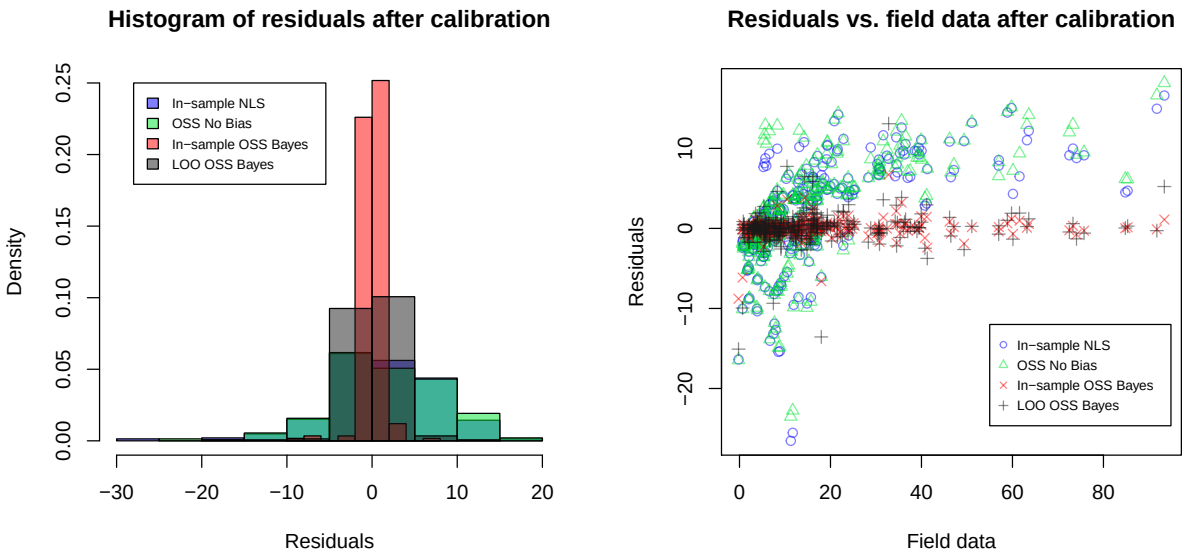


Figure 4.17: Residual plots over honeycomb field data. The left panel shows histograms comparing three approaches; right panel plots them versus the true response.

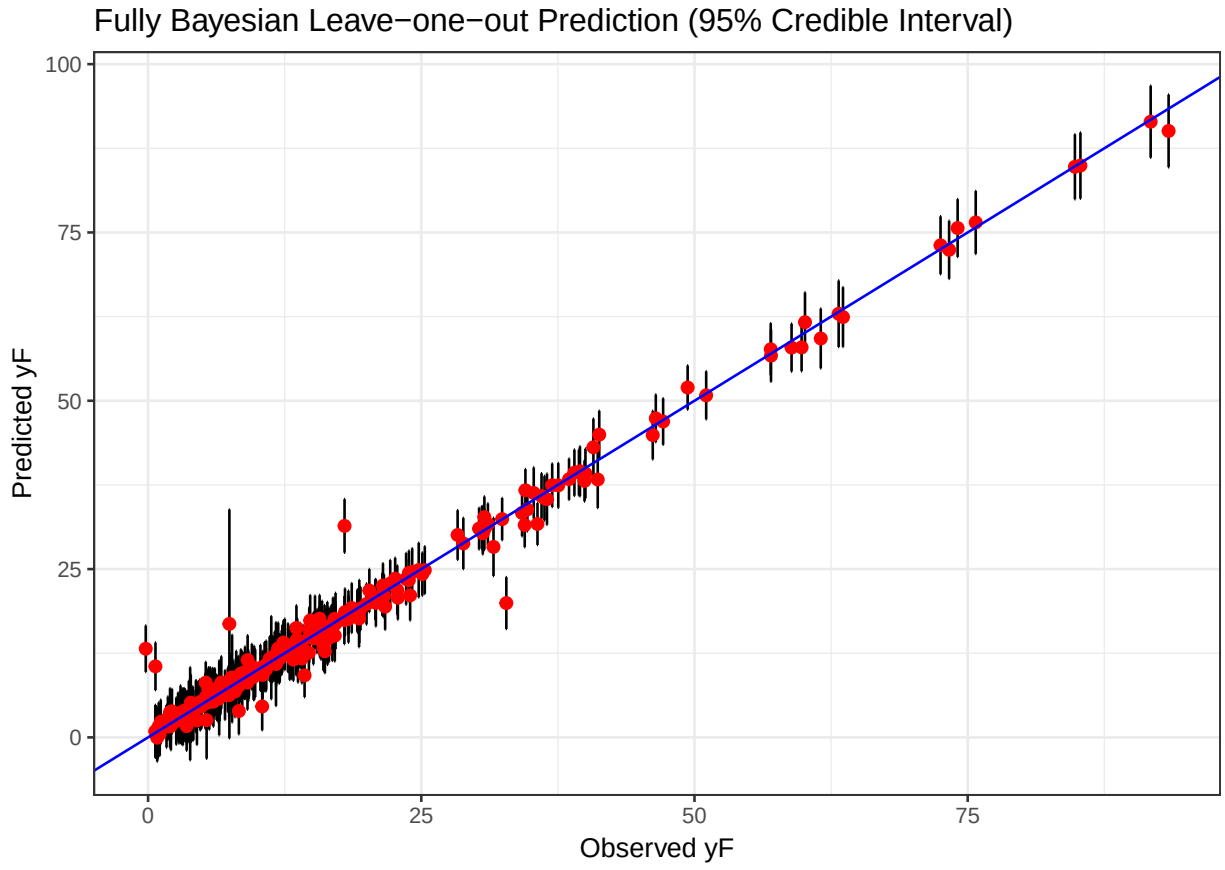


Figure 4.18: Fully Bayesian OSSs out-of-sample prediction. Fully Bayesian out-of-sample predicted $\mathbf{y}^F$ with 95% credible interval over observed honeycomb field data $\mathbf{y}^F$.

Chapter 5

Multiple output calibration using on-site surrogates

5.1 Introduction

In various applications of modern computer experiments, a wealth of information about the underlying physical and engineering systems can now be generated from computer models, in the form of high-dimensional outputs. For many researchers and practitioners actively working on computer experiments, "high-dimensional simulation output is the *rule*, rather than the exception"(Higdon et al. 2008). High dimensional outputs from computer models, including different forms of multivariate outputs and functional outputs, such as space, time, and spatial-temporal outputs, provide rich insights about the underlying physical reality.

In this chapter, we consider a model calibration problem for computer experiments with multiple outputs. Kennedy and O'Hagan 2001a proposed a univariate Bayesian calibration framework, combining the physical field observations $y^F(\mathbf{x})$ with computer model simulations $y^M(\mathbf{x}, \mathbf{u}^*)$ through a discrepancy term, or bias correction $b(\mathbf{x})$, between simulation and field as follows:

$$y^F(\mathbf{x}) = y^M(\mathbf{x}, \mathbf{u}^*) + b(\mathbf{x}) + \epsilon, \tag{5.1}$$

where $\mathbf{u}^*$ is the unknown "true" or "best" setting for the calibration input parameters, and $\epsilon \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\epsilon^2)$ represents random noise in the field measurements. Calibration with univariate output can be already fraught with different challenges, such as identifiability between the calibration parameter and model discrepancy (Kennedy and O'Hagan 2001a, Higdon et al. 2004, Bayarri et al. 2007a, Higdon et al. 2008, Tuo and Wu 2015, Plumlee 2017) and computational challenges from increasingly larger modern computer experiments (Liu, Bayarri, and Berger 2009, Gramacy et al. 2015, Huang et al. 2018).

Calibration problems with multivariate outputs and functional outputs can be even more challenging than univariate calibration problems, due to the additional burden of several factors. Multivariate outputs naturally bring orders of magnitude larger data size. In addition, the high-dimensional nature of the data requires dimension reduction and data compression. Nevertheless, combining multiple physically meaningful outputs together offers the potential for improvement of model calibration. Multivariate data provides richer insights from different sources for better Bayesian learning and further uncertainty reduction.

Different calibration methodology for computer experiments with multiple outputs

have been developed in the literature for different purposes, due to specific characteristics of the underlying physical systems. Higdon et al. 2008 extends the Kenny and O'Hagan (KOH) Bayesian calibration framework from univariate output to high-dimensional output using basis representations. Highly multivariate simulation and experimental outputs are represented through principal-component decomposition and combined into a coherent Bayesian calibration framework. Once the high-dimensional outputs can be efficiently represented through basis decomposition, the multivariate calibration framework from Higdon et al. 2008 provides fully Bayesian tractability via Markov chain Moonte Carlo with computational efficiency.

Also developed around the same time, Bayarri et al. 2007b utilizes a wavelet representation of functional outputs, in order to incorporate highly irregular functional simulation outputs. When the computer model is computationally demanding and Gaussian process emulation is required and sufficient, Paulo, Garcia-Donato, and Palomo 2012 provide a KOH-style multivariate calibration framework through Linear Model of Coregionalization (LMC). The LMC structure of multivariate outputs keeps the calibration problem tractable within a Gaussian process-based framework. Moreover, there are also empirical results indicating that including multiple outputs in model calibration can further improve identifiability of calibration parameter and model discrepancy, comparing to its univariate counterparts (Arendt et al. 2012).

The rest of this chapter is organized as follows. Section 5.2 describes high-dimensional outputs of the honeycomb gas seal application, the challenges stemming from its simulation, and subsequent attempts to calibrate via on-site surrogates. Section 5.3 applies the uni-

variate on-site surrogates calibration method developed in Chapter 4 individually to each of the multiple outputs in honeycomb application. A discussion of benefits and limitations of this univariate approach is also presented in Section 5.3. Section 5.4 extends the on-site surrogates calibration method to multiple output calibration through basis representations, where the basis is built in the on-site observed discrepancy space with a space-filling design on calibration parameters. New and improved multiple output calibration results are also presented in Section 5.4. Section 5.5 concludes this chapter with a brief discussion on potential future research directions.

5.2 Honeycomb gas seal with multiple outputs

The honeycomb seal is an important component widely used in BHGE's high-pressure centrifugal compressors to enhance rotor stability in oil and gas applications or to control leakage in aircraft gas turbines. The seal(s) and applications at BHGE are described by $p_x = 13$ design variables $\mathbf{x}$ characterizing geometry and flow dynamics: rotational speed, cell depth, seal diameter and length, inlet swirl, gas viscosity, gas temperature, compressibility factor, specific heat, inlet/outlet pressure, and clearance. The field experiment, from BHGE's component-level honeycomb seal test campaign, comprises $N_F = 292$ runs varying a subset of those conditions, $\mathbf{X}^F$, believed to have greatest variability during turbomachinery operation: clearance, swirl, cell depth, seal length, and seal diameter. Measured outputs include direct/cross stiffness and damping, at multiple frequencies.

5.2.1 Multiple outputs in honeycomb

These multiple physically meaningful outputs at six levels of frequency include: direct stiffness (K_{direct}), cross stiffness (k_{cross}), direct damping (C_{direct}), cross damping (c_{cross}), at frequencies 28 Hz, 70 Hz, 98 Hz, 126 Hz, 154 Hz, and 182 Hz. Due to missing and observational constraints on outputs at frequencies 98 Hz and 182 Hz from BHGE's physical experimental test campaign, we only consider multiple outputs at the remaining four frequencies, at 28 Hz, 70 Hz, 126 Hz and 154 Hz, in this chapter.

A proof-of-concept relationship between the four outputs, direct stiffness (K), cross stiffness (k), direct damping (C), and cross damping (c), has been investigated in turbomachinery literature based on bulk-flow model (Hirs 1973). Several studies on rotordynamic coefficient predictions have been conducted from a mechanical engineering perspective (Childs 1993, Kleynhans and Childs 1997, D'Souza and Childs 2002). Described by a classical transfer function in bulk-flow theory (Hirs 1973, D'Souza and Childs 2002), the multiple outputs of direct/cross stiffness and damping coefficients in honeycomb gas seal process can be expressed in a conventional linear motion/reaction-force model

$$-\begin{bmatrix} F_x \\ F_y \end{bmatrix} = \begin{bmatrix} K & k \\ -k & K \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} C & c \\ -c & C \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}. \tag{5.2}$$

In units newtons (N) for the force, meters (m) for length and seconds for time, the terms direct stiffness K in [N/m] and direct damping C in units [N.s/m] account for a reaction force in the direction of motion. The terms cross-coupled stiffness k in [N/m] and cross-coupled

damping c in [N.s/m] develop reaction forces that are orthogonal to the direction of motion.

A simulator called `ISOTSEAL`, developed at Texas A&M University (Kleynhans and Childs 1997), offers a relatively speedy evaluation (about one second) of the response(s) of interest for the honeycomb seal under study at BHGE. `ISOTSEAL` is built on bulk-flow theory, calculating gas seal force coefficients based on seal flow physics. Our BHGE colleagues have developed an R interface mapping seventeen scalar inputs for the honeycomb seal experiment into the format required for `ISOTSEAL`. Thirteen of those inputs match up with the columns of $\mathbf{X}^F$ (i.e., they are $\mathbf{x}$'s); four are tuning parameters $\mathbf{u}$, which could not be controlled in the field. We further build an R interface with `ISOTSEAL` to simulate at any specified frequency levels. Four different outputs including direct/crossing damping and stiffness (K, k, C, c) can be simulated at once through `ISOTSEAL` by setting the seventeen dimensional inputs $(\mathbf{x}, \mathbf{u})$ at a specified frequency level using this interface. Also, see Section 4.2.1 in Chapter 4 for more `ISOTSEAL` simulator details.

5.2.2 On-site surrogates for multiple outputs

As demonstrated in Section 4.2.1, several challenging issues exist in this honeycomb calibration problem: high-dimensional inputs, nonstationary outputs, missing data, and numerical instability in `ISOTSEAL` computer model. Traditional global surrogate modeling methods fail to account for local heterogeneities due to both local nonstationarities and input-dependent numerical instability, even at prohibitively expensive computational cost. The challenges of global nonstationarity, local features, numerical artifacts, and high input

dimension together inhibit direct adoption of the canonical Kennedy and O'Hagan calibration apparatus in this honeycomb problem.

As presented in Chapter 4, we develop a new divide-and-conquer approach to large-scale calibration problems based on on-site experimental design and surrogate modeling, called *on-site surrogates* approach. On-site surrogates reduce a $p = p_x + p_u = 17$-dimensional problem into a $p_u = 4$-dimensional problem by building as many surrogates as there are field data observations, $N_F = 292$. The mapping from one big surrogate to many smaller but better focused ones may be conceptualized by the following chart:

$$\hat{y}^M(\mathbf{x}, \mathbf{u}) \longrightarrow \hat{y}^M(\mathbf{x}_i, \mathbf{u}) \longrightarrow \hat{y}_i^M(\mathbf{u}), \quad \text{for } \; i = 1, 2, \ldots, N_F. \tag{5.3}$$

That is, rather than building one big emulator for the entire p-dimensional input space $\hat{y}^M(\mathbf{x}, \mathbf{u})$, we instead train separate emulators $\hat{y}_i^M(\mathbf{u})$ focused on each site $\mathbf{x}_i$ where field data has been collected. In this way, OSSs are a divide-and-conquer scheme that swap joint modeling in a large $(\mathbf{x}, \mathbf{u})$-space, where design coverage and modeling fidelity could at best be thin, for many smaller models in which, separately, ample coverage is attainable with modestly sized design in $\mathbf{u}$-space only. Specific on-site features can be captured in parallel, since the fitting and simulation for each field data site $\mathbf{x}_i$, $i = 1, \ldots, N_F$ are both operationally and statistically independent. Nonstationary modeling is implicit, since each surrogate focuses on a different part of the input space. See Section 4.3 for more details.

Here, we further simulate all on-site `ISOTSEAL` multiple outputs for direct/crossing damping and stiffness (K, k, C, c) at multiple frequency levels 28 Hz, 70 Hz, 126 Hz and 154

Hz, in a similar fashion as described in Section 4.3.1 for direct stiffness output $y \equiv K$ at 28 Hz. At each of the $N_F = 292$ field data sites, we create the same 1000-run maximin distance LHS designs for friction factors in $p_u = 4$-dimensional $\mathbf{u}$-space. In this way, we design individually four sets of $N_M = 292,000$ `ISOTSEAL` simulation runs at four different frequency levels 28 Hz, 70 Hz, 126 Hz and 154 Hz.

Let $\mathbf{Y}_i^M = [\mathbf{y}_1^M(\mathbf{U}_i), \mathbf{y}_2^M(\mathbf{U}_i), \ldots, \mathbf{y}_{16}^M(\mathbf{U}_i)]$, for $i = 1, 2, \ldots, N_F$ be 16-column matrix holding the n_i rows of converged `ISOTSEAL` runs for the i^{th} site, where the 16 columns are from 4 outputs at 4 frequencies and $n_i \leq 1,000$. $\mathbf{U}_i$ is the corresponding $n_i \times p_u$ on-site design matrix. In our `ISOTSEAL` experiment, where $N_F = 292$, a total of $N_M = \sum_{i=1}^{N_F} n_i = 286,282$ runs terminated successfully. Most sites (241) had $n_i = 1,000$ successful runs from a complete on-site maximin LHS. Of the 51 with missing responses of varying multitudes, the smallest was $n_{238} = 574$.

On each site, the OSSs comprise of multiple fitted GP regressions between successful on-site `ISOTSEAL` run outputs $\mathbf{y}_i^M$ and $\mathbf{U}_i$. Specifically, $\hat{y}_i^M(\mathbf{U}_i)$ is built by fitting a stationary zero-mean GP using a scaled and nugget-augmented separable Gaussian power exponential kernel

$$V_i(\mathbf{u}, \mathbf{u}') = \tau_i^2 \exp\left\{ -\sum_{k=1}^{p_u} \frac{||\mathbf{u}_{ik} - \mathbf{u}'_{ik}||^2}{\theta_{ik}} + \delta_{u,u'}\eta_i \right\}, \tag{5.4}$$

where τ_i^2 is a site-specific scale parameter, $\boldsymbol{\theta}_i = (\theta_{i1}, \theta_{i2}, \ldots, \theta_{ip_u})^\top$ is vector of site-specific lengthscales, η_i is a nugget parameter, and $\delta_{u,u'}$ is the Kronecker delta. Denote the set of hyperparameters of the i^{th} OSS as $\boldsymbol{\phi}_i = \{\tau_i^2, \boldsymbol{\theta}_i, \eta_i\}$, for $i = 1, 2, \ldots, N_F$. Although nuggets

η_i are usually fit to smooth over noise, here we are including them to smooth over any deterministic numerical "jitters." Other mean and covariance structures may be reasonable, so in what follows let $\boldsymbol{\phi}_i$ stand in generically for the estimable quantities of each OSS. Although numerous options for inference exist, we prefer plug-in maximum likelihood estimates (MLEs) $\hat{\boldsymbol{\phi}}_i$, calculated in parallel for each $i = 1, \ldots, N_F = 292$ via `L-BFGS-B` (Byrd et al. 1995) using analytic derivatives via `mleGPsep` in the `laGP` package (Gramacy and Sun 2018; Gramacy 2016) for R.

In total, we build up 16 corpora with 292 on-site surrogates for each outputs, at 4 frequencies for four different outputs direct stiffness (K_{direct}), cross stiffness (k_{cross}), direct damping (C_{direct}), and cross damping (c_{cross}). This simple OSS strategy provides far more accurate emulation out-of-sample than does the best global alternative we could muster with a commensurate computational effort. One obvious explanation for better emulation is that the OSS structure can incorporate a better focused design with much larger sized on-site simulated data. In addition, the OSS structure naturally brings model flexibility to capture on-site nonstationary features through all site-specific Gaussian processes hyperparameters.

5.3 On-site surrogates univariate calibration for multiple outputs

In initial efforts, I implement the on-site surrogate calibration with both optimization and fully Bayesian approaches presented in Chapter 4 for each of the other three outputs

individually: cross stiffness, direct damping, and cross damping at frequency 28 Hz.

5.3.1 OSSs univariate calibration under Beta priors

Firstly, I implemented on-site surrogate calibration with both optimization and fully Bayesian approaches for all the four different outputs individually, using regularizing independent Beta priors, $u_j \overset{\text{iid}}{\sim} \text{Beta}(2, 2)$, as described in Chapter 4. Combining the results from Figures 4.12 with new results in Figures 5.1, 5.2, and 5.3 for all four outputs, there is a complicated relationship between multiple outputs in honeycomb. Calibrating them individually result in different best settings of calibration parameters $\hat{\mathbf{u}}$ and, obviously, with different corresponding model discrepancy terms.

5.3.2 OSSs univariate calibration under uniform priors

Next, we implement on-site surrogate calibration with both optimization and fully Bayesian approaches for all the four different outputs individually, using independent uniform priors, $u_j \overset{\text{iid}}{\sim} \text{Uniform}(0, 1)$. Combining the results from Figures 4.13 with the new results in figures 5.4, 5.5, and 5.6 for all four outputs, there is a complicated relationship between multiple outputs of honeycomb. Calibrating them individually result in different best settings of calibration parameters $\hat{\mathbf{u}}$ and also different corresponding model discrepancies.

5.3.3 Discussion

From these eight Figures under two sets of priors, several interesting patterns can be observed. Firstly, all four outputs of direct/cross stiffness and damping contain different level of uncertainties in the calibration parameters, from both the MAP and the full posterior distributions. Secondly, it seems that the two direct outputs have more similar calibration parameters to each other (see Figures 4.12, 4.13 for direct stiffness and Figures 5.2, 5.5 for direct damping) while the calibration parameters of the two cross outputs are also similar to each other (see Figures 5.1, 5.4 for for cross stiffness Figures 5.3, 5.6 for for cross damping). Thirdly, both the optimization approach and the fully Bayesian calibration yield consistent results, with the fully Bayesian approach providing a complete posterior uncertainty evaluation and the optimization approach showing many local optima. Lastly, the regularizing Beta prior provides more robust posterior distributions and facilitates learning for calibration parameters. However, prior information substantially affect the posterior distributions and the MAP values under these two sets of priors, suggesting the information in the likelihood function here can not sufficiently uncover both the unknown, and possibly confounded, calibration parameters and model discrepancy at once.

Furthermore, from these plots it is clear that the best settings of the four calibration parameters are still very different, even given at only one frequency level 28 Hz. While calibrating each of multiple outputs individually at different frequency level can take tremendous amount of computational efforts, the final results can not directly be combined into one uniform solution. This inconsistency in solution and the potential to improve motivate

us to develop new multivariate calibration methodology.

Combining multiple outputs for calibration can be a more challenging task than univariate calibration, especially when each output yields a different best setting of calibration parameters. On the other hand, higher dimensional data can provide richer information for learning calibration parameter and model discrepancy. High dimensional data can be especially informative when only a limited amount field data is available in a relatively high-dimensional inputs $\mathbf{X}^F$ space under a highly nonlinear response surface. This is the case in our motivating honeycomb example. For this honeycomb problem, we intend to learn the nonlinear discrepancy term in $p_x = 13$ dimensional input space with only $N_F = 292$ runs of field physical observations. Multiple outputs can provide richer information from different angles for better calibration and further uncertainty reduction.

5.4 OSS multiple output calibration via basis representations

5.4.1 Basis representations for multiple frequencies

After several iterations of careful exploratory data analyses, we observed that the multiple frequency levels of each of the four outputs from both the `ISOTSEAL` simulation y^M and field observations y^F are all highly linearly correlated. See the pairwise scatterplots for the multiple outputs at multiple frequencies, as demonstrated in Figure 5.7 for direct stiffness (K_{direct}), Figure 5.8 for cross stiffness (k_{cross}), Figure 5.9 for direct damping (C_{direct}), and

Figure 5.10 for cross damping (c_{cross}).

In particular, we found out that principal-component basis representations can effectively combine the four frequencies 28 Hz, 70 Hz, 126 Hz, and 154 Hz for four outputs direct stiffness (K_{direct}), cross stiffness (k_{cross}), direct damping (C_{direct}), and cross damping (c_{cross}). For the purpose of calibration, we are also in face of the uncertainty from unknown calibration parameters $\mathbf{u}$. Thus, we combine different frequencies using a principal-component basis in the space of the observed discrepancy between the field data $\mathbf{y}^F(\mathbf{X}_F)$ and the multiple on-site emulations $\hat{\mathbf{y}}^M(\mathbf{X}_F, \mathbf{U})$ on a space-filling design on the unknown calibration parameter $\mathbf{u}$ built from Section 5.2.2

$$\mathbf{y}^F_{ij}(\mathbf{X}_F) - \hat{\mathbf{y}}^M_{ij}(\mathbf{X}_F, \mathbf{U}) \tag{5.5}$$

where i labels one of the four output types with $i \in \{K_{direct}, k_{cross}, C_{direct}, c_{cross}\}$ and j indicates the level of frequency $j \in \{28$ Hz, 70 Hz, 126 Hz, 154 Hz$\}$. We further summarize the first two component variance representations in Table 5.1 with all the scree plots in Figure 5.11

Output	K_{direct}	k_{cross}	C_{direct}	c_{cross}
Variation represented in the first PC	96.46%	94.89%	91.49%	84.19%
Variation represented in the second PC	2.843%	4.658%	6.414%	12.68%

Table 5.1: Variation representations from principal-component bases in the observed discrepancy on four outputs K_{direct}, k_{cross}, C_{direct}, and c_{cross}, combing 4 levels of frequency 28 Hz, 70 Hz, 126 Hz, and 154 Hz for each outputs.

5.4.2 Multiple output calibration combining different frequencies through basis representations

Acting on all the 16 corpora of 292 on-site surrogates built as described from Section 5.2.2, we perform optimization calibration in their basis representations for each of these four outputs: direct stiffness (K_{direct}), cross stiffness (k_{cross}), direct damping (C_{direct}), and cross damping (c_{cross}). Even with a full corpus of accurate and extremely parsimonious on-site surrogates focusing at all field data locations with multiple outputs, Bayesian calibration can still be computationally challenging in large-scale computer experiments. In the KOH framework (4.1), both $\mathbf{u}^*$ and a bias correcting GP $b(\mathbf{x})$, via hyperparameters $\boldsymbol{\phi}_b$, are unknown and must jointly be estimated. Here, we take the calibration as optimization approach described in Section 4.3.3 as an initial step to alleviate the computational burden.

As an alternative to the fully Bayesian method, we implement here an adaptation of Gramacy et al. 2015's modularized calibration as optimization in the basis space of the first principal component representation for the observed discrepancy in equation 5.5. Instead of sampling a full posterior distribution, $\hat{b}(\cdot)$ and $\hat{\mathbf{u}}$ are calculated as

$$\hat{\mathbf{u}} = \arg\max_{\mathbf{u}} \left\{ p(\mathbf{u}) \left[\max_{\boldsymbol{\phi}_b} p_b(\boldsymbol{\phi}_b \mid \mathbf{D}^B_{N_F}(\mathbf{u})) \right] \right\}, \tag{5.6}$$

which explores different values of $\hat{\mathbf{u}}$ via the resulting posterior probability of discrepancy hyperparameters $p_b(\phi_b \mid \mathbf{D}^B_{N_F}(\mathbf{u}))$ applied to a data set of residuals $\mathbf{D}^B_{N_F}(\mathbf{u})$. Specifically, $\mathbf{D}^B_{N_F}(\mathbf{u}) = (\mathbf{X}^F_{N_F}, \hat{\mathbf{y}}^{B|u}_{N_F})$ is the observed field inputs $\mathbf{X}^F_{N_F}$ and discrepancies $\hat{\mathbf{y}}^{B|\mathbf{u}}_{N_F} = \mathbf{y}^F_{N_F} - \hat{\mathbf{y}}^{M|\mathbf{u}}_{N_F}$

in its first principal component space given a particular $\mathbf{u}$. The probability $p_b(\cdot \mid \cdot)$ refers to the marginal likelihood of the GP with parameters $\hat{\boldsymbol{\phi}}_b$ fit to those residuals via their own "inner" derivative-based optimization routine. The object in Eq. (5.6) basically encodes the idea that $\mathbf{u}$-settings leading to better-fitting GP bias corrections are preferred. We prefer independent $u_j \sim \text{Beta}(2, 2)$ in each coordinate as a means of regularizing the search by mildly penalizing boundary solutions, in part because we know that frictions factors at the boundaries of $\mathbf{u}$-space lean heavily on the surrogate as runs of `ISOTSEAL` fail to converge there.

In practice, the log of the criteria in Eq. (5.6) can be optimized numerically with robust library methods such as "L-BFGS-B", via `optim` (R Core Team 2018), or `nloptr` (Ypma, Borchers, and Eddelbuettel 2017). Since the optimizations are fast but local and since the surface being optimized can have many local optima, we entertain a large set of random initializations—in parallel—in search for the best (most global) solution for $\hat{\mathbf{u}}$ and $\hat{b}(\cdot)$. We summarize all these optimization calibration results from 500 random space-filling initialization in Figure 5.12 for direct stiffness (K_{direct}), Figure 5.13 for cross stiffness (k_{cross}), Figure 5.14 for direct damping (C_{direct}), and Figure 5.15 for cross damping (c_{cross}).

Comparing to these univariate calibration results from Section 5.3, combining multiple frequencies in outputs enhance Bayesian learning for both unknown calibration parameters and model discrepancy. It is clear that the posterior distributions of $\mathbf{u}$ in Figures 5.12, 5.14, and 5.15 are much more concentrated in a smaller dense green area, compared to their corresponding univariate counterparts in the these upper triangle plots from 4.12, 5.2, and

5.3. The overall similar shapes of posterior distribution with more concentrated higher posterior probability regions suggests that the combined outputs yield consistent yet improved calibration results, with further reduced uncertainty in the calibration parameters $\mathbf{u}$. The only exception is the output cross stiffness k_{cross}: combining multiple outputs does not help too much here, as the best setting of calibration parameter $\mathbf{u}$ does not seem to be constant across the input space. As shown in Figures 5.1 and 5.13, both univariate calibration and multivariate calibration lead to a correlation structure between the two pairs of parameters in more complicated structures: u_1 seems to be negatively correlated with u_2 while u_3 seems to be negatively but also more nonlinearly correlated with u_4.

5.5 Discussion

In this chapter, we present both the OSS univariate calibration results under two different priors and OSS multiple outputs calibration using basis representations. The results from OSS multiple outputs calibration using basis representations provide improved and more concentrated distribution of calibration parameters with reduced uncertainty, compared to results from corresponding univariate alternatives. Combining the multiple outputs at four frequencies 28 Hz, 70 Hz, 126 Hz and 154 Hz using basis representations, we obtain uniform and improved solutions for three out of four outputs at multiple frequencies. Now, we have combined 16 different sources of multiple outputs into 4 separate calibration frameworks: direct stiffness (K_{direct}), cross stiffness (k_{cross}), direct damping (C_{direct}), and cross damping (c_{cross}). Are there any better way to further combine these four basis representations of

direct/cross stiffness and damping into one single unified calibration framework? Much work remains to assess the potential for such an approach, say via *co-kriging* (Ver Hoef and Barry 1998), linear model of co-regionalization e.g., Wackernagel 1998, or through other multi-objective optimization algorithms, such as Pareto-optimal framework in Binois, Ginsbourger, and Roustant 2015. Last but not least, the patterns from cross stiffness k_{cross} indicates another venue for future investigation: input-dependent calibration.

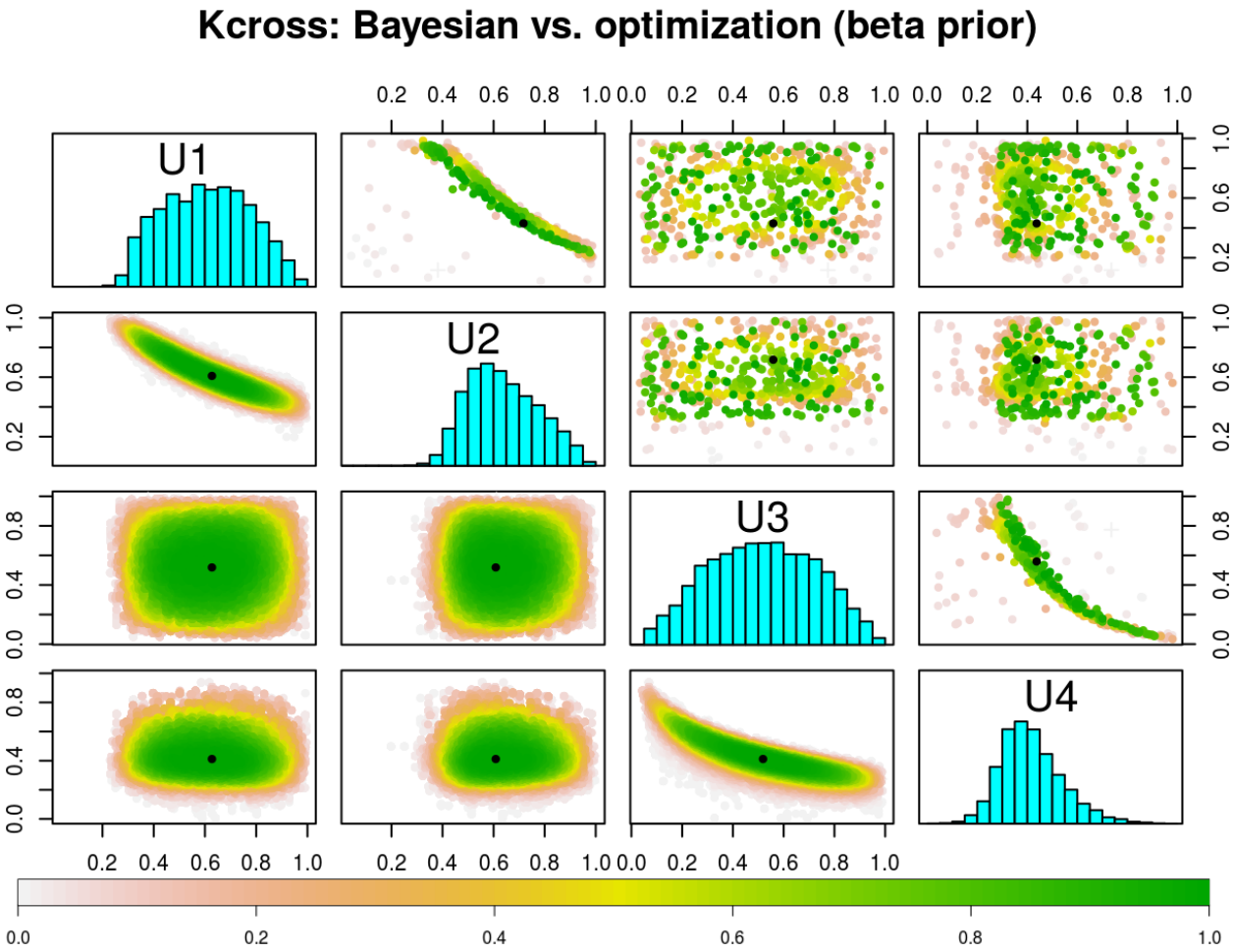


Figure 5.1: Calibration results for honeycomb with Beta prior on k_{cross}. Bayesian KOH (lower and diagonal) posterior for $\mathbf{u}$ vs. modularlized optimization (upper) analog, both under an independent Beta$(2, 2)$ prior for $\mathbf{u}$ on k_{cross}. Colors are derived from ranks of posterior probabilities to aid in visualization. Modularized results are from 500 converged optimization under random initialization. Black dots indicate MAP values.

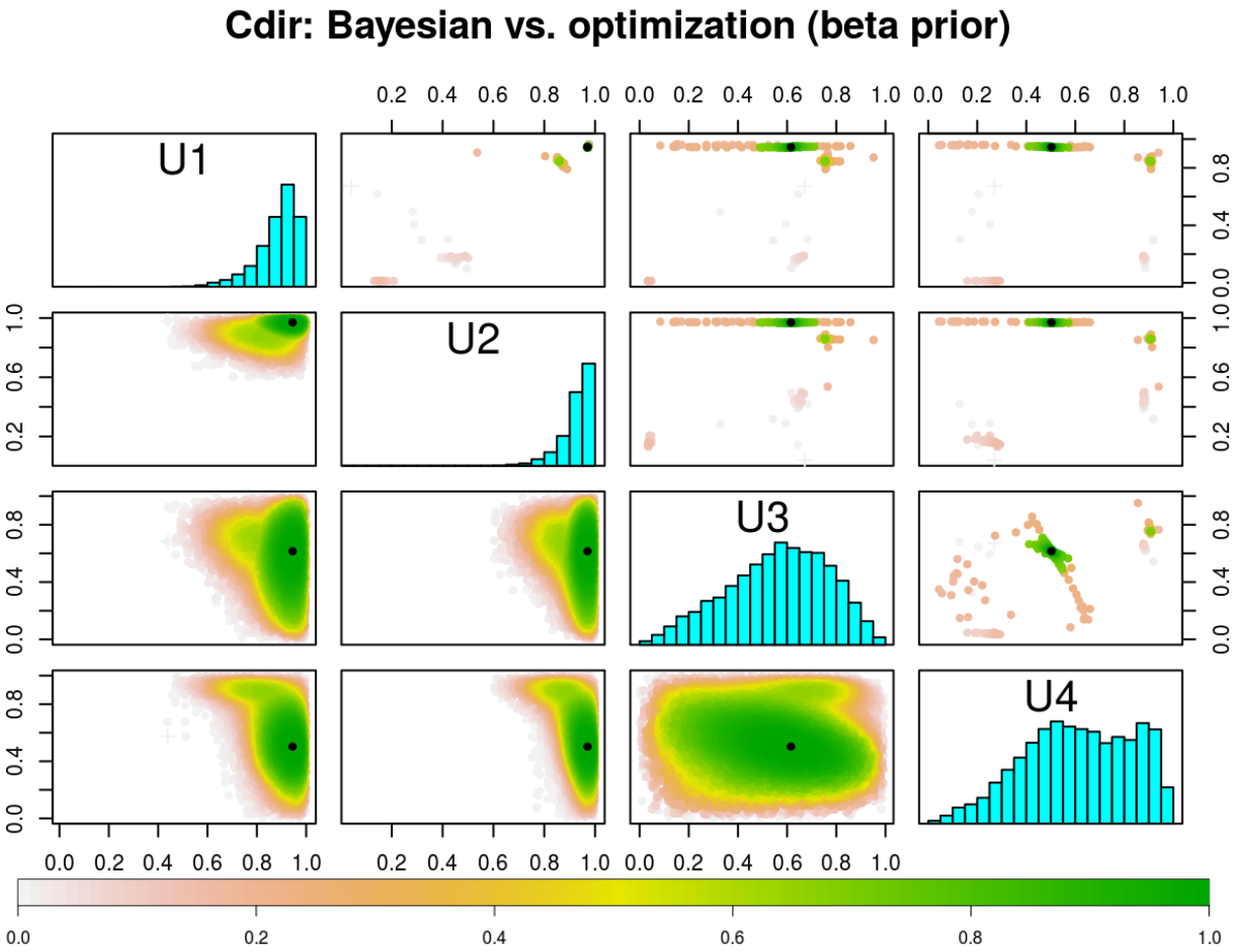


Figure 5.2: Calibration results for honeycomb with Beta prior on C_{direct}. Bayesian KOH (lower and diagonal) posterior for $\mathbf{u}$ vs. modularlized optimization (upper) analog, both under an independent Beta$(2, 2)$ prior for $\mathbf{u}$ on C_{direct}. Colors are derived from ranks of posterior probabilities to aid in visualization. Modularized results are from 500 converged optimization under random initialization. Black dots indicate MAP values.

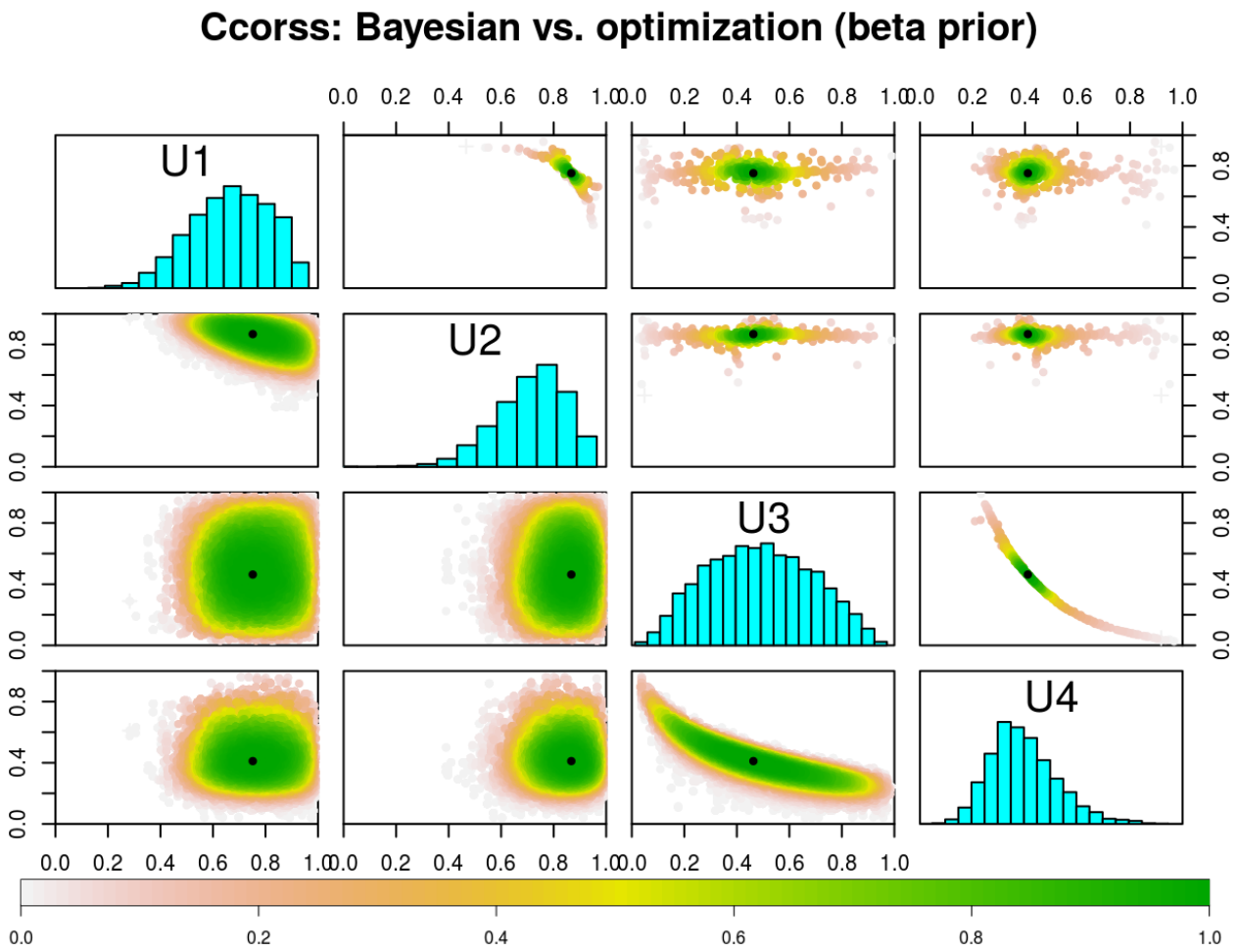


Figure 5.3: Calibration results for honeycomb with Beta prior on c_{cross}. Bayesian KOH (lower and diagonal) posterior for $\mathbf{u}$ vs. modularlized optimization (upper) analog, both under an independent Beta$(2, 2)$ prior for $\mathbf{u}$ on c_{cross}. Colors are derived from ranks of posterior probabilities to aid in visualization. Modularized results are from 500 converged optimization under random initialization. Black dots indicate MAP values.

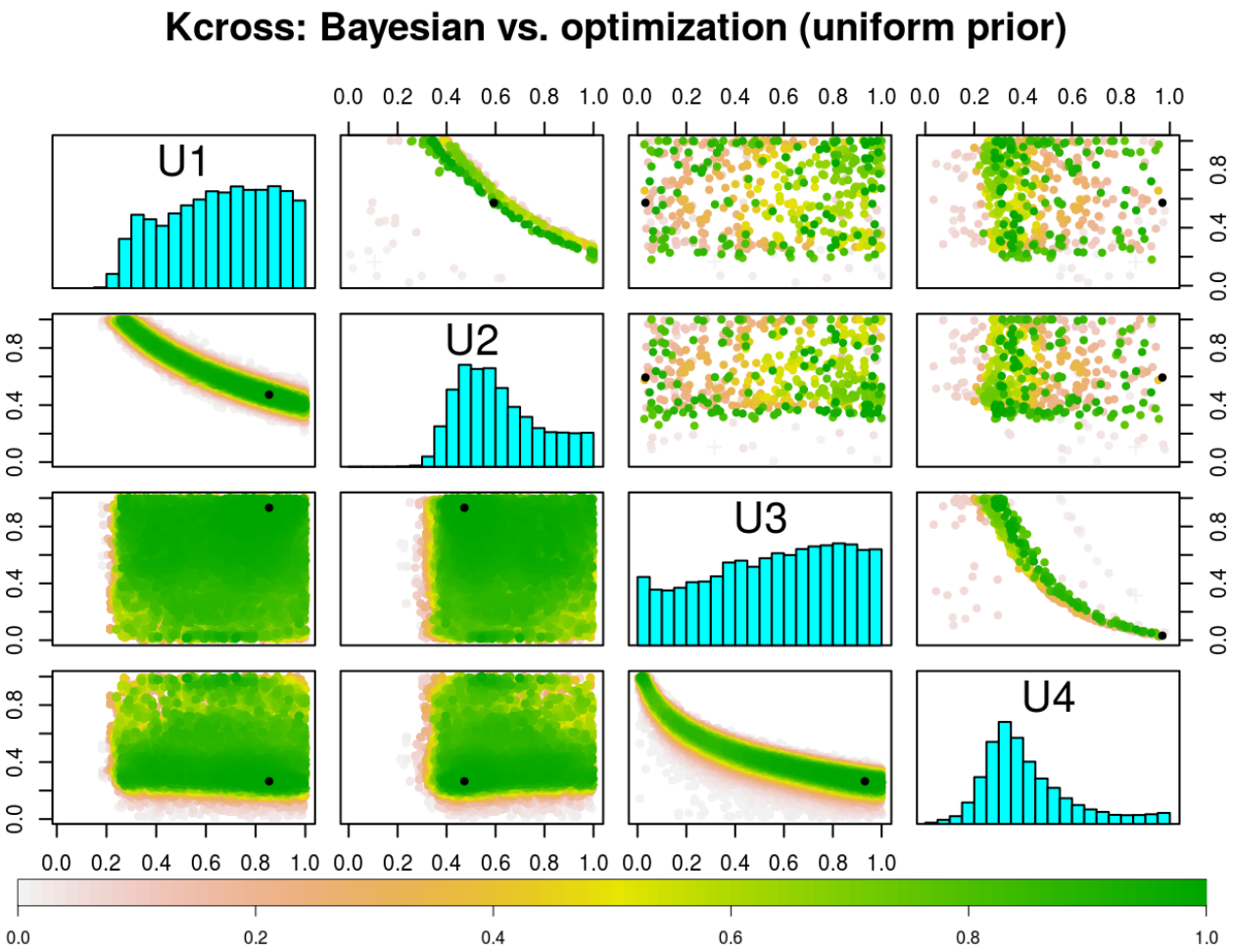


Figure 5.4: Calibration results for honeycomb with uniform prior on k_{cross}. Bayesian KOH (lower and diagonal) posterior for $\mathbf{u}$ vs. modularlized optimization (upper) analog, both under an independent uniform$(0, 1)$ prior for $\mathbf{u}$ on cross stiffness k_{cross}. Colors are derived from ranks of posterior probabilities to aid in visualization. Modularized results are from 500 converged optimization under random initialization. Black dots indicate MAP values.

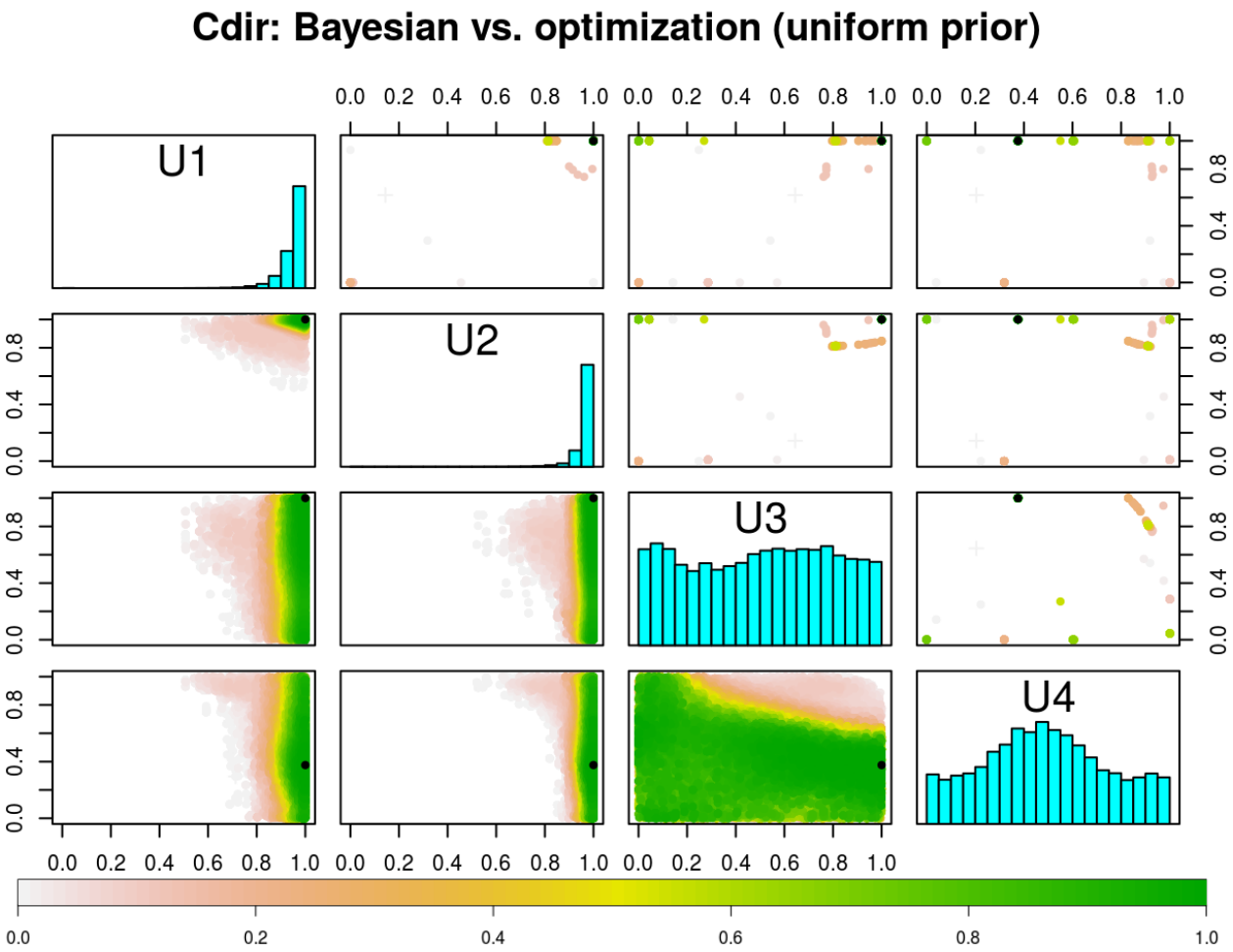


Figure 5.5: Calibration results for honeycomb with uniform prior on C_{direct}. Bayesian KOH (lower and diagonal) posterior for $\mathbf{u}$ vs. modularlized optimization (upper) analog, both under an independent uniform$(0, 1)$ prior for $\mathbf{u}$ on direct damping C_{direct}. Colors are derived from ranks of posterior probabilities to aid in visualization. Modularized results are from 500 converged optimization under random initialization. Black dots indicate MAP values.

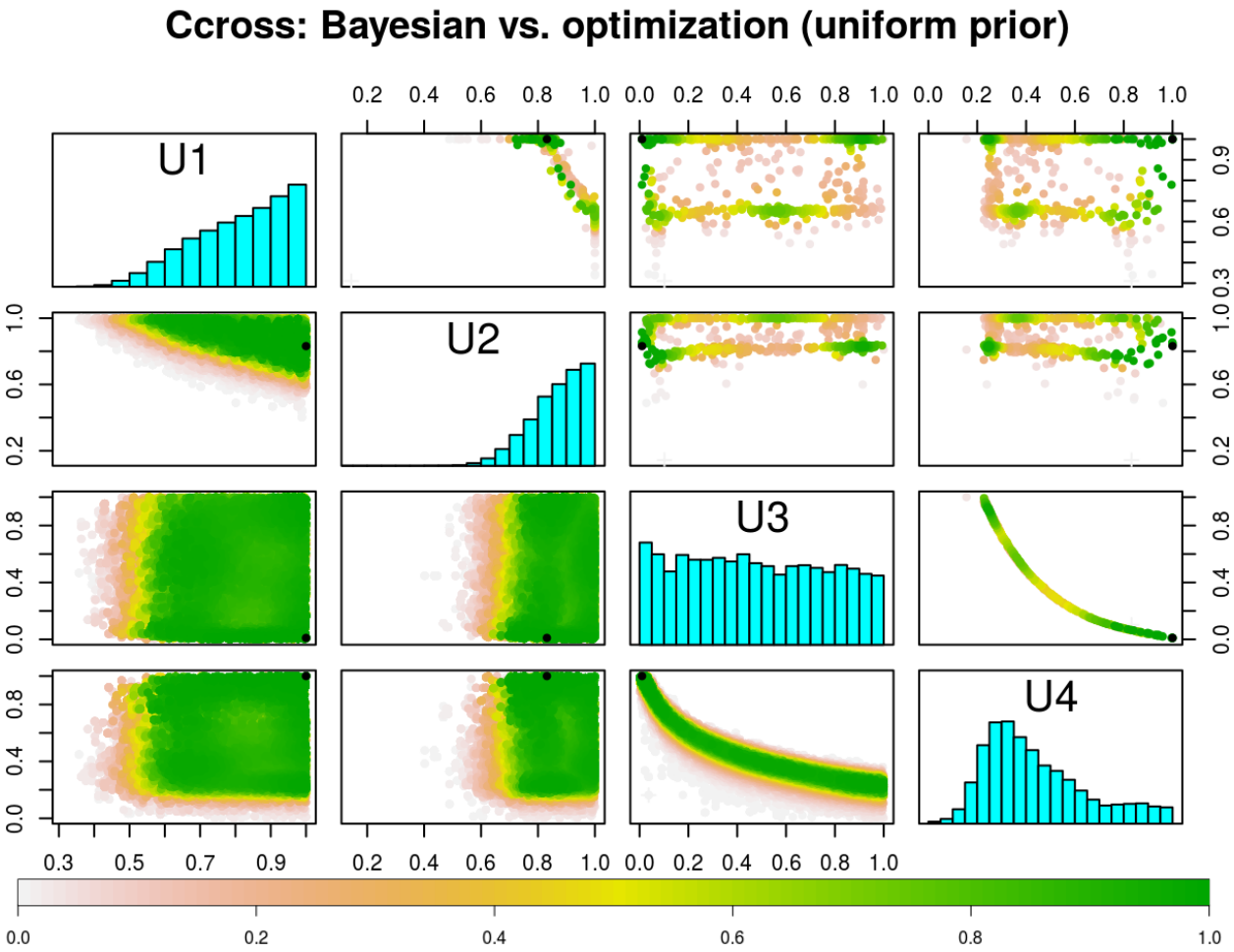


Figure 5.6: Calibration results for honeycomb with uniform prior on c_{cross}. Bayesian KOH (lower and diagonal) posterior for $\mathbf{u}$ vs. modularlized optimization (upper) analog, both under an independent uniform$(0, 1)$ prior for $\mathbf{u}$ on cross damping c_{cross}. Colors are derived from ranks of posterior probabilities to aid in visualization. Modularized results are from 500 converged optimization under random initialization. Black dots indicate MAP values.

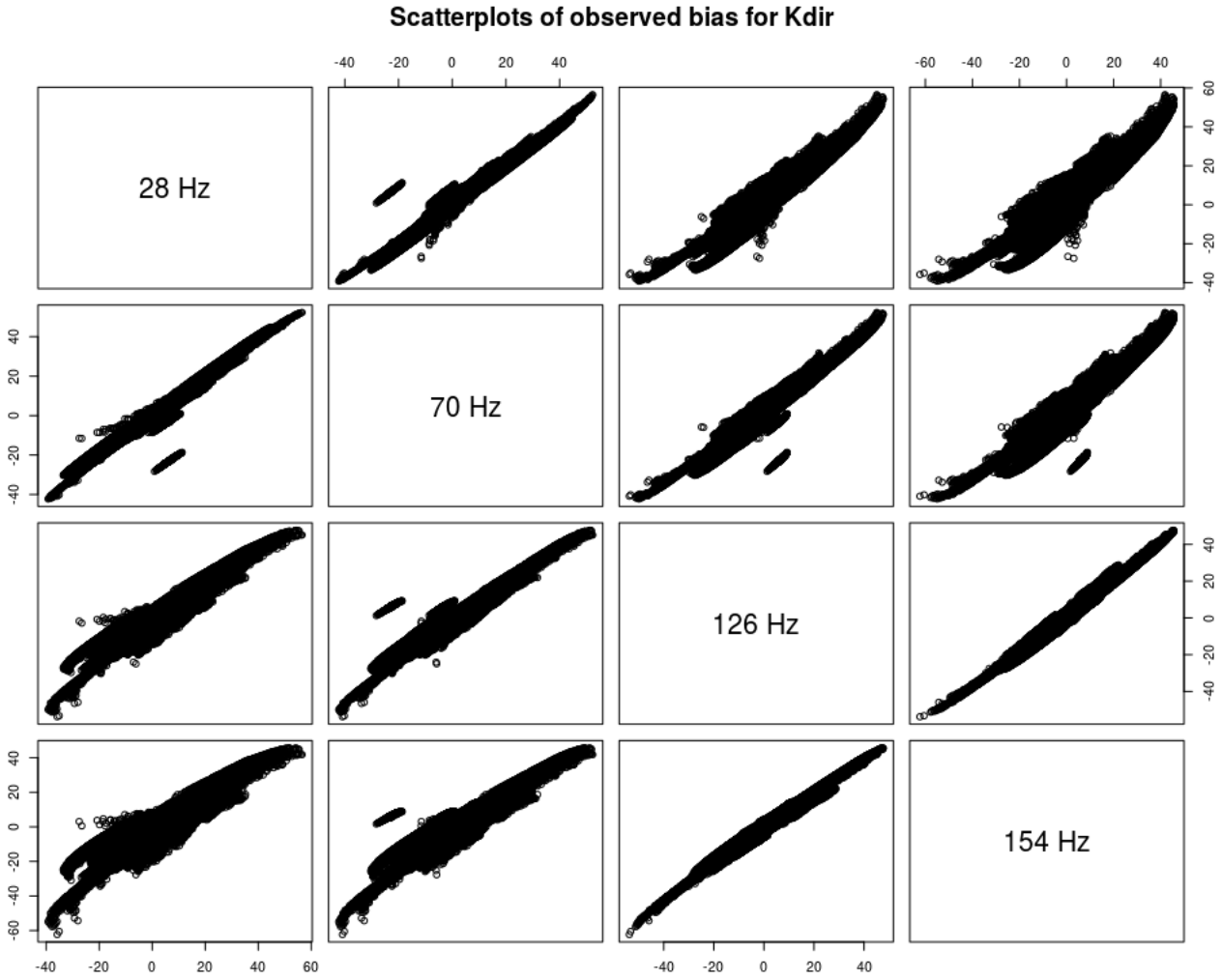


Figure 5.7: Scatterplots of observed discrepancy for direct stiffness. Scatterplots of $N_M = \sum_{i=1}^{N_F} n_i = 286,282$ observed discrepancy between on-site simulation and physical observation for direct stiffness K_{direct} at frequencies 28 Hz, 70 Hz, 126 Hz, and 154 Hz.

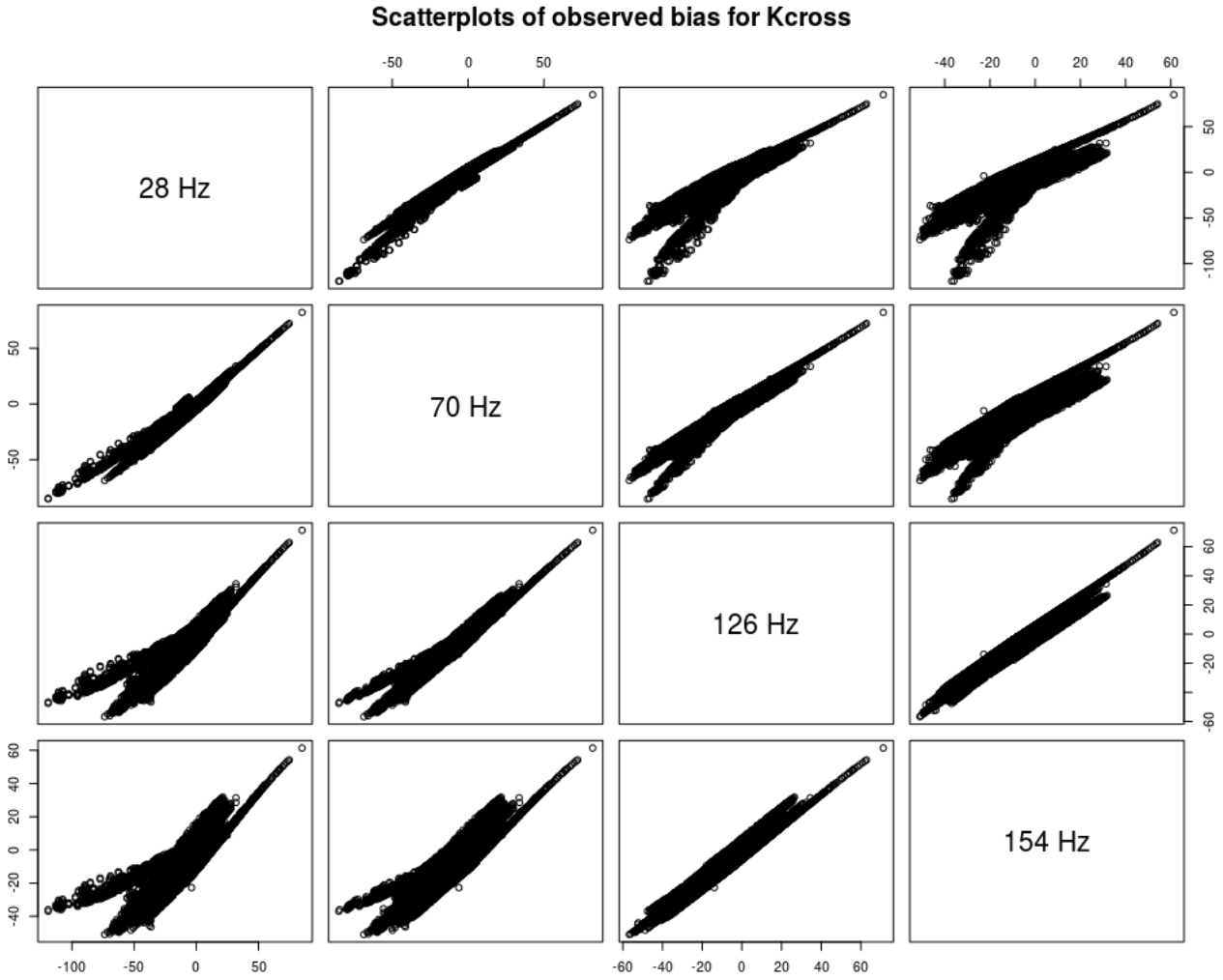


Figure 5.8: Scatterplots of observed discrepancy for cross stiffness. Scatterplots of $N_M = \sum_{i=1}^{N_F} n_i = 286,282$ observed discrepancy between on-site simulation and physical observation for cross stiffness k_{cross} at frequencies 28 Hz, 70 Hz, 126 Hz, and 154 Hz.

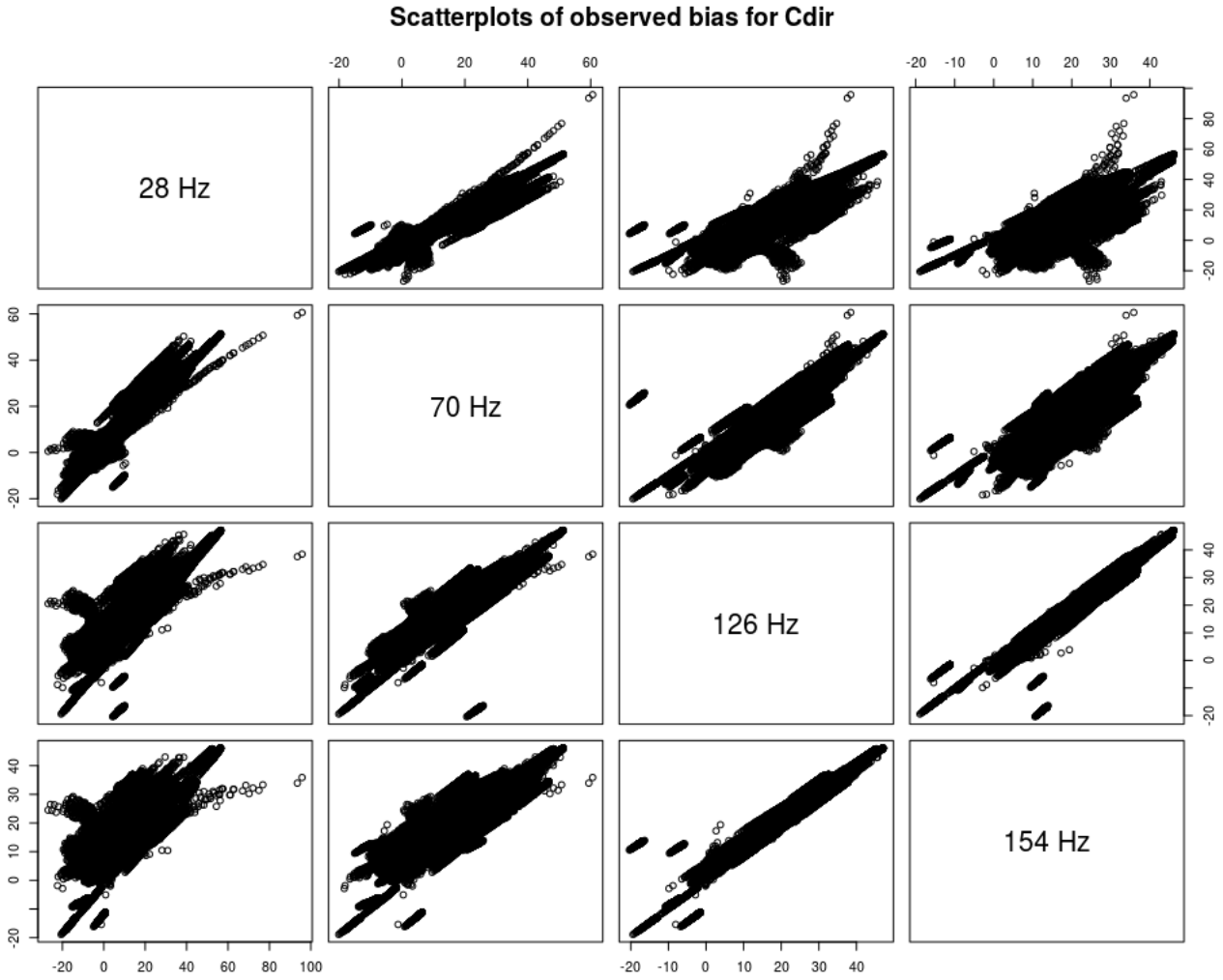


Figure 5.9: Scatterplots of observed discrepancy for direct damping. Scatterplots of $N_M = \sum_{i=1}^{N_F} n_i = 286,282$ observed discrepancy between on-site simulation and physical observation for direct damping C_{direct} at frequencies 28 Hz, 70 Hz, 126 Hz, and 154 Hz.

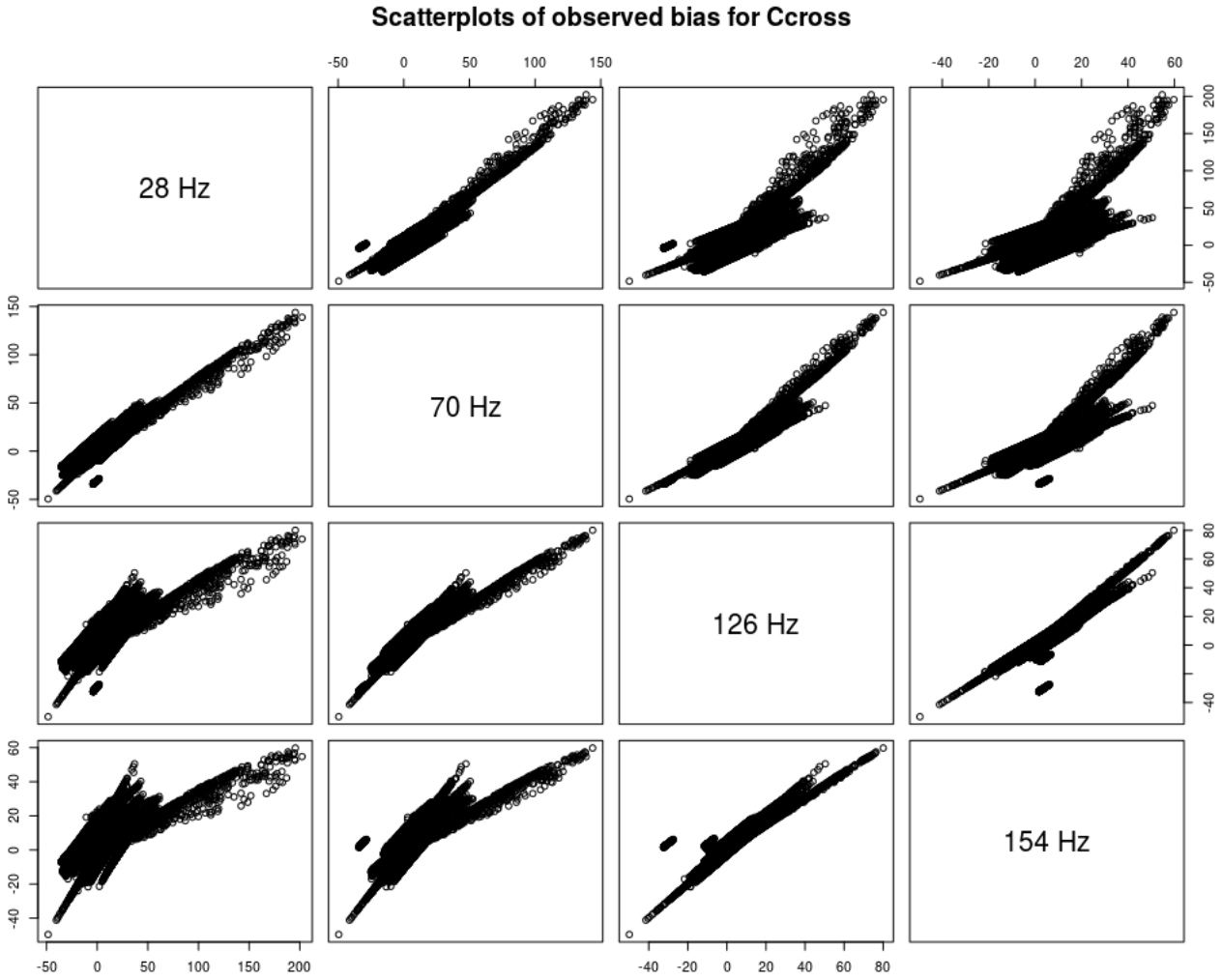


Figure 5.10: Scatterplots of observed discrepancy for cross damping. Scatterplots of $N_M = \sum_{i=1}^{N_F} n_i = 286,282$ observed discrepancy between on-site simulation and physical observation for cross damping c_{cross} at frequencies 28 Hz, 70 Hz, 126 Hz, and 154 Hz.

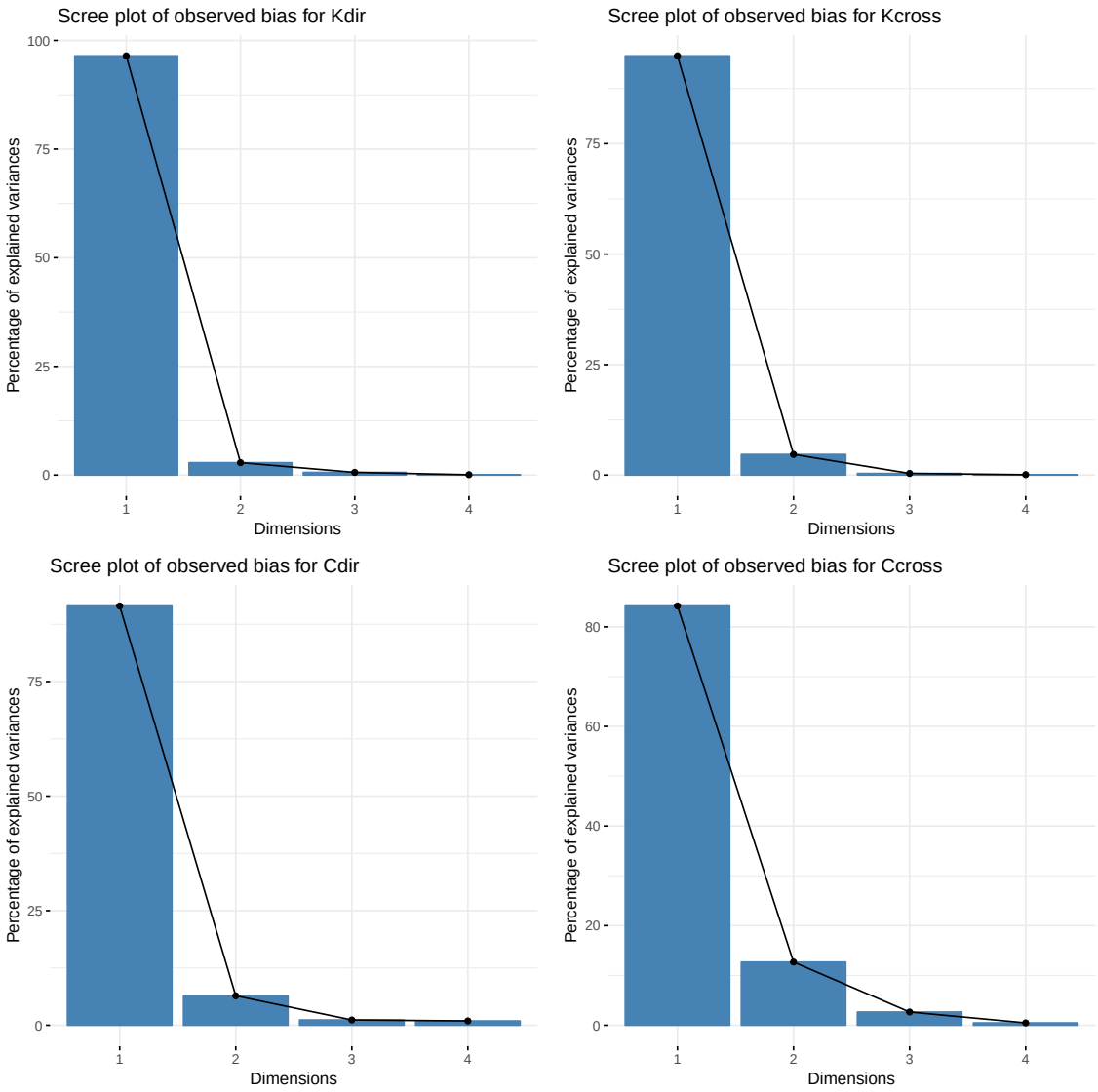


Figure 5.11: Scree plots of principal components of different frequencies. Scree plots of principal components of different frequencies 28 Hz, 70 Hz, 126 Hz, and 154 Hz for direct/cross stiffness and damping: K_{direct}, k_{cross}, C_{direct}, c_{cross}. A clear first principal component exists for all four outputs: the first dimension of principal components can represent 96.46% of K_{direct}, 94.89% of k_{cross}, 91.49% of C_{direct}, and 84.19% of c_{cross} variances.

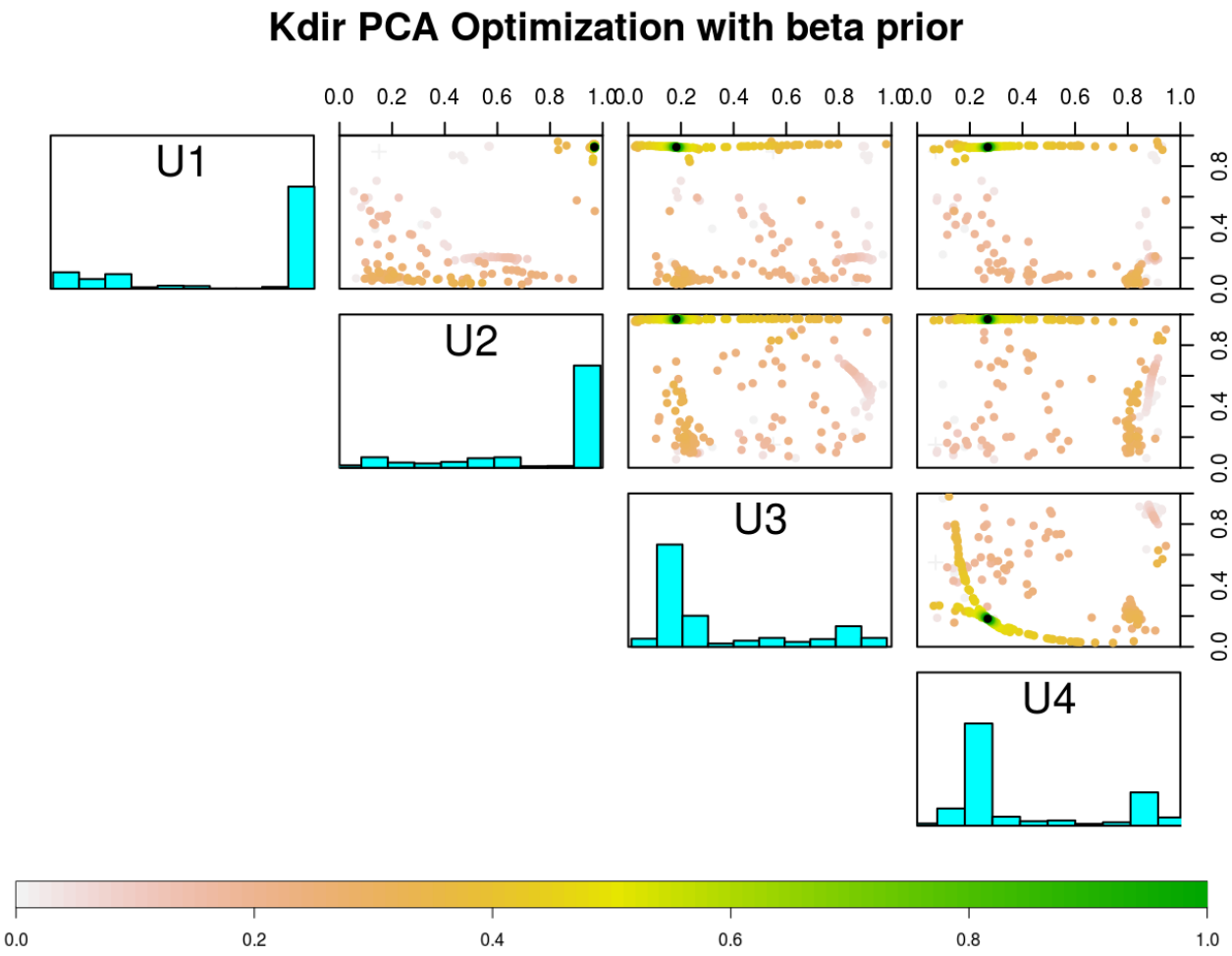


Figure 5.12: PC calibration results for honeycomb on K_{direct} under Beta prior. Optimization calibration on $\mathbf{u}$ using OSSs through principal-component representation on observed discrepancy of four frequencies for direct stiffness K_{direct} under Beta prior. Colors are derived from ranks of posterior probabilities to aid in visualization. The results are from 500 converged optimization under random initialization. Black dots indicate MAP values.

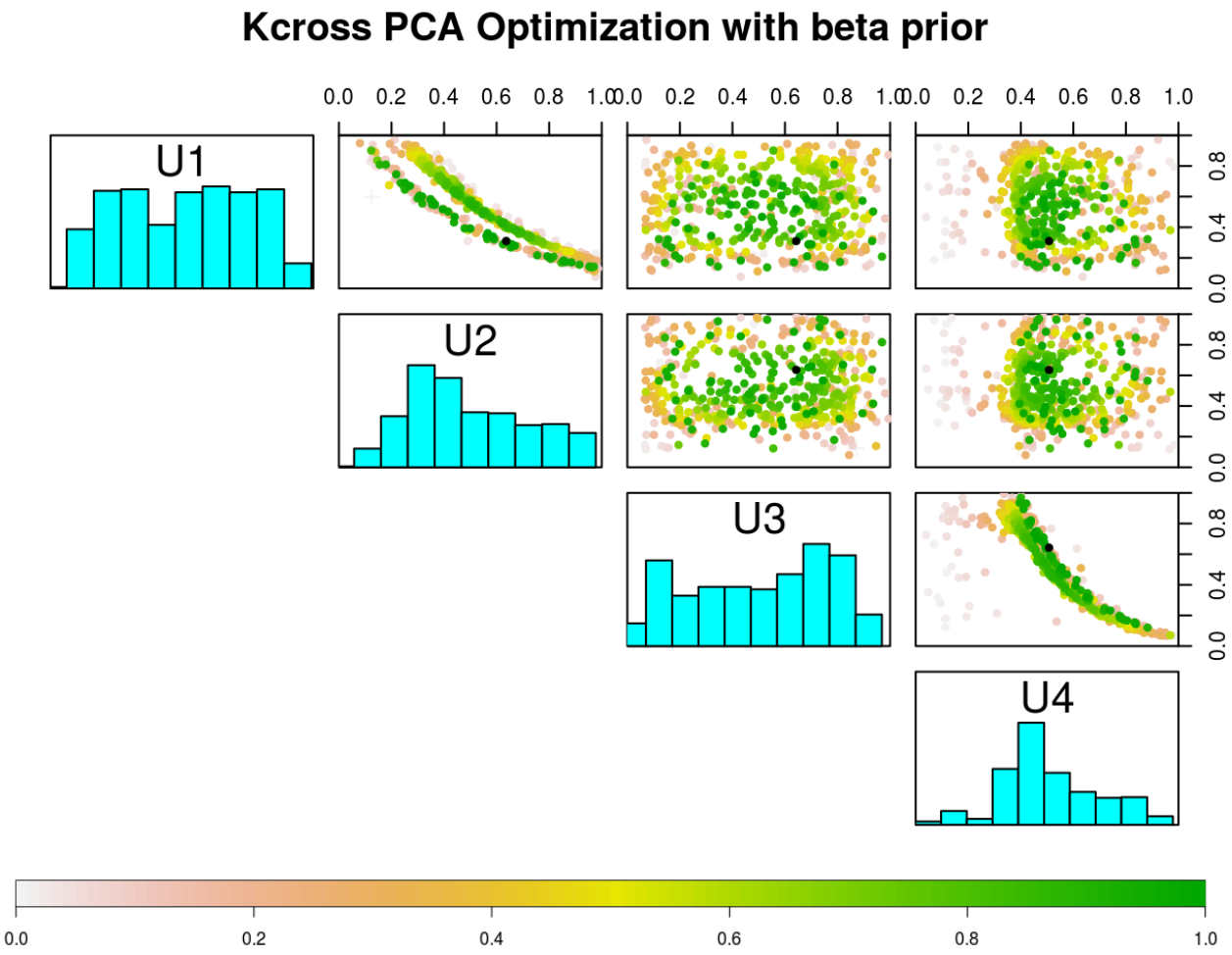


Figure 5.13: PC calibration results for honeycomb on k_{cross} under Beta prior. Optimization calibration on $\mathbf{u}$ using OSSs through principal-component representation on observed discrepancy of four frequencies for cross stiffness k_{cross} under Beta prior. Colors are derived from ranks of posterior probabilities to aid in visualization. The results are from 500 converged optimization under random initialization. Black dots indicate MAP values.

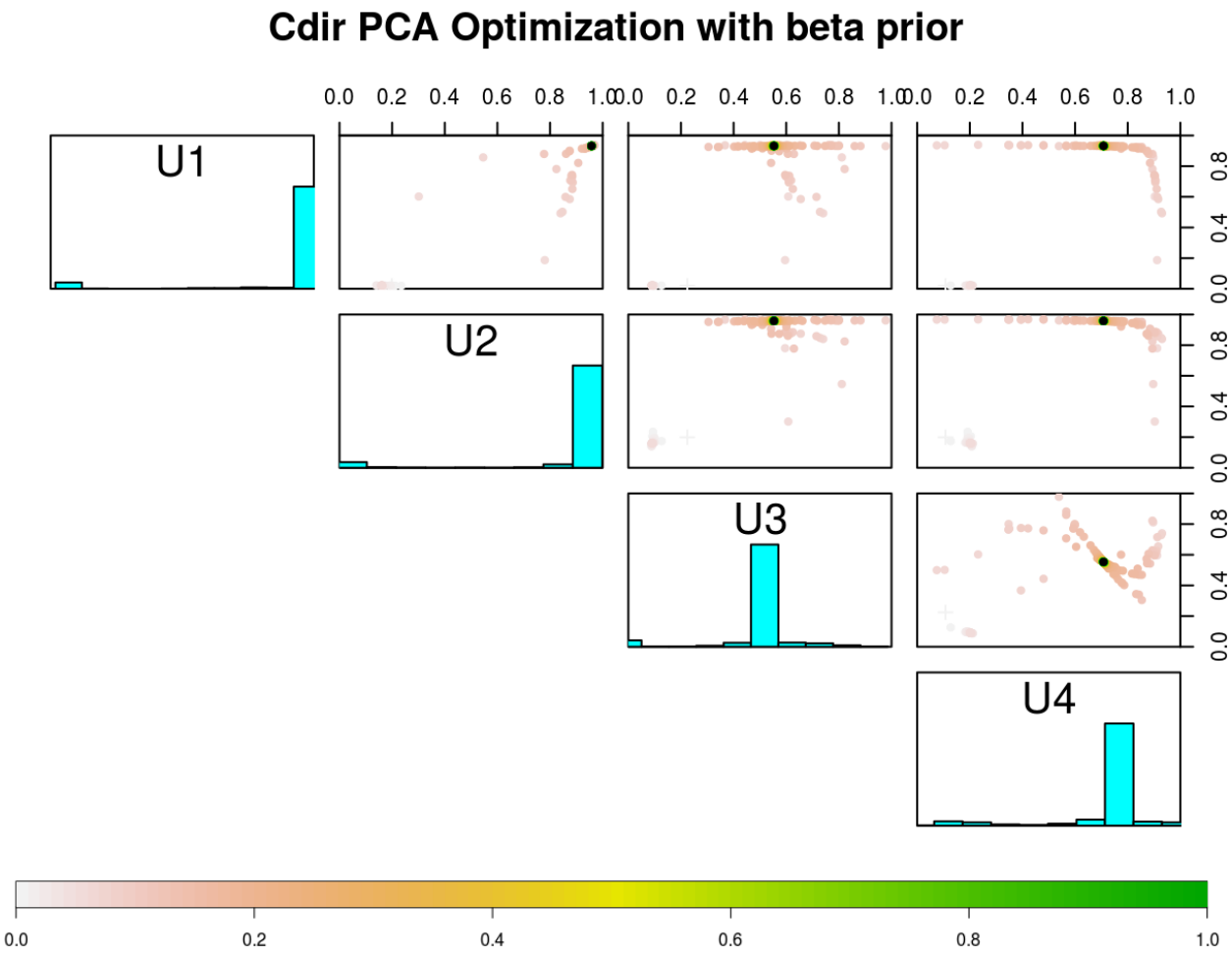


Figure 5.14: PC calibration results for honeycomb on C_{direct} under Beta prior. Optimization calibration on $\mathbf{u}$ using OSSs through principal-component representation on observed discrepancy of four frequencies for direct damping C_{direct} under Beta prior. Colors are derived from ranks of posterior probabilities to aid in visualization. The results are from 500 converged optimization under random initialization. Black dots indicate MAP values.

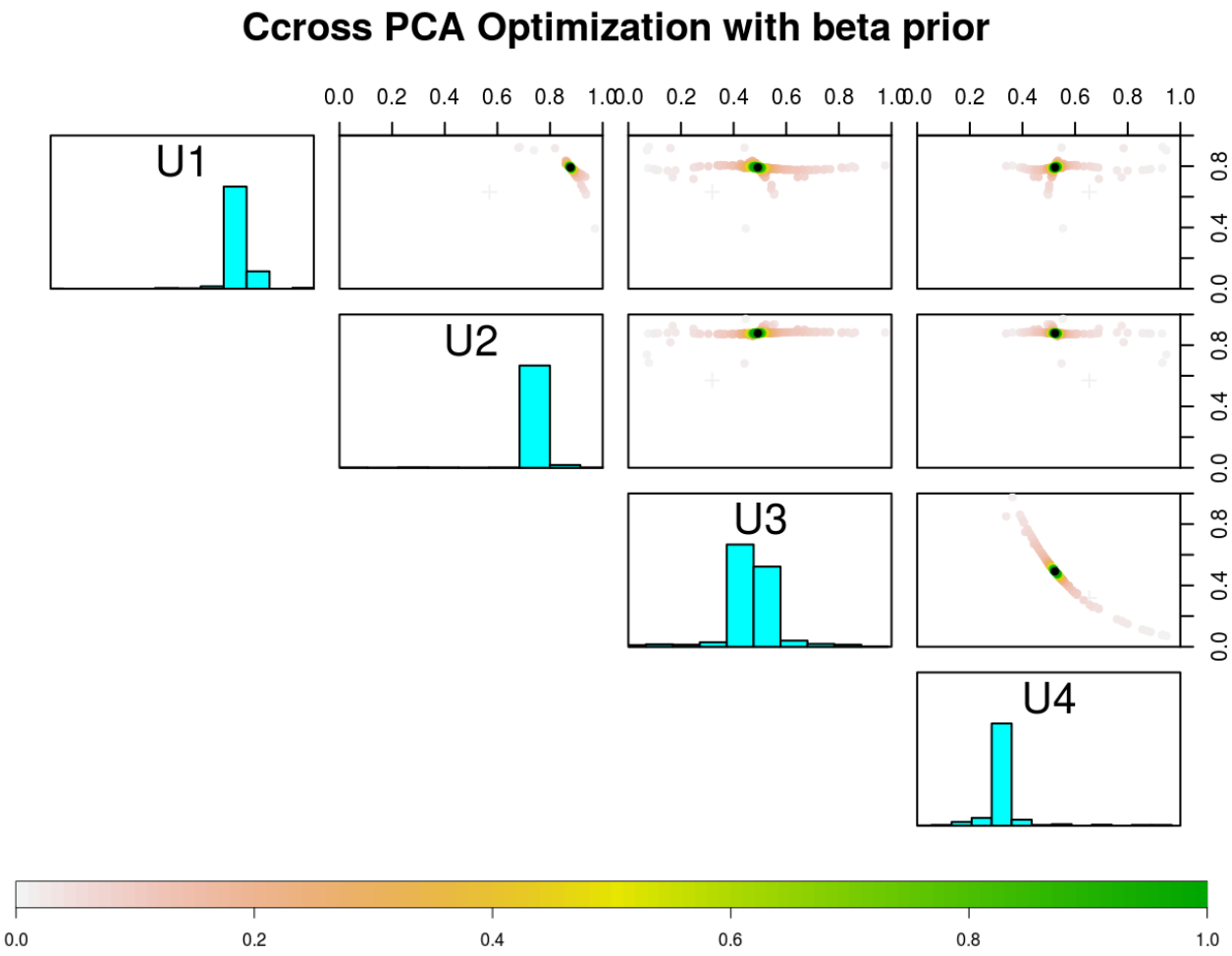


Figure 5.15: PC calibration results for honeycomb on c_{cross} under Beta prior. Optimization calibration on $\mathbf{u}$ using OSSs through principal-component representation on observed discrepancy of four frequencies for cross damping c_{cross} under Beta prior. Colors are derived from ranks of posterior probabilities to aid in visualization. The results are from 500 converged optimization under random initialization. Black dots indicate MAP values.

Chapter 6

Conclusion and future directions

6.1 Conclusion

In this dissertation, we have presented several new statistical methods on sequential learning for heteroskedastic Gaussian processes, on-site calibration and uncertainty quantification for large-scale computer experiments. In Chapter 3, we presented an optimal look-ahead based sequential learning strategy for stochastic simulation experiments, balancing replication and exploration to facilitate sequential learning in heteroskedastic systems with input-dependent signal-to-noise ratio. In Chapter 4, we presented highly accurate and computationally efficient divide-and-conquer calibration method based on on-site experimental design and surrogate modeling for large-scale computer model calibration problems, in order to extract key information from massive simulation and to make better prediction for the reality with fully Bayesian posterior uncertainty quantification. This new on-site surrogates

calibration method has been applied to calibrate challenging real-world large-scale computer experiments developed in the oil and gas unit at Bake Hughes, a G.E. company. In Chapter 5, the on-site surrogate calibration method has been extended to multiple output calibration settings. Combing multiple outputs together can improve calibration results and enhance Bayesian learning with further uncertainty reduction.

6.2 Future directions

6.2.1 Input-dependent calibration

One direction for future research is on input-dependent calibration for this honeycomb project. Input-dependent calibration was actually in our initial wishlist for the honeycomb project. We were initially optimistic that we could tackle the issue of input-dependent calibration quickly with ready progress. A careful exploratory analysis on computer model `ISOTSEAL` and honeycomb field data revealed challenges hidden just below the surface: data size (simultaneously too little and too much), dimensionality, computer model reliability, and the nonstationary nature of the dynamics under study. Taken separately, each stretches the limits of the canonical computer model calibration setup, especially in our favored Bayesian setting. Taken all at once, these challenges demanded a fresh perspective.

To address all these challenges for honeycomb, we developed a new Bayesian calibration methodology called on-site surrogate for large-scale calibration described in Chapter 4. Now, with the highly accurate and computationally efficient on-site surrogates calibration

method, we are ready to address the topic of input-dependent calibration for the motivating honeycomb example. Also as indicated in Chapter 5, the output cross stiffness k_{cross} appears to have obvious non-constant calibration parameter settings. It is clear from Figures 5.1 and 5.13 that the best fitted values for output cross stiffness k_{cross} are not unique.

For the honeycomb gas seal, there is both empirical evidence from our BHGE collaborators and theoretical rotordynamics justification from turbomachinery literature, suggesting that the best settings of the calibration parameter are dependent on several important controllable field inputs, see D'Souza and Childs 2002 and Kleynhans and Childs 1997. In particular, there is potential for unknown functional relationships to exist between controllable physical inputs clearance and pressure on both rotoric friction coefficients $n_r = u_3$ and $m_r = u_4$.

For input-dependent calibration, we want to explore different functional relationships between these important physical inputs $\mathbf{x}^*$, such clearance and pressure in the honeycomb seal, and the corresponding input-dependent calibration parameter $\mathbf{u}$. Instead of a constant setting of calibration parameter $\mathbf{u}$ for the entire space of $\mathbf{X}$, we want to learn and fit input-dependent calibration parameters $\mathbf{u}(\mathbf{x}^*)$, which are functions of $\mathbf{x}^*$:

$$\mathbf{u} \longrightarrow \mathbf{u}(\mathbf{x}^*). \tag{6.1}$$

The best fitted input-dependent calibration parameter $\mathbf{u}(\mathbf{x}^*)$ can leverage additional knowledge from related important inputs $\mathbf{x}^*$ to generate more specific and accurate simulation from computer models. Meanwhile, input-dependent calibration parameters can also improve the

flexibility of the simulator and thus decrease the model discrepancy to reality.

Our initial attempt on this input-dependent calibration direction involves a linear function between physical input clearance $\mathbf{x}^*$ and rotoric friction coefficients $n_r = u_3$, $m_r = u_4$:

$$u_3(\mathbf{x}^*) = b_0 + b_1\mathbf{x}^* \quad \text{and} \quad u_4(\mathbf{x}^*) = b_0 + b_1\mathbf{x}^*. \tag{6.2}$$

Figures 6.1 and 6.2 show two sets of optimization solutions of posterior distribution of input-dependent calibration parameters from a linear functional approach. In this case, the calibration parameters are mapped from 4 dimensional $\mathbf{u} = (u_1, u_2, u_3, u_4)^\top$ to 6 dimensional $\mathbf{u}_{new} = (u_1, u_2, b_{0n}, b_{1n}, b_{0m}, b_{1m})^\top$, through the following straightforward linear functional relationship,

$$\mathbf{u} = (u_1, u_2, u_3, u_4)^\top \longrightarrow \mathbf{u}_{new} = (u_1, u_2, b_{0n}, b_{1n}, b_{0m}, b_{1m})^\top, \tag{6.3}$$

implicitly defining $u_3(\mathbf{x}^*)$ and $u_4(\mathbf{x}^*)$ through equations 6.2

For honeycomb, we searched for two sets of initial values using the on-site surrogate optimization approach, first with wider range for $b \in (-5, 5)$ and second with narrower ranges for $b \in (-1, 1)$. Searching from random 500 space-filling initial values in the whole spaces of $\mathbf{u}_{new}$, we get two very similar maximum of a posterior solutions for the transformed calibration parameters $\mathbf{u}_{new}$:

- For OSS optimization search with $b \in (-5, 5)$: $\hat{\mathbf{u}} = (u_1 = .947, u_2 = .981, b_{0n} =$

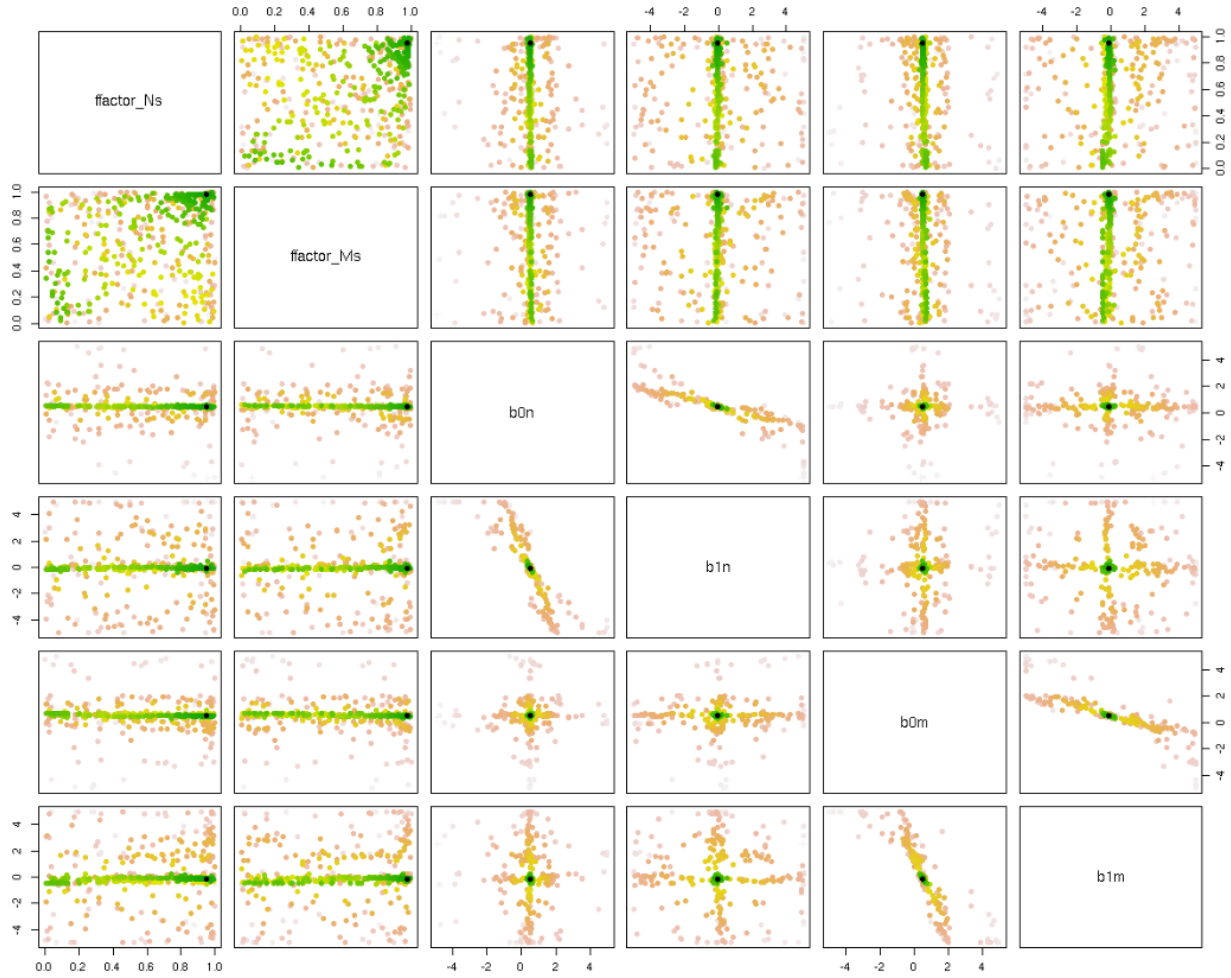


Figure 6.1: Linear functional input-dependent calibration, $b \in (-5, 5)$. Linear functional input-dependent calibration using on-site surrogate optimization approach, searching for $b \in (-5, 5)$: rotoric friction coefficients $n_r = u_3$, $m_r = u_4$ as linear functions of physical input clearance $\mathbf{u}(\mathbf{x}) = b_0 + b_1\mathbf{x}$. Terrain colors are derived from ranks of log-scaled posteriors as a visual aid; black dots indicate the MAP setting: $\hat{\mathbf{u}} = (u_1 = .947, u_2 = .981, b_{0n} = .489, b_{1n} = -.060, b_{0m} = .490, b_{1m} = -.130)^\top$.

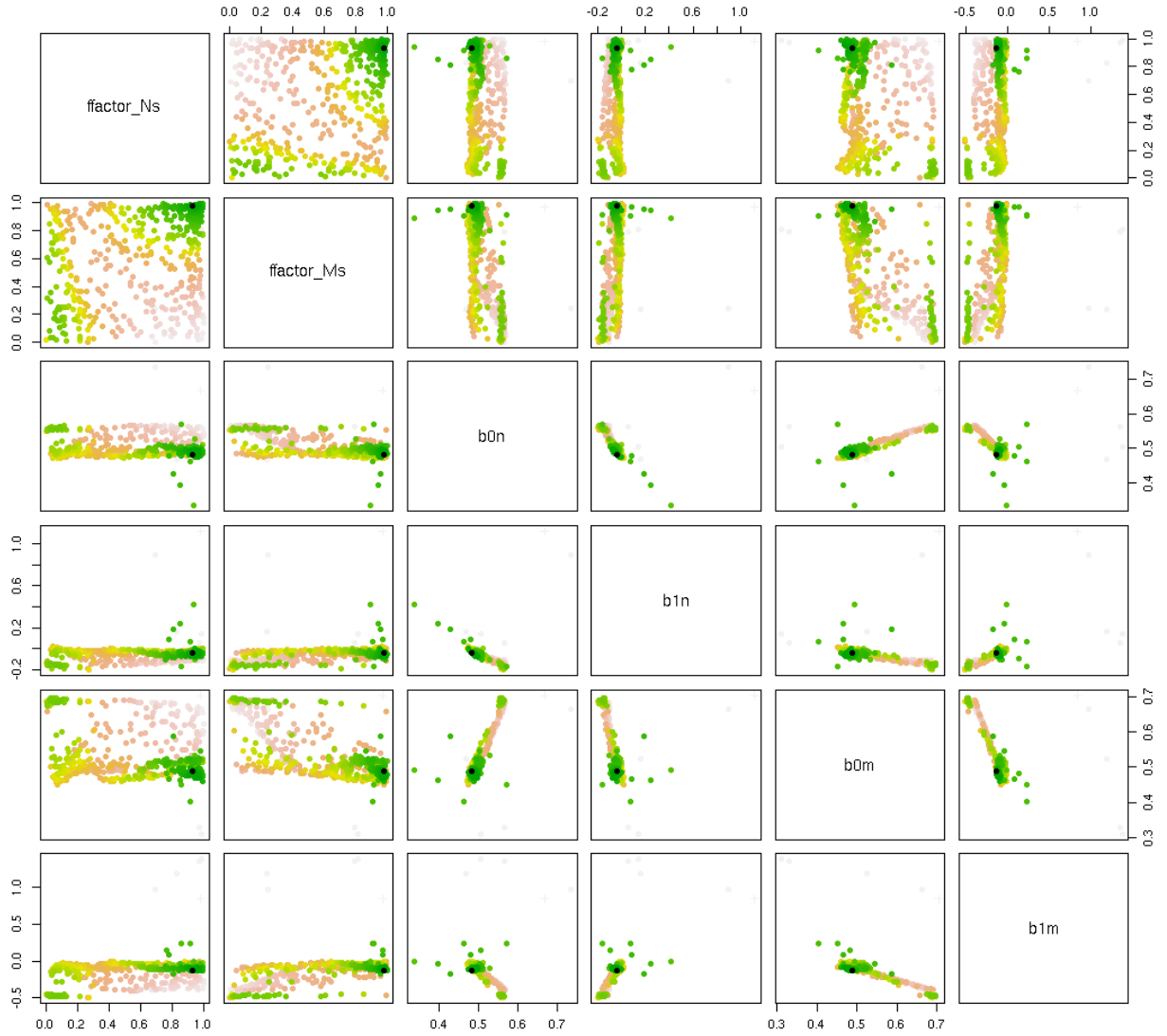


Figure 6.2: Linear functional input-dependent calibration, $b \in (-1, 1)$. Linear functional input-dependent calibration using on-site surrogate optimization approach, searching for $b \in (-1, 1)$: rotoric friction coefficients $n_r = u_3$, $m_r = u_4$ as linear functions of physical input clearance $\mathbf{u}(\mathbf{x}) = b_0 + b_1\mathbf{x}$. Terrain colors are derived from ranks of log-scaled posteriors as a visual aid; black dots indicate the MAP setting: $\hat{\mathbf{u}} = (u_1 = .933, u_2 = .981, b_{0n} = .481, b_{1n} = -.040, b_{0m} = .487, b_{1m} = -.127)^\top$.

$.489, b_{1n} = -.060, b_{0m} = .490, b_{1m} = -.130)^\top$

- For OSS optimization search with $b \in (-1, 1)$: $\hat{\mathbf{u}} = (u_1 = .933, u_2 = .981, b_{0n} = .481, b_{1n} = -.040, b_{0m} = .487, b_{1m} = -.127)^\top$

This linear functional approach is straightforward to implement and can be a good starting point to test the strength and direction of input dependence between calibration parameters $\mathbf{u}$ and important physical inputs $\mathbf{x}^*$. Specifically, the value and sign of slopes b_1 in the corresponding linear models can be informative about this input dependence relationship.

In the next step, we can also further explore more nonparametric functional relationships between calibration parameters $\mathbf{u}$ and important physical inputs $\mathbf{x}^*$. Whether a simple partition-based or linear scheme might be appropriate, or if something more generally nonparametric like Brown and Atamturktur 2018 is required, remains an open question and requires continuing investigation.

6.2.2 Extensions for on-site surrogates

Another direction for future research is to use more advanced tools to further improve the fidelity of these on-site surrogates for calibration presented in Chapters 4 and 5. Despite its many attractive features, including both of its substantially improved accuracy and highly computational efficiency, there is clearly much potential to refine this on-site approach to calibration, in particular the design and modeling behind the on-site surrogates.

While simple maximin distance Latin hypercube samples and stationary Gaussian processes with exponential power kernels with nuggets effect substantially improved the

fidelity of surrogate models for `ISOTSEAL` as shown in Figure 4.5, several simple extensions could be quite powerful. Both other forms of more reasonable experimental design and surrogate modeling strategies can be further implemented under this on-site idea, with the potential to even better capture the on-site features with more uncertainty reduction.

The need for such extensions, along at least one avenue, is perhaps revealed by the final row of Figure 4.6 from Chapter 4. Those plots show bifurcating `ISOTSEAL` runs due to numerical instabilities. These bifurcating `ISOTSEAL` sites actually also match to these outliers in the right panel boxplot in the out-of-sample prediction comparison in Figure 4.5, suggesting there still exist some level of remaining uncertainty of `ISOTSEAL` on these sites.

Although inflated nuggets, as described in Equation 4.4, enable the fitted surrogates to smooth over those regimes, the result contains uniformly high uncertainty for all inputs rather than just near the trouble spot. The reason is that the GP formulation being used is still (locally) stationary. Specifically, the underlying error structure is homoskedastic. Global homoskedastic assumption on the whole error structure across the entire input space can be unrealistic and inappropriate for many modern large-scale simulators with special local features, such as indicated in the motivating `ISOTSEAL` simulator.

Using a heteroskedastic GP model instead (Binois, Gramacy, and Ludkovski 2018), say via `hetGP` on CRAN (Binois and Gramacy 2018), could offer a potential remedy to better separate the signal from input-dependent error structure. As a follow-in work, Chapter 3 showed how designs for effective `hetGP` modeling could be built up sequentially, balancing an appropriate amount of exploration and replication in order to effectively learn signal-to-noise

relationships in the data. Such a sequential `hetGP` approach could represent an attractive alternative to simple space-filling designs, such as maximin LHSs, in **u**-space for further uncertainty reduction in these on-site surrogates.

Bibliography

Abrahamsen, Petter (1997). *A Review of Gaussian Random Fields and Correlation Functions, Second Edition*. Technical Report 917, Norwegian Computing Center. URL: https:http://publications.nr.no/directdownload/917_Rapport.pdf.

Abramson, M.A., C. Audet, G. Couture, J.E. Dennis, Jr., S. Le Digabel, and C. Tribes (2013). *The NOMAD project*. Software available at http://www.gerad.ca/nomad. URL: http://www.gerad.ca/nomad.

Anagnostopoulos, C. and R.B. Gramacy (2013). "Information-Theoretic Data Discarding for Dynamic Trees on Data Streams". In: *Entropy* 15.12. arXiv:1201.5568, pp. 5510–5535.

Ankenman, Bruce, Barry L Nelson, and Jeremy Staum (2010). "Stochastic kriging for simulation metamodeling". In: *Operations research* 58.2, pp. 371–382.

Antognini, Alessandro Baldi and Maroussa Zagoraiou (2010). "Exact optimal designs for computer experiments via Kriging metamodelling". In: *Journal of Statistical Planning and Inference* 140.9, pp. 2607–2617.

Arendt, Paul D, Daniel W Apley, Wei Chen, David Lamb, and David Gorsich (2012). "Improving identifiability in model calibration using multiple responses". In: *Journal of Mechanical Design* 134.10, p. 100909.

Ba, Shan and V. Roshan Joseph (2012). "Composite Gaussian process models for emulating expensive functions". In: *Annals of Applied Statistics* 6.4, pp. 1838–1860.

Bachoc, François (2013). "Cross Validation and Maximum Likelihood estimations of hyper-parameters of Gaussian processes with model misspecification". In: *Computational Statistics & Data Analysis* 66, pp. 55–69.

Barnett, S. (1979). *Matrix Methods for Engineers and Scientists*. McGraw-Hill.

Bastos, LL.S. and A. O'Hagan (2009). "Diagnostics for Gaussian Process Emulators". In: *Technometrics* 51.4, pp. 425–438.

Bayarri, Maria J, James O Berger, Rui Paulo, Jerry Sacks, John A Cafeo, James Cavendish, Chin-Hsu Lin, and Jian Tu (2007a). "A framework for validation of computer models". In: *Technometrics* 49.2, pp. 138–154.

Bayarri, M.J., J. Berger, G. Garcia-Donato, F. Liu, R. Paulo, Jerome Sacks, J Palomo, D. Walsh, J. Cafeo, and R. Parthasarathy (2007b). "Computer Model Validation with Functional Output". In: *Annals of Statistics* 35.5, pp. 1874–1900.

Binois, Mickaël, David Ginsbourger, and Olivier Roustant (2015). "Quantifying uncertainty on Pareto fronts with Gaussian process conditional simulations". In: *European Journal of Operational Research* 243.2, pp. 386–394.

Binois, Mickaël and Robert B. Gramacy (2018). `hetGP`*: Heteroskedastic Gaussian Process Modeling and Design under Replication*. R package version 1.0.3.

Binois, Mickaël, Robert B Gramacy, and Michael Ludkovski (2018). "Practical heteroskedastic Gaussian process modeling for large simulation experiments". In: *Journal of Computational and Graphical Statistics* 27.4, pp. 1–41.

Binois, Mickaël, Jiangeng Huang, Robert B Gramacy, and Michael Ludkovski (2019). "Replication or exploration? Sequential design for stochastic simulation experiments". In: *Technometrics* 61.1, pp. 7–23.

Bornn, L., G. Shaddick, and J. Zidek (2012). "Modelling Nonstationary Processes Through Dimension Expansion". In: *J. of the American Statistical Association* 107.497, pp. 281–289.

Boukouvalas, Alexis (2010). "Emulation of random output simulators". PhD thesis. Aston University.

Boukouvalas, Alexis, Dan Cornford, and M Stehlík (2014). "Optimal design for correlated processes with input-dependent noise". In: *Computational Statistics & Data Analysis* 71, pp. 1088–1102.

Box, George E. P., J. Stuart Hunter, and William Gordon Hunter (2005). *Statistics for experimenters: design, innovation, and discovery*. New York: John Wiley and Sons, Inc.

Box, G.E.P. and D.R. Draper (1987). *Empirical Model-building and Response Surfaces*. New York: John Wiley & Sons, Inc.

Box, G.E.P. and D.R. Draper (2007). *Response Surfaces, Mixtures, and Ridge Analyses, 2nd Edition*. New York: John Wiley & Sons, Inc.

Brown, D. A. and S. Atamturktur (2018). "Nonparametric Functional Calibration of Computer Models". In: *Statistica Sinica* 28, pp. 721–742.

Burnaev, Evgeny and Maxim Panov (2015). "Adaptive design of experiments based on Gaussian processes". In: *Statistical Learning and Data Sciences*. Springer, pp. 116–125.

Byrd, R.H., P. Qiu, J. Nocedal, and C. Zhu (1995). "A Limited Memory Algorithm for Bound Constrained Optimization". In: *Journal on Scientific Computing* 16.5, pp. 1190–1208.

Carnell, Rob (2018). `lhs`*: Latin Hypercube Samples*. `R` package version 0.16. URL: `https://CRAN.R-project.org/package=lhs`.

Chen, Xi and Qiang Zhou (2014). "Sequential experimental designs for stochastic kriging". In: *Proceedings of the 2014 Winter Simulation Conference*. IEEE Press, pp. 3821–3832.

— (2017). "Sequential design strategies for mean response surface metamodeling via stochastic kriging with adaptive exploration and exploitation". In: *European Journal of Operational Research* 262.2, pp. 575–585.

Chevalier, Clément, David Ginsbourger, and Xavier Emery (2014). "Corrected kriging update formulae for batch-sequential data assimilation". In: *Mathematics of Planet Earth*. Springer, pp. 119–122.

Childs, D. (1993). *Turbomachinery Rotordynamics: Phenomena, Modeling, and Analysis*. New York, NY: John Wiley & Sons, Inc.

Cioffi-Revilla, Claudio (2014). *Introduction to computational social science*. Berlin/New York: Springer.

Cohn, D. A. (1994). "Neural network exploration using optimal experimental design". In: *Advances in Neural Information Processing Systems* 6. Morgan Kaufmann Publishers, pp. 679–686.

Cohn, D. A., Z. Ghahramani, and M. I. Jordan (1996). "Active Learning with Statistical Models". In: *Journal of Artificial Intelligence Research* 4, pp. 129–145.

Cressie, Noel (1988). "Variogram". In: *Encyclopedia of Statistical Sciences*.

— (1993). *Statistics for Spatial Data*. New York: John Wiley and Sons, Inc.

Das, Abhimanyu and David Kempe (2008). "Algorithms for subset selection in linear regression". In: *Proceedings of the fortieth annual ACM symposium on Theory of computing*. ACM, pp. 45–54.

D'Souza, Rohan J. and Dara W. Childs (2002). "A Comparison of Rotordynamic-Coefficient Predictions for Annular Honeycomb Gas Seals Using Three Different Friction-Factor Models". In: *Journal of Tribology* 124.3, pp. 524–529.

Fang, Kai-Tai, Runze Li, and Agus Sudjianto (2006). *Design and Modeling for Computer Experiments*. Boca Raton, FL: Chapman & Hall/CPC.

Forrester, Alexander, Andrs Sobester, and Andy Keane (2008). *Engineering Design via Surrogate Modelling: A Practical Guide*. Wiley.

Franco, Jessica, Delphine Dupu, Olivier Roustant, Patrice Kiener, Guillaume Damblin, and Bertrand Iooss (2018). `DiceDesign`*: Space-Filling Designs and Uniformity Criteria*. `R` package version 1.8. URL: `https://cran.r-project.org/web/packages/DiceDesign/`.

Frazier, Peter, Warren Powell, and Savas Dayanik (2009). "The Knowledge-Gradient Policy forCorrelated Normal Beliefs". In: *INFORMS Journal on Computing* 21.4, pp. 599–613.

Gauthier, Bertrand and Luc Pronzato (2014). "Spectral approximation of the IMSE criterion for optimal designs in kernel-based interpolation models". In: *SIAM/ASA Journal on Uncertainty Quantification* 2.1, pp. 805–825.

Gelfand, Alan E. and Adrian F. M. Smith (1990). "Sampling-Based Approaches to Calculating Marginal Densities". In: *Journal of the American Statistical Association* 85.410, pp. 398–409.

Genton, Marc G. (2001). "Classes of Kernels for Machine Learning: a Statistical perspercctive". In: *Journal of Machine Learning Research* 2, pp. 299–312.

Ginsbourger, David and Rodolphe Le Riche (2010). "Towards Gaussian process-based optimization with finite time horizon". In: *mODa 9–Advances in Model-Oriented Design and Analysis*. Springer, pp. 89–96.

Gneiting, Tilmann and Adrian E Raftery (2007). "Strictly proper scoring rules, prediction, and estimation". In: *Journal of the American Statistical Association* 102.477, pp. 359–378. DOI: `10.1198/016214506000001437`.

Goldberg, Paul W., Christopher K.I. Williams, and Christopher M. Bishop (1998). "Regression with input-dependent noise: A Gaussian process treatment". In: *Advances in Neural Information Processing Systems*. Vol. 10. Cambridge, MA: MIT press, pp. 493–499.

Gonzalez, Javier, Michael Osborne, and Neil Lawrence (2016). "GLASSES: Relieving The Myopia Of Bayesian Optimisation". In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pp. 790–799.

Gorodetsky, Alex and Youssef Marzouk (2016). "Mercer kernels and integrated variance experimental design: connections between Gaussian process regression and polynomial approximation". In: *SIAM/ASA Journal on Uncertainty Quantification* 4.1, pp. 796–828.

Gramacy, R. B. and D. W. Apley (2015). "Local Gaussian process approximation for large computer experiments". In: *Journal of Computational and Graphical Statistics* 24.2, pp. 561–578.

Gramacy, R.B. and H.K.H. Lee (2012). "Cases for the nugget in modeling computer experiments". In: *Statistics and Computing* 22.3, pp. 713–722. DOI: 10.1007/s11222-010-9224-x. URL: http://dx.doi.org/10.1007/s11222-010-9224-x.

Gramacy, R.B. and N.G. Polson (2011). "Particle Learning of Gaussian Process Models for Sequential Design and Optimization". In: *Journal of Computational and Graphical Statistics* 20.1, pp. 102–118. DOI: 10.1198/jcgs.2010.09171.

Gramacy, Robert (2016). "laGP: Large-Scale Spatial Modeling via Local Approximate Gaussian Processes in R". In: *Journal of Statistical Software, Articles* 72.1, pp. 1–46. ISSN: 1548-7660. DOI: 10.18637/jss.v072.i01. URL: https://www.jstatsoft.org/v072/i01.

Gramacy, Robert B. and Herbert K. H. Lee (2008). "Bayesian treed Gaussian process models with an application to computer modeling". In: *Journal of the American Statistical Association* 103, pp. 1119–1130. URL: http://arxiv.org/abs/0710.4536.

— (2009). "Adaptive Design and Analysisof Supercomputer Experiments". In: *Technometrics* 51.2, pp. 130–145.

— (2010). "Optimization Under Unknown Constraints". In: *Valencia discussion paper, in Bayesian Statistics 9. Oxford University Press*. preprint on arXiv:1004.4027.

Gramacy, Robert B., Jarad Niemi, and Robin M. Weiss (2014). "Massively Parallel Approximate Gaussian Process Regression". In: *SIAM/ASA J. Uncertainty Quantification* 2.1, pp. 564–584.

Gramacy, Robert B. and Furong Sun (2018). laGP*: Local approximate Gaussian process regression*. R package version 1.5-2. URL: http://bobby.gramacy.com/r_packages/laGP.html.

Gramacy, Robert B. and Matt A. Taddy (2016). tgp*: Bayesian Treed Gaussian Process Models*. R package version 2.4-14.

Gramacy, Robert B., Derek Bingham, James Paul Holloway, Michael J. Grosskopf, Carolyn C. Kuranz, Erica Rutter, Matt Trantham, and R. Paul Drake (2015). "Calibrating a

large computer experiment simulating radiative shock hydrodynamics". In: *Annals of Applied Statistics* 9.3, pp. 1141–1168.

Gramacy, Robert B., Genetha Gray, A. Sbastien Le Digabel, Herbert K. H. Lee, Pritam Ranjan, Garth Wells, and Stefan M. Wild (2016). "Modeling an Augmented Lagrangian for Blackbox Constrained Optimization". In: *Technometrics* 58.1, pp. 1–11.

Gul, Evren, V. Roshan Joseph, Huang Yan, and Shreyes N. Melkote (2018). "Uncertainty quantification of machining simulations using an in situ emulator". In: *Journal of Quality Technology* 50.3, pp. 253–261.

Hastings, W. K. (1970). "Monte Carlo Sampling Methods Using Markov Chains and Their Applications". In: *Biometrika* 57.1, pp. 97–109.

Higdon, D., J. Swall, and J. Kern (1999). *Non-stationary spatial modeling.* Bayesian Statistics 6(eds J. M.Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith) Oxford: Clarendon, pp. 743–758.

Higdon, Dave, Marc Kennedy, James C. Cavendish, John A. Cafeo, and Robert D. Ryne (2004). "Combining field data and computer simulations for calibration and prediction". In: *SIAM Journal on Scientific Computing* 26.2, pp. 448–466.

Higdon, Dave, James Gattiker, Brian Williams, and Maria Rightley (2008). "Computer model calibration using high-dimensional output". In: *Journal of the American Statistical Association* 103.482, pp. 570–583.

Hirs, G. G. (1973). "A Bulk-Flow Theory for Turbulence in Lubricant Films". In: *Journal of Lubrication Technology* 95.2, pp. 137–145.

Hoff, Peter D (2009). *A first course in Bayesian statistical methods*. Springer Science & Business Media.

Hong, L.J. and B.L. Nelson (2006). "Discrete optimization via simulation using COMPASS". In: *Operations Research* 54.1, pp. 115–129.

Horn, Daniel, Melanie Dagge, Xudong Sun, and Bernd Bischl (2017). "First Investigations on Noisy Model-Based Multi-objective Optimization". In: *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, pp. 298–313.

Huan, Xun and Youssef M Marzouk (2016). "Sequential Bayesian optimal experimental design via approximate dynamic programming". In: *arXiv preprint arXiv:1604.08320*.

Huang, Jiangeng, Robert B. Gramacy, Mickaël Binois, and Mirko Libraschi (2018). "On-site surrogates for large-scale calibration". In: preprint on arXiv:1810.01903.

Iman, R. L. and W. J. Conover (1980). "Small sample sensitivity analysis techniques for computer models, with an application to risk assessment (with discussions)". In: *Comunication in Statistics, theory and methods* 9, pp. 1749–1874.

Jalali, Hamed, Inneke Van Nieuwenhuyse, and Victor Picheny (2017). "Comparison of Kriging-based algorithms for simulation optimization with heterogeneous noise". In: *European Journal of Operational Research* 261.1, pp. 279 –301.

Johnson, L.R. (2008). "Microcolony and Biofilm Formation as a Survival Strategy for Bacteria". In: *Journal of Theoretical Biology* 251, pp. 24–34.

Johnson, M.E., L.M. Moore, and D. Ylvisaker (1990). "Minimax and maximin distance designs". In: *Journal of Statistical Planning and Inference* 26.2, pp. 143–151.

Jones, Donald R., Matthias Schonlau, and William J. Welch (1998). "Efficient Global Optimization of ExpensiveBlack-Box Functions". In: *Journal of Global Optimization* 13.4, pp. 455–492.

Joseph, V. Roshan, Evren Gul, and Shan Ba (2015). "Maximum projection designs for computer experiments". In: *Biometrika* 102.2, pp. 371–380.

Kamiński, Bogumił (2015). "A method for the updating of stochastic Kriging metamodels". In: *European Journal of Operational Research* 247.3, pp. 859–866.

Kass, Robert E., Bradley P. Carlin, Andrew Gelman, and Radford M. Neal (1998). "Markov Chain Monte Carlo in Practice: A Roundtable Discussion". In: *The American Statistician* 52.2, pp. 93–100.

Kennedy, Marc C. and Anthony O'Hagan (2001a). "Bayesian calibration of computer models (with discussion)". In: *Journal of the Royal Statistical Society, Series B* 63.3, pp. 425–464.

— (2001b). "Supplementary details on Bayesian calibration of computer models". In: *Technical report, University of Sheffield*. URL: `ttp://www.tonyohagan.co.uk/academic/ps/calsup.pdf`.

Kersting, Kristian, Christian Plagemann, Patrick Pfaff, and Wolfram Burgard (2007). "Most likely heteroscedastic Gaussian process regression". In: *Proceedings of the International Conference on Machine Learning*. New York, NY: ACM, pp. 393–400.

Kim, Hyoung-Moon, Bani K Mallick, and C. C Holmes (2005). "Analyzing Nonstationary Spatial Data Using Piecewise Gaussian Processes". In: *Journal of the American Statistical Association* 100.470, pp. 653–668.

Kleijnen, Jack PC (2015). *Design and Analysis of Simulation Experiments*. Vol. 230. Springer.

Kleynhans, G. and D. Childs (1997). "The Acoustic Influence of Cell Depth on the Rotordynamic Characteristics of Smooth-Rotor/Honeycomb-Stator Annular Gas Seals". In: *ASME Journal of Engineering for Gas Turbines and Power*, pp. 949–957.

Krause, Andreas and Carlos Guestrin (2007). "Nonmyopic active learning of gaussian processes: an exploration-exploitation approach". In: *Proceedings of the 24th international conference on Machine learning*. ACM, pp. 449–456.

Krause, Andreas, Ajit Singh, and Carlos Guestrin (2008). "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies". In: *Journal of Machine Learning Research* 9.Feb, pp. 235–284.

Lam, Remi, Karen Willcox, and David H Wolpert (2016). "Bayesian Optimization with a Finite Budget: An Approximate Dynamic Programming Approach". In: *Advances In Neural Information Processing Systems*, pp. 883–891.

Law, Averill M. (2015). *Simulation Modeling and Analysis*. 5th ed. McGraw-Hill.

Le Digabel, S. (2011). "Algorithm 909: NOMAD: Nonlinear Optimization with the MADS algorithm". In: *ACM Transactions on Mathematical Software* 37.4, 44:1–44:15. DOI: 10.1145/1916461.1916468. URL: http://dx.doi.org/10.1145/1916461.1916468.

Leatherman, Erin R, Thomas J Santner, and Angela M Dean (2017). "Computer experiment designs for accurate prediction". In: *Statistics and Computing*, pp. 1–13.

Letham, Benjamin and Eytan Bakshy (2019). "Bayesian Optimization for Policy Search via Online-Offline Experimentation". In: *Journal of Machine Learning Research*. URL: arXiv:1904.01049.

Letham, Benjamin, Brian Karrer, Guilherme Ottoni, and Eytan Bakshy (2019). "Constrained Bayesian Optimization with Noisy Experiments". In: *Bayesian Analysis* 14.2, pp. 495–519.

Liu, F., M.J. Bayarri, and J.O. Berger (2009). "Modularization in Bayesian analysis, with emphasis on analysis of computer models". In: *Bayesian Analysis* 4.1, pp. 119–150.

Liu, Ming and Jeremy Staum (2010). "Stochastic kriging for efficient nested simulation of expected shortfall". In: *The Journal of Risk* 12.3, p. 3.

MacKay, D. J. C. (1992). "Information-Based Objective Functions for Active Data Selection". In: *Neural Computation* 4.4, pp. 590–604.

Matheron, G. (1963). "Principles of geostatistics". In: *Economic Geology* 58, pp. 1246–1266.

McKay, M. D., R. J. Beckman, and W. J. Conover (1979). "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code". In: *Technometrics* 21.2, pp. 239–245.

Mehdad, Ehsan and Jack P.C. Kleijnen (2018). "Stochastic intrinsic Kriging for simulation metamodeling". In: *Applied Stochastic Models in Business and Industry*, in press. ISSN: 1526-4025.

Morris, Max D. and Toby J. Mitchell (1995). "Exploratory designs for computational experiments". In: *Journal of Statistical Planning and Inference* 43, pp. 381–402.

Müller, Werner G, Luc Pronzato, and Helmut Waldl (2012). "Relations between designs for prediction and estimation in random fields: an illustrative case". In: *Advances and Challenges in Space-time Modelling of Natural Events*. Springer, pp. 125–139.

Myers, Raymond H, Douglas C. Montgomery, and Christine M. Anderson-Cook (2016). *Response Surface Methodology, fourth edition*. New York: John Wiley & Sons, Inc.

Nash, John C. (2016). `nlmrt`*: Functions for Nonlinear Least Squares Solutions*. R package version 2016.3.2. URL: `https://cran.r-project.org/web/packages/nlmrt/`.

Neal, Radford M. (1997). *Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification*. Technical Report No.9702, Department of Statistics, University of Toronto.

O'Hagan, Anthony (2006). "Bayesian analysis of computer code outputs: A tutorial". In: *Reliability Engineering & System Safety* 91.10–11, pp. 1290–1300.

Paulo, Rui, Gonzalo Garcia-Donato, and Jesus Palomo (2012). "Calibration of computer models with multivariate output". In: *Computational Statistics and Data Analysis* 56.12, pp. 3959–3974.

Petersen, Kaare Brandt and Michael Syskind Pedersen (2008). "The matrix cookbook". In: *Technical University of Denmark* 7, p. 15.

Picheny, Victor and David Ginsbourger (2013). "A nonstationary space-time Gaussian process model for partially converged simulations". In: *SIAM/ASA Journal on Uncertainty Quantification* 1, pp. 57–78.

Picheny, Victor, Robert B. Gramacy, Stefan Wild, and Sebastien Le Digabel (2016). "Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian". In: *Advances in on Neural Information Processing Systems (NIPS)* 29, pp. 1435–1443.

Plumlee, Matthew (2017). "Bayesian calibration of inexact computer models". In: *Journal of the American Statistical Association* 112.519, pp. 1274–1285.

Plumlee, Matthew and Rui Tuo (2014). "Building accurate emulators for stochastic simulations via quantile Kriging". In: *Technometrics* 56.4, pp. 466–473.

Pratola, Matthew T, Ofir Harari, Derek Bingham, and Gwenn E Flowers (2017). "Design and Analysis of Experiments on Nonconvex Regions". In: *Technometrics*, pp. 1–12.

Pronzato, Luc and Werner G Müller (2012). "Design of computer experiments: space filling and beyond". In: *Statistics and Computing* 22.3, pp. 681–701.

Qian, P. Z. G. (2009). "Nested Latin hypercube designs". In: *Biometrika* 96.4, pp. 957–970.

— (2012). "Sliced Latin Hypercube Designs". In: *Journal of the American Statistical Association* 107.497, pp. 393–399.

Qian, P. Z. G., M. Ai, and C. F. J. Wu (2009). "Construction of Nested Space-Filling Designs". In: *Annals of Statistics* 37.6A, pp. 3616–3643.

Qian, P. Z. G. and C. F. J. Wu (2009). "Sliced Space-Filling Designs". In: *Biometrika* 96.4, pp. 945–956.

Quan, Ning, Jun Yin, Szu Hui Ng, and Loo Hay Lee (2013). "Simulation optimization via kriging: a sequential search using expected improvement with computing budget constraints". In: *IIE Transactions* 45.7, pp. 763–780.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: `https://www.R-project.org/`.

— (2018). `optim`*: General-purpose Optimization*. R package version 3.6.0. URL: `https://stat.ethz.ch/R-manual/R-devel/library/stats/html/optim.html`.

Rasmussen, Carl E. and Christopher Williams (2006). *Gaussian Processes for Machine Learning*. MIT Press. URL: `http://www.gaussianprocess.org/gpml/`.

Sacks, Jerome, Susannah Schiller, and William J Welch (1989). "Design for computer experiments". In: *Technometrics* 31.1, pp. 41–47.

Sacks, Jerome, William J Welch, Toby J Mitchell, and Henry P Wynn (1989). "Design and analysis of computer experiments". In: *Statistical science* 4, pp. 409–423.

Santner, Thomas J, Brian J Williams, and William I Notz (2003). *The design and analysis of computer experiments*. Springer Science & Business Media.

— (2018). *The design and analysis of computer experiments, 2nd Edition*. Springer Science & Business Media.

Schmidt, Alexandra M. and Anthony O'Hagan (2003). "Bayesian inference for nonstationary spatial covariance structure via spatial deformations". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 65.3, pp. 743–758.

Seo, S., M. Wallat, T. Graepel, and K. Obermayer (2000a). "Gaussian process regression: active data selection and test point rejection". In: *Proceedings of the International Joint Conference on Neural Networks. IEEE*, pp. 241–246.

— (2000b). "Gaussian Process Regression: Active Data Selection and Test Point Rejection". In: *Proceedings of the International Joint Conference on Neural Networks*. Vol. III. IEEE, pp. 241–246.

Shewry, Michael C. and Henry P. Wynn (1987). "Maximum entropy sampling". In: *Journal of Applied Statistics* 14.2, pp. 165–170.

Stein, M. (1987). "Large sample properties of simulations using Latin hypercube sampling". In: *Technometrics* 29, pp. 143–151.

Stein, Michael L. (1999). *Interpolation of Spatial Data*. New York: Springer.

Sun, Furong, Robert B. Gramacy, Benjamin Haaland, Earl Lawrence, and Andrew Walker (2019a). "Emulating satellite drag from large simulation experiments". In: *SIAM/ASA J. Uncertainty Quantification* 7.2. arXiv:1712.00182.

Sun, Furong, Robert B. Gramacy, Benjamin Haaland, Siyuan Lu, and Youngdeok Hwang (2019b). "Synthesizing simulation and field data of solar irradiance". In: *Statistical Analysis and Data Mining*. in press, arXiv:1806.05131.

Tuo, Rui and C. F. Jeff Wu (2015). "Efficient calibration for imperfect computer models". In: *The Annals of Statistics* 43.6, pp. 2331–2352.

— (2016). "A Theoretical Framework for Calibration in Computer Models: Parameterization, Estimation and Convergence Properties". In: *Journal of Uncertainty Quantification* 4, pp. 767–795.

Vannarsdall, Michael Lloyd (2011). "Measured Results for a New Hole-Pattern Annular Gas Seal Incorporating Larger Diameter Holes, Comparisons to Results for a Traditional Hole-Pattern Seal and Predictions". Available electronically from http://hdl.handle.net/1969.1/E TAMU-2011-08-9759. Texas A&M University.

Ver Hoef, J. and R. P. Barry (1998). "Constructing and Fitting Models for Cokriging and Multivariate Spatial Prediction". In: *Journal of Statistical Planning and Inference* 69, pp. 275–294.

Wackernagel, Hans (1998). *Multivariate Geostatistics*. New York: Springer.

Wang, Wenjia and Benjamin Haaland (2017). "Controlling Sources of Inaccuracy in Stochastic Kriging". In: *arXiv preprint arXiv:1706.00886*.

Weaver, Brian P, Brian J Williams, Christine M Anderson-Cook, David M Higdon, et al. (2016). "Computational enhancements to Bayesian design of experiments using Gaussian processes". In: *Bayesian Analysis* 11.1, pp. 191–213.

Williams, Christopher KI and Matthias Seeger (2001). "Using the Nyström method to speed up kernel machines". In: *Advances in neural information processing systems*, pp. 682–688.

Wong, Raymond K. W., Curtis B. Storlie, and Thomas C. M. Lee (2017). "A Frequentist Approach to Computer Model Calibration". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79.2, pp. 635–648.

Xie, J., P.I. Frazier, and S. Chick (2012). *Assemble to Order Simulator*. URL: `http://simopt.org/wiki/index.php?title=Assemble_to_Order&oldid=447` (visited on 07/15/2016).

Ypma, Jelmer, Hans W. Borchers, and Dirk Eddelbuettel (2017). `nloptr`*: R interface to NLopt*. R package version 1.0.4. URL: `https://cran.r-project.org/web/packages/nloptr/`.