POSTERIOR APPROXIMATION BY INTERPOLATION FOR BAYESIAN INFERENCE IN COMPUTATIONALLY EXPENSIVE STATISTICAL MODELS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Nikolay Bliznyuk

August 2008

POSTERIOR APPROXIMATION BY INTERPOLATION FOR BAYESIAN INFERENCE IN COMPUTATIONALLY EXPENSIVE STATISTICAL MODELS

Nikolay Bliznyuk, Ph.D.

Cornell University 2008

Markov Chain Monte Carlo (MCMC) is nowadays a standard approach to numerical computation of integrals of the posterior density π of the parameter vector η. Unfortunately, Bayesian inference using MCMC is computationally intractable when π is expensive to evaluate. In this work, we develop practical methods that approximate π with radial basis functions (RBFs) and Gaussian processes (GPs) interpolants and use the resulting cheap-to-evaluate surfaces in MCMC.

In Chapter 1, π arises from a nonlinear regression model with transformation and dependence. To build the RBF approximation, we limit evaluation of the computationally expensive regression function to points chosen on a high posterior density (HPD) region found using a local quadratic approximation of $\log(\pi)$ at its mode. We illustrate our approach on simulated data for a pollutant diffusion problem and study the frequentist coverage properties of credible intervals.

In Chapter 2, we relax the assumptions about π made in Chapter 1 and develop a derivative-free procedure `GRIMA` to approximate the logarithm of π using RBF interpolation over a HPD region of π estimated using the RBF surface. We use `GRIMA` for Bayesian inference in a computationally intensive nonlinear regression model for real measured streamflow data in the Town Brook watershed.

In Chapter 3, we study statistical models where it is possible to identify a minimal subvector β of η responsible for the expensive computation in the evaluation of π. We propose two approaches to approximate π by interpolation that exploit this computational structure. Our primary contribution is derivation of a GP interpolant that provably improves over some of the existing approaches by reducing the effective dimension of the interpolation problem from $\dim(\eta)$ to $\dim(\beta)$. When $\dim(\eta)$ is high but $\dim(\beta)$ is low, this allows one to dramatically reduce the number of expensive evaluations necessary to construct an accurate approximation of π.

Our experiments indicate that our methods produce results similar to those when the true expensive posterior density is sampled by MCMC while reducing computational costs by well over an order of magnitude.

BIOGRAPHICAL SKETCH

Nikolay Bliznyuk was born on the 6th of March, 1981, in Moscow, Russia. In 2001, he obtained his Bachelor's degree in Economics from George Mason University in the US. He entered the doctoral program in Operations Research at Cornell University in 2002. Upon the defense of his Ph.D. thesis, Nikolay will join the ranks of postdoctoral fellows in the Department of Biostatistics at Harvard University.

I dedicate this work to my parents and grandparents.

ACKNOWLEDGEMENTS

I would like to express my utmost gratitude to my thesis advisor Professor David Ruppert for his superb guidance, remarkable patience and healthy dose of encouragement that enabled me to complete this work.

I wish to thank Professor Christine Shoemaker for exposing me to the practical modeling issues arising in environmental engineering, as well as for moral and financial support of my work. I am appreciative of the insightful discussions with members of her research group – Dillon Cowan, Rommel Regis and Stefan Wild, as well as with David Ruppert's students – Yingxing Li, Ben Shaby and Emmanuel Sharef.

I would like to thank Professors Shane Henderson and David Shmoys for their service on my thesis committee and for their helpful suggestions during my stay here at Cornell. Honorable mention goes to Professors Tanya Apanasovich, Jim Booth, Ciprian Crainiceanu, Gena Samorodnitsky and Rob Strawderman for their kind advice and everlasting willingness to help in some of the dilemmas that I faced. I am also indebted to Professor Sid Resnick for allowing me to use his lecture notes during my teaching of ENGRD270.

For the remarkable academic environment, I would like to thank the rest of the Operations Research department faculty and staff, as well as the graduate students in Statistics and in OR&IE, in particular, my office mates who heroically resisted my attempts to work in the office without all lights on. On the non-academic side, I thank my friends Vadim Zipunnikov, Lyuba Kuznetsova and Ilya Ganusov for all the great time we had together. Finally, I thank my parents and relatives for their unending support and faith in me.

This research work was made possible by support from the National Science Foundation under grant DMS-04-538.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

BAYESIAN CALIBRATION OF COMPUTATIONALLY EXPENSIVE MODELS USING OPTIMIZATION AND RADIAL BASIS FUNCTION APPROXIMATION

1.1 Introduction

[1] A common problem throughout science and engineering is the calibration of scientific models (e.g., Benaman, Shoemaker and Haith (2005), Tolson and Shoemaker (2007a, 2007b), Shoemaker, Regis and Fleming (2007)). Calibration means estimation of unknown parameters, for example, initial conditions or reaction and diffusion rates in a system modeled by partial differential equations. In this paper, we propose a Monte Carlo-based strategy for Bayesian calibration when the models are specified by computationally expensive computer codes, also referred to as simulators.

Our focus is on computer codes that, in a single run, produce deterministic d-dimensional output vectors $f(X, \boldsymbol{\beta})$ for all vector "indices" X in some specified set and a given parameter vector $\boldsymbol{\beta}$. For example, the numerical solution of a partial differential equation produces output at all points on a space-time grid for a fixed vector of coefficients $\boldsymbol{\beta}$. We assume that one has a sample $Y_1, \ldots, Y_n$ of observation vectors in $\mathbb{R}^d$ that correspond to the model values $f(X_1, \boldsymbol{\beta}), \ldots, f(X_n, \boldsymbol{\beta})$, and the goal is to make inferences about $\boldsymbol{\beta}$. The vector X_i, which may contain covariates for the statistical model, is assumed

[1] This chapter was published as a separate paper (Bliznyuk et al., 2008) with Nikolay Bliznyuk as the primary author. The copyright belongs to *American Statistical Association, Institute of Mathematical Statistics, and Interface Foundation of North America*. This chapter is included here with the permission from *Journal of Computational & Graphical Statistics*.

to be known to the experimenter and can thus be regarded as a label for the model $f(X_i; \boldsymbol{\beta})$ for Y_i. We are motivated by environmental engineering problems where Y_i's are vectors of observed concentrations of chemical species and X_i's include the temporal instants and spatial locations where the concentrations were measured, although our methodology is applicable to a wider range of problems. Evaluating $f(X_1, \boldsymbol{\beta}), \ldots, f(X_n, \boldsymbol{\beta})$ for a single value of $\boldsymbol{\beta}$ can be computationally expensive taking, for example, 2.5 hours of CPU time in a groundwater bioremediation problem studied by Mugunthan, Shoemaker and Regis (2005) and Mugunthan and Shoemaker (2006). Thus accurate calibration of such models is infeasible without special methods such as those introduced here.

Given that Y_i is $f(X_i, \boldsymbol{\beta}^{(0)})$ plus noise for value $\boldsymbol{\beta}^{(0)}$ of $\boldsymbol{\beta}$, calibration is seen to be a nonlinear regression problem. However, ordinary (nonlinear) least squares is not recommended since practitioners often find that the variation of Y_i about $f(X_i, \boldsymbol{\beta})$ is non-normally distributed with nonconstant variance and correlated across time and space. We accommodate the non-normality and heteroscedasticity by the transform-both-sides methodology of Carroll and Ruppert (1984) – we assume that, after a suitable transformation, Y_i's are normally distributed and homoscedastic. To model dependencies in the noise, we use a parametric space-time covariance model. The statistical model will be stated precisely in Section 1.2.

Specifying the likelihood of the data $Y_1, \ldots, Y_n$ and prior densities for parameters, we obtain the expression for the unnormalized posterior density. Even though our interest is in the models with the likelihood specified in Section 1.2, any alternative form of the likelihood can be used. We assume that the posterior

density has a single mode in the interior of the parameter space and is differentiable twice; however, derivatives of the simulator with respect to $\boldsymbol{\beta}$ are not assumed to be given.

Our algorithm has four main steps: (1) use numerical optimization to locate the region of the parameter space having high posterior probability; (2) evaluate the model on a suitable set of parameter values in the region of high posterior probability; (3) use the evaluations in steps (1) and (2) to construct a radial basis function (RBF) interpolant of the logarighm of the posterior density; and (4) draw a sample from the approximate posterior density using a Markov Chain Monte Carlo (MCMC) algorithm. As a result, the computational burden is reduced considerably since step (4) does not require expensive function evaluations. Using the sample from the approximate posterior distribution allows us to solve the problems of Bayesian calibration and of prediction for $F(\boldsymbol{\beta})$ by estimating moments and quantiles of the posterior distributions of $\boldsymbol{\beta}$ and $F(\boldsymbol{\beta})$. Here, F is a function whose computation, for a given $\boldsymbol{\beta}$, involves evaluation of $f(\cdot, \boldsymbol{\beta})$ for multiple values of X; more precisely, $F(\boldsymbol{\beta})$ is the value of a functional of $f(\cdot, \boldsymbol{\beta})$.

Empirical studies show that our algorithm can produce estimates of posterior densities for $\boldsymbol{\beta}$ and $F(\boldsymbol{\beta})$ that are nearly the same as when sampling from the exact posterior density. However, our methodology requires far fewer evaluations of the simulator than are needed if the exact posterior density were sampled, e.g., in our application approximately 150 expensive function evaluations are used but the RBF approximation is evaluated 10,000 times.

To the best of our knowledge, this is the first investigation that uses a nonparametric approximation to the posterior density on a region of high posterior

probability found by derivative-free optimization. In Section 1.4.1 we introduce a new transformation family, which is more attractive from the Bayesian perspective than the usual power family and allows a systematic treatment of data transformation, which is typically carried out in an *ad hoc* fashion.

The outline above reflects the organization of the paper: Section 1.2 specifies the statistical model for the data, Section 1.3 deals with the approximation to the posterior density and contains details of the algorithm, Section 1.4 reports the results of a simulation study of a synthetic diffusion problem, and Section 1.5 discusses alternative approaches to calibration of computationally intensive models.

1.2 Description Of The Statistical Model

We assume that Y_i is $f(X_i, \boldsymbol{\beta})$ perturbed by noise, which could include model misspecification and measurement error. In many applications, the components of Y_i show right-skewed variation about $f(X_i, \boldsymbol{\beta})$ with variability that increases with $f(X_i, \boldsymbol{\beta})$. The transform-both-sides methodology of Carroll and Ruppert (1984, 1988) is particularly well suited for such data.

Denote by $Y_{i,j}$ and $f_j(X_i, \boldsymbol{\beta})$ the jth components of Y_i and $f(X_i, \boldsymbol{\beta})$, respectively, where $j = 1, \ldots, d$. Let $\{h(\cdot, \lambda) : \lambda \in \Lambda\}$ be a parametric family of differentiable increasing transformations that are indexed by λ and whose range is the real line for every λ. We assume that, for some λ_j, $h(Y_{i,j}, \lambda_j)$ is distributed $N\left[h\{f_j(X_i, \boldsymbol{\beta}), \lambda_j\}, \sigma_j^2\right]$, where σ_j is constant as a function of X_i; later in this section we discuss a possible extension to account for simulator inadequacy. Stated differently, $h(\cdot, \lambda_j)$ is both a normalizing and variance-stabilizing transforma-

tion for $Y_{i,j}$. In addition, we require that the Y_i's can be transformed to have a joint multivariate normal (MVN) distribution. In Section 1.4.1 we describe a new transformation family that we have used in our application.

It is important to notice that both $Y_{i,j}$ and $f_j(X_i, \boldsymbol{\beta})$ are transformed in the same way. This implies that $f_j(X_i, \boldsymbol{\beta})$ is the conditional median of $Y_{i,j}$ given X_i, so, unlike when $Y_{i,j}$ alone is transformed as in Box and Cox (1964), $f_j(X_i, \boldsymbol{\beta})$ continues to be a model for $Y_{i,j}$. In fact, the model for $Y_{i,j}$ is

$$Y_{i,j} = h^{-1}\left[h\left\{f_j(X_i, \boldsymbol{\beta}), \lambda_j\right\} + \epsilon_{i,j}, \lambda_j\right], \tag{1.1}$$

where, for a fixed λ, $h^{-1}(\cdot, \lambda)$ is the inverse of $h(\cdot, \lambda)$, and $\epsilon_{i,j} \sim N(0, \sigma_j^2)$. For example, if $h(\cdot, \lambda_j)$ is the log transformation, then

$$Y_{i,j} = \exp\left[\log\{f_j(X_i, \boldsymbol{\beta})\} + \epsilon_{i,j}\right] = f_j(X_i, \boldsymbol{\beta}) \exp(\epsilon_{i,j}),$$

so the model has multiplicative, lognormal variation about the conditional median, $f_j(X_i, \boldsymbol{\beta})$.

Let $\boldsymbol{Y} = \left[Y_1^\mathsf{T}, \ldots, Y_n^\mathsf{T}\right]^\mathsf{T}$ be the nd-dimensional column vector of observed responses and $\boldsymbol{f}(\boldsymbol{\beta}) = \left[f(X_1, \boldsymbol{\beta})^\mathsf{T}, \ldots, f(X_n, \boldsymbol{\beta})^\mathsf{T}\right]^\mathsf{T}$ be the corresponding value of the regression function. Define $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_d)^\mathsf{T}$, and denote by $h\{\boldsymbol{Y}, \boldsymbol{\lambda}\}$ and $h\{\boldsymbol{f}(\boldsymbol{\beta}), \boldsymbol{\lambda}\}$ the coordinate-wise transformations of $\boldsymbol{Y}$ and $\boldsymbol{f}(\boldsymbol{\beta})$, where every coordinate corresponding to the jth outcome is transformed by $h(\cdot, \lambda_j)$. Our statistical model is then $h\{\boldsymbol{Y}, \boldsymbol{\lambda}\} \sim MVN\left[h\{\boldsymbol{f}(\boldsymbol{\beta}), \boldsymbol{\lambda}\}, \boldsymbol{\Sigma}(\boldsymbol{\theta})\right]$ with the corresponding likelihood function

$$[\boldsymbol{Y}|\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\theta}] = \frac{\exp\left(-0.5\|h\{\boldsymbol{Y}, \boldsymbol{\lambda}\} - h\{\boldsymbol{f}(\boldsymbol{\beta}), \boldsymbol{\lambda}\}\|^2_{\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}}\right)}{(2\pi)^{nd/2}|\boldsymbol{\Sigma}(\boldsymbol{\theta})|^{1/2}} \cdot |J_h(\boldsymbol{Y}, \boldsymbol{\lambda})|, \tag{1.2}$$

where $J_h(\boldsymbol{Y}, \boldsymbol{\lambda})$ is the Jacobian of the transformation from $\boldsymbol{Y}$ to $h\{\boldsymbol{Y}, \boldsymbol{\lambda}\}$ and $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ belongs to a family of covariance matrices parameterized by $\boldsymbol{\theta}$. Here we

use the now standard notation that [*list*] is the joint density of the random variables in *list* and [*list 1*|*list 2*] is the conditional density of the random variables in *list 1* given those in *list 2*. We also use the conventional notation for the generalized norm, $\|x\|^2_{\boldsymbol{A}} = x^\mathsf{T}\boldsymbol{A}x$.

Define the noise vectors $\epsilon_i = (\epsilon_{i,1}, \ldots, \epsilon_{i,d})^\mathsf{T} = h\{Y_i, \boldsymbol{\lambda}\} - h\{f(X_i, \boldsymbol{\beta}), \boldsymbol{\lambda}\}$ for $i = 1, \ldots, n$, $\boldsymbol{\epsilon}_{\bullet,j} = (\epsilon_{1,j}, \ldots, \epsilon_{n,j})^\mathsf{T}$ for $j = 1, \ldots, d$, and $\boldsymbol{\epsilon} = (\epsilon_1^\mathsf{T}, \ldots, \epsilon_n^\mathsf{T})^\mathsf{T}$. The covariance between $\epsilon_{i,j}$ and $\epsilon_{i',j'}$ is modeled parsimoniously using a separable covariance function of the form $\boldsymbol{C}_{j,j'} \cdot \rho_{ST}(X_i, X_{i'}; \boldsymbol{\gamma})$, where $\boldsymbol{C}$ is a $d \times d$ covariance matrix for ϵ_i and $\rho_{ST}(X_i, X_{i'}; \boldsymbol{\gamma})$ is a space-time correlation function parameterized by $\boldsymbol{\gamma}$. Let $\boldsymbol{S}(\boldsymbol{\gamma})$ be the $n \times n$ space-time correlation matrix with $\boldsymbol{S}_{i,i'}(\boldsymbol{\gamma}) = \rho_{ST}(X_i, X_{i'}; \boldsymbol{\gamma})$. Then $\text{Var}\{\boldsymbol{\epsilon}_{\bullet,j}\} = \boldsymbol{C}_{j,j} \cdot \boldsymbol{S}(\boldsymbol{\gamma})$ and, more generally, $\text{Var}\{\boldsymbol{\epsilon}\} = \boldsymbol{\Sigma}(\boldsymbol{\theta}) = \boldsymbol{S}(\boldsymbol{\gamma}) \otimes \boldsymbol{C}$, where $\boldsymbol{\theta} = (\boldsymbol{\gamma}, \boldsymbol{C})$ and $\otimes$ denotes the Kronecker product of two matrices.

In equation (1.1) we assume that the Gaussian noise term $\epsilon_{i,j}$ is the sum of a model misspecification error (model inadequacy function) and the observation error, which is in the spirit of Higdon, Lee and Holloman (2003) and Craig, Goldstein, Rougier and Seheult (2001). In general, it is impossible to separate these two types of errors, and only their sum is identified. Only with additional assumptions can the individual errors be identified. For example, Kennedy and O'Hagan (2001) assume that the observation errors are independent. In this case, if one assumes that the model misspecification errors are a continuous Gaussian process, then the observation error is a nugget effect and can be identified. Whether the observation errors are independent will, of course, be application-specific. In some cases, it will be not clear whether a specific type of error should be considered "observation error" or model inadequacy. For ex-

ample, in our current work with a stream runoff model, we have observed that often, after a large rainfall event, the residuals are consistently either positive or negative. This pattern is not surprising, since there are large sampling errors when rainfall is estimated from a few gauges. Sampling error in rainfall could be called observation error or model inadequacy, depending on one's viewpoint. In fact, we would rather consider it a third type of error, measurement error in a covariate (rainfall). (In our notation, covariates are included in X.) Because of the difficulties in identifying the different sources of error, we only model their sum. Hence, $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ is the sum of the various covariance matrices.

If there is evidence, a priori or from intermediate diagnostics (see, for example, Bates and Watts (1988)), that the simulator is a deficient representation of the underlying physical process that generates observations, equation (1.2) can be generalized as in Kennedy and O'Hagan (2001), by replacing $h\{\boldsymbol{f}(\boldsymbol{\beta}), \boldsymbol{\lambda}\}$ with $[h\{\boldsymbol{f}(\boldsymbol{\beta}), \boldsymbol{\lambda}\} + \boldsymbol{g}(\boldsymbol{\beta}, \boldsymbol{\eta})]$ or with $h\{\boldsymbol{f}(\boldsymbol{\beta}) + \boldsymbol{g}(\boldsymbol{\beta}, \boldsymbol{\eta}), \boldsymbol{\lambda}\}$. The vector-valued function $\boldsymbol{g}(\boldsymbol{\beta}, \boldsymbol{\eta})$ is the (statistical) model for the mean of the model inadequacy function that may involve X_i's as well as additional predictors. However, Beven (2001) cautions that "... experience with Monte Carlo simulations for complex environmental models used in the generalized likelihood uncertainty estimate (GLUE) ... suggest that it may be very difficult to formulate an inadequacy function."

Unless n is very large, computation of $\boldsymbol{f}$ usually presents the main computational challenge in evaluation of the likelihood. In our algorithm of Section 1.3, we take advantage of this to evaluate the likelihood for multiple values of non-simulator parameters for each run of the simulator.

When the goal of the study is the posterior distribution of the value, $F(\boldsymbol{\beta})$, of

some functional of $f(\cdot, \boldsymbol{\beta})$, as well as the joint posterior density for $\boldsymbol{\beta}$, it may be necessary to evaluate the expensive function for additional space-time indices $X_{n+1}, \ldots, X_{n^*}$, for which no response Y_i was observed, in order to compute or approximate $F(\boldsymbol{\beta})$; further details are found in Section 1.3.3. We assume that, for a single value of $\boldsymbol{\beta}$, a run of the expensive model produces the entire vector

$$\boldsymbol{f}^*(\boldsymbol{\beta}) = \left(\boldsymbol{f}\{\boldsymbol{\beta}\}^\mathsf{T}, f\{X_{n+1}, \boldsymbol{\beta}\}^\mathsf{T}, \ldots, f\{X_{n^*}, \boldsymbol{\beta}\}^\mathsf{T}\right)^\mathsf{T}. \tag{1.3}$$

This leads to computational savings, for example, in models relying on numerical meshes or grids, for which it is often more beneficial to obtain values of $f(X_i, \boldsymbol{\beta})$ for all values X_i of interest in a single step rather than to compute them in two stages (i.e., first for $\boldsymbol{f}(\boldsymbol{\beta})$ and then for $f(X_i, \boldsymbol{\beta})$ for all $i > n$).

1.3 Methodology

1.3.1 An Approximation to the Posterior Density

Since our procedure approximates the posterior density by an interpolant, it is subject to the curse of dimensionality. In this subsection we briefly review ways to lower the dimension of the argument of the posterior density and introduce some new notation.

Given a prior density $[\boldsymbol{\beta}, \boldsymbol{\zeta}]$, one has a posterior density

$$[\boldsymbol{\beta}, \boldsymbol{\zeta} | \boldsymbol{Y}] = \frac{[\boldsymbol{Y} | \boldsymbol{\beta}, \boldsymbol{\zeta}] \cdot [\boldsymbol{\beta}, \boldsymbol{\zeta}]}{\int [\boldsymbol{Y} | \boldsymbol{\beta}, \boldsymbol{\zeta}] \cdot [\boldsymbol{\beta}, \boldsymbol{\zeta}] d\boldsymbol{\beta}\, d\boldsymbol{\zeta}}. \tag{1.4}$$

As before, $\boldsymbol{\beta}$ is the argument of the simulator and $\boldsymbol{\zeta}$ is the vector of non-simulator parameters. We associate $\boldsymbol{\zeta}$ with nuisance parameters $\{\boldsymbol{\lambda}, \boldsymbol{\theta}\}$ from

the previous section. The case with model inadequacy function parameters $\boldsymbol{\eta}$ can be treated similarly and is not considered.

In applications, $\boldsymbol{\beta}$ is the primary parameter and interest centers on its marginal posterior density $[\boldsymbol{\beta}|\boldsymbol{Y}] = \int [\boldsymbol{\beta}, \boldsymbol{\zeta}|\boldsymbol{Y}] d\boldsymbol{\zeta}$. It is often possible to integrate out a sub-block of parameters in $\boldsymbol{\zeta}$ either analytically, by using a conjugate family of prior densities as shown in Appendix 1.6.4, or numerically. In what follows, let $\boldsymbol{\zeta}$ be the subvector of the remaining non-simulator parameters after the integration. Also, $\boldsymbol{Y}$ is always regarded as fixed and $[\boldsymbol{\beta}, \boldsymbol{Y}]$ and $[\boldsymbol{\beta}, \boldsymbol{\zeta}, \boldsymbol{Y}]$ refer, respectively, to arbitrary unnormalized marginal and joint posterior densities.

If $\boldsymbol{f}$ were inexpensive to evaluate, we could sample from $[\boldsymbol{\beta}, \boldsymbol{\zeta}|\boldsymbol{Y}]$ using $[\boldsymbol{\beta}, \boldsymbol{\zeta}, \boldsymbol{Y}]$ with a Metropolis-Hastings (M-H) algorithm, and the sample of $\boldsymbol{\beta}$ would be a sample from $[\boldsymbol{\beta}|\boldsymbol{Y}]$. However, drawing large samples is computationally prohibitive in our setting.

Our goal is to obtain an accurate and cheap-to-evaluate nonparametric approximation to $[\boldsymbol{\beta}, \boldsymbol{Y}]$ or $[\boldsymbol{\beta}, \boldsymbol{\zeta}, \boldsymbol{Y}]$ based on a relatively small number of evaluations of $\boldsymbol{f}$. One can use the resulting surface as a surrogate for the respective unnormalized posterior density in a M-H algorithm, as the sampler does not require specification of normalizing constants. (In Section 1.5, we contrast the proposed procedure with similar approaches in the literature.) When the expression for $[\boldsymbol{\beta}, \boldsymbol{Y}]$ is not available, we first approximate $[\boldsymbol{\beta}, \boldsymbol{Y}]$ heuristically. For a fixed value of $\boldsymbol{\beta}$, let $\widehat{\boldsymbol{\zeta}}(\boldsymbol{\beta})$ be the maximizer of $[\boldsymbol{\beta}, \boldsymbol{\zeta}, \boldsymbol{Y}]$ with respect to $\boldsymbol{\zeta}$. One possible heuristic approximation is the *profile* posterior density

$$\pi_{\max}(\boldsymbol{\beta}, \boldsymbol{Y}) = \sup_{\boldsymbol{\zeta}} [\boldsymbol{\beta}, \boldsymbol{\zeta}, \boldsymbol{Y}] = [\boldsymbol{\beta}, \widehat{\boldsymbol{\zeta}}(\boldsymbol{\beta}), \boldsymbol{Y}]. \tag{1.5}$$

A more sophisticated *Laplace* approximation of Tierney and Kadane (1986) mul-

tiplies (1.5) by a correction factor. A simplification to (1.5), referred to as *pseudoposterior* density, is obtained by replacing $\widehat{\boldsymbol{\zeta}}(\boldsymbol{\beta})$ by $\widehat{\boldsymbol{\zeta}}(\widehat{\boldsymbol{\beta}})$, where $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\zeta}}(\widehat{\boldsymbol{\beta}}))$ is the maximum a posteriori (MAP) estimator, the mode of the joint posterior density $[\boldsymbol{\beta}, \boldsymbol{\zeta}|\boldsymbol{Y}]$. Each of these approximations to $[\boldsymbol{\beta}, \boldsymbol{Y}]$ attempt to avoid the more difficult task of integrating out nuisance parameters by maximization. It should be kept in mind, though, that neither the integration nor the maximization requires extra evaluations of $\boldsymbol{f}$. In the sequel, the notation $\pi(\cdot, \boldsymbol{Y})$ will be used to refer to any of these heuristic approximations to $[\boldsymbol{\beta}, \boldsymbol{Y}]$.

As a nonparametric approximation, we use interpolation of the logarithms of $\pi(\cdot, \boldsymbol{Y})$ or $[\boldsymbol{\beta}, \boldsymbol{\zeta}, \boldsymbol{Y}]$ by radial basis functions (RBFs). We state our algorithm for $\pi(\cdot, \boldsymbol{Y})$ as the surface of interest. The treatment of $[\boldsymbol{\beta}, \boldsymbol{\zeta}, \boldsymbol{Y}]$ is similar.

1.3.2 The Algorithm

In our algorithm, $\boldsymbol{f}^*$ is evaluated only during the optimization stage in order to find the MAP estimate (**Step 1**) and for values of $\boldsymbol{\beta}$ in a high posterior density region (**Step 2**) in order to approximate the logarithm of the posterior density accurately by an RBF surface (**Step 3**). The approximate posterior surface is subsequently sampled using MCMC in **Step 4**.

For ease of exposition, we assume that the posterior density has a single mode located in the interior of the parameter space and that $\boldsymbol{f}$ is twice differentiable in a neighborhood $\widehat{\boldsymbol{\beta}}$, but we are currently generalizing the approach to multimodal densities. While selecting "design points" (the values of $\boldsymbol{\beta}$ at which to evaluate $\boldsymbol{f}^*$) we try to keep as small as possible the number of "uninformative" points – those very close to some point, at which the value of $\boldsymbol{f}^*$ is known,

or far away from the mode.

Finding the MAP (Step 1)

For a given value $\boldsymbol{\beta}^{(0)}$ of $\boldsymbol{\beta}$, the gradient and Hessian of $\log\{[\boldsymbol{\beta}^{(0)}, \boldsymbol{\zeta}, \boldsymbol{Y}]\}$ with respect to $\boldsymbol{\zeta}$ are available analytically, and so this function can be maximized efficiently to produce $\widehat{\boldsymbol{\zeta}}(\boldsymbol{\beta}^{(0)})$ and thus to compute $\log\{\pi_{\max}(\boldsymbol{\beta}^{(0)}, \boldsymbol{Y})\}$ from (1.5). We perform this maximization using a constrained minimization routine (sequential quadratic programming) with analytical gradients and Hessians, implemented in MATLAB's `fmincon`. Consequently, we maximize $\log\{\pi_{\max}(\boldsymbol{\beta}, \boldsymbol{Y})\}$ with respect to $\boldsymbol{\beta}$ to find $\widehat{\boldsymbol{\beta}}$ and then $\log\{[\widehat{\boldsymbol{\beta}}, \boldsymbol{\zeta}, \boldsymbol{Y}]\}$ with respect to $\boldsymbol{\zeta}$ to determine $\widehat{\boldsymbol{\zeta}}(\widehat{\boldsymbol{\beta}})$ and hence the MAP. The use of a gradient-based algorithm for maximization with respect to $\boldsymbol{\beta}$ is not recommended unless the Jacobian of $\boldsymbol{f}$ comes at low cost along with $\boldsymbol{f}$ because finite differencing to estimate derivatives produces clusters of "uninformative" design points. We maximize $\log\{[\widehat{\boldsymbol{\beta}}, \boldsymbol{\zeta}, \boldsymbol{Y}]\}$ using publicly available software `CONDOR` described by Vanden Berghen and Bersini (2005), which implements a derivative-free trust-region algorithm UOBYQA of Powell (2000). Other derivative-free optimization methods could also be used in this step including those applied (without accompanying uncertainty analysis) to environmental calibration problems as discussed in the papers by Shoemaker et al. (2007) and Tolson and Shoemaker (2007a).

The Experimental Design (Step 2)

Ideally, we would like to fit the RBF surface over a highest posterior density (HPD) region of $[\boldsymbol{\beta}|\boldsymbol{Y}]$, defined as $C_R(\alpha) = \{\boldsymbol{\beta} : [\boldsymbol{\beta}, \boldsymbol{Y}] > \kappa(\alpha)\}$, where $\kappa(\alpha)$ is

chosen so that the credible region $C_R(\alpha)$ contains the fraction $1-\alpha$ of the mass of $[\boldsymbol{\beta}, \boldsymbol{Y}]$. Here α is a tuning parameter, for example, 0.05 or 0.01.

The size $(1-\alpha)$ HPD region cannot be computed accurately – not only for $[\boldsymbol{\beta}, \boldsymbol{Y}]$, but also for $[\boldsymbol{\beta}, \boldsymbol{\zeta}, \boldsymbol{Y}]$ and for any of the heuristic approximations to $[\boldsymbol{\beta}, \boldsymbol{Y}]$ from the previous section – without a prohibitive number of evaluations of $\boldsymbol{f}$. We obtain an approximate HPD region, $\widehat{C}_R(\alpha)$, using a Taylor expansion of $\log\{[\boldsymbol{\beta}, \boldsymbol{\zeta}, \boldsymbol{Y}]\}$ near the MAP $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\zeta}})$, which corresponds to the approximation to $[\boldsymbol{\beta}, \boldsymbol{\zeta}|\boldsymbol{Y}]$ by a multivariate normal density. Specifically, let $\widehat{\boldsymbol{I}}$ be the negative of the Hessian of $\log\{[\boldsymbol{\beta}, \boldsymbol{\zeta}, \boldsymbol{Y}]\}$ with respect to $(\boldsymbol{\beta}, \boldsymbol{\zeta})$ evaluated at $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\zeta}})$. By partitioning $\widehat{\boldsymbol{I}}^{-1}$ into blocks corresponding to $\boldsymbol{\beta}$ and $\boldsymbol{\zeta}$ one gets

$$\begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\zeta} \end{bmatrix} \overset{approx.}{\sim} MVN\left(\begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\zeta}} \end{bmatrix}, \begin{bmatrix} \widehat{\boldsymbol{I}}^{\boldsymbol{\beta}\boldsymbol{\beta}} & \widehat{\boldsymbol{I}}^{\boldsymbol{\beta}\boldsymbol{\zeta}} \\ \widehat{\boldsymbol{I}}^{\boldsymbol{\zeta}\boldsymbol{\beta}} & \widehat{\boldsymbol{I}}^{\boldsymbol{\zeta}\boldsymbol{\zeta}} \end{bmatrix} \right), \text{ where} \tag{1.6}$$

$$\widehat{\boldsymbol{I}}^{-1} = \begin{bmatrix} \widehat{\boldsymbol{I}}_{\boldsymbol{\beta}\boldsymbol{\beta}} & \widehat{\boldsymbol{I}}_{\boldsymbol{\beta}\boldsymbol{\zeta}} \\ \widehat{\boldsymbol{I}}_{\boldsymbol{\zeta}\boldsymbol{\beta}} & \widehat{\boldsymbol{I}}_{\boldsymbol{\zeta}\boldsymbol{\zeta}} \end{bmatrix}^{-1} = \begin{bmatrix} \widehat{\boldsymbol{I}}^{\boldsymbol{\beta}\boldsymbol{\beta}} & \widehat{\boldsymbol{I}}^{\boldsymbol{\beta}\boldsymbol{\zeta}} \\ \widehat{\boldsymbol{I}}^{\boldsymbol{\zeta}\boldsymbol{\beta}} & \widehat{\boldsymbol{I}}^{\boldsymbol{\zeta}\boldsymbol{\zeta}} \end{bmatrix}. \tag{1.7}$$

Estimation of $\widehat{\boldsymbol{I}}$ by finite differences is wasteful as it does not produce new informative design points. We estimate $\widehat{\boldsymbol{I}}$ by fitting a quadratic surface to $\log\{[\boldsymbol{\beta}, \boldsymbol{\zeta}, \boldsymbol{Y}]\}$ in a neighborhood of $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\zeta}})$. This procedure, stated in detail in Appendix 1.6.1, allows one to reduce the number of wasteful design points and to reuse the points from the optimization trajectory from **Step 1**. To avoid new notation, from now on we use the old notation for the true $\widehat{\boldsymbol{I}}$ and its blocks from (1.7) to refer solely to the estimated Hessian and its blocks.

We define

$$\widehat{C}_R(\alpha) = \left\{ \boldsymbol{\beta} : (\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}})^{\mathsf{T}} \left[\widehat{\boldsymbol{I}}^{\boldsymbol{\beta}\boldsymbol{\beta}} \right]^{-1} (\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}) \leq \chi^2_{p,1-\alpha} \right\}, \tag{1.8}$$

where $\widehat{\boldsymbol{I}}^{\boldsymbol{\beta}\boldsymbol{\beta}} = \left[\widehat{\boldsymbol{I}}_{\boldsymbol{\beta}\boldsymbol{\beta}} - \widehat{\boldsymbol{I}}_{\boldsymbol{\beta}\boldsymbol{\zeta}} \cdot \widehat{\boldsymbol{I}}_{\boldsymbol{\zeta}\boldsymbol{\zeta}}^{-1} \cdot \widehat{\boldsymbol{I}}_{\boldsymbol{\zeta}\boldsymbol{\beta}} \right]^{-1}$ and $\chi^2_{p,1-\alpha}$ is the $(1-\alpha)$th quantile of the

χ^2_p distribution, with p being the dimension of $\boldsymbol{\beta}$. This approximate HPD region is the size-$(1-\alpha)$ minimum volume confidence ellipsoid for the (marginal) normal approximation to $[\boldsymbol{\beta}|\boldsymbol{Y}]$ based on equation (1.6). We will use evaluations of $\boldsymbol{f}$ on this region at the same set of values of $\boldsymbol{\beta}$ to fit RBF surfaces to any of the posterior surfaces from Section 1.3.1, possibly all of them. This substep of determining an approximate design region is referred to as **Step 2A**.

We remark that $\widehat{\boldsymbol{I}}$ is crucial for subsequent analysis. If $\boldsymbol{H}$ is any square full-rank matrix such that $\widehat{\boldsymbol{I}}^{\boldsymbol{\beta\beta}} = \boldsymbol{H}\boldsymbol{H}^{\mathsf{T}}$, for example, a Cholesky factor of $\widehat{\boldsymbol{I}}^{\boldsymbol{\beta\beta}}$, then we apply the linear transformation $\boldsymbol{H}^{-1}$ to $\boldsymbol{\beta}$ to ensure the same scale and to reduce correlation in parameters. Extra design points are chosen with respect to the maximum separation criteria on this transformed space. We fit our RBF surface on the transformed space as well, but choose not to introduce new notation to emphasize this. Finally, $\widehat{\boldsymbol{I}}$ is also used in the MCMC stage to define one of the scale parameters of the proposal density.

Let $\mathcal{B}_O$ and $\mathcal{B}_H$ be sets of values of $\boldsymbol{\beta}$ at which $\boldsymbol{f}^*$ is evaluated during optimization in **Step 1** and during estimation of $\widehat{\boldsymbol{I}}$ in **Step 2A**, respectively. In general, points in $\mathcal{B}_O \cup \mathcal{B}_H$ do not cover $\widehat{C}_R(\alpha)$ adequately to enable us to approximate the chosen posterior surface accurately over the whole approximate HPD region. We augment these points with an approximate *maximin* experimental design $\mathcal{B}_E$. Specifically, we require that points in $\mathcal{B}_E$ be well-separated and do not lie close to those in $\mathcal{B}_O \cup \mathcal{B}_H$, with between-point distances measured after the mentioned linear transformation. (When working with $[\boldsymbol{\beta}, \boldsymbol{\zeta}, \boldsymbol{Y}]$, we evaluate the joint posterior density for multiple values of non-simulator parameters $\boldsymbol{\zeta}$ for each given evaluation of the simulator.) Further details and motivation are provided in Appendix 1.6.2. We refer to the step of choosing $\mathcal{B}_E$ by **Step 2B**.

Finally, we let $\mathcal{B}_D = (\mathcal{B}_O \cup \mathcal{B}_H \cup \mathcal{B}_E) \cap \widehat{C}_R(\alpha')$ for $\alpha' \leq \alpha$ and define $N = |\mathcal{B}_D|$, the size of $\mathcal{B}_D$. The points in $\mathcal{B}_D$ will be used to build the RBF approximation. The motivation is that the optimization trajectory points $\mathcal{B}_O$ lying far outside of $\widehat{C}_R(\alpha)$ rarely improve the quality of approximation. We typically use $\alpha \leq 0.1$ and $\alpha' = 0.01$ or 0.005 in practice.

The RBF Approximation (Step 3)

We use radial basis functions (Buhmann 2003, Powell 1992) to approximate the logarithm of the posterior surface by an interpolant of $l(\cdot) = \log\{\pi(\cdot, \boldsymbol{Y})\}$ at the design points $\mathcal{B}_D = \{\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(N)}\}$ of the form

$$\widetilde{l}(\boldsymbol{\beta}) = \sum_{i=1}^{N} a_i \phi(\|\boldsymbol{\beta} - \boldsymbol{\beta}^{(i)}\|_2) + q(\boldsymbol{\beta}), \tag{1.9}$$

where $\boldsymbol{\beta} \in \mathbb{R}^p$, $a_1, \ldots, a_N \in \mathbb{R}$, $q \in \Pi_m^p$ (the space of polynomials in $\mathbb{R}^p$ of degree less than or equal to m), $\|\cdot\|_2$ denotes the Euclidean norm, and the basis function ϕ has one of the following forms: (1) *surface spline*: $\phi(r) = r^\kappa$, $\kappa \in \mathbb{N}$, κ odd, or $\phi(r) = r^\kappa \log r$, $\kappa \in \mathbb{N}$, κ even; (2) *multiquadric*: $\phi(r) = (r^2 + \gamma^2)^\kappa$, $\kappa > 0$, $\kappa \notin \mathbb{N}$; (3) *inverse multiquadric*: $\phi(r) = (r^2 + \gamma^2)^\kappa$, $\kappa < 0$; (4) *Gaussian*: $\phi(r) = \exp(-\gamma r^2)$; where $r \geq 0$ and γ is a positive constant. The purpose of the polynomial tail is to ensure that the interpolation matrix is invertible. In the numerical experiments, we use the cubic form $\phi(r) = r^3$ with a linear tail $q(\boldsymbol{\beta}) = (1, \boldsymbol{\beta}^\mathsf{T}) \cdot \boldsymbol{c}$.

Our choice of RBF approximation over alternatives was influenced by success with application of this method to related problems reported in Regis and Shoemaker (2007a, 2007b). However, other interpolation methods could be used. The closely related technique of kriging assumes that l is a realization of a Gaussian process (GP), determined by mean and covariance functions, and

uses best linear prediction as a means of interpolation. The RBF interpolation model is a form of universal kriging with a generalized (not necessarily positive definite) covariance function (Cressie 1991, sec. 4.4.5). Unlike a general RBF model, kriging allows one to use the covariance function of the GP (conditional on the process values at design points) to assess prediction uncertainty, which may be used to sequentially select design points for additional simulator runs.

In our case study with synthetic data in Section 1.4, we found that the RBF interpolant gave results that were virtually indistinguishable from the exact results. Nonetheless, the ability to assess the uncertainty of prediction is useful, especially in higher dimensional problems, or under other circumstances, where the RBF interpolant is likely to be less accurate. It may be possible to devise a similar measure in the case of RBF interpolation, and we intend to investigate this in the future.

Selection of extra design points $\mathcal{B}_E$ in our implementation of RBF model is guided by the convergence results of $\widetilde{l}$ to l (Buhmann 2003, chap. 5) that suggest that the rate of convergence is governed by the maximum (over all points in $\widehat{C}_R(\alpha)$) distance from any point in $\widehat{C}_R(\alpha)$ to the closest design point in $\mathcal{B}_D$ (coverage radius). We are not aware of similar convergence results for a kriging model, in part because the covariance function is typically chosen for other reasons. (However, see Appendix 1.6.2 for design optimality considerations for GP interpolation.)

MCMC Sampling (Step 4)

In **Step 4** of the algorithm we draw an MCMC sample from the density proportional to $\widetilde{\pi}(\cdot, \boldsymbol{Y}) = \exp\{\widetilde{l}(\cdot)\}$ restricted to the approximate HPD region $\widehat{C}_R(\alpha')$, see the end of Section 1.3.2 and equation (1.9). This is done to prevent sampling of $\widetilde{\pi}(\cdot, \boldsymbol{Y})$ in the low-probability regions of $[\boldsymbol{\beta}|\boldsymbol{Y}]$ where $\pi(\cdot, \boldsymbol{Y})$ is not approximated well enough. Sampling can be carried out using any MCMC algorithm that does not require the normalizing constant of $\widetilde{\pi}(\cdot, \boldsymbol{Y})$ to be known. We work with the autoregressive Metropolis-Hastings algorithm (Tierney (1994)) that uses a (vector) $AR(1)$ process to generate candidate points $\boldsymbol{\beta}^c$ given the current state $\boldsymbol{\beta}^{(t)}$ of the chain, *i.e.*, $\boldsymbol{\beta}^c = \boldsymbol{\mu} + \boldsymbol{\rho}(\boldsymbol{\beta}^{(t)} - \boldsymbol{\mu}) + \boldsymbol{e}_t$, where $\boldsymbol{\mu}$ is the location parameter, $\boldsymbol{\rho}$ is the autoregressive parameter (matrix), and $\boldsymbol{e}_t$'s are *i.i.d.* noise vectors from a density g. The algorithm allows much freedom in tuning its performance and includes the popular random walk M-H (when $\boldsymbol{\rho} = 1$) and the independence M-H (when $\boldsymbol{\rho} = 0$) algorithms as special cases. In our experiments, g is taken to be a finite mixture of multivariate normal and Student's t densities centered at zero with dispersion matrices proportional to $\widehat{\boldsymbol{I}}^{\boldsymbol{\beta\beta}}$. The location parameter $\boldsymbol{\mu}$ is set to the MAP $\widehat{\boldsymbol{\beta}}$. We observed that negative values of $\boldsymbol{\rho}$ help to reduce serial correlation in the Markov chain. To improve mixing, we recommend that the tuning parameters for the sampler be calibrated to a particular application individually by conventional methods reviewed, for example, in Gelman et al. (2004), as at this stage MCMC does not require evaluation of $\boldsymbol{f}$.

1.3.3 Bayesian Inference

Once the MCMC sample $\mathcal{B}_M$ from the approximate posterior density is obtained as discussed in Section 1.3.2, inference about $\boldsymbol{\beta}$ can proceed using standard methods. A problem of particular concern in environmental engineering is estimation of the value $F(\boldsymbol{\beta})$ of some functional of $f(\cdot, \boldsymbol{\beta})$, for example, $f(X, \boldsymbol{\beta})$ itself at values of X whose time coordinate is in the future. In this case, the set $\{F(\boldsymbol{\beta}) : \boldsymbol{\beta} \in \mathcal{B}_M\}$ is a sample from the approximate posterior distribution of $F(\boldsymbol{\beta})$.

Since $F(\boldsymbol{\beta})$ is determined by $f(\cdot, \boldsymbol{\beta})$, it is also computationally expensive, and hence approximation is necessary to evaluate it at the points from the MCMC run. However, assuming as in Section 1.2 that $F(\boldsymbol{\beta})$ is a function (or can be approximated by a function) of components of $\boldsymbol{f}^*(\boldsymbol{\beta})$, it may be sufficient to compute its values only on the approximate HPD region for $\boldsymbol{\beta}$. Since we have already evaluated $\boldsymbol{f}^*$ at the design points in $\mathcal{B}_D$, it is cheap computationally to interpolate F (or an approximation to it) at the points in $\mathcal{B}_D$ and to evaluate the resulting interpolant at the points in $\mathcal{B}_M$. This approximate sample from the posterior distribution of $F(\boldsymbol{\beta})$ can be subsequently used to estimate functionals of the posterior of $F(\boldsymbol{\beta})$.

1.4 An Environmental Application

In this section, we consider calibration of an environmental model for the concentrations of pollutants and illustrate our methodology on a synthetic test problem. The test problem was chosen such that $f(X, \boldsymbol{\beta})$ is given in closed form

and can be evaluated inexpensively. Unlike with the expensive model functions used in many applications, this allows us to carry out an extensive Monte Carlo study comparing the coverage properties of the approximate Bayesian credible intervals based on RBF surfaces that require a relatively small number of evaluations of $\boldsymbol{f}^*$ with those of the exact credible intervals that require thousands of evaluations of the expensive exact posterior density.

Examples of methods designed for calibration and uncertainty analysis of computationally expensive environmental models are Mugunthan et al. (2005) and Mugunthan and Shoemaker (2006), respectively. The methods in both of these papers are applied to a remediation problem at an US-DOD site that has been contaminated with chlorinated ethenes in the soil and groundwater. The simulation model there takes 2.5 hours to run. Neither of these earlier methods base analysis on the joint posterior density of the parameters as is done in this paper.

Before starting with the details of the environmental application, it is necessary to define a new transformation for positive data that are common in science.

1.4.1 A New Transformation Family

Since we are modeling concentrations, we assume in our application that both the vector of observed concentrations $\boldsymbol{Y}$ and the simulator $f(X, \boldsymbol{\beta})$ are positive. The usual transformation family used with the transform-both-sides method for such data is the Box-Cox family where $h_{BC}(y, \lambda)$ is $(y^\lambda - 1)/\lambda$ if $\lambda \neq 0$ and is $\log(y) = \lim_{\lambda \to 0}(y^\lambda - 1)/\lambda$ if $\lambda = 0$. In typical applications, λ takes values

between 0 and 1. Lower values of λ define more concave transformations.

Notice that the requirement of Section 1.2 that the range of $h(\cdot, \lambda)$ is the real line does not hold for $h_{BC}(\cdot, \lambda)$ except when $\lambda = 0$. Then one needs to "truncate" the normal distribution of $\epsilon_{i,j}$ in equation (1.1) to the set where the inverse of $h_{BC}(\cdot, \lambda)$ is defined. Consequently, to make the expression in equation (1.2) a valid density, one must multiply it by a normalizing constant, whose computation is feasible only for the simplest models.

To avoid this difficulty, we propose the *COnvex combination of Identity and Log (COIL)* family defined as

$$h_C(y, \lambda) = \lambda y + (1 - \lambda) \log(y), \quad 0 < \lambda \leq 1. \tag{1.10}$$

As in the Box-Cox family, λ similarly controls the degree of concavity.

Our simulation experiments with the transform-both-sides method show that the entire Box-Cox family for $\lambda \in [0, 1)$ can be approximated well by our family. The empirical study of the *COIL* family, including its generalizations to more concave transformations, will be reported in a separate paper.

1.4.2 Environmental Assessment of a Chemical Spill: Formulation

Consider a chemical accident that has caused a pollutant to spill at two locations into a long and narrow holding channel. Assume it is known that the same mass M was spilled at each location (0 and L) and that the vector of the location and time of the first spill is $(0, 0)$. However, the location L and time τ of the second

spill are unknown as is the value of M and the diffusion rate D in the channel. We want to estimate the average concentration of the pollutant at the one point the channel and assess the uncertainty associated with this value since harmful effects to the environment are usually estimated from pollutant concentrations. We want to know the joint posterior distribution of all the parameters, but the parameter L is of special interest because L locates the as-yet-unidentified industry that will need to pay for its share of the clean-up costs.

A first-order approach to modeling the concentration of substances in such channels is to assume that the channel can be approximated by an infinitely long one-dimensional system in which diffusion is the only transport device. We assume that the spills are each of mass M and occur instantaneously at space-time points $(s,t) = (0,0)$ and $(s,t) = (L,\tau)$ and that the diffusion coefficient D is constant in both time and space. This leads to the concentration representation (for $t > 0$ and $s \geq 0$):

$$C(s,t;M,D,L,\tau) = \frac{M}{\sqrt{4\pi Dt}} \exp\left[\frac{-s^2}{4Dt}\right] + \frac{M}{\sqrt{4\pi D(t-\tau)}} \exp\left[\frac{-(s-L)^2}{4D(t-\tau)}\right] \cdot \mathbb{I}(\tau < t), \tag{1.11}$$

where $\mathbb{I}$ is the indicator function. We take $\boldsymbol{\beta}$ to be the vector of the four unknown environmental parameters (M, D, L, τ) and consider the scaled concentration $f\{(s,t),\boldsymbol{\beta}\} = \sqrt{4\pi}C(s,t;\boldsymbol{\beta})$.

We assume that each of the five monitoring stations fixed at spatial locations $s_j = 0, 0.5, 1, 1.5, 2.5$ record 200 concentration readings at times $t_k = 0.3, 0.6, \ldots, 50.7, 60$. The corresponding expensive model function is $\boldsymbol{f}(\boldsymbol{\beta}) = \{f(X_1,\boldsymbol{\beta}), \ldots, f(X_{1000},\boldsymbol{\beta})\}^{\mathsf{T}}$, where $X_i = (s_j, t_k)$ if $i = (j-1)\cdot 200 + k$. In this example $f(X_i,\boldsymbol{\beta})$ is scalar because there is only one pollutant.

The ultimate goal of the study is to assess the space-time prediction uncer-

tainty associated with the average concentration at the end of the channel, corresponding to $s = 3$, over the time interval $[40, 140]$. To this end we consider the function $F(\boldsymbol{\beta}) = \sum_{i=0}^{20} f\{(3, 40 + 5i), \boldsymbol{\beta}\}$ which requires evaluation of f at the additional points $\{(3, 40), (3, 45), \ldots, (3, 140)\}$. As discussed in Section 1.2, the expensive model of equation (1.3) that we evaluate is

$$\boldsymbol{f}^*(\boldsymbol{\beta}) = \left(\boldsymbol{f}\{\boldsymbol{\beta}\}^\mathsf{T}, f\{(3, 40), \boldsymbol{\beta}\}, \ldots, f\{(3, 140), \boldsymbol{\beta}\}\right)^\mathsf{T}.$$

An intermediate goal is estimation of the posterior density of $\boldsymbol{\beta}$, which is partially captured by the marginal densities of its components.

Table 1.1: Parameter spaces and true parameter values, mean and (standard deviation) of Monte Carlo mean, mean and (standard deviation) of ratios of lengths of RBF to exact credible intervals, based on 1000 dataset replications and $[\boldsymbol{\beta}, \lambda, \boldsymbol{Y}]$ as the surface. The RBF approximations use, on average, 150 expensive function evaluations compared to 10,000 for the exact results.

			MC mean		ratio of lengths of cred. int.'s		
	domain	true	exact	RBF	size 0.9	size 0.95	size 0.99
$\boldsymbol{\beta}_1$	[7, 13]	10	10.0057	10.0061	0.9969	0.9961	0.9844
			(0.0866)	(0.0893)	(0.0602)	(0.0624)	(0.0738)
$\boldsymbol{\beta}_2$	[0.02, 0.12]	0.07	0.07008	0.07008	0.9910	0.9888	0.9687
			(0.00097)	(0.00101)	(0.0592)	(0.0612)	(0.0673)
$\boldsymbol{\beta}_3$	[.01, 3]	1	1.0005	1.0005	0.9671	0.9662	0.9604
			(0.0136)	(0.0134)	(0.0785)	(0.0765)	(0.0750)
$\boldsymbol{\beta}_4$	[30.01, 30.295]	30.16	30.1610	30.1610	0.9786	0.9709	0.9403
			(0.0096)	(0.0096)	(0.0779)	(0.0818)	(0.0835)
$F(\boldsymbol{\beta})$	–	128.998	129.063	129.067	0.9959	0.9937	0.9841
			(1.087)	(1.100)	(0.062)	(0.0628)	(0.0695)

The vector $\boldsymbol{Y}$ of observed concentrations is generated from models of equations (1.2) and (1.11) with $\lambda = 0.333$ for the COIL family given by (1.10) and the values of $\boldsymbol{\beta}_i$'s and respective parameter spaces (domains) given in Table 1.1. The likelihood from equation (1.2) is discontinuous since $C(s_i, t_j; \boldsymbol{\beta})$ in equation (1.11) explodes when $\boldsymbol{\beta}_3 \equiv L = s_i$ and $\boldsymbol{\beta}_4 \equiv \tau$ approaches t_j from below. To avoid discontinuities, the parameter space for $\boldsymbol{\beta}_4$ was restricted to the interval containing the true value of the parameter, given in Table 1.1. Components of $\boldsymbol{Y}$ are independent, with variance of $h(Y_i, \lambda)$ for every i equal to the sample variance of $h\{f(X_1, \boldsymbol{\beta}), \lambda\}, \ldots, h\{f(X_{1000}, \boldsymbol{\beta}), \lambda\}$, computed for the true (fixed) values of $\boldsymbol{\beta}$ and λ and multiplied by a scaling constant c^2. Here, c controls the amount of noise in the transformed observed data relative to the variability of the corresponding transformed model values. For illustrative purposes – to ensure that the likelihood has a single dominant mode – c is set to $.3$. We put a uniform prior density on $(\boldsymbol{\beta}, \lambda)$ over the parameter spaces mentioned earlier and an overdispersed inverse-gamma prior density on σ^2. This is a special case of the earlier model for multiple chemical species, and, as shown in Appendix 1.6.4, σ^2 can be integrated out analytically. Thus $[\boldsymbol{\beta}, \lambda, \boldsymbol{Y}]$ is the unnormalized posterior density from which we derive $\pi(\cdot, \boldsymbol{Y})$ as discussed in Section 1.3.1.

1.4.3 Analysis

We applied our algorithm to a large number of dataset replications with the same statistical model and parameter values but a different realization of the noise. The maximizer $(\widehat{\boldsymbol{\beta}}, \widehat{\lambda})$ of $[\boldsymbol{\beta}, \lambda, \boldsymbol{Y}]$ was found by `CONDOR` via maximization of $\pi_{\max}(\cdot, \boldsymbol{Y})$ given by equation (1.5), and $\widehat{\boldsymbol{I}}$ was estimated by fitting a quadratic, as explained in Sections 1.3.2 and 1.3.2 and Appendix 1.6.1. The mean and stan-

dard deviation of the number of function evaluations to find the MAP when started at a random point from the uniform distribution on the parameter space were around 100 and 20, respectively. Nearly all of the points in $\mathcal{B}_O$ produced in **Step 1** were sufficiently separated to be used as part of the experimental design. However, usually just over one third of them were actually valuable for Hessian estimation or surface approximation while the rest were outside of $\widehat{C}_R(\alpha')$ and hence too far away from the mode. Based on 1000 dataset replications, the mean and median numbers of new design points to estimate $\widehat{\boldsymbol{I}}$ by fitting a quadratic, including the 8 points corresponding to forward differencing to obtain an estimate of the diagonal, were 20 and 19, respectively. A small correlation between $\boldsymbol{\beta}$ and λ and small variance of λ, as estimated by the entries of $\widehat{\boldsymbol{I}}^{-1}$, indicate that $[\boldsymbol{\beta}|\boldsymbol{Y}]$ is likely to be close to the conditional density $[\boldsymbol{\beta}|\lambda = \widehat{\lambda}, \boldsymbol{Y}]$.

Experiments were run for elliptical design regions $\widehat{C}_R(\alpha)$ given by equation (1.8), with α in $\{0.2, 0.1, 0.05, 0.01\}$ and numbers of extra experimental design points ($|\mathcal{B}_E|$) in $\{0, 10, 20, \ldots, 100\}$. It was observed that the estimates of the posterior densities based on MCMC samples from approximate surfaces are not very sensitive to the volume of $\widehat{C}_R(\alpha)$ provided there are enough extra design points, with larger regions requiring greater numbers of extra design points. Also, for a fixed α, there is usually little (visual) improvement in density estimates when the size of $\mathcal{B}_E$ grows above 50, and the quality of approximation is often unsatisfactory for the sizes of $\mathcal{B}_E$ below 20.

All graphical summaries of posterior densities for $\boldsymbol{\beta}$ and $F(\boldsymbol{\beta})$ that we report for a single representative dataset correspond to the $\widehat{C}_R(0.1)$ region with 30 extra design points for $\boldsymbol{\beta}$, the same for every surface. In the case of RBF interpolation of $\log\{[\boldsymbol{\beta}, \lambda, \boldsymbol{Y}]\}$, for each $\boldsymbol{\beta}^{(i)} \in \mathcal{B}_D$, we choose 10 design points for λ and fit the

RBF surface at the design points $\{(\boldsymbol{\beta}^{(i)}, \lambda_{ij})$ for $i = 1, \ldots, N$ and $j = 1, \ldots, 10\}$, chosen as outlined in Appendix 1.6.2. All tabular summaries pertain to this RBF approximation to $[\boldsymbol{\beta}, \lambda, \boldsymbol{Y}]$ and the true surface $[\boldsymbol{\beta}, \lambda, \boldsymbol{Y}]$. All results are reported for the autoregressive M-H sampler with $\boldsymbol{\rho} = -0.25$, the density g being the equal-weight mixture of a multivariate normal and Cauchy distributions and with other parameters chosen as discussed in Section 1.3.2.

Samples from the approximate posterior distribution of $F(\boldsymbol{\beta})$ were obtained by first interpolating F at $\boldsymbol{\beta} \in \mathcal{B}_D$ by the (cubic) RBF surface of the form given by the right-hand side of equation (1.9) and then evaluating the resulting interpolant at the MCMC samples from the RBF approximations to the pseudoposterior, to the profile posterior (with and without Laplace correction) and to the joint posterior densities; see Section 1.3.3 for a discussion. Likewise, the sample from the true posterior distribution of $F(\boldsymbol{\beta})$ was obtained by evaluating F at the sample from $[\boldsymbol{\beta}|\boldsymbol{Y}]$.

Figure 1.4.3 presents plots of the *differences between sample quantiles* of the components of $\boldsymbol{\beta}$ based on MCMC samples from the approximate posterior surfaces and the corresponding sample quantiles based on an MCMC run using the exact joint posterior surface (ordinate) *against the sample quantiles* of the components of $\boldsymbol{\beta}$ based on an MCMC run using the exact joint posterior surface (abscissa). Figure 1.4.3 overlays similar plots based on sample quantiles for the exact and approximate posterior distributions of $F(\boldsymbol{\beta})$. (MCMC samples of length 30,000, rather than 10,000, were used for all plots to reduce the variability in the estimates of tail quantiles.) Comparing the magnitudes of the differences between the sample quantiles to the respective interquartile range or to some other measure of dispersion, one can appreciate the accuracy of these

RBF approximations. The plots of the differences between the sample quantiles appeared the most informative to us because the q-q plots of the MCMC sample quantiles from the RBF approximation against those from the exact surface looked like a straight line with slope 1.

Overlaid plots of kernel density estimates (not reported in this paper) for the marginal densities of the components of $\boldsymbol{\beta}$ (and similar plots for $F(\boldsymbol{\beta})$) using MCMC samples from the exact joint posterior density and the approximate posterior densities showed close agreement between the estimates of the exact and approximate densities. Striking similarities between the exact and RBF results in Table 1.1, Figure 1.4.3 and Figure 1.4.3 suggest that our method is capable of achieving nearly the full accuracy of estimation at the expense of only a small fraction of the computational cost required to carry out MCMC sampling using the exact posterior surface.

For each of the 1000 replications of $\boldsymbol{Y}$ under the same model and parameter values but a different realization of noise, we found the MAP and $\widehat{\boldsymbol{I}}$. We then took an MCMC sample of size 10,000 using the true surface $[\boldsymbol{\beta}, \lambda, \boldsymbol{Y}]$ and the respective RBF surface with approximate HPD region $\widehat{C}_R(0.1)$ and the number of extra experimental design points $|\mathcal{B}_E| = 30$. Table 1.2 reports the observed coverage proportions of components of $\boldsymbol{\beta}$ by symmetric credible intervals of sizes 0.9, 0.95 and 0.99, along with the standard errors. The last three columns of Table 1.1 give means and standard deviations for the ratios of the lengths of RBF and exact credible intervals over all datasets.

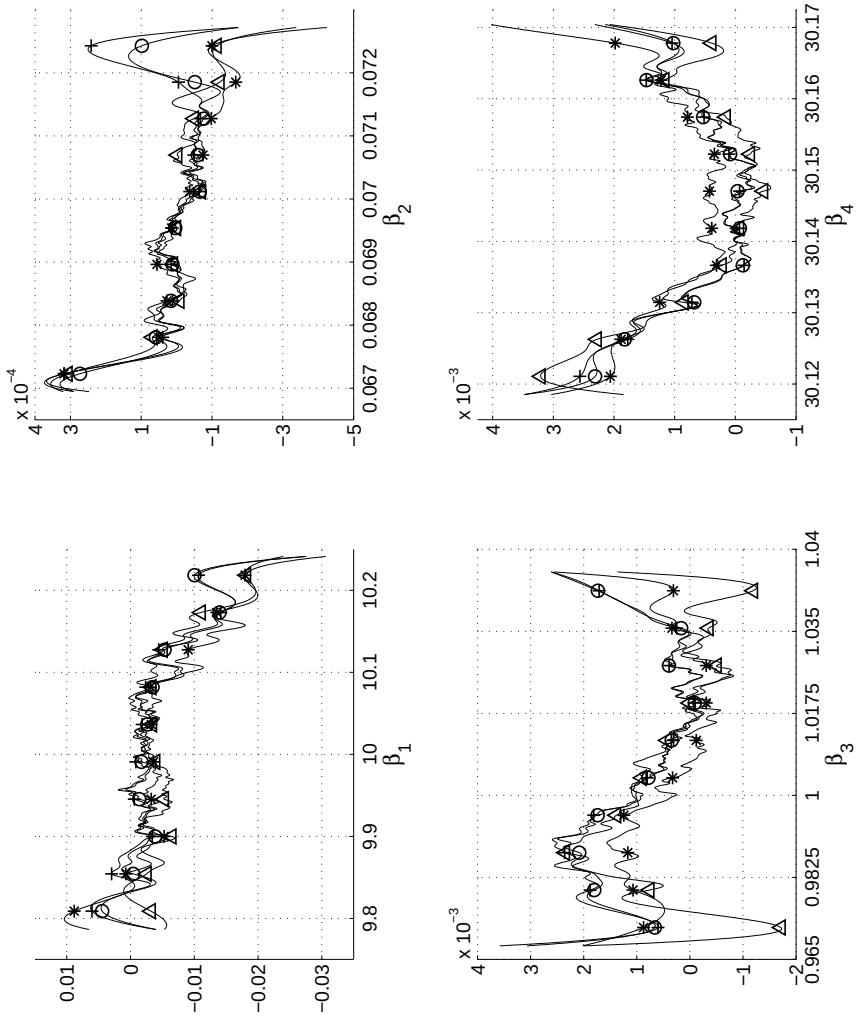

Figure 1.1: Interpolated *pairwise differences between sample quantiles* of $\boldsymbol{\beta}_i$ based on MCMC samples from each approximate posterior surface and the respective sample quantiles of $\boldsymbol{\beta}_i$ based on an MCMC run using the exact joint posterior surface (ordinate) *against the sample quantiles* of $\boldsymbol{\beta}_i$ based on an MCMC run using the exact joint posterior surface (abscissa). All plots are of the form `(approximate minus exact) vs exact` quantiles for the RBF approximations to the joint posterior (`*`), profile posterior with (`+`) and without (`o`) the Laplace correction and pseudoposterior (Δ) densities. Markers are placed at the $(-0.05 + 0.1 \cdot j)$th sample quantiles for $j = 1, 2, \ldots, 10$. These RBF approximations for a single representative dataset use $\widehat{C}_R(0.1)$ and $|\mathcal{B}_E| = 30$. MCMC run length is 30,000.

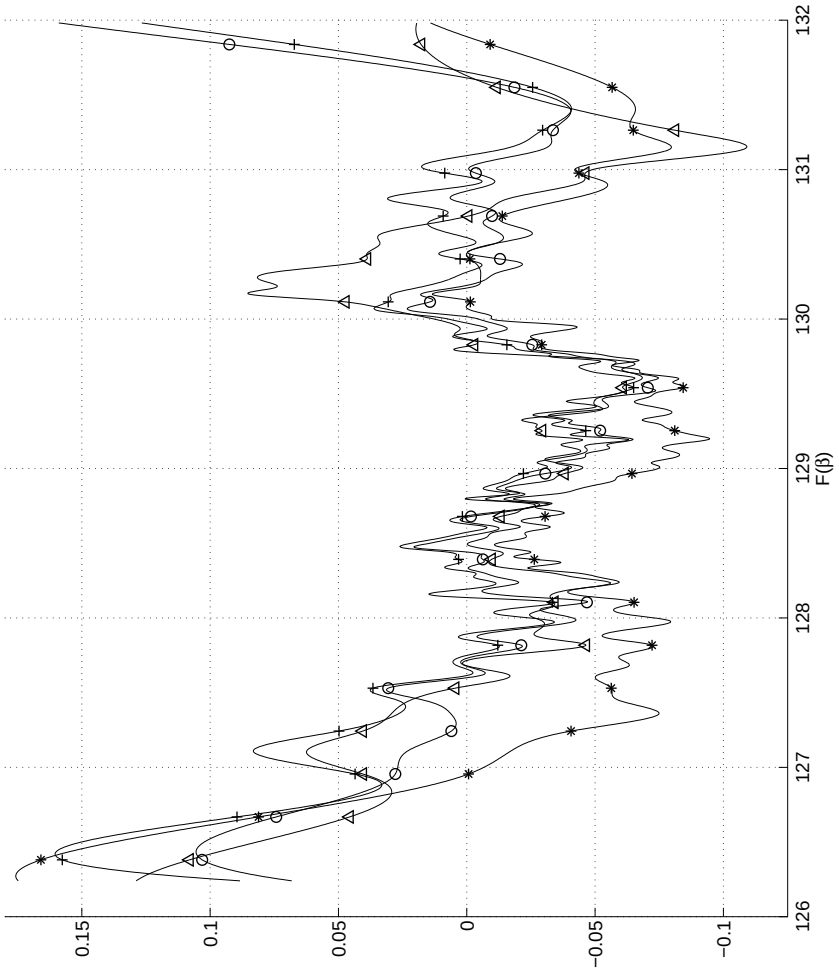

Figure 1.2: Interpolated *pairwise differences between sample quantiles* of $F(\boldsymbol{\beta})$ based on MCMC samples from each approximate posterior surface and the respective sample quantiles of $F(\boldsymbol{\beta})$ based on an MCMC run using the exact joint posterior surface (ordinate) *against the sample quantiles* of $F(\boldsymbol{\beta})$ based on an MCMC run using the exact joint posterior surface (abscissa). Markers are placed at the $(-0.025 + 0.05 \cdot j)$th sample quantiles for $j = 1, 2, \ldots, 20$. The dataset, exact and RBF surfaces and samples $\mathcal{B}_M$, as well as the plot identifiers, are the same as in Figure 1.4.3.

Table 1.2: Observed probabilities of coverage with (standard errors) of symmetric credible intervals based on 1000 dataset replications and joint posterior density as the surface for MCMC. The RBF approximations use, on average, 150 expensive function evaluations compared to 10,000 for the exact results.

	size 0.9 cred. int.			size 0.95 cred. int.			size 0.99 cred. int.	
	exact	RBF		exact	RBF		exact	RBF
β_1	0.905	0.904		0.950	0.944		0.986	0.990
	(0.009)	(0.009)		(0.007)	(0.007)		(0.004)	(0.003)
β_2	0.908	0.903		0.954	0.951		0.991	0.987
	(0.009)	(0.009)		(0.007)	(0.007)		(0.003)	(0.004)
β_3	0.916	0.899		0.953	0.954		0.989	0.988
	(0.009)	(0.010)		(0.007)	(0.007)		(0.003)	(0.003)
β_4	0.904	0.909		0.947	0.945		0.988	0.987
	(0.009)	(0.009)		(0.007)	(0.007)		(0.003)	(0.004)
$F(\boldsymbol{\beta})$	0.904	0.902		0.947	0.937		0.994	0.980
	(0.009)	(0.009)		(0.007)	(0.008)		(0.002)	(0.004)

These results show that the coverage properties and lengths of the credible intervals based on exact and approximate surfaces are similar. The tables also suggest that the approximate credible intervals can serve as frequentist confidence intervals, as the observed coverage proportions are close to the nominal confidence coefficients.

1.5 Discussion

1.5.1 Survey of Literature

Most of the literature dealing with Bayesian calibration of complex computer models focuses on reducing the number of expensive function evaluations via approximation of $\boldsymbol{f}$.

Papers can be roughly divided into two groups. In the first group, papers by O'Hagan, Kennedy and Oakley (1998) and Kennedy and O'Hagan (2000) assume that the model can be run at different levels of complexity and accuracy. (In Kennedy and O'Hagan (2001), the unobservable physical model, approximated by a complex code, plays the role of the top-level code.) Craig et al. (2001) model the unobserved physical process that generates measurements $\boldsymbol{Y}$ as a sum of simulator and inadequacy functions, the latter assumed to have mean zero and a covariance matrix determined by an expert. Goldstein and Rougier (2004, 2006) develop a logical framework for inference about the physical system using multiple simulators (some of them hypothetical) of different quality. In each of these three papers, the simulator $\boldsymbol{f}$ is approximated component-wise. We remark that so long as the likelihood of the data $\boldsymbol{Y}$ can be evaluated, our al-

gorithm of Section 1.3.2 can be applied to any of the models from these papers.

In the second group of papers, Higdon, Lee and Holloman (2003) run coarse and fine (corresponding to the original expensive model) Markov chains in tandem and use information from the faster-mixing coarse chain to improve mixing of the fine chain. Christen and Fox (2005) use a cheap-to-evaluate approximation to the unnormalized posterior density, to evaluate the expensive posterior density only for the MCMC moves that are likely to be accepted. The emphasis, however, is on the models for which approximation to the posterior density is obtained by replacing $\boldsymbol{f}$ by an approximation, for example, by linearization. The approach of Rasmussen (2003) uses a GP interpolant of the logarithm of the posterior density and of its first derivatives to generate proposal states for a Hybrid Monte Carlo algorithm. Each of these three papers requires at least one evaluation of the expensive posterior density for each accepted state.

1.5.2 Differences from Earlier Approaches

The route we take is significantly different from those mentioned. First, we do not use coarse and inexpensive versions of the expensive code, because in our experience these often do not exist. Second, given our interest in the sample from the posterior density for $\boldsymbol{\beta}$, we approximate the (scalar-valued) posterior density *directly*, and not through approximation of the high-dimensional model output $\boldsymbol{f}$. Third, we realize that in many problems sampling thousands of times from the exact posterior surface is not computationally feasible and thus work solely with the approximation, unlike Higdon et al. (2003) and Christen and Fox (2005).

By virtue of working with GP interpolants (see our Section 3.2.3), the paper of Rasmussen (2003) has a number of similarities with ours, although our attention is not restricted to Hybrid Monte Carlo. The main requirement that the two approaches share is that the logarithm of the posterior density must be approximated well on a HPD region. However, in order to make a GP or RBF approximation strategy practical, one needs to resolve several issues, which are not addressed in Rasmussen's work. First, since interpolation suffers from the curse of dimensionality, it makes sense to separate explicitly the argument of the posterior distribution into simulator ($\boldsymbol{\beta}$) and non-simulator ($\boldsymbol{\zeta}$) parameters. When evaluation of $\boldsymbol{f}$ is the main computational bottleneck, it is beneficial to evaluate the posterior density for multiple values of $\boldsymbol{\zeta}$ for the same $\boldsymbol{\beta}$, which can increase the number of design points by orders of magnitude. Second, the sequential design procedure of Rasmussen does not guarantee that the whole HPD region is covered when the number of allowed runs of $\boldsymbol{f}$ is fixed and small. We avoid this problem by defining the design region explicitly. Third, any sampler drawing from an approximate posterior density must be restricted to the region where the approximation is good (the approximate HPD region); otherwise, a large mass of the proposal density may be in a region of low probability under the exact posterior distribution. Fourth, if one attempts to generate variates from the exact posterior distribution, the computational budget needs to be split in advance into parts for approximation and sampling. If the approximation is accurate, there is not much extra benefit from evaluating the exact posterior density in the MCMC run, as Rasmussen's first example shows; otherwise, the sampler will be wasting simulator runs for the states rejected by M-H. Since accurate approximation is possible only at the expense of the MCMC run length, we allocate the whole budget to interpolation and sample only the ap-

proximate density. With our approach, optimality of placement of design points can (potentially) be enforced, whereas, even if design points from the sampling stage are re-used to improve the approximation (which Rasmussen does not consider), one has no control of their placement. Our work addresses all of these concerns.

1.5.3 Other Considerations (Limitations and Extensions)

A potential weakness of our algorithm is its reliance on the quadratic approximation of the logarithm of the posterior density used to define the design region. If the MAP happens to lie on the boundary of the parameter space, it will not be possible to obtain an approximate HPD region using equation (1.8); however, encountering this situation is likely to be a sign of a misspecified model or parameter space. If there is extreme skewness in the posterior density, then an asymmetric approximation to a HPD region is expected to be superior to our elliptical region.

In order to ensure that the design region $\widehat{C}_R(\alpha)$ covers the true HPD region adequately, the parameter α that controls the size of the region should be tuned in practice. Starting with a "smaller" $\widehat{C}_R(\alpha_0)$ and design points on it, let $\alpha_1 < \alpha_0$ and choose additional design points in the region $\widehat{C}_R(\alpha_1)$ that contains $\widehat{C}_R(\alpha_0)$. Then MCMC samples from the two approximate densities restricted to the respective design regions can be obtained and compared by means of a distribution test, controlling for dependence. Ideally, one would continue to "grow" the design region until a "discrepancy" between consecutive MCMC runs becomes small. To allow more general region shapes, one would start with an initial (e.g.,

elliptical) region and "grow" it outwards in the directions where the RBF surface is highest using the feedback from preceding MCMC runs. This is one of the lines of our current work.

Using an MCMC sample from the approximate posterior surface restricted to a HPD region is likely to produce accurate estimates of non-extreme quantiles, but estimated moments may be misleading. We are not aware of work on approximation of expensive models that resolves this issue.

In applications, the component-wise output of $\boldsymbol{f}$ may be discontinuous on a very fine scale due to discretization, e.g., when an appropriate system of differential equations is solved numerically. Theoretical convergence results for the corresponding exact posterior densities are not applicable. Nevertheless, our approach still captures the shape of the true surface – by maintaining separation of design points we are essentially interpolating a smooth version of the exact posterior density.

Consider two methods for approximating the posterior density. The *direct method*, which we use, approximates the logarithm of the posterior density itself. The *indirect method* approximates each component of $\boldsymbol{f}$ and plugs these into the logarithm of the posterior density. One advantage of the direct method is that it approximates the scalar-valued log-posterior surface, whereas the indirect method must approximate a surface whose dimension can be quite high, e.g., 1,000 in the example in Section 1.4—although this is a synthetic example, it is typical of many actual applications. With the indirect method, there is an interpolation error for each component of $\boldsymbol{f}$, and the cumulative effect of component-wise approximation errors is unclear but could be large. If the indirect method is applied by modeling $\boldsymbol{f}$ as a multivariate GP, then specifica-

tion of the cross-covariance matrix requires substantial subject-matter expertise, whereas interpolation of the log-posterior surface by the direct method can be automated.

A referee asked whether, because the direct method does not interpolate $\boldsymbol{f}$, one can check the goodness-of-fit of the model. For diagnostics, one would normally use the simulator output only at a single point estimate of $\boldsymbol{\beta}$, say the MAP $\widehat{\boldsymbol{\beta}}$, to compute residuals. This value $\boldsymbol{f}(\widehat{\boldsymbol{\beta}})$ is known from **Step 1** of the algorithm, so no extra computation is required.

1.5.4 Summary and Conclusions

This paper presented a Bayesian calibration method suitable when the allowed number of evaluations of the computationally expensive simulator $\boldsymbol{f}^*$ is relatively small and no inexpensive approximation to it is available. Sampling the exact posterior distribution many thousands of times during an MCMC run is questionable, if feasible, under such restrictions.

The main contribution is the algorithm of Section 1.3.2, which re-uses a subset of well-separated design points from a derivative-free optimization search (**Step 1**), augmented with additional design points (**Step 2**), to build an RBF approximation for the posterior density on the region of high posterior probability (**Step 3**). This allows one to draw arbitrarily long samples from the cheap proxy to the true expensive posterior density in **Step 4**. Derivative-based optimization routines that use finite differences are undesirable for **Step 1** since they produce clusters of nearby design points that carry little new information about the log-posterior surface once the surface value at any one of these points

is known. Furthermore, all points in each cluster cannot be re-used in the RBF interpolation without creating numerical instability in the linear system (1.12) of Appendix 1.6.3.

In our experiments presented in Section 1.4, a very accurate approximation to the exact posterior density was obtained, on average, using 150 runs of the simulator (**Step 1** and **Step 2**). The computational effort of our approach is well over an order of magnitude below that required to carry out several thousands of steps in an MCMC run using the exact posterior density. Our method hence shows promise as a means for doing a rigorous Bayesian uncertainty analysis on some functions (including simulation models) for which there currently does not exist a numerically feasible alternative method.

1.5.5 Further Developments

Our current work focuses on extending the approach to deal with $\boldsymbol{f}$ under less restrictive smoothness assumptions, posterior densities with multiple important modes and pronounced skewness, and on developing a sequential procedure for determining an approximate HPD region and an experimental design on it. Under the GP model, we have devised and are currently studying the properties of algorithms to sample from the densities determined by the individual realizations of the conditional GP and to integrate out the uncertainty due to approximation by MCMC. A systematic study of the effect of space-time dependence is also needed. After more insight into these issues is gained, the methodology will be applied to a truly expensive model.

1.6 Appendix

1.6.1 Estimation of $\widehat{\boldsymbol{I}}$

Let $p = \dim(\boldsymbol{\beta})$ and $u = \dim(\boldsymbol{\zeta})$. To approximate the Hessian $\widehat{\boldsymbol{I}}$ of $-\log\{[\boldsymbol{\beta}, \boldsymbol{\zeta}, \boldsymbol{Y}]\}$ at $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\zeta}})$ by forward differences generally requires (around) $(p+u+1)(p+u+2)/2$ function evaluations. Partition $\widehat{\boldsymbol{I}}$ as in equation (1.7). In our problem, $\widehat{\boldsymbol{I}}_{\boldsymbol{\zeta}\boldsymbol{\zeta}}$ can be found analytically, and the off-diagonal blocks of $\widehat{\boldsymbol{I}}$ can be computed entirely from the evaluations of $\boldsymbol{f}$ used to estimate the diagonal of $\widehat{\boldsymbol{I}}_{\boldsymbol{\beta}\boldsymbol{\beta}}$.

Unfortunately, the $(p+1)(p+2)/2$ points for evaluation of $\boldsymbol{f}$ by finite differences to compute $\widehat{\boldsymbol{I}}_{\boldsymbol{\beta}\boldsymbol{\beta}}$ are very close to $\widehat{\boldsymbol{\beta}}$ and are not valuable for surface approximation. However, one can lower the number of uninformative design points using the approach below.

Taylor's theorem suggests the approximation

$$\log\{[\boldsymbol{\beta}, \widehat{\boldsymbol{\zeta}}(\widehat{\boldsymbol{\beta}}), \boldsymbol{Y}]\} \approx \text{const} - \frac{1}{2}\|\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}\|^2_{\widehat{\boldsymbol{I}}_{\boldsymbol{\beta}\boldsymbol{\beta}}}$$

on the ellipsoid $\mathcal{E}(c) = \{\boldsymbol{\beta} : \|\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}\|^2_{\widehat{\boldsymbol{I}}_{\boldsymbol{\beta}\boldsymbol{\beta}}} \leq c^2\}$ for some c. We propose to choose $(p+1)(p+2)/2$ design points that are well-separated inside $\mathcal{E}(c)$, fit a quadratic surface through them and estimate $\widehat{\boldsymbol{I}}_{\boldsymbol{\beta}\boldsymbol{\beta}}$ by the Hessian of the quadratic. The task is to ensure that these points lie inside $\mathcal{E}(c)$ without knowing the shape and orientation of the ellipsoid. Denote by $\boldsymbol{e}_i$ the ith standard basis vector for $\mathbb{R}^p$ and notice that the boundary of $\mathcal{E}(c)$ passes through points $\widehat{\boldsymbol{\beta}} \pm \boldsymbol{b}_i$, where $\boldsymbol{b}_i = \boldsymbol{e}_i \cdot c/\sqrt{\widehat{\boldsymbol{I}}_{\boldsymbol{\beta}\boldsymbol{\beta}}(i,i)}$ and $\widehat{\boldsymbol{I}}_{\boldsymbol{\beta}\boldsymbol{\beta}}(i,i)$ is the ith diagonal entry of $\widehat{\boldsymbol{I}}_{\boldsymbol{\beta}\boldsymbol{\beta}}$. The convex

hull of these points

$$\mathcal{H}(c) = \left\{ \begin{array}{c} \boldsymbol{\beta}: \quad \boldsymbol{\beta} = \widehat{\boldsymbol{\beta}} + \sum_{i=1}^{p} (\psi_{i,1} - \psi_{i,2}) \boldsymbol{b}_i \text{ such that } \sum_{j=1}^{2} \sum_{i=1}^{p} \psi_{i,j} = 1 \\ \text{and } \psi_{i,j} \geq 0 \text{ for } i = 1, \ldots, p \text{ and } j = 1, 2 \end{array} \right\}$$

is a subset of $\mathcal{E}(c)$, and so it is guaranteed that any experimental design on $\mathcal{H}(c)$ also lies in $\mathcal{E}(c)$. Hence one only needs to estimate the diagonal of $\widehat{\boldsymbol{I}}_{\boldsymbol{\beta}\boldsymbol{\beta}}$ by forward differences using at most $2 \cdot p$ extra function evaluations, half of which are reused in computation of the off-diagonal blocks $\widehat{\boldsymbol{I}}_{\boldsymbol{\beta}\boldsymbol{\zeta}}$. (Thus we have reduced the number of "uninformative" design points roughly by $(p+1)(p-2)/2$.)

The argument c in the definition of the ellipsoid is to be chosen by the experimenters in accord with their beliefs. It is helpful to think of c^2 as a quantile of the χ^2_p distribution that defines a confidence ellipsoid for the multivariate normal approximation to $[\boldsymbol{\beta}|\boldsymbol{\zeta} = \widehat{\boldsymbol{\zeta}}, \boldsymbol{Y}]$. Large values of c often yield Hessians that are inaccurate or not positive definite, and very small values result in new design points close to $\widehat{\boldsymbol{\beta}}$. We had some success with the following procedure to estimate $\widehat{\boldsymbol{I}}$ and to produce the set $\mathcal{B}_H$ from Section 1.3.2:

1. Initialization:
   (a) Choose a moderate initial value of c, say, $\sqrt{\chi^2_{p,0.1}}$.
   (b) Set $\mathcal{B}_H = \mathcal{B}_O$ and remove from $\mathcal{B}_H$ points very close to each other. The set $\mathcal{B}_H$ will contain values of $\boldsymbol{\beta}$ for which $\boldsymbol{f}^*$ has been computed. Assume for now that $|\mathcal{B}_H| \leq (p+1)(p+2)/2$.
2. Augment $\mathcal{B}_H$ with new well-separated points so that $|\mathcal{B}_H \cap \mathcal{H}(c)| = (p+1)(p+2)/2$ and evaluate $\boldsymbol{f}^*$ at the new points.
3. Fit a quadratic surface through the points in $\mathcal{B}_H \cap \mathcal{H}(c)$ and plug its Hessian into the expression for $\widehat{\boldsymbol{I}}$. If the resulting estimate of $\widehat{\boldsymbol{I}}$ (not only that of $\widehat{\boldsymbol{I}}_{\boldsymbol{\beta}\boldsymbol{\beta}}$)

is not positive definite, reduce c and return to the previous step. Otherwise terminate; return $\widehat{\boldsymbol{I}}$ and (the set difference) $\mathcal{B}_H = \mathcal{B}_H - \mathcal{B}_O$.

If $|\mathcal{B}_H| > (p+1)(p+2)/2$ in step 1(b) above, then no new design points are necessary and one starts by working with subsets of $\mathcal{B}_H$. Once they are exhausted, one moves to Step 2.

1.6.2 Choice of Design Points

While forming $\mathcal{B}_H$ and $\mathcal{B}_E$, we require that the new design points lie far from each other and from the points where $\boldsymbol{f}^*$ has been evaluated previously ("fixed points"). This objective is related to the *maximin* criterion that attempts to *max*imize the *min*imum between-point distance over all pairs of design points (Santner, Williams and Notz (2003, sec. 5.3)). Generating such designs exactly is computationally difficult, and usually one is happy to obtain a good approximate maximin design (Trosset 1999).

As we remarked in Section 1.3.2, ideally we would like a design that minimizes the coverage radius of design points (minimax design). Johnson, Moore and Ylvisaker (1990) argue that minimax design minimizes maximum prediction variance in GP interpolation when intersite correlation is low, thereby linking optimality of designs for GP and RBF interpolation. As a heuristic, we bound the minimax distance by the intersite distance of the optimal maximin design, and then find an approximate maximin design.

We devised a simple "greedy" algorithm to update the set of fixed points with N_E extra design points: (i) choose $\kappa > 1$ and draw $\lceil \kappa \cdot N_E \rceil$ *candidate* points

uniformly at random on the design region; (ii) at the jth iteration, find the pair of points closest to each other that has at least one candidate point; if it has a single candidate point, delete it, otherwise delete the one that is closest to the remaining (fixed and candidate) points, until only N_E candidate points remain. This algorithm is applied to update $\mathcal{B}_O$ to produce $\mathcal{B}_H$ and then to augment $\mathcal{B}_O \cup \mathcal{B}_H$ with $\mathcal{B}_E$. As an intermediate step, one has to sample uniformly inside polytopes and spheres; for discussion, see Devroye (1986, chap. 5).

To obtain the (joint) experimental design for $\boldsymbol{\beta}$ and $\boldsymbol{\zeta}$ for fitting an RBF surface to $\log\{[\boldsymbol{\beta}, \boldsymbol{\zeta}, \boldsymbol{Y}]\}$, we start with a (marginal) design for $\boldsymbol{\beta}$ on $\widehat{C}_R(\alpha)$ as in Section 1.3.2 and augment each design point $\boldsymbol{\beta}^{(i)} \in \mathcal{B}_D$ with a (conditional) design for $\boldsymbol{\zeta}$ based on the multivariate normal approximation to $[\boldsymbol{\zeta}|\boldsymbol{\beta} = \boldsymbol{\beta}^{(i)}, \boldsymbol{Y}]$, derived from equation (1.6). As a consequence, the increase in the dimension of the argument of the posterior density (going from $\pi(\cdot, \boldsymbol{Y})$ to $[\boldsymbol{\beta}, \boldsymbol{\zeta}, \boldsymbol{Y}]$) in this problem need not translate into the increase in the number of evaluations of $\boldsymbol{f}$.

1.6.3 Details for Fitting the RBF Surface

We now describe the procedure for fitting the RBF interpolation model of Section 1.3.2 with the cubic basis function and a linear polynomial tail $q(\boldsymbol{\beta}) = (1, \boldsymbol{\beta}^{\mathsf{T}}) \cdot \boldsymbol{c}$. Discussion of fitting for other choices of basis functions is in Powell (1996).

Define the matrix $\boldsymbol{\Phi} \in \mathbb{R}^{N \times N}$ by: $\boldsymbol{\Phi}_{i,j} = \phi(\|\boldsymbol{\beta}^{(i)} - \boldsymbol{\beta}^{(j)}\|_2)$, for $i, j = 1, \ldots, N$. Let $\boldsymbol{P} \in \mathbb{R}^{N \times (p+1)}$ be the matrix with $(1, \{\boldsymbol{\beta}^{(i)}\}^{\mathsf{T}})$ as the ith row for $i = 1, \ldots, N$. The coefficients for the RBF surface that interpolates $l(\cdot) = \log(\pi(\cdot, \boldsymbol{Y}))$ at the

points $\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(N)}$ are obtained by solving the system

$$\begin{pmatrix} \boldsymbol{\Phi} & \boldsymbol{P} \\ \boldsymbol{P}^{\mathsf{T}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{c} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mathcal{L}} \\ \mathbf{0} \end{pmatrix}, \tag{1.12}$$

where $\boldsymbol{\mathcal{L}} = \left[l(\boldsymbol{\beta}^{(1)}), \ldots, l(\boldsymbol{\beta}^{(N)}) \right]^{\mathsf{T}}$, $\boldsymbol{a} \in \mathbb{R}^N$ and $\boldsymbol{c} \in \mathbb{R}^{p+1}$.

The coefficient matrix in equation (1.12) is invertible if and only if the rank of $\boldsymbol{P}$ is $p+1$ (Powell 1992). For the case of a cubic RBF with a linear tail, this holds if and only if the set of (distinct) design points contains $p+1$ points that are *affinely independent*. For stability purposes, we solve equation (1.12) by means of matrix factorizations, as described in Powell (1996).

1.6.4 Details on Integrating C out

Let $\boldsymbol{Z} = \boldsymbol{Z}(\boldsymbol{\beta}, \boldsymbol{\lambda})$ be the matrix with the ith row $[h\{Y_i, \boldsymbol{\lambda}\} - h\{f(X_i, \boldsymbol{\beta}), \boldsymbol{\lambda}\}]^{\mathsf{T}}$ for $i = 1, \ldots, n$, $\boldsymbol{R}$ be the upper-triangular Cholesky factor of $\boldsymbol{S}(\boldsymbol{\gamma})$ and $\widetilde{\boldsymbol{Z}} = \boldsymbol{R}^{-\mathsf{T}}\boldsymbol{Z}$. Notice that, under the separable covariance model $\boldsymbol{\Sigma}(\boldsymbol{\theta}) = \boldsymbol{S}(\boldsymbol{\gamma}) \otimes \boldsymbol{C}$ of Section 1.2, the likelihood equation (1.2) implies that the rows $\widetilde{\boldsymbol{Z}}_{1,\bullet}, \ldots, \widetilde{\boldsymbol{Z}}_{n,\bullet}$ of $\widetilde{\boldsymbol{Z}}$ are *i.i.d.* $MVN(\mathbf{0}, \boldsymbol{C})$. Notice that

$$\begin{aligned} [\boldsymbol{Y}|\boldsymbol{\beta}, \boldsymbol{\lambda}, (\boldsymbol{\gamma}, \boldsymbol{C})] &\propto |J_h(\boldsymbol{Y}; \boldsymbol{\lambda})| \cdot |\boldsymbol{S}(\boldsymbol{\gamma})|^{-d/2} |\boldsymbol{C}|^{-n/2} \prod_{j=1}^{n} \exp\left(-0.5 \cdot \|\widetilde{\boldsymbol{Z}}_{j,\bullet}\|^2_{\boldsymbol{C}^{-1}} \right) \\ &= |J_h(\boldsymbol{Y}; \boldsymbol{\lambda})| \cdot |\boldsymbol{S}(\boldsymbol{\gamma})|^{-d/2} |\boldsymbol{C}|^{-n/2} \exp\left(-0.5 \cdot tr\{\boldsymbol{C}^{-1} \widetilde{\boldsymbol{Z}}^{\mathsf{T}} \widetilde{\boldsymbol{Z}}\} \right). \end{aligned}$$

We put a Wishart prior density on $\boldsymbol{C}^{-1}$ and assume that, a priori, $\boldsymbol{C}$ and $(\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\gamma})$ are independent, so that

$$[\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\gamma}, \boldsymbol{C}^{-1}] \propto |\Delta|^a |\boldsymbol{C}^{-1}|^{a-(d+1)/2} \exp\left(-tr\{\Delta \boldsymbol{C}^{-1}\} \right) \cdot [\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\gamma}],$$

where $a > (d-1)/2$, $\Delta \in \mathcal{M}_d$, the space of $d \times d$ symmetric positive definite matrices, and $tr(\cdot)$ is the trace operator. This allows us to integrate $\boldsymbol{C}$ out of the joint posterior density analytically:

$$\begin{aligned}
[\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\gamma}, \boldsymbol{Y}] &= \int_{\mathcal{M}_d} [\boldsymbol{Y}|\boldsymbol{\beta}, \boldsymbol{\lambda}, (\boldsymbol{\gamma}, \boldsymbol{C})] \cdot [\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\gamma}, \boldsymbol{C}^{-1}] d\boldsymbol{C}^{-1} \propto c(\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\gamma}) \cdot \\
&\cdot \int_{\mathcal{M}_d} \exp\left(-tr\{\boldsymbol{C}^{-1}(\widetilde{\boldsymbol{Z}}^{\mathsf{T}}\widetilde{\boldsymbol{Z}}/2 + \Delta)\}\right) |\boldsymbol{C}^{-1}|^{a+(n-d-1)/2} d\boldsymbol{C}^{-1} \\
&\propto c(\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\gamma}) \cdot |\widetilde{\boldsymbol{Z}}^{\mathsf{T}}\widetilde{\boldsymbol{Z}}/2 + \Delta|^{-(a+n/2)} \\
&= [\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\gamma}] \cdot |J_h(\boldsymbol{Y}; \boldsymbol{\lambda})| \cdot |\boldsymbol{S}(\boldsymbol{\gamma})|^{-d/2} \cdot |\boldsymbol{Z}^{\mathsf{T}}[\boldsymbol{S}(\boldsymbol{\gamma})]^{-1}\boldsymbol{Z}/2 + \Delta|^{-(a+n/2)}.
\end{aligned}$$

CHAPTER 2

A DERIVATIVE-FREE APPROACH TO APPROXIMATION OF COMPUTATIONALLY EXPENSIVE POSTERIOR DENSITIES, WITH APPLICATION TO PARAMETER UNCERTAINTY ANALYSIS FOR A WATERSHED MODEL

2.1 Introduction

The key step of Bayesian inference is expression of uncertainty about model parameters η given the observed data Y using the posterior density π of η. In most practical applications, functionals of π, such as integrals with respect to π, cannot be found in closed form and can only be estimated from a Markov Chain Monte Carlo (MCMC) sample from π. However, when π is computationally expensive to evaluate, long MCMC runs are intractable, which causes the estimation error to be high. Therefore, it is necessary to construct an approximation $\widetilde{\pi}$ of π.

The focus of our work is reduction of the cost of MCMC via nonparametric approximation of computationally expensive posterior densities for which derivatives are not available. After an accurate approximate posterior density $\widetilde{\pi}$ has been produced, one can sample $\widetilde{\pi}$ using MCMC, which is computationally inexpensive, to obtain arbitrarily large effective sample sizes.

Of particular interest to us are posterior densities that arise in computationally intensive nonlinear regression problems. In such problems, the nonlinear regression function is simulated by a complex computer code f, known as *simulator*. For example, in the watershed modeling problems analyzed in Shoe-

maker *et al.* (2007) and Tolson and Shoemaker (2007a, 2007b), for a given parameter value β a single run of the simulator produces a vector $f(\beta)$ of time series of daily average water flows. The vector of observed data Y is modeled as $Y = f(\beta) + e$, where f is a nonlinear regression function and e is the vector of errors whose distribution depends on additional parameters ζ. Specification of the density of e and of the prior densities for β and ζ determines the joint posterior density π. (The exact posterior density specification for our model is provided in Section 2.4.) Obtaining large MCMC samples from π is computationally intractable since acceptance of a candidate state in the Markov chain requires an evaluation of f. Each run of f can take from several seconds to a few hours depending on the application.

The primary contribution of this paper is an algorithm to approximate π by interpolation using radial basis functions (RBFs) on the high probability density (HPD) region under π. Approximation of π over the whole parameter space $\mathfrak{E}$ for η is computationally wasteful since the volume of $\mathfrak{E}$ typically exceeds that of the HPD region by orders of magnitude. Building the approximation is nontrivial since the shape of the HPD region is unknown and only a limited number of evaluations of π is allowed because of the constraints on the computational budget. The problem is further complicated by potential nonsmoothness of π due to discretization in the simulator (Shoemaker *et al.* (2007), Benaman *et al.* (2005), Mugunthan *et al.* (2005), Mugunthan and Shoemaker (2006)). This roughness makes existing approximation approaches that assume derivatives (Rasmussen, 2003; Bliznyuk et al., 2008) unattractive.

After having introduced our algorithm in Section 2.3, we apply it to estimate the posterior density of the simulator parameters given the real data from the

Town Brook watershed. The statistical model that we introduce in Section 2.4 can accomodate non-normality of errors via transformations, as well as dependence in the data.

The procedure we propose is a derivative-free extension of the approach of Bliznyuk *et al.* (2008). The initial approximate HPD region is reached by a derivative-free optimization algorithm. Our iterative algorithm `GRIMA` is based on the interplay of two steps: (i) determination of the approximate HPD region (*design region*) using a cheap-to-obtain MCMC sample from the surrogate density $\widetilde{\pi}$ and (ii) choice of additional knots for interpolation on the design region in order to improve the approximation $\widetilde{\pi}$ to π. This typically entails enlarging ("growing") the initial design region during early stages of the algorithm and improvement of the approximation during the later stages. (Hence the acronym `GRIMA`—"`G`row the (design) `R`egion and `IM`prove the `A`pproximation".) The algorithm is terminated when a discrepancy measure, such as the total variation norm, between consecutive approximate densities becomes negligible.

Because of its generality, our algorithm can be applied without modifications to approximation of posterior densities in contexts other than nonlinear regression. For example, in our current work we successfully approximated the posterior density for parameters in a high-dimensional linear model with space-time dependence (Bliznyuk et al, 2008, in preparation or submitted). `GRIMA` only requires that one can evaluate the unnormalized posterior density for a given parameter value; closed-form specification of the posterior density is not necessary.

This paper is organized as follows: Section 2.2 contains a brief survey of the current literature, contrasts our approach with existing methods and explains

why GRIMA is necessary. In Section 2.3.1 we concentrate on the main ideas behind our algorithm and summarize the features of the problem of density approximation by interpolation. We outline the GRIMA algorithm and discuss its practical implementation in Section 2.3.2. It is illustrated and is compared to the results of Rasmussen (2003) in Section 2.3.3 on his 2-dimensional test problem. This section enables the reader to visualize the progress of GRIMA without the need to follow all technical details of Section 2.3.2. In Section 2.4 we search for an adequate statistical model for the real time series Y of the water flow in the Town Brook watershed and apply GRIMA to estimate the posterior density of the simulator parameters β given the data Y. Possible extensions are briefly outlined in Section 2.5. Appendices provide information necessary for efficient implementation of the procedures discussed in this paper.

2.2 Earlier Work and Our Contribution

Bayesian inference about parameters of the nonlinear regression function is known in the literature as *Bayesian calibration* (Kennedy and O'Hagan, 2001). A recent review of the work on Bayesian calibration appears in Section 5 of Bliznyuk et al. (2008). Here we review some of this literature relevant to the approximation of the computationally intensive posterior densities in nonlinear regression problems.

2.2.1 Literature Review and Our Contribution

The landmark paper by Kennedy and O'Hagan (2001) emphasized the importance of finding the set of sites where to run the simulator for calibration (*design points*). Even though Kennedy and O'Hagan argue that design points for the vector of calibration parameters needs to be chosen sequentially "over the range covered by its posterior distribution" (Kennedy and O'Hagan, 2001, p. 441), they do not pursue this path and instead use a Latin hypercube design over the whole parameter space. Doing this may be meaningful for their application but we have serious reservations about the appropriateness of this design when the HPD region is a small subset of the parameter space. For example, in our application in Section 2.4 it turns out that the volume of the HPD region containing nearly all of the mass of the posterior density constitutes only about .002% of the volume of the parameter space. This implies that nearly all of the design points would be wasted if the approximation were done over the whole parameter space. In Section 5.1, Kennedy and O'Hagan conjectured that a sequential design procedure similar to that used for optimization of complex computer codes using surrogate surfaces (Bernardo et al. 1992, Aslett et al. 1998) may be used for experimental design. Unfortunately, this is not the case because knowledge of the values of the objective function (posterior density in our paper) for the optimization trajectory by itself does not contain any information about the HPD region. To approximate the HPD region, it is necessary to use the design points to specify an approximate probability density to which the HPD region conforms. Such a surrogate density can be obtained using interpolation.

Two papers that approximate the scalar-valued density by interpolation *directly*, not through the component-wise approximation of the multidimensional

simulator f, are Rasmussen (2003) and Bliznyuk *et al.* (2008). Rasmussen (2003) uses *best linear unbiased prediction (BLUP)* under a *Gaussian processes (GPs)* model to interpolate the logarithm l of π and its gradient (assumed to be available) to obtain a proposal density for the Hybrid Monte Carlo sampler. However, since his approximation is not restricted to a high probability density (HPD) region, the sampler is susceptible to wasting evaluations for the candidate states where proposal density is high due to approximation error but the target density is low. The paper of Bliznyuk *et al.* (2008) uses an RBF interpolant of l over an estimated HPD region of π to obtain an approximation $\widetilde{\pi}$ to π. The HPD region is estimated by a confidence ellipsoid from the local quadratic approximation of l at the mode of π, assumed to be unique, which is reached using formal optimization. However, if the curvature of l near the mode carries little information about the shape of the HPD region or the mode is at the boundary of the parameter space, the estimated HPD region may be misleading or even undefined. To summarize, the major drawback of both of these approaches is in their reliance on the existence of at least one derivative of l, which is not available in a host of practical problems that we pointed at in Section 2.1.

In this work, we build on the approach of Bliznyuk et al. (2008). Our practical iterative scheme constructs a surrogate surface $\widetilde{l}$ by interpolation of the logarithm l of the posterior density π. Our goal is to choose design points on the unknown HPD region under π. Notice that $\exp(\widetilde{l})$ is a valid unnormalized probability density that is cheap to evaluate. We estimate the minimum height of $\widetilde{l}$ over the HPD region (of a given level) for $\exp(\widetilde{l})$ using an MCMC sample from $\exp(\widetilde{l})$. This information is used to decide whether $\widetilde{l}$ is "high enough" at a candidate design point η^* (chosen a certain distance away from the existing design points) and whether the true log-posterior l should be evaluated at η^* to

refine the surrogate density. This is the main idea of our approach.

The choice of which class of interpolants to use, RBFs or BLUPs under GPs (known in spatial statistics as *kriging*), as well as whether to interpolate l directly or via interpolation of each component of f, is of secondary importance. Any of these interpolants may be used to define a surrogate density in our framework. Below we outline why the *direct* RBF model may be preferable.

2.2.2 Choice of the Interpolant

For a given value of basis or covariance function parameters, coefficients of the RBF or kriging interpolants can be computed by solving the linear system of dual kriging equations (chap. 4.4.5 of Cressie, 1991), which have the same form for either interpolant. An advantage of RBF interpolants is in the lack of basis function parameters for some practical RBF choices, which makes RBF fitting easier than fitting a kriging model that requires estimation of parameters of the covariance function.

We found the assumption that the log-posterior l is a realization of a GP to be often misleading and, consequently, estimation of the covariance function parameters by MLE to be inefficient. For example, approximation to a fixed accuracy of a quadratic function (that corresponds to the logarithm of a multivariate normal density) by kriging with parameters estimated by MLE under a GP model requires considerably more (e.g., twice as many for a 10-dimensional problem) design points than when the parameters are estimated by K-fold cross-validation under the same model without the assumption of normality. An RBF interpolant used in this paper performs on par with or bet-

ter than kriging with K-fold cross-validation.

Bliznyuk et al. (2008) noted in Section 3.2.3 that prediction error variance from the kriging predictor may be used to select sites for new simulator runs. This does not appear to us to be a great advantage of the kriging over RBF model because the prediction error variance under the GP model is completely determined by the (subjective) choice of the covariance function and locations of the design points (e.g., Cressie, 1991). Put differently, unlike the BLUP itself, the variance of the error from prediction with the BLUP does not depend on the GP values at the design points. Choosing new design points to reduce the maximum prediction error variance can be used to maintain separation between design points. In our work, we control the interpoint distance explicitly.

Indirect approximation of l via interpolation of each component of the simulator f is also possible but is not considered in this paper. In our other work, indirect approximation of l using RBFs produced results similar to the direct interpolation of l under RBF and kriging models. Modeling and fitting issues for the indirect approximation under the kriging model are nontrivial when the dimension of f is high.

To summarize, we did not find kriging to perform better than RBF interpolation. We use RBFs mainly because of the simplicity of fitting.

2.3 GRIMA Algorithm for Density Approximation

2.3.1 Initial Observations and Main Ideas Behind the Algorithm

Prior to formally stating the algorithm to find the high probability density (HPD) region, it is helpful to make a few observations to illustrate main ideas behind the algorithm and to summarize our experience in approximating probability densities by interpolants. It is assumed that the probability density to be approximated is unimodal or multimodal but the HPD region is connected.

As we noted in Section 2.1, applicability of our approximation algorithm is not limited to nonlinear regression problems. For this reason, the exposition below treats π as a "black-box" unnormalized density.

Estimation of HPD Regions for Unnormalized Probability Densities

Suppose one has a continuous multivariate unnormalized probability density π with support $\mathfrak{E}$. Let $C_R(\alpha) := \{\eta : \pi(\eta) \geq c(\alpha)\}$ for some constant $c(\alpha)$ chosen in such a way that

$$\int_{C_R(\alpha)} \pi(\eta) d\eta = (1-\alpha) \cdot \int_{\mathfrak{E}} \pi(\eta) d\eta, \tag{2.1}$$

so that $C_R(\alpha)$ is a HPD region of size $(1-\alpha)$ for π.

If $\widehat{c}(\alpha)$ is an estimator for $c(\alpha)$, then the set $\widehat{C}_R(\alpha) := \{\eta : \pi(\eta) \geq \widehat{c}(\alpha)\}$ is an approximation to $C(\alpha)$. For example, if $\eta^{(1)}, \ldots, \eta^{(k)}$ is (possibly but not necessarily) an *i.i.d.* sample from π, then the α-th sample quantile of $\pi(\eta^{(1)}), \ldots, \pi(\eta^{(k)})$

is a reasonable estimator for $c(\alpha)$. Furthermore, if $\widetilde{\pi}$ is an unnormalized probability density that is "close" to π with respect to some measure of distance and $\widetilde{\eta}^{(1)}, \ldots, \widetilde{\eta}^{(k)}$ is a sample from $\widetilde{\pi}$, then it is meaningful to approximate $c(\alpha)$ by the α-th sample quantile of $\widetilde{\pi}(\widetilde{\eta}^{(1)}), \ldots, \widetilde{\pi}(\widetilde{\eta}^{(k)})$.

Estimation of $c(\alpha)$ using an MCMC sample from a cheap-to-evaluate surrogate density is one of the key steps of our algorithm that will be stated later in this section.

Constructing Approximate Posterior Densities by Interpolation

The approach of Bliznyuk *et al.* (2008) can be used to obtain accurate approximations $\widetilde{\pi}$ to π when the exact HPD region is approximately elliptical, but it is not robust if the curvature of logatithm of π near *the* mode does not capture the shape of the HPD region well. (For example, this is the case when the dominant mode of the exact density is located at or very close to the boundary of the parameter space $\mathfrak{E}$.) Earlier papers assume existence (Bliznyuk *et al.*, 2008) and availability (Rasmussen, 2003) of derivatives of the logarithm of π, making the methods proposed therein inapplicable to a host of practical problems (Shoemaker *et al.*, 2007; Tolson and Shoemaker, 2007a; 2007b).

The approach developed in this paper does not make any smoothness assumptions about the exact posterior. The only restriction is that π is "practically continuous", meaning that it is either continuous or is obtained from a continuous function using a discretization on a fine grid.

As in Bliznyuk *et al.* (2008) and in Rasmussen (2003), the approximation to the logarithm of π is based on an interpolant at *design points* (or *knots*), which is a

form of universal kriging with generalized covariance functions (Cressie, 1991). As in the former paper, however, our focus is on interpolation using radial basis functions, but any other class of interpolants can be used.

It is instructive to make a few observations about the main features of density interpolation problem that a practical approximation algorithm must take into account.

1. Design points must be chosen in the region of high probability under the density π. Choice of the knots in the low-probability region is meaningless if π is poorly approximated over the HPD.
2. Since the approximation $\widetilde{\pi}$ becomes less reliable as one moves away from design points, the support of the approximate density must be restricted explicitly to some neighborhood of the set of design points.
3. Our experiments indicate that the quality of interpolation by radially symmetric functions (such as RBFs) is sensitive to the parameterization of π. Approximation of π to a fixed level of accuracy typically requires a significantly greater number of design points if there are directions around the mode(s) of the posterior density along which the logarithm of π changes much more rapidly than along others. For example, if $\log(\pi)$ is a (positive definite) quadratic function, then the higher the condition number of the matrix that defines the quadratic, the more design points are required. Therefore, it may be problematic to obtain an accurate approximation to π given a fixed number (but not the locations) of knots, unless a (linear) change of variables is done prior to interpolation.
4. Enforcing separation between design points is necessary to avoid clumps of nearby knots that provide little new information (relative to their neigh-

bors) about the function being interpolated.

Keeping these observations in mind, we move on to the formal definition of the GRIMA algorithm, which addresses all of them.

2.3.2 Outline of the Algorithm

Notation and Definitions

The purpose of this subsection is to collect all relevant notation in one place for easy reference.

All variables are assumed to be (column) vectors or matrices of size specified in the appropriate definition; this will *not* be emphasized by bold-face notation. The notation $\|\cdot\|$ will refer to a generalized Euclidean norm, defined as $\|x\|_A = \sqrt{x^\mathsf{T} A x}$ for a column vector x and a positive definite matrix A. Unless specified explicitly, A is taken as the identity matrix of appropriate size. Also, define the distance between a point x and a set $\mathcal{S}$ as $\text{dist}(x, \mathcal{S}) = \inf_{x' \in \mathcal{S}} \|x - x'\|$. Only when applied to a vector, a single subscript notation is used to "extract" components, e.g., x_i is the i-th component of x. Finally, for sets $\mathcal{S}_1$ and $\mathcal{S}_2$, $\mathcal{S}_1 \backslash \mathcal{S}_2$ will denote the set of elements of $\mathcal{S}_1$ that are not in $\mathcal{S}_2$ and $|\mathcal{S}_1|$ will give the number of elements in $\mathcal{S}_1$.

As before, π is the exact unnormalized posterior density for the vector of parameters $\eta \in \mathfrak{E} \subset \mathbb{R}^{\dim(\eta)}$, where $\mathfrak{E}$ is the parameter space.

Let $\widetilde{l}_i$ be an RBF interpolant of $l := \log(\pi)$ at the set of design points $\mathcal{D}_i$

available at the stage i of the algorithm, defined as

$$\widetilde{l}_i(\eta) := \sum_{j=1}^{|\mathcal{D}_i|} a_j \phi(\|\eta - \eta^{(j)}\|) + p(\eta), \tag{2.2}$$

where ϕ is a basis function and p is a low-order polynomial. For details of fitting, see our Appendix A.2 or Appendix A.3 of Bliznyuk *et al.* (2008, in press).

The corresponding approximation $\widetilde{\pi}$ to π is defined as

$$\widetilde{\pi}_i(\eta) := \exp\{\widetilde{l}_i(\eta)\} \cdot \mathbb{I}(\eta \in \mathcal{N}_i) \text{ for all } \eta \in \mathfrak{E}, \tag{2.3}$$

where, for a positive coverage radius $r > 0$ (to be discussed shortly),

$$\mathcal{N}_i := \{\eta \in \mathfrak{E} : \text{dist}(\eta, \mathcal{D}_i) \leq r\} \tag{2.4}$$

is a neighborhood of $\mathcal{D}_i$ and $\mathbb{I}$ is the indicator function. Thus the approximation is restricted to a neighborhood of design points as was noted in Section 2.3.1, observation 2.

`GRIMA` Algorithm to Approximate π by $\widetilde{\pi}$

The algorithm of this section is a derivative-free alternative to the *exploratory* stage of `GPHMC` algorithm of Rasmussen (2003) or to steps 2 and 3 of the procedure of Bliznyuk *et al.* (2008, in press). It assumes that the maximum a posteriori (MAP) estimator $\widehat{\eta}$ for η has been found (or, more generally, that some of the initial design points $\mathcal{D}_0$ are "not far" from the true HPD region). Such information is often available from preliminary non-Bayesian parameter estimation (e.g., by MLE).

After outlining the initialization and the main iteration of `GRIMA`, we state the it formally as Algorithm 2.3.1 and, in the next subsection 2.3.2, provide motivation and details for practical implementation.

Maximization of $l := \log(\pi)$ by derivative-free optimization routines produces a set of widely separated design points $\mathcal{D}_{OPT}$. We retain a subset $\mathcal{D}_0$ of $\mathcal{D}_{OPT}$, by discarding design points at which the values of l are "extremely low", and obtain an interpolant $\widetilde{l}_0$ of l at $\mathcal{D}_0$ as in equation (2.2). Choosing the starting radius r in such a way that the neighborhood $\mathcal{N}_0$ of $\mathcal{D}_0$ is connected (since the true HPD is assumed to be), we define $\widetilde{\pi}_0$ by means of equation (2.3).

At the i-th iteration of `GRIMA`, an MCMC sample from the cheap-to-evaluate $\widetilde{\pi}_i$ is obtained and is used to determine the lower bound on $\widetilde{l}_i$ over the size-$(1-\alpha)$ HPD region for $\widetilde{\pi}_i$ (lines 4-5 of Algorithm 2.3.1). Subsequently, $\widetilde{l}_i$ is maximized over the boundary of $\mathcal{N}_i$ and if the value of $\widetilde{l}_i$ is "high enough" at the maximizer η^*, l is evaluated at η^* and the RBF interpolant $\widetilde{l}_i$ is updated (lines 6-16). By allowing coverage radius r to shrink or to increase depending the height of $\widetilde{l}_i$ at a point η^* at a given iteration (lines 17-21), `GRIMA` algorithm attempts to choose candidate points η^* on the boundary of $\mathcal{N}_i$ as far as possible from $\mathcal{D}_i$, provided that the prediction $\widetilde{l}_i(\eta^*)$ for $l(\eta^*)$ is above the threshold $\widetilde{c}_i(\alpha) - \delta$. If r grows (more knots from the boundary of $\mathcal{N}_i$ are added), so does the approximate HPD region; as r shrinks, the approximation to π over the HPD region tends to improve. The algorithm is allowed to terminate before the computational budget has been exhausted if extra evaluations of π do not improve the quality of approximation significantly as judged from diagnostics (lines 22-26).

Algorithm 2.3.1 GRIMA

Require: $r > 0$, $\delta > 0$, $\alpha \in (0,1)$, $\rho \in (0,1)$, J, T as specified below

Require: $i = 0$, $\mathcal{D}_0$, $\mathcal{N}_0$, $\widetilde{l}_0$, $\widetilde{\pi}_0$ as defined above

1: **while** computational budget has not been exceeded **do**

2: $\quad i \leftarrow i+1$, $\mathcal{D}_i \leftarrow \mathcal{D}_{i-1}$, $\widetilde{l}_i \leftarrow \widetilde{l}_{i-1}$

3: set $\mathcal{N}_i$ as in equation (2.4) and $\widetilde{\pi}_i$ as in equation (2.3)

4: obtain an MCMC sample $\mathcal{M}_i$ from $\widetilde{\pi}_i$ of length T

5: find $\widetilde{c}_i(\alpha)$, the α-th sample quantile of the sample $\{\widetilde{l}_i(\eta) : \eta \in \mathcal{M}_i\}$

6: **for** $j = 1, \ldots, J$ **do**

7: $\mathcal{C} \leftarrow \{\eta \in \mathcal{N}_i : \text{dist}(\eta, \mathcal{D}_i) = r\}$

8: find $\eta^* \in \mathcal{C}$ such that $\widetilde{l}_i(\eta^*) \approx \max_{\eta \in \mathcal{C}} \widetilde{l}_i(\eta)$

9: **if** $\widetilde{l}_i(\eta^*) \geq \widetilde{c}_i(\alpha) - \delta$ **then**

10: $\mathcal{D}_i \leftarrow \mathcal{D}_i \cup \{\eta^*\}$

11: evaluate l at η^*

12: update $\widetilde{l}_i$ so that it interpolates l at $\mathcal{D}_i$ (in particular, at η^*)

13: **else**

14: set $j \leftarrow j - 1$ and **break** the **for** loop

15: **end if**

16: **end for**

17: **if** $j < J$ **then**

18: set $r \leftarrow \rho \cdot r$

19: **else**

20: set $r \leftarrow \rho^{-1} \cdot r$

21: **end if**

22: **if** diagnostics suggest that sequence $\{\widetilde{\pi}_k\}_{k=1}^{i}$ has "practically" converged **then**

23: **break** the **while** loop

24: **else**

25: re-estimate scaling matrix and parameters of the MCMC sampler

26: **end if**

27: **end while**

28: **return** $\widetilde{\pi} \leftarrow \widetilde{\pi}_i$

Discussion and Practical Implementation

As follows from earlier works (Bliznyuk *et al.* (2008, in press), Rasmussen (2003)), the design points from the optimization run, $\mathcal{D}_{OPT}$, at which l takes extremely low values hardly improve the approximation to l on the HPD region, and often make the quality of approximation deteriorate if not removed. To obtain the initial set $\mathcal{D}_0$ of knots for interpolation, we discard such points using a cutoff value q, which may be suggested by the asymptotic properties (as the dimension of Y increases) of the MAP estimator and of the likelihood ratio test statistic in nonlinear regression models (see Wu (1981) for an example of regularity conditions). In the frequentist framework, $2\left(l(\eta) - l(\widehat{\eta})\right)$ may often be approximated by a chi-squared distribution with $\dim(\eta)$ degrees of freedom. Consequently, we define

$$\mathcal{D}_0 := \{\eta \in \mathcal{D}_{OPT} : -2l(\eta) \leq -2l(\widehat{\eta}) + q\}, \tag{2.5}$$

where q is chosen to be a tail (e.g., .99-th) quantile of this distribution. (If necessary, additional points from $\mathcal{D}_{OPT}$ may be incorporated into the approximation at later stages of `GRIMA`.)

The initial coverage radius r is chosen in such a way that the initial design region $\mathcal{N}_0$, defined in equation (2.4), is connected. This is necessary to ensure that $\mathcal{N}_0$ can be traversed by an MCMC sampler and since it was assumed that the true HPD region is connected. If $M = \max_j \text{dist}(\eta^{(j)}, \mathcal{D}_0 \backslash \{\eta^{(j)}\})$, the *maximin* distance between design points in $\mathcal{D}_0$, then connectedness of $\mathcal{N}_0$ is ensured if $r \geq 0.5 \cdot M$. In applications, we use (and recommend) initialization $r = M$.

As in Bliznyuk *et al.* (2008, in press), we fit an RBF surface with cubic basis functions $\phi(x) := x^3$ and a linear tail $p(\eta) := [1, \eta^{\mathsf{T}}] \cdot b$, where $b \in \mathbb{R}^{\dim(\eta)+1}$. Our experiments with other basis functions produced results similar to those we report.

Having defined the approximate posterior $\widetilde{\pi}_0$, let's consider the i-th iteration of `GRIMA`.

In lines 4-5, we generate an MCMC sample of size T from $\widetilde{\pi}_i$ to determine the minimum height of $\widetilde{l}_i$ over the HPD of size $(1-\alpha)$ for $\widetilde{\pi}_i$. The value T can be estimated based on existing consistency and asymptotic normality results for sample quantiles (e.g., Shao (1999), sec. 5.3) controlling for serial correlation in the Markov chain. In our experience, very accurate estimation of the true quantile is not necessary and several thousand Markov chain steps are sufficient under fast mixing. The MCMC sampler is always initialized at the knot $\eta^{(k)} \in \mathcal{D}_i$ at which l is highest, which makes the length of the burn-in period negligible.

We remind that $\widetilde{\pi}_i$ is restricted to the neighborhood $\mathcal{N}_i$ in order to prevent the MCMC sampler from escaping to the poorly approximated low-probability (with respect to π) regions, where $\widetilde{l}_i$ may be high due to approximation error.

The purpose of the **for** loop (lines 6-16) is to attempt to choose J design points from $\mathcal{N}_i$ in such a way that (i) the candidate points η^* are no closer than distance r to any point in $\mathcal{D}_i$ and (ii) the value $l(\eta^*)$ is at least as high as in the approximate HPD region. Since $l(\eta^*)$ is unknown before it is computed, we use the known value $\widetilde{l}_i(\eta^*)$ of the logarithm of the approximate density to make the decision (line 9), adjusting for the uncertainty with the help of δ. The candidate points η^* that maximize $\widetilde{l}_i$ are chosen by uniform random sampling on $\mathcal{C}$, which

is easy to do efficiently; formal optimization (over the boundary of the union of ellipsoids) is non-trivial and is not recommended. In applications, we choose δ to be a small multiple (e.g., .3 as in subsequent sections) of the maximum change of l over $\eta \in \mathcal{D}_i$, for which l exceeds $\widetilde{c}_i(\alpha)$. If the condition of line 9 is satisfied, the exact density is evaluated at η^* and the RBF interpolant of l is updated. An update can be performed using $\mathcal{O}(|\mathcal{D}_i|^2)$ rather than $\mathcal{O}(|\mathcal{D}_i|^3)$ flops if the factorization of the interpolation matrix is re-used as outlined in Appendix A.2.

The value J needs to be chosen based on comparison of the computational costs to carry T Markov chain steps and to evaluate π once. If a single evaluation of π takes significantly more time, then re-estimation of $\widetilde{c}_i(\alpha)$ after each update of $\widetilde{l}_i$ can be enforced by setting $J = 1$.

Although it is omitted in the statement of the `GRIMA` algorithm, there is a possibility that

$$\widetilde{l}_i(\eta^\bullet) > \max\{l(\eta) : \eta \in \mathcal{D}_i\}, \tag{2.6}$$

where $\eta^\bullet := \arg\max\{\widetilde{l}_i(\eta) : \eta \in \mathcal{M}_i\}$. Typically this signifies that either r is too large or that $l(\eta^\bullet)$ exceeds the right-hand side of equation (2.6) (as it turns out to be the case in the application of Section 2.4.2). As a rule of thumb, if $\widetilde{l}_i(\eta^\bullet)$ exceeds $\max\{l(\eta) : \eta \in \mathcal{D}_i\}$ by more than 1, we evaluate l at $\eta^\bullet$, update the approximation, reduce r and move on to the next iteration of the **while** loop.

If the **for** loop terminates because fewer than J new design points could be added for a given value of r, r is reduced by a factor $\rho \in (0, 1)$; otherwise, `GRIMA` attempts to increase r. A value around $.9$ is suggested for ρ, in order not to have many design points becoming disconnected in the design region $\mathcal{N}_{i+1}$ in the following iteration if r shrinks.

If a substantial number of well-separated design points have been chosen, and the sequence of the approximate densities $\widetilde{\pi}_i$ appears to have converged, there is little value in continuing the selection of new knots. One may use a combination of graphical and numerical tools for diagnostics. Among graphical summaries, we recommend examining plots of estimates of marginal densities of components of η based on the MCMC samples from approximate (joint) densities $\widetilde{\pi}_i$ from each iteration. As far as numerical summaries are concerned, we use an estimate of the total variation (*TV*) norm between marginal densities of the components η_i of η of subsequent approximate densities; see Appendix A.1 for details.

As we remark in Section 2.3.1, fitting of radially symmetric interpolants (like those using RBFs) is sensitive to the parameterization of π. We recommend to estimate the sample covariance matrix C_i from the MCMC sample $\mathcal{M}_i$ (or from a longer run during the diagnostic step) (lines 22-26) and to refit the RBF surface from scratch after the linear change of variables $\eta \mapsto H_i^{-1}\eta$, where H_i is any square matrix satisfying $H_i H_i^{\mathsf{T}} = C_i$ (e.g., a Cholesky factor of C_i). This is especially important at early stages of the algorithm, and the role of refitting is diminished after the approximation has stabilized. In what follows, the surface $\widetilde{l}_i$ is always fitted after a linear change of variables; we emphasize this by using the generalized ($\|\cdot\|$) rather than standard Euclidean ($\|\cdot\|_2$) norm in equation (2.2). However, when evaluating $\widetilde{\pi}$ and $\widetilde{l}_i$, we pass the argument η on the original scale.

In applications, we do diagnostics and change parameterization (roughly) every $2 \cdot J - 3 \cdot J$ evaluations of π.

2.3.3 Illustration on a Synthetic Problem

Before applying GRIMA to a model with real data in Section 2.4, we compare our algorithm with the results presented by Rasmussen (2003) for the density

$$\pi(\eta) \propto \exp\{-0.5 \cdot [(\eta^{\mathsf{T}}\eta - a)/b]^2\} \quad \text{for } \eta \in \mathbb{R}^2, \tag{2.7}$$

where $a = 1$ and $b = 1/4$. The set of modes of π is the boundary of the unit circle, and the HPD region is a two-dimensional torus (Figure 2.3.3). Even though l is twice differentiable, the curvature at any of the (infinitely many) modes carries little information about the shape of the HPD region, so the approach of Bliznyuk *et al.* (2008, in press) is not applicable.

Since GRIMA algorithm does not assume availability of derivatives of l, we use a derivative-free minimization routine CONDOR (Vanden Berghen and Bersini (2005)) in order to find a MAP $\widehat{\eta}$ for initialization, although any other algorithm can been used. Optimization search was started far from the HPD region at $\eta = [20; 20]$ and took 26 evaluations of π to converge.

The initial set of design points $\mathcal{D}_0$, chosen as discussed in the beginning of Section 2.3.2 with q taken to be the .99-th quantile of chi-squared distribution with $\dim(\beta) = 2$ degrees of freedom, contains 9 points from the optimization run. Even though the χ^2_2 approximation to the likelihood ratio statistic is not justified here, it provides some guidance as to which values of the log-posterior density are high.

The parameters of GRIMA were chosen as $T = 5 \cdot 10^3$, $J = 4$, $\alpha = .01$ and $\rho = .9$; r and δ are updated as discussed in Section 2.3.2. We use random walk Metropolis-Hastings algorithm for MCMC (Tierney (1994)). Diagnostics of approximation were done periodically (every 7-10 evaluations of the exact poste-

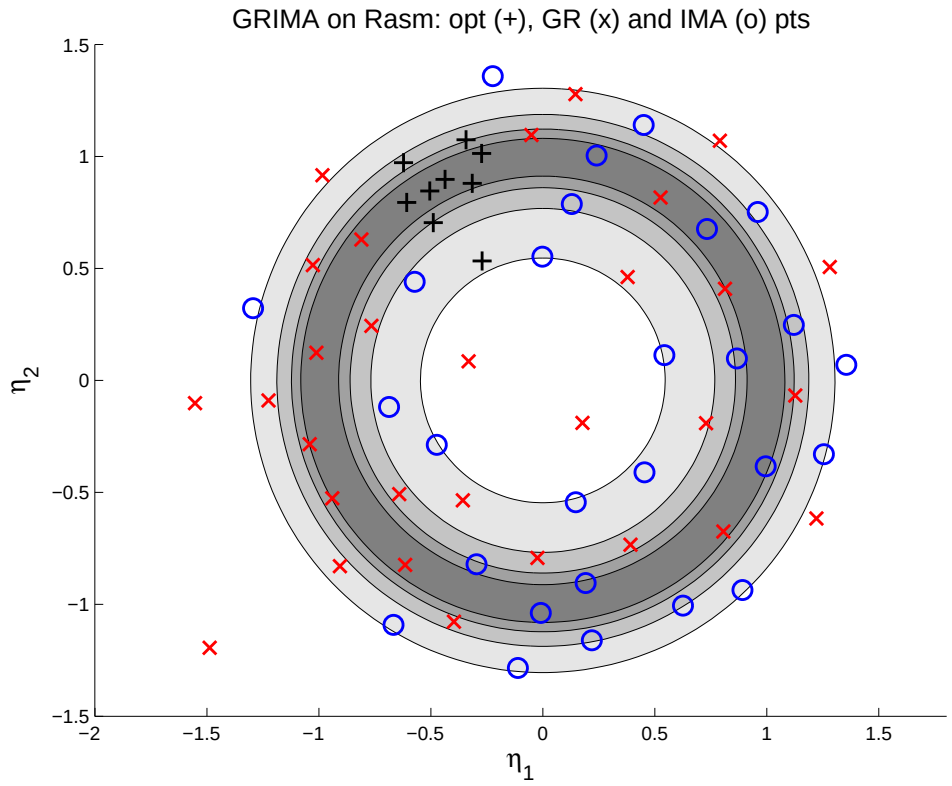

Figure 2.1: True HPD region and progress of `GRIMA` algorithm on Rasmussen's density of equation (2.7). The HPD regions, listed in the order of diminishing average intensity of grey, contain .5, .7, .9 and .995 of the total mass of the density. Markers denote initial design points after *optimization* (+) stage (9 knots), additional knots added in the *design region growth* (x) stage (40-9=31 knots) and additional knots added in the *approximation improvement* (o) stage (67-40=27 knots).

rior) using an MCMC run of length 10^5. The scaling matrix H_i for the linear change of variables when fitting the RBF surface (Section 2.3.2) was taken to be the lower-triangular Cholesky factor of the MCMC sample covariance matrix C_i after each diagnostic run. A multiple of C_i that ensures acceptance rate between .2 and .3 and lag one autocorrelation coefficient of around .9 in Markov chain was used as a covariance matrix in the multivariate normal proposal density of the random walk sampler.

We illustrate the progress of GRIMA in Figure 2.3.3. It can be seen that the knots retained from the optimization run (marked with +) cover only a small part of the HPD region. The approximate HPD region tends to grow until about 31 new knots (marked with x) are added, after which the *maximin* distance between the knots tends to shrink as extra knots (marked with o) are added.

The decision to terminate GRIMA was based entirely on examination of plots similar to that of Figure 2.3.3. The plot shows how different (with respect to the estimated *TV* norm — see Appendix A.1) the successive MCMC samples from marginal densities obtained from $\widetilde{\pi}_j$ (based on design points $\mathcal{D}_j$ for $j = 0, \ldots, i$) are, relative to the MCMC sample from the approximate density from $\widetilde{\pi}_i$ that uses all available design points $\mathcal{D}_i$. It is seen that the approximations change little after 58 knots are used; the algorithm is terminated at 67 design points (9 out of which came from optimization).

Figure 2.3.3 compares plots of the estimates of the marginal densities of η_1 and η_2 based on initial, intermediate and terminal approximate densities with those based on an *i.i.d.* sample from the exact density π that we draw efficiently using Algorithm 2.6.2 of our Appendix 2.6.1. The closeness of the terminal approximate marginal densities to their exact counterparts gives support to our termination criterion.

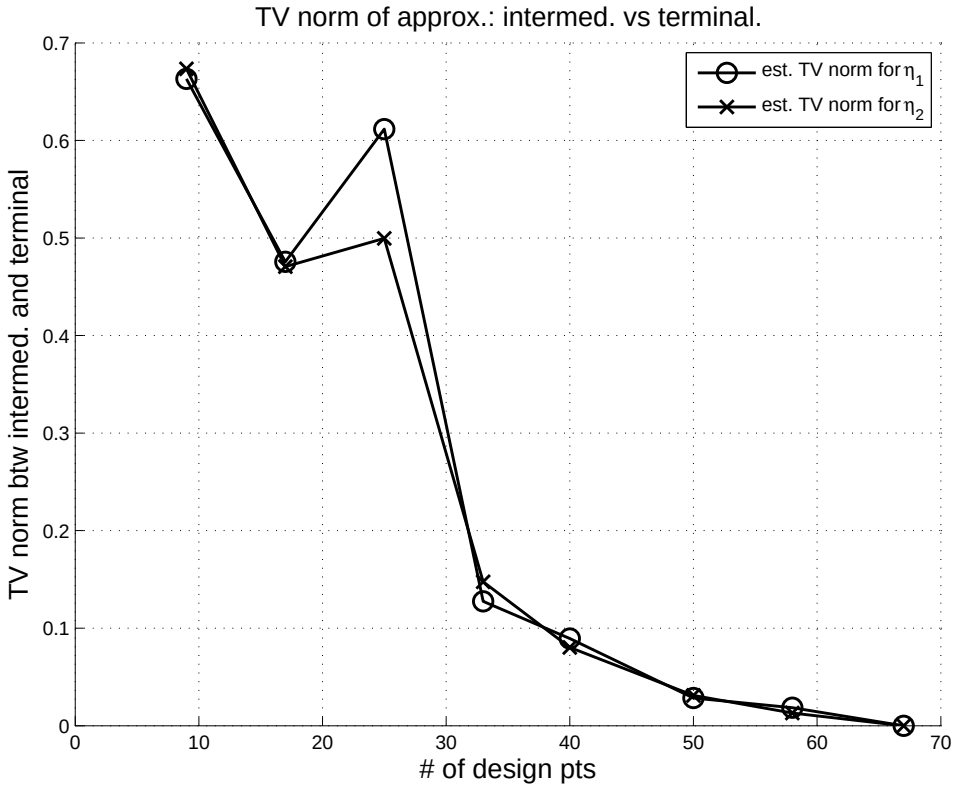

Figure 2.2: Estimated total variation norm between intermediate and terminal (based on 67 knots) approximate densities for η_1 and η_2 (for Rasmussen's density) as a function of the number of knots used to obtain intermediate approximate densities.

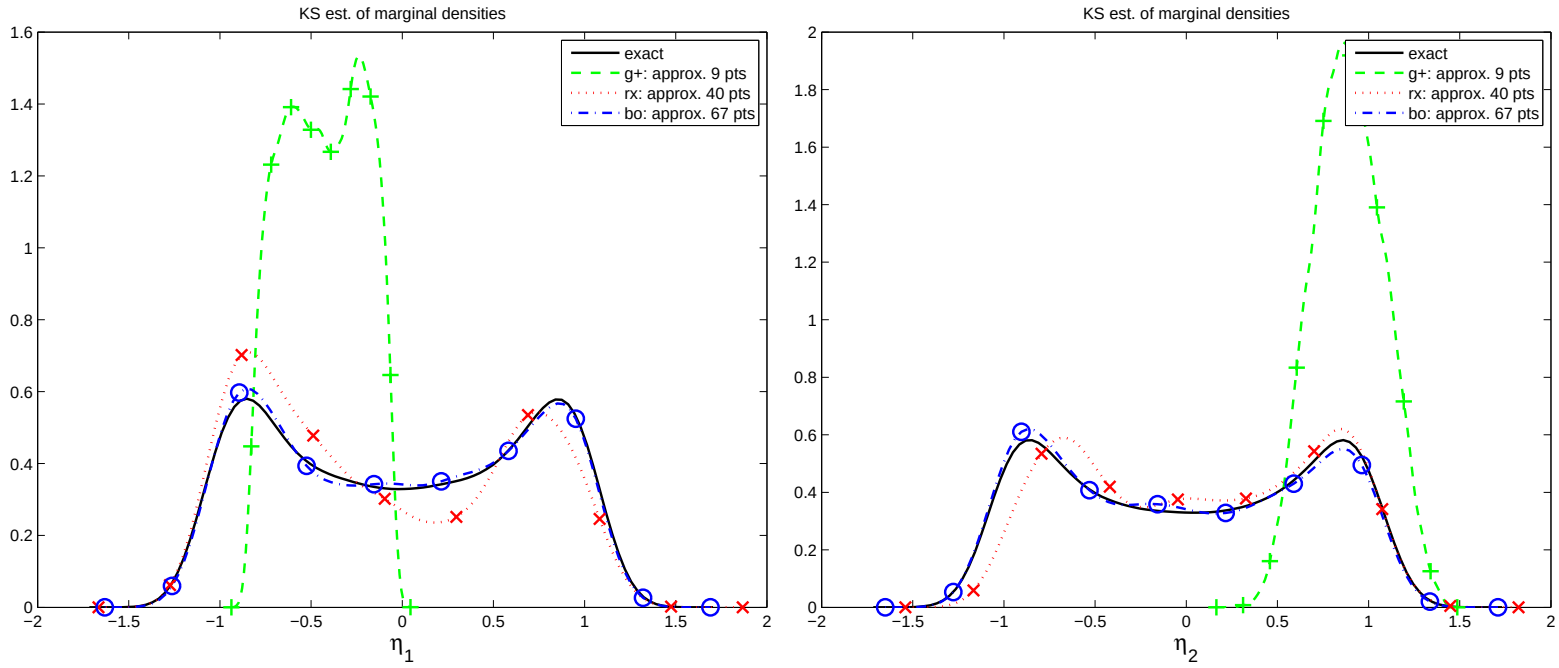

Figure 2.3: Kernel-smoothed estimates of marginal densities of η_1 and η_2 (for Rasmussen's density) using 10^5 samples from exact (no marker) and approximate initial (+, 9 knots), intermediate (x, 40 knots) and terminal (o, 67 knots) bivariate densities. For the exact density, the samples are *i.i.d.*; for approximate - drawn using random-walk MCMC.

Our results (84=26-9+67 knots and no gradients) compare favorably to those reported by Rasmussen (2003), where for the interpolation of the logarithm of the exact posterior, l, by Gaussian processes he uses 100 evaluations of l and of its gradient, although it is not clear where he initializes the optimization search. Apart from applicability of our approach in the contexts where derivatives are computationally expensive to obtain (or are not available or meaningful), we provide a measure of approximation quality and a termination criterion, without which any non-trivial practical application of the density approximation is hardly possible. Our additional experiments (not reported here) suggest that our algorithm is not very sensitive to the choice of the optimization routine and of the starting point.

2.4 Case study: Town Brook

2.4.1 Background

The Town Brook watershed is a 37 km^2 subwatershed inside the larger Cannonsville watershed (1200 km^2) located in the Catskills Region of New York State. The vector Y of measured flow data used in the analysis consists of 1096 daily observations (from October 1997 to September 2000) for water entering the Delaware River from Town Brook watershed based on readings by the U.S. Geological Survey. The nonlinear regression function is produced by the `SWAT2000` simulator (Arnold *et al.* (1998)), which has been used by over a thousand agencies and academics worldwide in analysis of water flow and nutrient transport in watersheds (e.g., Eckhardt *et al.* (2002), Grizzetti *et al.* (2003),

Shoemaker *et al.* (2007), Tolson and Shoemaker (2007b)). The water draining the Town Brook and rest of the Cannonsville watershed collects in the Cannonsville Reservoir, from which it is piped over a hundred miles to New York City for drinking water. The quality of this drinking water is threatened by phosphorus pollution and, if not protected, could result in the need for a water filtration plant estimated to cost over $8 billion. For this economic reason as well as for general environmental concerns, there is great interest in quantifying the parameter uncertainty for this model.

The input information of the `SWAT2000` computer code for the Town Brook model formulation, subsequently referred to as the Town Brook simulator f, is discussed briefly in Tolson and Shoemaker (2007a) and in more detail in Tolson and Shoemaker (2004). Earlier work (Shoemaker *et al.* (2007) and Tolson and Shoemaker (2007a)) revealed that the output of f is discontinuous due to discretizations inside `SWAT2000`. Additional experiments showed that, when all parameters are varying, the output has very large jump discontinuities that cannot be attributed to discretization alone. After fixing some of the input parameters at meaningful values after consultation with a subject matter expert, the jump discontinuity appears to be absent. Currently we collaborate with hydrologic scientists in order to understand the reason for the discontinuities and to expand the set of calibration variables. In order to illustrate `GRIMA`, we work with a subset of 4 parameters for which uncertainty assessment is most critical. The values of the fixed parameters are provided in Table 2.1.

Table 2.1: Fixed and variable flow-related parameters for the Town Brook simulator. Ranges of parameters that vary during calibration are in Table 2.2.

#	SWAT ID	Brief description (units)	Value (if fixed)
1.	SFTMP	Snowfall temperature (C)	1
2.	SMTMP	Snow melt temperature threshold (C)	1.75
3.	SMFMX	Melt factor for snow (mm H_2O/C/d)	3
4.	TIMP	Snow pack temperature lag factor	1
5.	SURLAG	Surface runoff lag coefficient	1
6.	GW_DELAY	Groundwater delay time (days)	β_1
7.	ALPHA_BF	Baseflow alpha factor (days)	β_2
8.	GWQMN	Threshold groundwater depth for return flow (mm)	100
9.	LAT_TIME	Lateral flow travel time (days)	β_3
10.	ESCO	Soil evaporation compensation factor	.7
11.	CN2_f	Runoff curve number multiplicative factor	β_4
12.	AWC_f	Available water capacity factor	0
13.	Ksat_f	Saturated hydraulic conductivity factor	0
14.	DepthT_f	Soil profile depth factor	0

2.4.2 Statistical Model and Analysis

In this subsection we use a statistical model that transforms both the vector of observations Y and the environmental model (simulator) values $f(\beta)$. We start with an initial statistical model that assumes that the errors in the observed flow are independent; subsequently, the model is refined to explain temporal correlation in Y. Once an adequate model is found, we illustrate the `GRIMA` algorithm to approximate the posterior density of the simulator parameters β conditional on Y and the vector of best-fitting non-simulator parameters.

Initial Model and Refinements

Following the suggestions in Bliznyuk *et al.* (2008, in press), we use the general transform-both-sides (Carroll and Ruppert (1984, 1988)) model of the form

$$h(Y_i, \lambda) = h\{f_i(\beta), \lambda\} + \epsilon_i, \tag{2.8}$$

where $h(\cdot, \lambda)$ is an element of a transformation family indexed by λ, f_i is the non-linear regression model for the observed flow Y_i at the i-th temporal instant, and $\epsilon_1, \ldots, \epsilon_n$ are errors that have a multivariate normal distribution with mean zero and covariance matrix $\Sigma(\theta)$, parameterized by θ.

Since we are modeling flow in the stream, the vector of responses Y and the values of the simulator $f(\beta)$ are positive. A popular family of transformations for such data is the Box-Cox power family $\{h_{BC}(\cdot, \lambda) : \lambda \in \mathbb{R}\}$ (Box and Cox (1964)), defined for a positive scalar y as

$$h_{BC}(y, \lambda) := \begin{cases} (y^\lambda - 1)/\lambda & \text{if } \lambda \neq 0 \\ \log(y) = \lim_{\lambda \to 0}(y^\lambda - 1)/\lambda & \text{if } \lambda = 0 \end{cases} \tag{2.9}$$

For right-skewed data, such as those in our application, $\lambda \leq 1$.

However, since the support of ϵ_i in the equation (2.8) is $\mathbb{R}$, it is implied that the image of $h(\cdot, \lambda)$ must be $\mathbb{R}$ for every value of the parameter λ (otherwise, the inverse of $h(\cdot, \lambda)$ is undefined), which does not hold for h_{BC}. We "repair" this defect of the Box-Cox family for concave transformations by perturbing h_{BC} using the logarithmic transformation

$$h(y, \lambda) := (1 - \Delta) \cdot h_{BC}(y, \lambda) + \Delta \log(y), \tag{2.10}$$

where Δ is a small positive constant (e.g., 10^{-4}) and $\lambda \leq 1$. (An advantage of this transformation over the one suggested by Bliznyuk *et al.* (2008, in press) is that the parameter λ retains its conventional interpretation.)

The logarithm of the (unnormalized) likelihood of parameters β and ζ given the data Y is

$$L(\beta, \zeta|Y) := -\frac{1}{2} \log \det \Sigma(\theta) - \frac{1}{2} \|h(Y, \lambda) - h\{f(\beta), \lambda\}\|^2_{\Sigma(\theta)^{-1}} + \sum_{i=1}^{n} \log \frac{\partial h(Y_i, \lambda)}{\partial Y_i}, \tag{2.11}$$

where $\zeta = [\lambda, \theta]$ is a vector of non-simulator parameters and $h(\cdot, \lambda)$ is applied component-wise to vectors. Further, we define the *profile log-likelihood* as

$$\widehat{L}(\beta) := \sup_{\zeta} L(\beta, \zeta|Y). \tag{2.12}$$

Since derivatives of L with respect to the vector ζ of non-simulator parameters are available analytically, the value $\widehat{L}(\beta)$ typically is cheap to compute numerically once $f(\beta)$ has been computed. (We do this by quadratic programming routine `FMINCON` in Matlab.)

We fit the initial model that assumes that the errors ϵ_i are *i.i.d.* $N(0, \theta_1^2)$. Following the suggestions in Bliznyuk *et al.* (2008, in press), we estimate β by maximization of $\widehat{L}$ to obtain the MLE $\widehat{\beta}$ for β using minimization routine `CONDOR` and

recover the MLE $\widehat{\zeta}$ for ζ by maximizing $L(\widehat{\beta}, \cdot)$ with respect to ζ. In the absence of information about the location of $\widehat{\beta}$, `CONDOR` is initialized at the center of the parameter space $\mathfrak{B}$. Optimization required 91 runs of the simulator to converge. (The starting and terminal estimates for β for the models we consider, as well as parameter spaces for model parameters, are reported in Table 2.2.)

Even though the simulated flow $f(\widehat{\beta})$ predicts the observed flow Y well (the hydrograph of Figure 2.4.2 is similar to that in Shoemaker *et al.* (2007)), the residuals

$$e_i := h(Y_i, \widehat{\lambda}) - h\{f_i(\widehat{\beta}), \widehat{\lambda}\} \tag{2.13}$$

exhibit serial correlation: the plot of the autocorrelation function (ACF) shows (roughly) exponential decay, and the plot of the partial autocorrelation function (PACF) has a spike of height close to .8 at lag 1.

Consequently, correlation was incorporated into the statistical model of equation (2.8) by modeling ϵ_i as an autoregressive process of order 1 (AR(1) process). Even though

$$Cov(\epsilon_i, \epsilon_j) = \theta_1^2 \cdot \theta_2^{|i-j|} \tag{2.14}$$

under the AR(1) model implies that $\Sigma(\theta)$ is a dense matrix, the inverse of $\Sigma(\theta)$ is tridiagonal (e.g., Hamilton (1993), chap. 5). Hence, for a known $f(\beta)$, $L(\beta, \cdot)$ can be evaluated in time $\mathcal{O}(n)$, and the overhead to maximize $L(\beta, \cdot)$ in order to compute $\widehat{L}(\beta)$ is insignificant.

Table 2.2: Values of β and ζ (with appropriate parameter spaces) that maximize the log-likelihood L found by optimization by `CONDOR` for models with *i.i.d.* and AR(1) errors and in the course of `GRIMA` for AR(1) model with non-simulator parameters ζ held fixed at $\widehat{\zeta}$.

stage	β				ζ			$-2L(\widehat{\beta}, \widehat{\zeta})$	# of extra runs of f
	$\widehat{\beta}_1$	$\widehat{\beta}_2$	$\widehat{\beta}_3$	$\widehat{\beta}_4$	$\widehat{\lambda}$	$\widehat{\theta}_1$	$\widehat{\theta}_2$		
`CONDOR` *i.i.d.* ϵ_i	6.12	.646	34	.892	$-.039$	.647	0	-112.6	91
`CONDOR` AR(1) ϵ_i	6.97	.66	27.11	.75	$-.152$	.454	.814	-1135.7	65
`GRIMA` AR(1) ϵ_i	7.85	.995	26.27	.751	$-.152$	.454	.814	-1145	113
lower bound	.001	.001	.001	.75	-10	0	-1	–	–
upper bound	500	1	180	1.25	1	100	1	–	–

The second run of CONDOR to maximize $\widehat{L}$ under the AR(1) model for errors was initialized at the MLE $\widehat{\beta}$ under the *i.i.d.* model for errors. This second stage of maximization took 65 runs of the simulator. Figure 2.4.2 shows plots of the observed flow Y and of the simulated flow $f(\widehat{\beta})$ for the MLE $\widehat{\beta}$ under the AR(1) model.

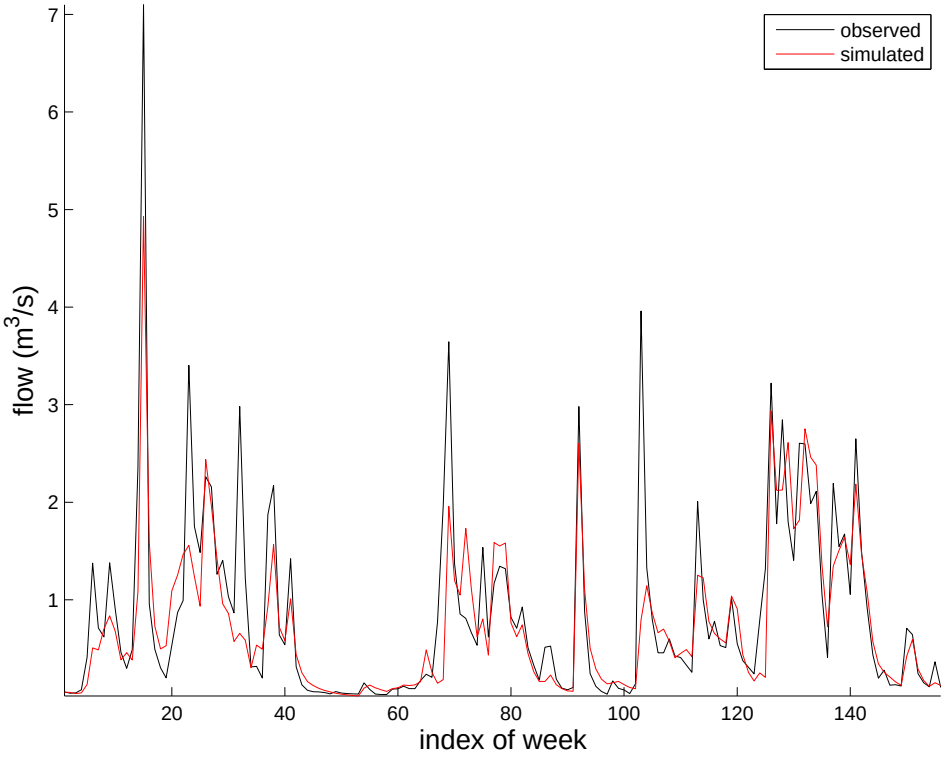

Figure 2.4: Hydrograph of the average weekly observed flow for the Town Brook watershed data and of the average weekly simulated flow obtained from $f(\widehat{\beta})$ for the MLE $\widehat{\beta}$ obtained using CONDOR for the log-likelihood of equation (2.11) under the AR(1) model for errors.

Examination of the ACF and PACF plots of the AR(1)-corrected residuals, obtained from the residuals e_i of equation (2.13) as

$$u_i := e_{i+1} - \widehat{\theta}_2 \cdot e_i \tag{2.15}$$

for $i = 1, \ldots, n-1$, reveals a little (less than .2) correlation at lag 2. The starting

and terminal estimates for β and estimates for ζ for the two models that we consider are reported in Table 2.2. We choose not to refine the model further so as not to obscure the main goal of the paper which is density approximation.

Approximation

Having settled on the statistical model, we move on to approximation of π using `GRIMA` algorithm.

We focus our attention entirely on the conditional distribution of the simulator parameters β given the data Y keeping the non-simulator parameters ζ fixed at their estimated values $\widehat{\zeta}$. Intelligent exploitation of the computational savings derived from L being cheap to evaluate with respect to ζ when selecting extra design points is technical and will be reported in a separate paper.

After putting a uniform prior distribution on β over the parameter space $\mathfrak{B}$, we define

$$\pi(\beta) := \exp\{L(\beta, \widehat{\zeta}|Y)\} \cdot \mathbb{I}(\beta \in \mathfrak{B}). \tag{2.16}$$

We associate η with β and $\mathfrak{E}$ with $\mathfrak{B}$ of Section 2.3. The rest of the definitions for the application of the `GRIMA` algorithm were presented in Section 2.3.2.

We use the approximation of equation (2.5) and choose q to be .99-th quantile of the chi-squared distribution with $\dim(\beta) = 4$ degrees of freedom, which allows us to reuse 22 points from the optimization run to create an initial approximation $\widetilde{l}_0$ to l. (Recall the notation from Section 2.3.2.)

We initialize $T = 10^4$, $J = 4$ and the rest of the parameters as in Section 2.3.3. Every 12-14 evaluations of the exact posterior π we do a long MCMC run of

length 10^5 to assess the quality of approximation and to re-estimate the scaling matrix H_i for the MCMC sampler and in order to refit the RBF surface; we reset r after this new linear change of variables. The Markov chain mixes well, with lag one autocorrelation being less than .9 for each β_i and the overall acceptance rate between .2 and .3.

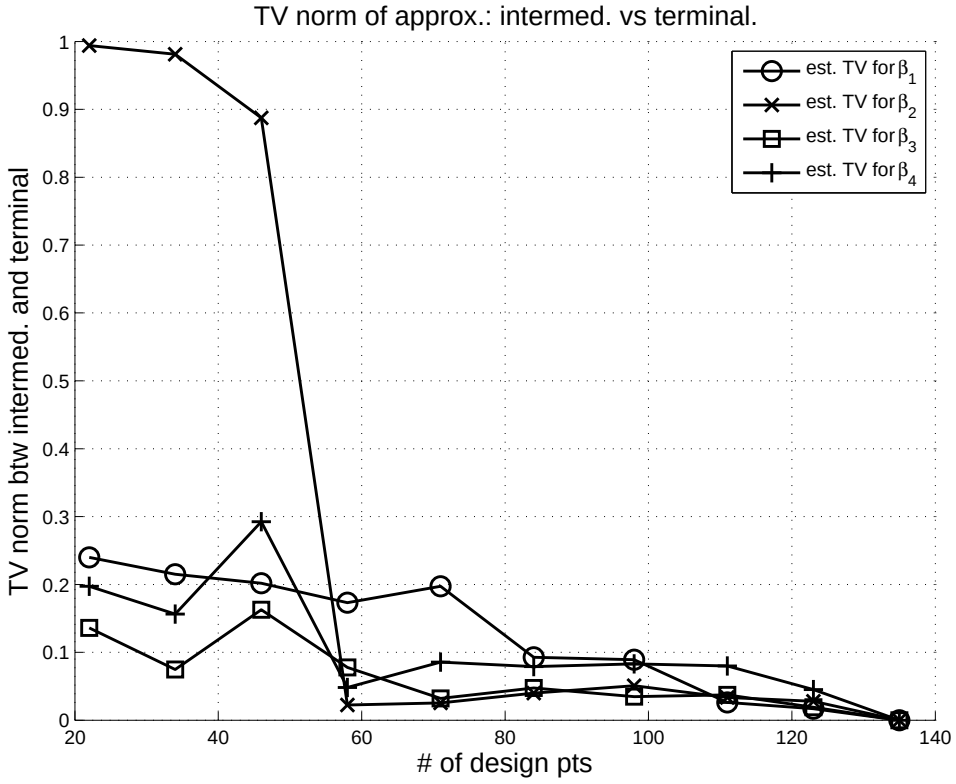

Figure 2.5: Estimated *TV* norm between intermediate and terminal (based on 135 knots for the Town Brook posterior density with AR(1) errors) approximate densities for β_i, $i = 1, \ldots, 4$, as a function of the number of knots used to obtain intermediate approximate densites.

Figure 2.4.2 compares estimates of the *TV* distance between terminal (i.e., "most recent") and preceding approximate marginal distributions of β_k. (More precisely, we compare samples from marginal distributions of β_k based on $\widetilde{\pi}_j$ with $22, 34, \ldots, 123$ knots with those from $\widetilde{\pi}_i$ with 135 knots.) Examination of

this plot and of plots of estimates of preceding densities suggest that, for each β_k the approximation improves little after the number of design points used in interpolation grows beyond 111. (Recall that 22 of these points come from the `CONDOR` run.) Consequently, we terminate the algorithm at 135 knots.

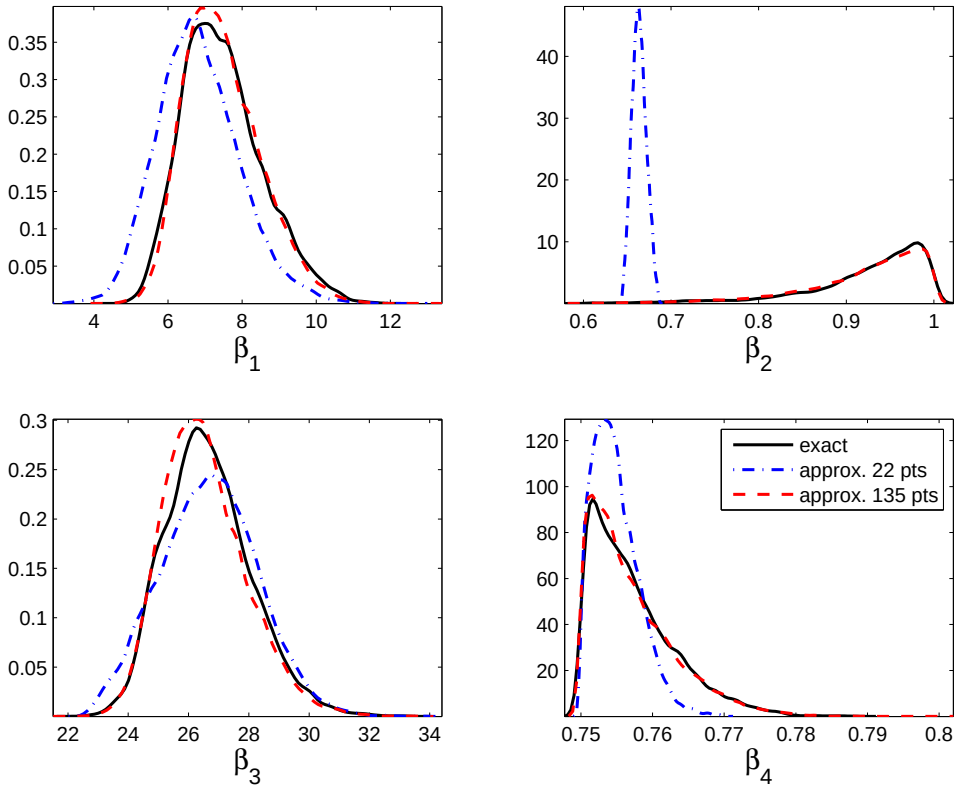

Figure 2.6: Kernel-smoothed estimates of marginal densities of β_i, $i = 1, \ldots, 4$, using MCMC samples from exact (solid line) and approximate initial (dash-and-dot line, 22 knots), and terminal (dashed line, 135 knots) multivariate posterior densities (for the Town Brook statistical model with AR(1) errors). For the exact density, the sample size is $2 \cdot 10^4$; for approximate - 10^5 (drawn using random-walk MCMC in all cases).

For the sake of comparison, we do an MCMC run of length $2 \cdot 10^4$ using the exact posterior density π. In Figure 2.4.2, we overlay plots of estimates (from respective MCMC runs) of marginal densities of β_k's using the initial approximate posterior density $\widetilde{\pi}_0$, the terminal approximate posterior density $\widetilde{\pi}$ and the exact

posterior density π. The plots in Figure 2.4.2 and in Figure 2.4.2 and Table 2.2 suggest that π is maximized when β is at its upper boundary, but `CONDOR` terminated prematurely. Remarkably, `GRIMA` was able to recover from this deficiency and to produce a very accurate approximation to π in 113 (or fewer) extra runs of the simulator f. (The value of β at which π is highest, reached within 41 extra evaluations of π by `GRIMA`, is reported in the third row of Table 2.2.)

From Figure 2.4.2 one can also appreciate the virtues of local (done over a HPD region) rather than global approximation to π. It is seen that nearly all of the mass of the posterior is contained in the hyper-rectangle with lower bound $[4,\ .7,\ 22,\ .75]$ and upper bound $[12,\ 1,\ 32,\ .79]$ of volume $.96$, which constitutes about $.002\%$ of the volume of the parameter space $\mathfrak{B}$. Therefore, the naïve approach that approximates π over the whole $\mathfrak{B}$ would waste nearly all of the computational budget on the unimportant low-probability region.

2.5 Discussion and Conclusions

In this paper, we presented a major extension of the procedure of Bliznyuk *et al.* (2008) for Bayesian treatment of computationally expensive nonlinear regression problems. The method proposed herein uses interpolation of the logarithm l of the exact posterior density using RBFs in order to construct a gradually improving sequence of logarithms $\widetilde{l}_i$ of the approximate posterior densities $\widetilde{\pi}_i$. As the approximate posterior densities become more accurate, so do the approximate HPD regions. As a consequence, the knots for RBF fitting, selected to satisfy a maximum separation criterion, are chosen on a true HPD region for π. Our approach has a considerable advantage over the approximation over the whole

parameter space $\mathfrak{E}$ (Kennedy and O'Hagan, 2001) in keeping low the proportion of knots chosen on the unimportant low-probability region. Unlike in the earlier papers, `GRIMA` does not require derivatives of π and is applicable to the problems with connected HPD regions of arbitrary shapes. Our stopping criterion, which is based on the estimated total variation norm (Appendix A.1) between the "most recent" and all of the preceding approximate densities, allows termination of the algorithm when the approximate densities become sufficiently accurate, thereby reducing the waste of simulator runs.

We illustrated the progress and robustness of `GRIMA` on a synthetic test problem of Rasmussen (2003) in Section 2.3.3. Subsequently, our algorithm was applied to solve the Bayesian parameter calibration problem for a real data set from the Town Brook watershed. Our results indicate that our algorithm is capable of reducing the computational load (relative to MCMC sampling from the exact posterior density), at least, by an order of magnitude.

In the applications considered in this paper, we attribute the success of `GRIMA` to doing the linear change of variables before RBF fitting (discussed in the end of Section 2.3.1 and in the end of Section 2.3.2) and on doing the approximation only over the approximate HPD region. The merits of updating of the RBF surface (Appendix A.2) over refitting it from scratch, although not manifested in these applications, will be realized when the number of knots for interpolation is on the order of thousands.

In this paper, we assumed that the HPD region is connected, which is crucial for ensuring that the random walk MCMC sampler traverses the support of the target distribution easily. However, it is possible to extend the algorithm to deal with multimodal posterior densities with disconnected modes: one would

apply GRIMA locally around each high-probability mode, represent the approximate posterior density as a mixture density and then proceed as discussed in Tjelmeland and Hegstad (2001). Thus, one can view GRIMA as a procedure for local parameter uncertainty analysis.

It is notable that, with only minor modifications, GRIMA can be parallelized. Indeed, one just needs to replace the **for** loop of the Algorithm 2.3.1 (lines 6–16) with an assignment of (at most) J candidate points to (at most) J processors, so that the exact posterior density π can be evaluated in parallel at the candidate points. It is also not hard to spread the computational load of the MCMC simulation over multiple processors, e.g., by running several shorter Markov chains in parallel.

Currently, our research involves the exploitation of computational gains that arise from having the likelihood function being cheap to evaluate with respect to non-simulator parameters (held fixed in the application of Section 2.4.2) when constructing the RBF approximation, in order to reduce the impact of the curse of dimensionality inherent to interpolation.

2.6 Appendix

2.6.1 Independent Sampling of Rasmussen's Density

In this section we outline an algorithm for *i.i.d.* sampling from an unnormalized probability density defined as

$$\pi(\eta) := \exp\{-0.5 \cdot [(\eta^{\mathsf{T}}\eta - a)/b]^2\} \quad \text{for } \eta \in \mathbb{R}^2.$$

Algorithm 2.6.2 sample from Rasmussen's density

Ensure: X is a realization from Rasmussen's donut density π.

1: draw $Z \sim Normal(a, b^2)$

2: **while** $Z < 0$ **do**

3: draw $Z \sim Normal(a, b^2)$

4: **end while**

5: draw $W \sim Normal(0, I_2)$ and set $W \leftarrow W/\|W\|_2$

{alternatively, draw $U \sim Uniform(0, 2\pi)$ and set $W \leftarrow [\cos(U), \sin(U)]^\mathsf{T}$}

6: **return** $X \leftarrow W \cdot \sqrt{Z}$

Examination of π reveals that $\|\eta\|_2^2$ follows a truncated normal distribution and that π is radially symmetric. These two observations, incorporated in the Algorithm 2.6.2, completely determine the probability density. It is required that the collection of random variables generated in the course of algorithm is independent.

CHAPTER 3

BAYESIAN INFERENCE USING EFFICIENT INTERPOLATION OF COMPUTATIONALLY EXPENSIVE DENSITIES WITH VARIABLE PARAMETER COSTS

3.1 Introduction

The core of Bayesian inference is formalization of beliefs about model parameters η given the observed data Y using the posterior density π of η. For most nontrivial problems, analytical derivation of characteristics of individual components η_i, such as posterior moments, quantiles or other functionals of the marginal density of η_i, is intractable and one has to resort to Markov Chain Monte Carlo (MCMC) to sample from π in order to estimate the desired quantities from the sample. Each transition of the Markov chain typically requires an evaluation of the target density π at a candidate state drawn from a proposal density. Therefore, when π is computationally expensive to evaluate, only short MCMC runs are feasible, which is not sufficient for accurate estimation.

The focus of our work is reduction of computational burden of MCMC via efficient construction of approximate posterior densities in settings where η is high-dimensional but there is structure in the computation to evaluate the exact posterior density π or its logarithm l. In many such problems, it is possible to identify in η the minimal subset β of variables responsible for the expensive computation, and thereby to partition η as $\eta = [\beta, \zeta]$. Consequently, l can be evaluated at a new parameter value $\eta^* = [\beta^*, \zeta^*]$ in two steps: (i) a computationally expensive step $v = G_E(\beta^*)$, followed by (ii) a cheap calculation $G_C(v, \beta^*, \zeta^*)$ or even $G_C(v, \zeta^*)$, so that $l([\beta^*, \zeta^*]) = G_C[G_E(\beta^*), \beta^*, \zeta^*]$.

For example, consider the linear model $Y = Xb + e$, where e has a multivariate normal (*MVN*) distribution with a zero mean and a covariance matrix $V := V(\gamma)$ parameterized by γ. If the dimension of the vector of observations Y is large, V^{-1} is not available in closed form and V does not have exploitable sparsity structure, as is often the case in spatio-temporal models, the cost of evaluation of the posterior density of $[b, \gamma]$ is dominated by the factorization of V, which plays the role of G_E, while the cost to complete the rest of the computations G_C is of smaller magnitude.

Another class of examples comes from the field of computationally expensive *inverse problems*, discussed in Kennedy and O'Hagan (2001). In the simplest case, the vector of observed data Y is modeled as $Y = f(\beta) + e$, where f is the vector-valued computationally expensive "black-box" nonlinear regression function (known as *simulator*) and e is the vector of errors that has a multivariate normal density. Evaluation of f at β^* presents the main computational challenge which we associate with G_E, and once the value $f(\beta^*)$ is known, the remaining computation G_C to evaluate l is cheap.

In this paper, we are concerned with systematic examination of computationally tractable approaches to approximate l when its argument η separates into the "expensive" and "cheap" blocks. The main idea is simple: evaluate G_E at a set of points on a high-probability region for β and use the values $G_E(\beta^{(i)})$ to approximate l by an interpolant $\widetilde{l}$. The resulting cheap-to-evaluate *surrogate* surface $\widetilde{l}$ can be used to define a proposal density for MCMC sampling from π that produces candidate states with a high probability of being accepted (Christen and Fox, 2005; Rasmussen, 2003), or as a substitute for l if the approximation is accurate enough (Bliznyuk et al., 2008).

Reduction of computational burden for such models via approximations to π (or its logarithm l) attracted considerable attention in recent years. To improve the efficiency of MCMC, Rasmussen (2003) uses best linear unbiased prediction (BLUP, known in geostatistics as *kriging*) to interpolate l *directly*, i.e., at the knots chosen on the η-space, under the assumption that l is a realization of a Gaussian process (GP). As a consequence, his heuristic approach is sensitive to the "curse of dimensionality" and only posterior densities with $\dim(\eta)$ around 15 are conjectured to be tractable (Rasmussen, 2003, p. 659).

The main contribution of our paper is extension of Rasmussen's interpolant to high-dimensional models where only a subvector β of η is "expensive". In the class of zero-mean GPs with separable correlation functions (as defined in Section 3.3), we are able to derive a direct optimal interpolant, for which the interpolation error is controlled only by the placement of knots in β, rather than in η. This causes the effective dimension of the interpolation problem to drop from $\dim(\eta)$ to $\dim(\beta)$, and is capable of reducing the number of expensive computations G_E necessary to construct a direct interpolant by orders of magnitude when $\dim(\beta)$ is low and $\dim(\eta)$ is high. As we illustrate in Section 3.4.2 for the above linear model example with $\dim(\eta) = 34$ and $\dim(\beta) = 3$, fewer than 50 β-knots are required to obtain a very accurate approximation to l. To address situations in which these assumptions on the GP may be overly restrictive, we discuss generalizations in Section 3.5.

In addition, we extend the idea of the *indirect* approximation from the field of inverse problems (e.g., Kennedy and O'Hagan, 2001) to general computationally intensive statistical problems with variable parameter costs. We propose to use the indirect interpolants of l of the form $\widetilde{l}([\beta^*, \zeta^*]) = G_C[\widetilde{G_E}(\beta^*), \beta^*, \zeta^*]$,

where the ith component of $\widetilde{G_E}$ interpolates the ith component of the "output" of G_E. To the best of our knowledge, this simple idea has not been considered outside of the literature on approximation of the simulator f in inverse problems. The dimension of each subproblem of interpolating a component of G_E is $\dim(\beta)$. We recommend and use interpolation under a radial basis function (RBF) model, which is a lot cheaper to fit than kriging models when the dimension of the "output" of G_E is very large (Sections 3.2 and 3.5.2).

The paper is structured as follows: Necessary notation and definitions for the interpolants we use are introduced in Section 3.2. Section 3.3 is devoted to derivation and analysis of properties of the optimal direct interpolant. Application of both of the proposed methods, *direct* and *indirect*, is illustrated in Section 3.4. Possible extensions of the proposed direct interpolant and relative merits of direct and indirect approximations are discussed in Section 3.5.

3.2 Notation and Definitions of Interpolants

In this section, we introduce relevant notation and we define the RBF and kriging interpolants that are used in this paper.

3.2.1 Notation

All variables are assumed to be (column) vectors or matrices of size specified in the appropriate definition; this will *not* be emphasized by bold-face notation. We define the distance between a point x and a set $\mathcal{S}$ as $\text{dist}(x, \mathcal{S}) = \inf_{x' \in \mathcal{S}} \|x - x'\|_2$. Only when applied to a vector, a single subscript notation

is used to "extract" components, e.g., x_i is the ith component of x. For sets $\mathcal{S}_1$ and $\mathcal{S}_2$, $\mathcal{S}_1 \backslash \mathcal{S}_2$ will denote the set of elements of $\mathcal{S}_1$ that are not in $\mathcal{S}_2$ and $|\mathcal{S}_1|$ will give the number of elements in $\mathcal{S}_1$. We represent sets as lists with lexicographic ordering of elements. The direct (Cartesian) product operator $\oplus$ is used to "merge" elements from lists $\mathcal{S}_1$ and $\mathcal{S}_2$ as

$$\mathcal{S}_1 \oplus \mathcal{S}_2 := \{[x^{(i)}, y^{(1)}], \ldots, [x^{(i)}, y^{(|\mathcal{S}_2|)}] : i = 1, \ldots, |\mathcal{S}_1|\}$$

(To emphasize the ordering of elements in the list $\mathcal{S}_1 \oplus \mathcal{S}_2$ necessary for the proof of Proposition 2 of Appendix 3.7.1, we did not use the conventional notation, $\times$, for the Cartesian product of two sets.)

For a scalar-valued function $g : (x, y) \mapsto g(x, y)$, we extend its definition to finite sets as $g : (\mathcal{S}_1, \mathcal{S}_2) \mapsto g(\mathcal{S}_1, \mathcal{S}_2)$, where $g(\mathcal{S}_1, \mathcal{S}_2)$ is a $|\mathcal{S}_1| \times |\mathcal{S}_2|$ matrix whose ijth element is $g(x^{(i)}, y^{(j)})$ for $x^{(i)} \in \mathcal{S}_1$ and $y^{(j)} \in \mathcal{S}_2$. We use an analogous extension for functions of a single vector argument.

3.2.2 Definitions of Interpolants

In the most general form, an RBF or a kriging interpolant of a scalar-valued function g at a set of points $\mathcal{D} = \{x^{(1)}, \ldots, x^{(K)}\}$ is given by

$$\widetilde{g}(x) = \sum_{i=1}^{K} a_i \phi(x, x^{(i)}; \theta) + q(x; c), \tag{3.1}$$

where ϕ is a basis function parameterized by θ and q is a "model" for the systematic variation in g. We restrict attention to "tails" q that are linear in c such as low-degree polynomials in x with coefficients c. The *basis function parameters* θ enter into the Equation (3.1) in a nonlinear way, whereas the interpolant is linear in the *coefficients* $a = [a_1, \ldots, a_K]^{\mathsf{T}}$ and c.

In the case of kriging, $\phi(\cdot, \cdot; \theta)$ is a positive definite function. In the instances of *simple kriging* we consider later, we use a Gaussian basis function defined as

$$\phi(x, y; \theta) = \exp\left\{ - \sum_{j=1}^{\dim(x)} \theta_j (x_j - y_j)^2 \right\} \tag{3.2}$$

Simple kriging assumes that the mean function of the GP is known (Cressie, 1991) and the BLUP is computed after it has been subtracted from the GP. For RBF interpolation, we use a cubic basis function $\phi(x, y; \theta) := \|x - y\|_2^3$ and a linear tail $q(x; c) := [1, x^\mathsf{T}] \cdot c$. Kriging and RBF interpolation with these choices of basis functions were used by Rasmussen (2003) and Bliznyuk et al. (2008), respectively. Merits of kriging and RBF interpolation were discussed in Cressie (1991, sec. 4.4) and Bliznyuk et al. (2008).

For a fixed vector θ of basis function parameters, the vectors of coefficients $a = [a_1, \ldots, a_K]^\mathsf{T}$ and c can be obtained by solving the system of dual kriging equations given in Cressie (1991, sec. 4.4.5), which requires $\mathcal{O}(K^3)$ flops. The right-hand side of this system is determined by the values g takes at $\mathcal{D}$ and linear constraints on a and c to ensure existence and uniqueness of the solution.

Our focus is interpolation of the log-posterior l that can be represented as $l(\beta, \zeta) = G_C[G_E(\beta), \beta, \zeta]$, where evaluation of G_E is expensive, but that of G_C is cheap. The *indirect a*pproximation (`INDA`) we consider has the form $\widetilde{l}(\beta, \zeta) = G_C[\widetilde{G}_E(\beta), \beta, \zeta]$, where $\widetilde{G}_E$ is the component-wise interpolant of the "output" of G_E. We use the cubic RBF defined above since fitting does not require estimation of basis function parameters (because ϕ does not depend on θ) and, consequently, only a single matrix factorization is required to solve the interpolating equations for multiple right-hand sides determined by values of components of G_E at the knots $\mathcal{D}$. The *direct* optimal interpolant of l by simple kriging (`DOSKA`) after l has been "recentered" to have a zero mean will be

derived in Section 3.3.1 and fitting issues will be addressed in Section 3.3.3.

As we noted in the introduction, knots for `DOSKA` and `INDA` are chosen on β-space rather than η-space. It is crucial that these be selected on a high probability density (HPD) region for β. The true HPD region for β is unknown but can be approximated using a local quadratic fit or a more general nonparametric approximation of l as discussed in Bliznyuk et al. (2008) and Bliznyuk et al. (2008, submitted), respectively. Here we follow fitting recommendations outlined in these papers. In particular, to reduce the sensitivity of the interpolants to scaling of variables, we fit the interpolants upon a "sphering" (sec. 7.3 of Scott, 1992) transformation $\beta \mapsto H^{-1}\beta$, where H is any square matrix satisfying $HH^{\mathsf{T}} \approx Var(\beta)$.

If $\widetilde{l}$ is a direct or an indirect interpolant of l, the approximate posterior density $\widetilde{\pi}$ is defined as

$$\widetilde{\pi}([\beta^*, \zeta^*]) = \exp\{\widetilde{l}([\beta^*, \zeta^*])\} \cdot \mathbb{I}\{\beta^* \in \mathcal{N}\}, \tag{3.3}$$

where $\mathbb{I}$ is the indicator function and $\mathcal{N}$ is some neighborhood of the β-knots $\mathcal{B}$ used for interpolation. Thus we restrict the support of $\widetilde{\pi}$ to the region where $\widetilde{l}$ is well-approximated. A more extensive discussion of the knot selection and fitting issues can be found in Bliznyuk et al. (2008) and Bliznyuk et al. (2008, submitted).

3.3 DOSKA — Direct Optimal Separable (Simple) Kriging Approximation

The focus of this section is derivation and study of the properties of the *direct* interpolant of l that distinguishes between expensive and cheap computations in the evaluation of l. We proceed under the *assumption* that l is a realization of a GP with mean 0, constant variance σ^2 and a correlation function satisfying the separability condition

$$C_\eta([\beta^{(1)}, \zeta^{(1)}], [\beta^{(2)}, \zeta^{(2)}]) = C_\beta(\beta^{(1)}, \beta^{(2)}) \cdot C_\zeta(\zeta^{(1)}, \zeta^{(2)}). \tag{3.4}$$

The implications and possible relaxations of this assumption are discussed in Sections 3.3.2 and 3.5. In Section 3.3.1, we derive an optimal interpolant as a solution to the following adaptive design problem: given a finite set $\mathcal{B}$ of β-knots, construct a set of knots

$$\mathcal{D}([\beta^*, \zeta^*]) = \{[\beta^{(j)}, \zeta^{(i,j)}] : 1 \leq i \leq K_j, \beta^{(j)} \in \mathcal{B}\} \tag{3.5}$$

to minimize the error of prediction of $l([\beta^*, \zeta^*])$ with the best linear unbiased predictor (BLUP) $E\{l([\beta^*, \zeta^*])|l(\mathcal{D})\}$. Thus the set $\mathcal{B}$ is "fixed" and the "expensive" subvector of each element of $\mathcal{D}([\beta^*, \zeta^*])$ is an element of $\mathcal{B}$. In Section 3.3.2 we study the properties of the proposed interpolant. Fitting issues are addressed in Section 3.3.3.

3.3.1 Derivation of DOSKA

Let $\mathcal{D} = \{[\beta^{(j)}, \zeta^{(i,j)}] : 1 \leq i \leq K_j, \beta^{(j)} \in \mathcal{B}\}$ be any finite set of η-knots that can be created using β-knots from $\mathcal{B}$. Define $\mathcal{Z}^* := \{\zeta^*\} \cup \{\zeta^{(i,j)} : [\beta^{(j)}, \zeta^{(i,j)}] \in \mathcal{D} \text{ for some } j\}$

By Proposition 1 proved Appendix 3.7.1,

$$Var\{l([\beta^*, \zeta^*])|l(\mathcal{D})\} \geq Var\{l([\beta^*, \zeta^*])|l(\mathcal{B} \oplus \mathcal{Z}^*)\},$$

as $\mathcal{D} \subset \mathcal{B} \oplus \mathcal{Z}^*$. Since $E\{l([\beta^*, \zeta^*])|l(\mathcal{D})\}$ and $E\{l([\beta^*, \zeta^*])|l(\mathcal{B} \oplus \mathcal{Z}^*)\}$ are both unbiased, the latter predictor improves over the former.

From Proposition 2 proved in Appendix 3.7.1 it follows that, under separability of C_η of Equation (3.4), $Var\{l([\beta^*, \zeta^*])|l(\mathcal{B} \oplus \mathcal{Z}^*)\} = Var\{l([\beta^*, \zeta^*])|l(\mathcal{B} \oplus \zeta^*)\}$. Hence $E\{l([\beta^*, \zeta^*])|l(\mathcal{B} \oplus \zeta^*)\}$ improves over $E\{l([\beta^*, \zeta^*])|l(\mathcal{D})\}$ and cannot be improved upon no matter what $\mathcal{D}$ is constructed using the knots in $\mathcal{B}$. The resulting Direct Optimal Separable (Simple) Kriging Approximant (DOSKA) has the form

$$\widetilde{l}_D([\beta^*, \zeta]) := E\{l([\beta^*, \zeta^*])|l(\mathcal{B} \oplus \zeta^*)\} = C_\beta(\beta^*, \mathcal{B}) \cdot C_\beta(\mathcal{B}, \mathcal{B})^{-1} \cdot l(\mathcal{B} \oplus \zeta^*) \quad (3.6)$$

We suppressed dependence of C_β on the correlation function parameters θ, which will be examined in Section 3.3.3 when we discuss fitting.

In Figure 3.3.1, we visualize the derivation of DOSKA when $\dim(\eta) = 2$ and $\dim(\beta) = 1$. One is given the set $\mathcal{B}$ of 10 β-knots, denoted by Δ. The goal is to predict $l(\eta^*)$ at a new site $\eta^* = [\beta^*, \zeta^*]$, marked by x. A reasonable strategy for creation of the set $\mathcal{D}$ of η-knots (marked by o) attempts to cover the (elliptical) HPD region for η. To improve the prediction error of the BLUP given $\mathcal{D}$, one (i) projects $\{\eta^*\} \cup \mathcal{D}$ onto the ζ-space to obtain $\mathcal{Z}^* = \{\zeta^*\} \cup \mathcal{Z}$ (with ζ^* marked by $*$ and $\mathcal{Z}$ marked by $\triangleright$) and (ii) constructs $\mathcal{B} \oplus \mathcal{Z}^*$ (marked by +, large and small). Under the above assumptions on l, the BLUP given the knots $\mathcal{B} \oplus \mathcal{Z}^*$ is the same as BLUP given the knots $\mathcal{B} \oplus \zeta^*$ (marked by large +).

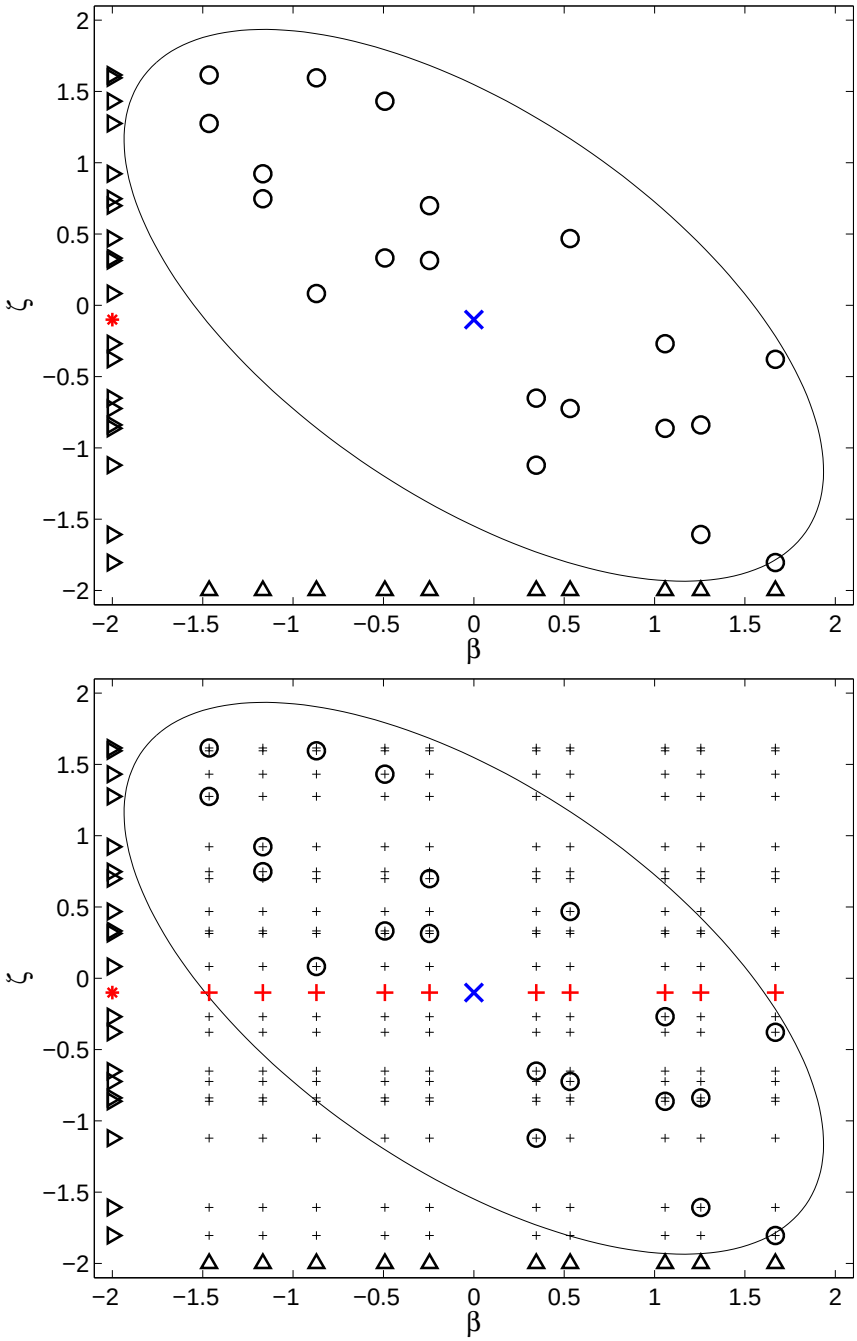

Figure 3.1: Illustration of derivation of DOSKA: The goal is to obtain a prediction of l at $\eta^* = [\beta^*, \zeta^*]$ (x) using the set $\mathcal{B}$ of β knots (Δ). *Top:* the knots $\mathcal{D}$ (o) are selected to cover the elliptical HPD region. $\{\eta^*\} \cup \mathcal{D}$ is projected onto the ζ-space to produce $\mathcal{Z}^*$ ($\triangleright$ and $*$). *Bottom:* $\mathcal{B} \oplus \mathcal{Z}^*$ is marked by large and small +; $\mathcal{B} \oplus \zeta^*$ is marked by large +.

3.3.2 Analysis

In this section we make a few important observations about DOSKA.

1. The predictor of Equation (3.6) does not assume that either C_β or C_ζ is separable, although this assumption is often made out of convenience in applications of kriging (Rasmussen, 2003). Remarkably, under the above assumptions on the GP, DOSKA does not depend on the choice of C_ζ at all, as can be seen from Equation (3.6).

2. It follows by Taylor's expansion of $\widetilde{l}_D$ of Equation (3.6) in the neighborhood of its maximizer $[\widehat{\beta}, \widehat{\zeta}]$ that, under the assumed separability of the correlation function of Equation (3.4), the unnormalized approximate posterior density $\exp(\widetilde{l}_D)$ implies neither the independence of β and ζ, nor the separability of the covariance matrix of β and ζ.

3. If $\beta^* \in \mathcal{B}$, then $\widetilde{l}_D([\beta^*, \zeta^*]) = l([\beta^*, \zeta^*])$. In this case, the gradients with respect to ζ of the left- and right-hand sides are equal, but the gradients with respect to β are not.

4. The derivatives of DOSKA are available analytically so long as C_β is differentiable and l is differentiable in ζ. They can be used for efficient sampling from the approximate posterior density using gradient-based MCMC samplers such as Langevin diffusions (Robert and Casella, 1999).

5. Given a compact set S and a set of n knots $\mathcal{D}_n \subset S \subset \mathbb{R}^d$, the *minimax* distance between $\mathcal{D}_n$ and S is defined as $m(\mathcal{D}_n, S) = \max_{x \in S} \text{dist}(x, \mathcal{D}_n)$. This is the minimum value of the coverage "radius" r that ensures that every point in S is within distance r from $\mathcal{D}_n$. Convergence of interpolants to the underlying function g is governed by $m(\mathcal{D}_n, S)$, and often the rate

is $\mathcal{O}(m(\mathcal{D}_n, S)^\alpha)$, where $\alpha > 0$ is determined by the smoothness g, by the choice of the interpolant and by the L_p norm used to measure distance between g and the interpolant. It is possible to show that the fastest rate, at which $m(\mathcal{D}_n, S)$ shrinks is $\mathcal{O}(n^{-1/d})$. For example, if $S = [0,1] \subset \mathbb{R}$, $m(\mathcal{D}_n, S) \geq 1/(2n)$.

If the full direct approximation to (continuous) l is used (like in Rasmussen, 2003) and the set $\mathcal{D}_n$ of η-knots is chosen on some subset S of $\mathbb{R}^{\dim(\eta)}$ to minimize $m(\mathcal{D}_n, S)$, the point-wise convergence rate of the kriging interpolant to l is $\mathcal{O}(n^{-1/\dim(\eta)})$, where $\dim(\eta) = \dim(\beta) + \dim(\zeta)$. On the other hand, the corresponding convergence rate for DOSKA is not influenced by $\dim(\zeta)$, and is $\mathcal{O}(n^{-1/\dim(\beta)})$. Stated differently, DOSKA interpolates each element of the family of functions $\{l([\cdot, \zeta]) : \zeta \in \mathfrak{Z}\}$ using the same set of knots $\mathcal{D}_n$ chosen in $\mathbb{R}^{\dim(\beta)}$, and is optimal within the rich class of kriging interpolants under the assumptions of this section. (Of course, direct interpolants other than DOSKA are possible for this family of functions and we discuss extensions in Section 3.5).

We are not aware of the results about L_p convergence rates for interpolation by kriging, but we expect that the results similar to those for RBFs (e.g., Buhmann 2002, chap. 5) may be possible.

3.3.3 Fitting of DOSKA

Unlike many popular RBF interpolants that involve no basis function parameters, successful application of kriging requires estimation of parameters θ of the correlation function C_β. In this section we review two methods of estimation, maximum likelihood and K-fold cross-validation. We assume that one has (i)

knots $\mathcal{D} = \mathcal{B} \oplus \mathcal{Z}$ in a high-probability region of π and (ii) values of l at these points. For consistency with the assumption of zero mean Gaussian process made about l, we re-center l by subtracting from it the mean of $l(\mathcal{B} \oplus \mathcal{Z})$, as was done in Rasmussen (2003). This does not influence the interpretation of the log-posterior l since it is only known up to an additive constant.

The assumption that l is a realization of a Gaussian process allows one to write down the likelihood of $l(\mathcal{B} \oplus \mathcal{Z})$. This is a multivariate normal density with mean 0 and covariance matrix $\sigma^2 \cdot C_\beta(\mathcal{B}, \mathcal{B}) \otimes C_\zeta(\mathcal{Z}, \mathcal{Z})$, by separability of C_η and our choice of knots $\mathcal{D}$. Thanks to the Kronecker product representation, the log-likelihood can be evaluated efficiently.

An alternative K-fold cross-validation criterion (KfCV) reuses subsets of the "data" $l(\mathcal{D})$ for validation, thereby guarding against overfitting. In our setting, its form is

$$F(\theta) := \sum_{i=1}^{K} \|\widetilde{l}_{i,\theta}(\mathcal{B}_i \oplus \mathcal{Z}) - l(\mathcal{B}_i \oplus \mathcal{Z})\|_F^2, \tag{3.7}$$

where

$$\widetilde{l}_{i,\theta}([\beta^*, \zeta^*]) := C_\beta(\beta^*, \mathcal{B}_{-i}; \theta) \cdot C_\beta(\mathcal{B}_{-i}, \mathcal{B}_{-i}; \theta)^{-1} \cdot l(\mathcal{B}_{-i} \oplus \zeta^*), \tag{3.8}$$

$\{\mathcal{B}_1, \ldots, \mathcal{B}_K\}$ is a partition of $\mathcal{B}$, $\mathcal{B}_{-i} := \mathcal{B} \backslash \mathcal{B}_i$ is the set difference and, for a matrix A, $\|A\|_F^2 = \sum_{i,j} A_{i,j}^2$ (squared Frobenius norm of A). To compute $\widetilde{l}_{i,\theta}(\mathcal{B}_i \oplus \mathcal{Z})$ for a given value of θ, it is necessary to obtain a factorization of $C_\beta(\mathcal{B}_{-i}, \mathcal{B}_{-i}; \theta)$ and to evaluate $l(\mathcal{B}_{-i} \oplus \zeta)$ for all $\zeta \in \mathcal{Z}$. The overall cost of factorizing $C_\beta(\mathcal{B}_{-i}, \mathcal{B}_{-i}; \theta)$ for all i can be made equal to a small multiple of $|\mathcal{B}|^3$, as opposed to $\mathcal{O}(K \cdot |\mathcal{B}|^3)$ in a naïve implementation, if one computes QR or Cholesky factorizations of $C_\beta(\mathcal{B}_{-i}, \mathcal{B}_{-i}; \theta)$ by downdating a single factorization of $C_\beta(\mathcal{B}, \mathcal{B}; \theta)$ for each i (Golub and Van Loan, 1996, sec. 12.5). For example, for the choice $K = |\mathcal{B}|/4$ that we use, computational savings can be enormous if $|\mathcal{B}|$ is large.

Many of the popular correlation functions are differentiable in θ, and so both F and the negative of the log-likelihood function can be minimized efficiently by numerical optimization. In our experiments, either of these criteria often has multiple minimizers, and so multiple starting points for optimization are necessary.

In our experiments to determine which method requires fewer knots, we had more success with KfCV. In particular, for Rasmussen's test problem 2 discussed below in Section 3.4.1, MLE required roughly twice as many β-knots as KfCV when $\dim(\beta)$ is above 7 and $\dim(\eta) = 10$. For this reason, we use the KfCV criterion in the experiments of this paper.

3.4 Simulation Studies

In this section, we examine our direct and indirect interpolants on cheap-to-evaluate densities from which a long sample can be obtained efficiently for reference purposes. Section 3.4.1 contrasts `DOSKA` with the direct interpolant that ignores distinction between "expensive" and "cheap" parameters. Section 3.4.2 explains and illustrates how `DOSKA` and `INDA` can be applied to a computationally expensive linear model problem outlined in the introduction, for which application of the direct interpolant of Rasmussen is infeasible.

The entering paragraphs of each subsection provide high-level overviews of the contents.

3.4.1 MVN Density with Correlation

The study of this section was inspired by the work of Rasmussen (2003) that chose knots on the η-space for his GP interpolant under the Gaussian correlation function. We adopt his "equicorrelation" test problem 2 that assumes that l is the logarithm of a 10-dimensional $MVN(0, \Sigma)$ density, with the entries of Σ on the main diagonal equal to 1 and all off-diagonal entries equal to $.908$. We treat this MVN density as a "black-box" posterior density for a 10-dimensional parameter vector η. We investigate the *impact of partitioning* of the argument $\eta = [\beta, \zeta]$ of l into the "expensive" and "cheap" blocks *on the number of knots* required to ensure an accurate approximation of l by DOSKA. In our experiments, $\dim(\eta) = 10$ and $\dim(\beta)$ ranges from 1 to 10. (The distinction between "expensive" and "cheap" parameters is artificial in this syntethetic test problem.)

The β-knots are chosen to cover the exact .99 HPD region for β after "sphering" (see Section 3.2.2). To select knots we use the "greedy" maximin heuristic from Appendix A.2 of Bliznyuk et al. (2008). For each value of $\dim(\beta)$ studied, we start with the terminal number of knots for $\dim(\beta) - 1$ and add extra knots in 20% increments until a discrepancy measure between the exact and approximate posterior densities falls below a specified threshold δ. As the discrepancy measure, we use the total variation (*TV*) norm – defined in Appendix A.1 – for each component of η under the exact and approximate posterior densities. More precisely, suppose that $N(d-1, \delta)$ β-knots are sufficient to ensure that the *TV* norm for each component of η is below δ when $\dim(\beta) = d - 1$. When $\dim(\beta)$ is increased from $d-1$ to d, we initially choose $K = N(d-1, \delta)$ β-knots and estimate the component-wise *TV* norms for η. If the *TV* norm between the "exact" and "approximate" samples for some η_i exceeds δ, we augment the

set of K knots with additional $K_0 \approx .2K$ knots, set $K \leftarrow K + K_0$, and refit DOSKA so that a new "approximate" sample can be collected from the updated interpolant for comparison with the "exact" sample. This process is repeated until the maximum component-wise *TV* norm falls below δ, in which case we set $N(d, \delta) = K$ and increase $\dim(\beta)$ from d to $d + 1$. We note that $N(d, \delta)$ is a random variable because the knot selection procedure is stochastic.

We estimate the component-wise *TV* norms as outlined in Appendix A.1. For that purpose, we use an *i.i.d.* sample from the exact posterior density $\exp(l)$ and a sample from the approximate posterior density $\exp(\widetilde{l}_D)$ obtained using a Metropolis-Hastings independence sampler (Tierney, 1993) with $\exp(l)$ as the proposal. If DOSKA is accurate, $\exp(l) \approx \exp(\widetilde{l}_D)$ and an MCMC sample from $\exp(\widetilde{l}_D)$ is essentially an *i.i.d.* sample. Each sample is of size 10^4.

For each *trial*, $\dim(\eta) = 10$ and $\dim(\beta)$ varies from 1 to 9. We repeated each trial 9 times with different placements of knots. The parameters θ of the Gaussian correlation function [Equation (3.2)] used in DOSKA were estimated by KfCV as discussed in Section 3.3.3. The value $\delta = .03$ was used as an upper bound on the maximum component-wise *TV* norm, which corresponds to a very accurate approximation. For example, the *TV* norm between two *i.i.d.* normal samples of size 10^4 (estimated using 100 simulated data sets) has sample mean and standard deviation of about $.015$ and $.004$, respectively.

The results of our study are summarized in Figure 3.4.1, where we plot against $\dim(\beta)$ the sample median and confidence bounds of level .9 for $N(\dim(\beta), \delta)$. Based on a separate experiment with 20 trials, $N(10, .03)$ is about 70 with little variability about this value. Comparison of values of $N(\dim(\beta), .03)$ on this plot with $N(10, .03)$ allows one to appreciate the potential

computational savings of exploiting the separation of η into the "expensive" and "cheap" blocks. For example, DOSKA requires between 10 and 18 knots when $\dim(\beta) = 3$, whereas 70 knots are necessary if separation is ignored.

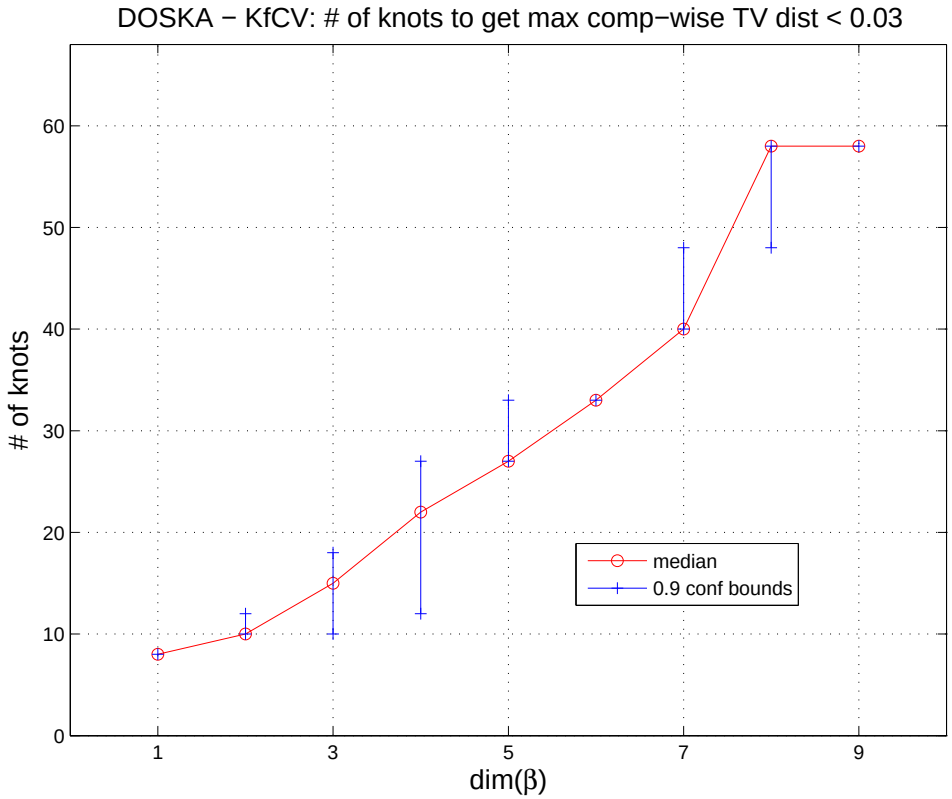

Figure 3.2: MVN problem of Section 3.4.1: Sample median and confidence bounds of level .9 for the estimated minimum required number of β-knots necessary to achieve maximum component-wise TV norm less than $\delta = .03$. The plot is based on 9 trials. KfCV is used to estimate DOSKA parameters.

3.4.2 A Linear Model With Unstructured Covariance Matrix

Here we examine the impact of separation of the "expensive" and "cheap" computations in evaluation of the log-likelihood (or log-posterior) function for a high-dimensional linear model (LM) with normal errors. The parametric corre-

lation matrix for the errors is assumed to have little or no structure which makes evaluation of the log-likelihood computationally expensive. Such problems are abundant in spatio-temporal modeling when the dependence is modeled parsimoniously using a low-dimensional parametric correlation function. As can be seen from Gneiting (2002b), it is unusual to have more than 5 correlation function parameters in such models.

This section is organized as follows: In Section 3.4.2 we provide a "big picture" view of approximation framework without a particular correlation function in mind. Our strategy is illustrated in Section 3.4.2 for a linear model with errors following an autoregressive (AR) process.

Approximation Framework for Linear Model (LM)

The log-likelihood function given by

$$L(\gamma, b, \sigma^2) = -\frac{1}{2}\left\{n\log(\sigma^2) + \log|V| + (Y - Xb)^{\mathsf{T}}(\sigma^2 V)^{-1}(Y - Xb)\right\}, \qquad (3.9)$$

where Y is the $n \times 1$ vector of observations, X is the $n \times p$ design matrix (of predictors), and $V := V(\gamma)$ is the $n \times n$ correlation matrix parameterized by a vector γ, so that $Cov(Y_i, Y_j) = \sigma^2 V_{ij}(\gamma)$. For many correlation functions of interest, V does not have computationally exploitable structure, and the cost to factorize V in order to evaluate the *determinant* $|V|$ and the quadratic form $(Y - Xb)^{\mathsf{T}} V^{-1}(Y - Xb)$ is $\mathcal{O}(n^3)$ in the worst case. For example, if $n = 10^4$, computing a Cholesky factorization of V takes about 100 seconds on a modern workstation, and so a fully Bayesian analysis using MCMC sampling of all of the parameters is computationally prohibitive.

Examination of the log-likelihood reveals that after $\log|V(\gamma)|$, $Y^{\mathsf{T}}[V(\gamma)]^{-1}Y$,

$Y^{\mathsf{T}}[V(\gamma)]^{-1}X$ and $X^{\mathsf{T}}[V(\gamma)]^{-1}X$ have been computed using a single factorization of $V(\gamma)$ for a given value of γ, the extra cost to evaluate the log-likelihood is $\mathcal{O}(p^2)$. Thus, we associate the parameter vector β with γ responsible for the "expensive" computation G_E of all the quantities above that involve γ, and the parameter ζ with $[b, \sigma^2]$ responsible for the remaining "cheap" computation G_C. Since the dimension of the "output" of G_E does not depend on n, the cost to store the values of G_E at the points $\beta^{(i)}$, at which $G_E(\beta^{(i)})$ has been computed, is negligible, and so is the cost to evaluate $L(\beta^{(i)}, \zeta^*)$ for a new value of ζ^*.

Unlike in Section 3.4.1, here we do *not* assume that the β-knots for interpolation are available. We need to produce them on the unknown HPD region of the true marginal posterior density of β. We do this using our recent algorithm `GRIMA` that has shown good performance on "irregular" densities (those having non-elliptical high-probability regions and modes occurring on the boundary of the parameter space), on which existing approaches, such as that of Bliznyuk et al. (2008), can fail. After a run of an optimization algorithm to reach the HPD region, `GRIMA` reuses the points from the optimization trajectory to build a response surface that is used to select those sites for new expensive evaluations that are likely to belong to the true HPD region. The response surface is updated after each new expensive evaluation thereby becoming more accurate. The approximate HPD region is being refined until an accurate approximation to the exact HPD region and to the true posterior density over it are obtained. (More details are available in the attached draft for the `GRIMA` paper, submitted to another journal.)

As a heuristic approximation to the logarithm of the marginal posterior density for β, we use the profile log-likelihood for β, which is available analytically

as

$$\widehat{L}(\beta) := \sup_{\zeta} L(\beta, \zeta) = -\frac{1}{2}\left\{\log\det V + n\log([Y - X\widehat{b}]^{\mathsf{T}} V^{-1}[Y - X\widehat{b}])\right\} + const, \tag{3.10}$$

where $V := V(\beta)$ and $\widehat{b} := \widehat{b(\beta)} = (X^{\mathsf{T}}V^{-1}X)^{-1}X^{\mathsf{T}}V^{-1}Y$. Alternatively, one can use the *profile log-posterior* defined similarly as $\sup_{\zeta} l([\beta, \zeta])$, which would be similar to $\widehat{L}$ if the effect of the prior is small. This surface is used in `GRIMA` to generate β-knots for `DOSKA` and `INDA`, but these interpolants are fitted to the exact joint posterior density of η as was discussed in Sections 3.2.2 and Section 3.3.

Illustration on LM with AR(3) errors

Below we compare the Bayesian inference using `DOSKA` and `INDA` on a test problem with V determined by the correlation function of the $AR(d)$ process. This test problem was chosen so that the likelihood can be evaluated using $\mathcal{O}(n)$ flops (due to the conditional independence property of the AR processes) and a long MCMC run from the exact posterior density could be drawn inexpensively for reference purposes. To avoid forming V which would take $\mathcal{O}(n^2)$ flops, we used a different parameterization of the model

$$L(\beta, b, \tau^2) = const - \frac{1}{2}\cdot\left\{(n-d)\log\tau^2 + \frac{1}{\tau^2}\sum_{i=d+1}^{n}\left(e_i - \sum_{j=1}^{d}\beta_j e_{i-j}\right)^2\right\}, \tag{3.11}$$

where $e_i = Y_i - X_i b$ and X_i is the ith row of X. This is the conditional log-likelihood of $e_{d+1}, \ldots, e_n$ given $e_1, \ldots, e_d$ (Hamilton, 1993, sec. 5.3). We put flat bounded priors on all parameters so that, for practical purposes, the log-posterior density is the same as the log-likelihood of Equation (3.11). (These restrictions are made to simplify the exposition, and can be easily relaxed in a serious application.) The Equation (3.11) can be maximized analytically with

respect to $\zeta := [b, \tau^2]$ to define the profile log-likelihood like we did in Equation (3.10).

The dataset Y was simulated for $d = 3$, $p = 30$, $n = 10^3$. The true parameter values we used are $\beta = [.5, .3, .1]^\mathsf{T}$, $\tau^2 = 100$, entries of X being *i.i.d.* standard normal and b being the 30×1 vector of zeros.

To produce β-knots on the high-probability region in order to initialize `GRIMA`, the profile log-likelihood was maximized by a derivative-free optimization algorithm `CONDOR` (Vanden Berghen and Bersini, 2005), but unlike in the earlier work of Bliznyuk et al. (2008), obtaining an accurate solution is unnecessary. The search was initalized at $\beta = [0, 0, 0]^\mathsf{T}$ and took 32 evaluations of G_E to complete; 4 well spread-out knots were used to build an initial direct cubic RBF approximation in `GRIMA`. (In the retrospect, it is notable that only 2 of the 4 knots belonged to the true HPD region of level .99 and these 2 interior knots were quite far from the mode.)

`GRIMA` was allowed to add new β-knots sequentially until the *improvement in the response surface* approximation of the profile log-likelihood from adding new knots *became negligible*. More precisely, we monitored the component-wise TV norms between samples from the "current" approximate density and the preceding ones that used fewer knots. For example, judging from Figure 3.4.2, it is seen that the extra reduction in the TV norms from using more than 38 knots is negligible, and we terminate `GRIMA` after it has added 46 new knots to the 4 that came from optimization.

`DOSKA` and `INDA` were fitted to the logarithm l of the full joint posterior density of $\eta = [\beta, b, \sigma^2]$ using the same set of 50 β-knots that were produced by

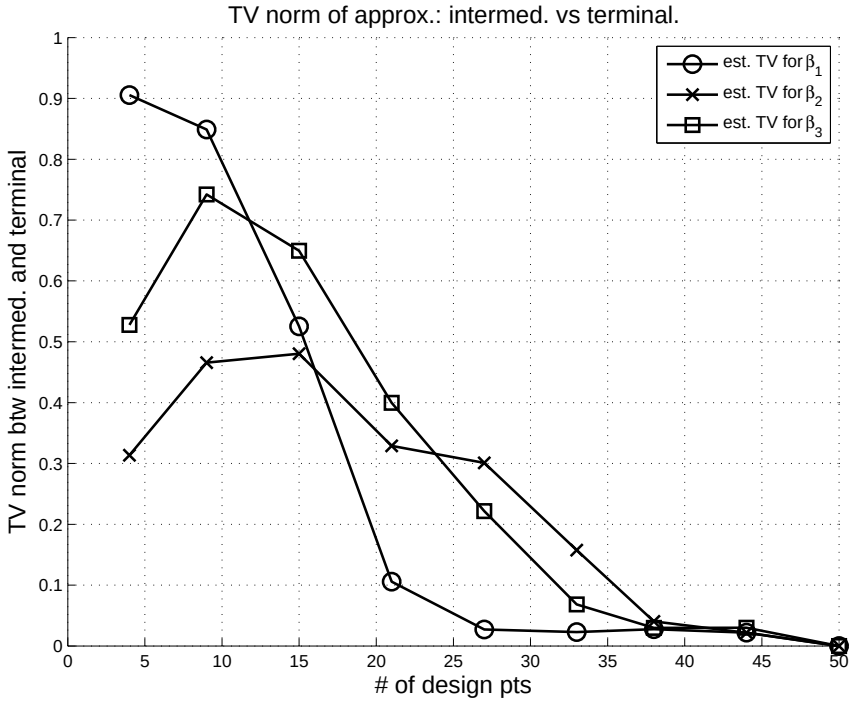

Figure 3.3: Summaries for the linear model: estimated TV norms between samples from RBF approximations to profile likelihood of Equation (3.10) with 50 knots and with smaller numbers of knots. The sample size is $3 \cdot 10^4$.

`GRIMA`. The "sphering" matrix as well as the approximate HPD region, to which the direct and indirect interpolants are restricted (see the end of Section 3.2.2) were produced by `GRIMA`.

For the purpose of reference, an MCMC sample $\mathcal{M}$ of size 10^5 from the exact joint posterior density of Equation (3.11) was collected using a random walk Metropolis-Hastings (M-H) algorithm (Tierney, 1993). The parameters of the sampler, estimated using a pilot run, resulted in a rapidly mixing Markov chain with a lag-1 autocorrelation of about .8 for each component of η. We obtained samples of size 10^5 from the approximate densities by resampling the available "exact" sample $\mathcal{M}$ and the corresponding values $l(\mathcal{M})$. The resulting independence M-H sampler reduced the typical component-wise lag-1 autocorrela-

tion in the Markov chain from .8 to .2. The same resample of size 10^5 from $\mathcal{M}$ was used both for `INDA` and `DOSKA`. This reduced the MCMC variability of the component-wise TV norms between the "exact" and "approximate" samples for the two approximations. The use of the same resample can be viewed as an application of the *common random numbers* technique (e.g., see Asmussen and Glynn, 2007).

From the plot of component-wise TV norms for the two approximations in Figure 3.4.2, it is seen that either of them is very accurate, with typical TV norm values of about .01. To summarize, only 32+46=78 evaluations of G_E were necessary for very accurate fully Bayesian inference using MCMC when $\dim(\eta) = 34$ and $\dim(\beta) = 3$.

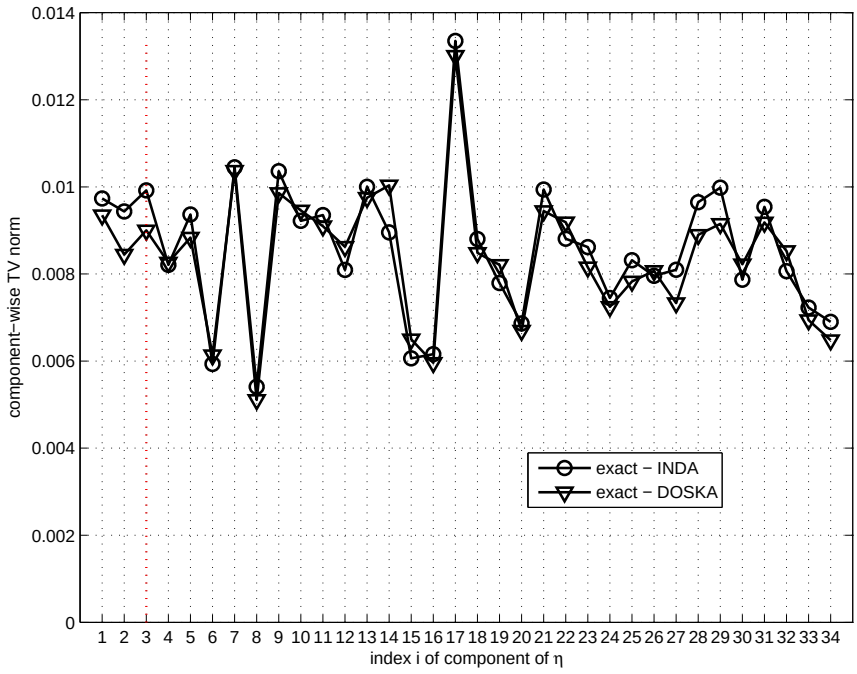

Figure 3.4: Summaries for the linear model: estimated component-wise TV norms between samples from the exact and approximate densities for `DOSKA` (∇) and `INDA` (o). MCMC sample size is 10^5.

3.5 Extensions and Computational Issues

3.5.1 Extensions

Recall that, to ensure optimality, DOSKA uses all of the knots $\mathcal{B}$, at which the expensive computation G_E has been performed. Therefore, fitting of the model is infeasible if the size of $\mathcal{B}$ measures in tens or hundreds of thousands, as is the case in some applications (Taddy et al., 2008, submitted). Also, separability of the basis function assumed by Equation (3.4) may be unappealing for the models with very high degree of dependence between β and ζ, especially when the number of allowed β-knots is small. These concerns can be resolved by using the following generalization by *localization*: Instead of constructing $\mathcal{D}$ of Equation (3.1) from the full set of knots $\mathcal{B}$ as was done in Section 3.3, $\mathcal{D}$ can be chosen *adaptively* depending on the new prediction site $\eta^* = [\beta^*, \zeta^*]$ as

$$\mathcal{D}([\beta^*, \zeta^*]) = \{[\beta^{(j)}, \zeta^{(i,j)}] : 1 \leq i \leq K_j, \beta^{(j)} \in \mathcal{B}(\beta^*)\} \tag{3.12}$$

where $\mathcal{B}(\beta^*) \subset \mathcal{B}$ are knots in some neighborhood of β^*, and $\zeta^{(i,j)}$'s are knots in a neighborhood of ζ^*. Notice that fitting is now done on the η-space, rather than β-space, and a general interpolant of Equation (3.1) with a nonseparable basis function and a nonzero tail q may be used. This local interpolant can arise naturally when one uses basis or covariance functions with bounded supports discussed in Buhmann (2002, chap. 6) and Gneiting (2002a), respectively, since the influence of the knots whose supports do not include $[\beta^*, \zeta^*]$ is expected to be negligible on prediction of $l([\beta^*, \zeta^*])$. (More precisely, the estimated coefficients a and c of Equation (3.1) depend on all knots but the basis functions are non-zero only for neighbors of η^*.) The cost to estimate parameters of the local interpolant using the knots $\mathcal{D}([\beta^*, \zeta^*])$ is low if the size of $\mathcal{D}([\beta^*, \zeta^*])$ is modest.

However, this cost is incurred every time the evaluation of the approximation at a new site is required.

It is expected that if $\mathcal{D}([\beta^*, \zeta^*])$ includes the knots $\mathcal{B}(\beta^*) \oplus \zeta^*$, the local direct interpolant will behave similarly to DOSKA. In the simplest case when $\mathcal{D}([\beta^*, \zeta^*]) \equiv \mathcal{B}(\beta^*) \oplus \zeta^*$, this local approximation amounts to interpolating the function $l(\cdot, \zeta^*)$ at the knots $\mathcal{B}(\beta^*)$ as we remarked near the end of Section 3.3.2. Because of the loose assumptions on the form of this local interpolant, it is unlikely that any kind of optimality can be proved.

3.5.2 Computational Considerations

We conclude this section with a detailed account of the relative computational advantages of direct and indirect interpolants under the assumption of ideal practical implementation of each method. Computationally, the choice only matters if very large MCMC samples from the surrogate density are required, since neither approximation evaluates G_E. As far as the quality of the interpolation is concerned, we doubt that a definitive recommendation can be given as to when to use each type of approximation.

Let $\dim(G_E)$ be the dimension of the "output" of G_E and $\mathbf{cost}(G_C)$ be the flop count of the cheap computation. If K is the number of β-knots and c is the cost to evaluate a basis function once, the cost to evaluate an interpolant of a one-dimensional function is cK. Each evaluation of DOSKA costs $K \cdot \mathbf{cost}(G_C)$ flops to compute $l[\mathcal{B}(\beta^*) \oplus \zeta^*]$ plus K^2 flops to solve the dual kriging system (Section 3.2) with the right-hand side $l[\mathcal{B}(\beta^*) \oplus \zeta^*]$ using a precomputed factorization of the interpolation matrix. The cost of evalution of INDA is $cK \cdot \dim(G_E)$ flops to

obtain $\widetilde{G}_E(\beta^*)$ plus **cost**(G_C) flops to compute $G_C[\widetilde{G}_E(\beta^*), \beta^*, \zeta^*]$. Therefore, the "global" version of DOSKA is less attractive than INDA when K and **cost**(G_C) are high and is more attractive when K and **cost**(G_C) are low but $\dim(G_E)$ is large. As an illustration, consider the inverse problem example from Section 3.1. If $n = \dim(f)$ is large and the covariance matrix for e is unstructured so that **cost**$(G_C) = \mathcal{O}(n^3)$, then DOSKA is roughly K times more "expensive" than INDA. On the other hand, if the covariance matrix for e is diagonal, DOSKA may be preferable (depending on the magnitude of c).

When "local" direct and indirect interpolants are used with the same basis function and tail [recall Equation (3.1)], DOSKA becomes more attractive because of the lower cost to refit the interpolant for each new evaluation. If F is the cost to fit a local interpolant with K knots, the refitting cost for DOSKA is F flops. However, INDA costs $F + K^2 \cdot \dim(G_E)$ flops under some RBF models (if factorization of interpolation matrix can be reused – see end of Section 3.2) and $F \cdot \dim(G_E)$ under most kriging models, since a separate interpolant needs to be fitted for each component of the "output" of G_E. Because $F = \mathcal{O}(K^3)$, the difference in overall fitting and evaluation cost can be quite considerable for the two "local" interpolants.

3.6 Conclusions

In this paper we presented two interpolation approaches, direct and indirect, that allow one to carry out fully Bayesian inference with the help of the approximate density when the exact posterior density π of the parameter vector η is computationally expensive to evaluate. The key to success is identification of

the subvector β of η that is responsible for the dominant computational cost in the evaluation of π. This identification can be done in a host of practical problems such as large-scale inverse problems and high-dimensional models with parametric spatio-temporal dependence.

The primary contribution is derivation of the optimal direct interpolant DOSKA (in Section 3.3) that provably improves over the existing direct GP interpolants of the logarithm l of π such as that of Rasmussen (2003). Since the quality of approximation by our interpolant of l is governed by $\dim(\beta)$ rather than by $\dim(\eta)$, a gain of several orders of magnitude over the naïve approaches that interpolate l on the η-space is expected when $\dim(\eta)$ is high but $\dim(\beta)$ is low.

We supported our analytical findings by simulation experiments of Section 3.4. In Section 3.4.1 we showed that intelligent exploitation of separation of η into the "expensive" and "cheap" subvectors allows one to decrease the number of expensive evaluations G_E by roughly an order of magnitude relative to the already very efficient approach of Rasmussen's. In Section 3.4.2 we provided an example of accurate fully Bayesian inference using the proposed direct and indirect interpolants for the linear model that has $\dim(\eta) = 34$ and $\dim(\beta) = 3$ using fewer than 80 evaluations of G_E.

These very encouraging results support application of our methods to high-dimensional structured statistical problems for which there currently do not exist computationally tractable alternatives.

3.7 Appendix

3.7.1 Proofs

The notation used in these proofs was introduced in Section 3.2.

Let l be a GP indexed by η, with mean 0 and covariance function C_η. Let $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3$ be any finite disjoint sets of values of η. Define $\Sigma_{ij} = C_\eta(\mathcal{E}_i, \mathcal{E}_j)$ for $1 \leq i, j \leq 3$. Since all finite-dimensional distributions of l are Gaussian, it is seen that

- $$E[l(\mathcal{E}_i)|l(\mathcal{E}_j)] = \Sigma_{ij}\Sigma_{jj}^{-1} l(\mathcal{E}_j), \tag{3.13}$$

- $$Var\{E[l(\mathcal{E}_i)|l(\mathcal{E}_j)]\} = \Sigma_{ij}\Sigma_{jj}^{-1}\Sigma_{ji}, \tag{3.14}$$

- $$l(\mathcal{E}_i)|l(\mathcal{E}_j) \sim MVN\left(E[l(\mathcal{E}_i)|l(\mathcal{E}_j)], \Sigma_{ii} - Var\{E[l(\mathcal{E}_i)|l(\mathcal{E}_j)]\}\right) \tag{3.15}$$

Notice that $Var[l(\mathcal{E}_i)|l(\mathcal{E}_j)]$ is the variance of the error from prediction of $l(\mathcal{E}_i)$ using the BLUP $E[l(\mathcal{E}_i)|l(\mathcal{E}_j)]$.

Proposition 1: *Under the above assumptions,*

$$Var[l(\mathcal{E}_1)] \geq Var[l(\mathcal{E}_1)|l(\mathcal{E}_3)] \geq Var[l(\mathcal{E}_1)|l(\mathcal{E}_2), l(\mathcal{E}_3)],$$

where $A \geq B$ iff $A - B$ is non-negative definite.

Proof: The first inequality follows from Equation (3.15). The inequality on the right-hand side is obtained by deriving the conditional distribution of $\{l(\mathcal{E}_1), l(\mathcal{E}_2)\}$ given $l(\mathcal{E}_3)$ and then applying Equation (3.15) again to notice that $Var(l(\mathcal{E}_1)|l(\mathcal{E}_2), l(\mathcal{E}_3)) = Var[l(\mathcal{E}_1)|l(\mathcal{E}_3)] - Var\{E[l(\mathcal{E}_1)|l(\mathcal{E}_2), l(\mathcal{E}_3)]\}$, where $Var\{E[l(\mathcal{E}_1)|l(\mathcal{E}_2), l(\mathcal{E}_3)]\}$ is a non-negative definite matrix.

Proposition 2: *Let $\mathcal{B}$ be a finite set of β-points and $\mathcal{Z}$ be a finite set of ζ-points. Define $\mathcal{Z}^* = \{\zeta^*\} \cup \mathcal{Z}$. If the covariance function for l is separable in a sense of Equation (3.4), then*

$$E[l([\beta^*, \zeta^*])|l(\mathcal{B} \oplus \mathcal{Z}^*)] = E[l([\beta^*, \zeta^*])|l(\mathcal{B} \oplus \zeta^*)]. \tag{3.16}$$

Proof: Without loss of generality, assume that $\sigma^2 = 1$ (because of Equation (3.13)), and that ζ^* is the first element of the list $\mathcal{Z}^*$.

Notice that, under the assumed separability,

- $Var[l(\mathcal{B} \oplus \mathcal{Z}^*)] = C_\eta(\mathcal{B} \oplus \mathcal{Z}^*, \mathcal{B} \oplus \mathcal{Z}^*) = C_\beta(\mathcal{B}, \mathcal{B}) \otimes C_\zeta(\mathcal{Z}^*, \mathcal{Z}^*)$, where $\otimes$ is the Kronecker product,
- $Cov[l(\mathcal{B} \oplus \mathcal{Z}^*), l(\mathcal{B} \oplus \zeta^*)] = C_\eta(\mathcal{B} \oplus \mathcal{Z}^*, \beta^* \oplus \zeta^*) = C_\beta(\mathcal{B}, \beta^*) \otimes C_\zeta(\mathcal{Z}^*, \zeta^*)$.
- $$[Var(l(\mathcal{B} \oplus \mathcal{Z}^*))]^{-1} Cov[l(\mathcal{B} \oplus \mathcal{Z}^*), l(\beta^* \oplus \zeta^*)] =$$
  $$C_\beta(\mathcal{B}, \mathcal{B})^{-1} C_\beta(\mathcal{B}, \beta^*) \otimes C_\zeta(\mathcal{Z}^*, \mathcal{Z}^*)^{-1} C_\zeta(\mathcal{Z}^*, \zeta^*) =$$
  $$C_\beta(\mathcal{B}, \mathcal{B})^{-1} C_\beta(\mathcal{B}, \beta^*) \otimes e_1,$$
  where e_1 is the first standard basis vector for $(|\mathcal{Z}^*| + 1)$-dimensional vector space. [This is true since $C_\zeta(\mathcal{Z}^*, \zeta^*)$ is the first column of $C_\zeta(\mathcal{Z}^*, \mathcal{Z}^*)$, by definition of $\mathcal{Z}^*$.]

Therefore, $E[l(\eta^*)|l(\mathcal{B} \oplus \mathcal{Z}^*)]$

$$= l(\mathcal{B} \otimes \mathcal{Z}^*)^{\mathsf{T}} C_\beta(\mathcal{B}, \mathcal{B})^{-1} C_\beta(\mathcal{B}, \beta^*) \otimes e_1$$

$$= [C_\beta(\mathcal{B}, \beta^*)^{\mathsf{T}} C_\beta(\mathcal{B}, \mathcal{B})^{-1} \otimes e_1^{\mathsf{T}}] \cdot l(\mathcal{B} \otimes \mathcal{Z}^*)$$

$$= vec\{e_1^{\mathsf{T}} \cdot unvec[l(\mathcal{B} \oplus \mathcal{Z}^*)] \cdot C_\beta(\mathcal{B}, \mathcal{B})^{-1} C_\beta(\mathcal{B}, \beta^*)\}$$

$$= l(\mathcal{B} \oplus \zeta^*)^{\mathsf{T}} C_\beta(\mathcal{B}, \mathcal{B})^{-1} C_\beta(\mathcal{B}, \beta^*).$$

In this equation, $vec(\cdot)$ is the vectorization operator defined for a $m \times n$ matrix A as $vec(A) = [A_1^{\mathsf{T}}, \ldots, A_n^{\mathsf{T}}]^{\mathsf{T}}$, where A_i is the ith column of A. The jth column of $unvec[l(\mathcal{B} \oplus \mathcal{Z}^*)]$ is the column vector $l(\mathcal{B} \oplus \zeta^{(j)})$, where $\zeta^{(j)}$ is the jth element of $\mathcal{Z}^*$. We are using the identity $vec(ABC) = (C^{\mathsf{T}} \otimes A) \cdot vec(B)$ for any matrices A, B, C of such dimensions that the product ABC is defined (Harville 1997, chap. 16).

The proof follows by observing that $E[l(\eta^*)|l(\mathcal{B} \oplus \mathcal{Z}^*)]$ does not depend on $\mathcal{Z}$, and is equal to $E[l(\eta^*)|l(\mathcal{B} \oplus \zeta^*)]$, which can be verified by taking $\mathcal{Z}$ to be an empty set.

APPENDIX A

COMPUTATIONAL DETAILS

A.1 Estimation of the Total Variation Norm by Importance Sampling

A Monte Carlo (MC) method to estimate the total variation (*TV*) norm is presented in this section.

For probability measures G_X and G_Y with densities g_X and g_Y the *TV* norm is defined as

$$TV(G_X, G_Y) = \sup_{A \in \mathbb{R}} |G_X(A) - G_Y(A)| = \frac{1}{2} \int_{\mathbb{R}} |g_X(t) - g_Y(t)| dt.$$

Notice that

$$\int_{\mathbb{R}} |g_X(t) - g_Y(t)| dt = \int_{\mathbb{R}} \frac{|g_X(t) - g_Y(t)|}{g(t)} g(t) dt \approx \frac{1}{M} \sum_{i=1}^{M} \frac{|g_X(V_i) - g_Y(V_i)|}{g(V_i)},$$

where $V_1, \ldots, V_M$ are *i.i.d.* from g. If the importance density is $g = \frac{1}{2} g_X + \frac{1}{2} g_Y$, the random variable $|g_X(V_i) - g_Y(V_i)|/g(V_i)$ is supported on the interval $[0, 2]$ and, as a consequence, its variance is bounded by 1 from above. (The variance is much lower if the true *TV* norm is small.) Hence, an MC estimate of the *TV* norm to a desired accuracy can be easily obtained.

If the densities g_X and g_Y are unknown, but the respective univariate samples $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$ are available, estimates of g_X and g_Y can be used as in Algorithm A.1.3 below. (It is assumed that the sample quantiles for the two samples are consistent; independence is not necessary.)

Algorithm A.1.3 sample from Rasmussen's density

Require: $x_1, \ldots, x_n \sim g_X$; $y_1, \ldots, y_m \sim g_Y$; M

1: estimate g_X and g_Y using kernel smoothing by $\widetilde{g}_X$ and $\widetilde{g}_Y$ from $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$
2: **for** $i = 1, \ldots, M$ **do**
3: draw $B_i \sim Bernoulli(1/2)$
4: **if** $B_i = 0$ **then**
5: set $V_i \leftarrow x_j$ with probability $1/n$ for $j = 1, \ldots, n$
6: **else**
7: set $V_i \leftarrow y_j$ with probability $1/m$ for $j = 1, \ldots, m$
8: **end if**
9: set

$$Z_i \leftarrow \frac{|\widetilde{g}_X(V_i) - \widetilde{g}_Y(V_i)|}{\widetilde{g}_X(V_i) + \widetilde{g}_Y(V_i)}$$

10: **end for**
11: **return** sample mean and sample variance of $Z_1, \ldots, Z_M$

We use a pilot run to estimate the variance of Z_i and choose M to make the MC error of the estimated *TV* norm negligible. In our applications, x_i's and y_i's are produced by MCMC runs from the cheap-to-evaluate approximate posterior densities whose length can be chosen by the user to control the accuracy of $\widetilde{g}_X$ and $\widetilde{g}_Y$.

For components of η_i of η, we estimate the *TV* norm between the "most recent" diagnostic MCMC sample and each of the preceding diagnostic MCMC samples. The plot of the estimated *TV* norm values against the number of knots used in the intermediate approximate densities is examined to make the deci-

sion to terminate GRIMA. Monitoring of scalar-valued functions of η other than projections η_i and extensions of Algorithm A.1.3 to multivariate samples are possible but are not pursued in this work.

A.2 Efficient Updating of the Response Surface

Review

In what follows, all quantities in bold font are vectors or matrices. All vectors (except zero) without a transposition operator are column vectors.

Fitting of RBF or kriging interpolant to data is accomplished by solving a nonsingular linear system

$$\boldsymbol{A}_n \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\gamma} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_n & \boldsymbol{F}_n \\ \boldsymbol{F}_n^{\mathsf{T}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\gamma} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Y}_n \\ \mathbf{0} \end{bmatrix}, \tag{A.1}$$

where $\boldsymbol{\Phi}_n$ is an $n \times n$ Gram (e.g., covariance in case of kriging) matrix, $\boldsymbol{F}_n$ is an $n \times q$ matrix of predictors (in case of kriging) or a tail part of interpolant in case of RBF, $\boldsymbol{Y}_n$ is an $n \times 1$ vector of observations/function values, and $\mathbf{0}$ is a matrix of zeros of appropriate size. (This formulation allows $\boldsymbol{F}_n$ to be empty. Also, usually $n >> q$.)

Computational load to solve the system (A.1) is $O((n+q)^3)$. A straightforward approach is via LU factorization of $\boldsymbol{A}_n$. A more elaborate (and stable) approach is as follows:

1. Compute a QR factorization of $\boldsymbol{F}_n$:

$$\boldsymbol{F}_n = \boldsymbol{Q}_n \boldsymbol{R}_n = [\boldsymbol{C}_n, \boldsymbol{N}_n] \boldsymbol{R}_n, \tag{A.2}$$

where $\boldsymbol{R}_n$ is upper-triangular, $\boldsymbol{C}_n$ is an orthogonal basis for column space of $\boldsymbol{F}_n$ and $\boldsymbol{N}_n$ is an orthogonal basis for the null space of $\boldsymbol{F}_n^{\mathsf{T}}$, so that $\boldsymbol{C}_n^{\mathsf{T}}\boldsymbol{N}_n = \boldsymbol{0}$. The work is $O(n^3)$.

2. Notice that the second equation of the system (A.1) implies that $\boldsymbol{F}_n^{\mathsf{T}}\boldsymbol{\alpha} = 0$, so that $\boldsymbol{\alpha} = \boldsymbol{N}_n\boldsymbol{a}$.

3. Solve the system for $\boldsymbol{a}$

$$\boldsymbol{\Psi}_n\boldsymbol{a} := \left(\boldsymbol{N}_n^{\mathsf{T}}\boldsymbol{\Phi}_n\boldsymbol{N}_n\right)\boldsymbol{a} = \boldsymbol{N}_n^{\mathsf{T}}\boldsymbol{Y}_n \tag{A.3}$$

   and use definition of previous item to get $\boldsymbol{\alpha} = \boldsymbol{N}_n\boldsymbol{a}$. Here, $\boldsymbol{\Psi}_n$ is a $(n-q)\times(n-q)$ matrix. Operations of computing $\boldsymbol{\Psi}_n$ and factorizing it each takes work $O((n-q)^3)$.

4. Solve the system

$$\boldsymbol{F}_n\boldsymbol{\gamma} = \boldsymbol{Y}_n - \boldsymbol{\Phi}_n\boldsymbol{\alpha} \quad \text{via QR factorization as} \tag{A.4}$$

$$\boldsymbol{\gamma} = \boldsymbol{R}_n(1:q, 1:q)\backslash[\boldsymbol{C}_n^{\mathsf{T}}(\boldsymbol{Y}_n - \boldsymbol{\Phi}_n\boldsymbol{\alpha})]. \tag{A.5}$$

   Solving this system is cheap once $\boldsymbol{R}_n$ and $\boldsymbol{Q}_n$ are known.

The approach to build design region (for the Town Brook problem) requires sequential selection of each new design point using the fit of interpolant to the previous design points. Therefore, refitting the surface from scratch (a $O(n^3)$ operation) is wasteful. What follows below proposes an approach that uses QR factorization to update solution of the system (A.1) with $(n+1)$ points (from all information about factorizations from previous system with n points) that requires $O(n^2)$ work per update.

An updating procedure

The goal of the section is to show how to reduce the work required by each $O(n^3)$ step above to $O(n^2)$. The approach is stated for the case when a new design point is added to the "top" of previous design points, e.g., $\boldsymbol{Y}_{n+1} = [Y_{n+1}; \boldsymbol{Y}_n]$, $\boldsymbol{F}_{n+1}(2 : n+1, :) = \boldsymbol{F}_n$, etc. It assumes that one needs to solve the system (A.1) for $\boldsymbol{A}_{n+1}$ and has at the disposal

- QR factorization $\boldsymbol{F}_n = \boldsymbol{Q}_n \boldsymbol{R}_n = [\boldsymbol{C}_n, \boldsymbol{N}_n]\boldsymbol{R}_n$;
- $\boldsymbol{\Psi}_n = \boldsymbol{N}_n^{\mathsf{T}} \boldsymbol{\Phi}_n \boldsymbol{N}_n$;
- Cholesky factorization either (i) of $\boldsymbol{\Psi}_n$ or (ii) of $\boldsymbol{P}_n^{\mathsf{T}} \boldsymbol{\Psi}_n \boldsymbol{P}_n$, where $\boldsymbol{P}_n$ is a square permutation matrix.

  Recall that "products" $\boldsymbol{P}_n^{\mathsf{T}} \boldsymbol{\Psi}_n$ and $\boldsymbol{\Psi}_n \boldsymbol{P}_n$ do not require $O(n^3)$ work since effectively one just permutes rows or columns of $\boldsymbol{\Psi}_n$. The cost is $O(n^2)$ since one needs just to read and write the elements of $\boldsymbol{\Psi}_n$, which takes $O((n-q)^2)$ work.

We also assumed that $\boldsymbol{\Phi}_n$ is positive definite so that $\boldsymbol{\Psi}_n$ is also positive definite. Relaxation of this assumption is discussed in the end of this section.

The method has the following steps:

1. Use Givens rotations to compute the QR factorization of $\boldsymbol{F}_{n+1}$ using that of $\boldsymbol{F}_n$. A conventional approach described in Golub & Van Loan's "Matrix Computations" requires $O(nq)$ work.

   An important *observation* is that the resulting basis for the null space of

$\boldsymbol{F}_{n+1}^{\mathsf{T}}$ has the following structure:

$$\boldsymbol{N}_{n+1} = \begin{bmatrix} b_1 & \mathbf{0} \\ \boldsymbol{b}_2 & \boldsymbol{N}_n \end{bmatrix}$$

2. Using the *observation* above, compute $\boldsymbol{\Psi}_{n+1} = \boldsymbol{N}_{n+1}^{\mathsf{T}} \boldsymbol{\Phi}_{n+1} \boldsymbol{N}_{n+1}$ using the above partition of $\boldsymbol{N}_{n+1}$ and of

$$\boldsymbol{\Phi}_{n+1} = \begin{bmatrix} \phi_1 & \boldsymbol{\phi}_2^{\mathsf{T}} \\ \boldsymbol{\phi}_2 & \boldsymbol{\Phi}_n \end{bmatrix}$$

in $O(n^2)$ time to obtain $\boldsymbol{\Psi}_{n+1}$ partitioned as

$$\boldsymbol{\Psi}_{n+1} = \begin{bmatrix} \psi_1 & \boldsymbol{\psi}_2^{\mathsf{T}} \\ \boldsymbol{\psi}_2 & \boldsymbol{\Psi}_n \end{bmatrix}$$

3. Recompute the Cholesky factorization.

   (a) If one has a Cholesky factorization $\boldsymbol{\Psi}_n = \boldsymbol{G}_n^{\mathsf{T}} \boldsymbol{G}_n$, where $\boldsymbol{G}_n$ is upper-triangular, then notice that

$$\begin{aligned} \boldsymbol{\Psi}_{n+1} &= \boldsymbol{G}_{n+1}^{\mathsf{T}} \cdot \boldsymbol{G}_{n+1} \\ \begin{bmatrix} \psi_1 & \boldsymbol{\psi}_2^{\mathsf{T}} \\ \boldsymbol{\psi}_2 & \boldsymbol{G}_n^{\mathsf{T}} \boldsymbol{G}_n \end{bmatrix} &= \begin{bmatrix} u_1 & \mathbf{0} \\ \boldsymbol{u}_2 & \boldsymbol{U}_3^{\mathsf{T}} \end{bmatrix} \cdot \begin{bmatrix} u_1 & \boldsymbol{u}_2^{\mathsf{T}} \\ \mathbf{0} & \boldsymbol{U}_3 \end{bmatrix} \\ &= \begin{bmatrix} u_1^2 & u_1 \boldsymbol{u}_2^{\mathsf{T}} \\ u_1 \boldsymbol{u}_2 & \boldsymbol{U}_3^{\mathsf{T}} \boldsymbol{U}_3 + \boldsymbol{u}_2 \boldsymbol{u}_2^{\mathsf{T}} \end{bmatrix}, \end{aligned} \tag{A.6}$$

   where $\boldsymbol{U}_3$ is upper-triangular. Finding u_1 and $\boldsymbol{u}_2$ is trivial using $O(n)$ work. Computing $\boldsymbol{U}_3$ seems daunting, but it can be done using $O(n^2)$ effort if one notices that $\boldsymbol{U}_3^{\mathsf{T}} \boldsymbol{U}_3 = \boldsymbol{\Psi}_n - \boldsymbol{u}_2 \boldsymbol{u}_2^{\mathsf{T}}$ and reuses the Cholesky factorization of $\boldsymbol{\Psi}_n$ (`cholupdate` in Matlab).

(b) If one has a Cholesky factorization

$$\widetilde{\boldsymbol{\Psi}}_n := \boldsymbol{P}_n^{\mathsf{T}} \boldsymbol{\Psi}_n \boldsymbol{P}_n = \boldsymbol{G}_n^{\mathsf{T}} \boldsymbol{G}_n,$$

where $\boldsymbol{G}_n$ is upper-triangular, it is easy to find a permutation matrix $\boldsymbol{P}_{n+1}$ such that

$$\begin{aligned} \widetilde{\boldsymbol{\Psi}}_{n+1} = \boldsymbol{P}_{n+1}^{\mathsf{T}} \boldsymbol{\Psi}_{n+1} \boldsymbol{P}_{n+1} &= \boldsymbol{G}_{n+1}^{\mathsf{T}} \boldsymbol{G}_{n+1} \\ \begin{bmatrix} \widetilde{\boldsymbol{\Psi}}_n & \boldsymbol{k}_2 \\ \boldsymbol{k}_2^{\mathsf{T}} & k_3 \end{bmatrix} &= \begin{bmatrix} \boldsymbol{U}_1^{\mathsf{T}} & \boldsymbol{0} \\ \boldsymbol{u}_2^{\mathsf{T}} & u_3 \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{U}_1 & \boldsymbol{u}_2 \\ \boldsymbol{0} & u_3 \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{U}_1^{\mathsf{T}} \boldsymbol{U}_1 & \boldsymbol{U}_1^{\mathsf{T}} \boldsymbol{u}_2 \\ \boldsymbol{u}_2^{\mathsf{T}} \boldsymbol{U}_1 & \boldsymbol{u}_2^{\mathsf{T}} \boldsymbol{u}_2 + u_3^2 \end{bmatrix}. \end{aligned} \tag{A.7}$$

Therefore, $\boldsymbol{U}_1 = \boldsymbol{G}_n$ and one can compute $\boldsymbol{u}_2 = \boldsymbol{U}_2^{\mathsf{T}} \backslash \boldsymbol{k}_2$ and u_3 using work $O(n^2)$.

It is as easy to solve the linear system (A.1) using a Cholesky factorization of $\widetilde{\boldsymbol{\Psi}}_n$ *as it is* with the Cholesky factorization of $\boldsymbol{\Psi}_n$.

Remark

In the case of RBF interpolation, $\boldsymbol{\Phi}_n$ is not necessarily positive definite. However, if one chooses the polynomial tail as in Powell (1996), then either $\boldsymbol{\Psi}_n$ or $-\boldsymbol{\Psi}_n$ is positive definite for all n such that the RBF interpolant exists and is unique. If $\boldsymbol{\Psi}_n$ is positive definite, the above updating scheme is applied without modification. Otherwise, one applies our updating scheme upon multiplication of the left-hand and right-hand sides of Equation A.1 by -1.

BIBLIOGRAPHY

[1] Arnold, J. G., Srinivasan, R., Muttiah, R. R., and Williams, J. R. (1998), "Large Area Hydrologic Modeling And Assessment. Part I: Model Development," *Journal of the American Water Resources Association*, 34, 73–89.

[2] Asmussen, S., and Glynn, P. (2007), *Stochastic Simulation: Algorithms and Analysis*, New York: Springer.

[3] Bates, D.M., and Watts, D.G. (1988), *Nonlinear Regression Analysis and Its Applications*. New York: Wiley.

[4] Benaman, J., Shoemaker, C. A., and Haith, D. A. (2005), "Calibration and Validation of Soil and Water Assessment Tool on an Agricultural Watershed in Upstate New York," *ASCE Journal of Hydrologic Engineering*, 10, 363–374.

[5] Beven, K. (2001), discussion of Kennedy and O'Hagan (2001), "Bayesian Calibration of Computer Models," *Journal of the Royal Statistical Society, Series B*, 63, 456.

[6] Bliznyuk, N., Ruppert, D., Shoemaker, C. A., Regis, R., Wild, S., and Mugunthan, P. (2008), "Bayesian Calibration and Uncertainty Analysis for Computationally Expensive Models Using Optimization and Radial Basis Function Approximation", *Journal of Computational & Graphical Statistics*, 17, 1–25.

[7] Box, G. E. P., and Cox, D. R. (1964), "An Analysis of Transformations," *Journal of the Royal Statistical Society, Series B*, 26, 211–246.

[8] Buhmann, M. D. (2003), *Radial Basis Functions*, New York: Cambridge University Press.

[9] Carroll, R. J., and Ruppert, D. (1984), "Power Transformation When Fitting Theoretical Models to Data," *Journal of the American Statistical Association*, 79, 321–328.

[10] ———— (1988), *Transformation and Weighting in Regression*, New York: Chapman & Hall.

[11] Christen, J. A., and Fox, C. (2005), "Markov Chain Monte Carlo Using an Approximation," *Journal of Computational & Graphical Statistics*, 14, 795–810.

[12] Craig, P. S., Goldstein, M., Rougier, J., and Seheult, A. H. (2001), "Bayesian Forecasting for Complex Systems Using Computer Simulators," *Journal of the American Statistical Association*, 96, 717–729.

[13] Cressie, N. (1991), *Statistics for Spatial Data*, New York: Wiley.

[14] Devroye, L. (1986), *Non-Uniform Random Variate Generation*, New York: Springer-Verlag.

[15] Eckhardt, K., Haverkamp, S., Fohrer, N. and Frede, H. G. (2002), "SWAT-G, A Version of SWAT99.2 Modified for Application to Low Mountain Range Catchments", *Physics And Chemistry of the Earth*, 27, 641–644.

[16] Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004), *Bayesian Data Analysis* (2nd ed.), Boca Raton: Chapman & Hall/CRC.

[17] Gneiting, T. (2002a), "Compactly Supported Correlation Functions", *Journal of Multivariate Analysis*, 83, 493–508.

[18] Gneiting, T. (2002b), "Nonseparable, Stationary Covariance Functions for Space-Time Data", *Journal of the American Statistical Association*, 97, 590–600.

[19] Goldstein, M., and Rougier, J. C. (2004), "Probabilistic Formulations for Transferring Inferences from Mathematical Models to Physical Systems," *SIAM Journal on Scientific Computing*, 26, 467–487.

[20] Goldstein, M. and Rougier, J. (2006), "Reified Bayesian Modelling and Inference for Physical Systems". Submitted to the *Journal of Statistical Planning and Inference*, available at http://www.maths.dur.ac.uk/stats/people/jcr/Reify.pdf

[21] Golub, G. H., and Van Loan, C. F. (1996), *Matrix Computations*, 3 ed., Baltimore: John Hopkins University Press.

[22] Grizzetti, B., Bouraoui, F., Granlund, K., Rekolainen, S., and Bidoglio, G., (2003), "Modelling Diffuse Emission and Retention of Nutrients in the Vantaanjoki Watershed (Finland) Using the SWAT Model," *Ecological Modelling*, 169, 25–38.

[23] Hamilton, J. D. (1994), *Time Series Analysis*, Princeton: Princeton University Press.

[24] Harville, D. A. (1997), *Matrix Algebra from a Statisticians Perspective*, New York: Springer.

[25] Higdon, D., Lee H., and Holloman, C. (2003), "Markov Chain Monte Carlo-Based Approaches for Inference in Computationally Intensive Inverse Problems," in *Bayesian Statistics 7*, eds. J. M. Bernardo, J. O. Berger, A. P. Berger and A. F. M. Smith, pp. 181–197.

[26] Johnson, M., Moore,L., and Ylvisaker, D. (1990), "Minimax and Maximin Distance Designs," *Journal of Statistical Planning and Inference*, 26, 131–148.

[27] Kennedy, M. C., and O'Hagan, A. (2000), "Predicting the Output From a Complex Computer Code When Fast Approximations are Available," *Biometrika*, 87, 1–13.

[28] ———— (2001), "Bayesian Calibration of Computer Models," *Journal of the Royal Statistical Society, Series B*, 63, 425–464.

[29] Morris, M., Mitchell, T., and Ylvisaker, D. (1993), "Bayesian Design and Analysis of Computer Experiments: Use of Derivatives in Surface Prediction," *Technometrics*, 35, 243–255.

[30] Mugunthan, P. and Shoemaker, C. A. (2006), "Assessing the Impacts of Parameter Uncertainty for Computationally Expensive Groundwater Models," *Water Resources Research*, 42, W10428, doi: 10.1029/2005WR004640.

[31] Mugunthan, P., Shoemaker, C. A., and Regis, R. G. (2005), "Comparison of Function Approximation, Heuristic and Derivative-Based Methods for Automatic Calibration of Computationally Expensive Groundwater Bioremediation Models," *Water Resources Research*, 41, W11427, doi:10.1029/2005WR004134.

[32] O'Hagan, A., Kennedy, M. C., and Oakley, J. E. (1998), "Uncertainty Analysis and Other Inference Tools for Complex Codes," in *Bayesian Statistics 6*, eds. J. M. Bernardo, J. O. Berger, A. P. Berger and A. F. M. Smith, pp. 503–524.

[33] Powell, M. J. D. (1992), "The Theory of Radial Basis Function Approximation in 1990," in *Advances in Numerical Analysis, Volume 2: Wavelets, Subdivision Algorithms and Radial Basis Functions*, ed. W. Light, New York: Oxford University Press, pp. 105–210.

[34] ———— (1996), "A Review of Algorithms for Thin Plate Spline Interpolation in Two Dimensions," in *Advanced Topics in Multivariate Approximation*, eds. F. Fontanella, K. Jetter and P. J. Laurent, River Edge, NJ: World Scientific Publishing, pp. 303–322.

[35] ———— (2002), "UOBYQA: Unconstrained Optimization by Quadratic Approximation," *Mathematical Programming*, 92, 555–582.

[36] Rasmussen, C.E. (2003), "Gaussian Processes to Speed up Hybrid Monte Carlo for Expensive Bayesian Integrals," in *Bayesian Statistics 7*, eds. J. M. Bernardo, J. O. Berger, A. P. Berger and A. F. M. Smith, pp. 651–659.

[37] Regis, R. G., and Shoemaker, C. A. (2007a), "A Stochastic Radial Basis Function Method for the Global Optimization of Expensive Functions," *INFORMS Journal of Computing*.

[38] Regis, R. G., and Shoemaker, C. A. (2007b), "Parallel Radial Basis Function Methods for the Global Optimization of Computationally Expensive Functions," *European Journal of Operations Research*

[39] Robert, C. P., and Casella, G. (1999), *Monte Carlo Statistical Methods*, New York: Springer.

[40] Santner, T.J., Williams, B.J. and Notz, W. (2003), *The Design and Analysis of Computer Experiments*. New York: Springer–Verlag.

[41] Scott, D. W. (1992), *Multivariate Density Estimation: Theory, Practice, and Visualization*, New York: Wiley.

[42] Shao, J. (1999), *Mathematical Statistics*, New York: Springer.

[43] Shoemaker, C., Regis, R., and Fleming, R. (2007), "Watershed Calibration Using Multistart Local Optimization and Evolutionary Optimization with Radial Basis Function Approximation," *Journal Of Hydrologic Science*.

[44] Taddy, M. A., Sanso, B., and Lee, H. K. H. (2008, submitted), "Fast Bayesian inference for Computer Simulation Inverse Problems", submitted to *Inverse Problems*.

[45] Tierney, L. (1994), "Markov Chains for Exploring Posterior Distributions," *The Annals of Statistics*, 22, 1701–1786.

[46] Tierney, L., and Kadane, J. B. (1986), "Accurate Approximations for Posterior Moments and Marginal Densities," *Journal of the American Statistical Association*, 81, 82–86.

[47] Tjelmeland, H., and Hegstad, B. K. (2001), "Mode Jumping Proposals in MCMC," *Scandinavian Journal of Statistics*, 28, 205–223.

[48] Tolson, B., and Shoemaker, C. A. (2004), Watershed Modeling of the Cannonsville Basin using SWAT2000: Model Development, Calibration and Validation for the Prediction of Flow, Sediment and Phosphorus Transport to the Cannonsville Reservoir, Version 1, Technical Report, School of Civil and Environmental Engineering, Cornell University. Available at `http://ecommons.library.cornell.edu/handle/1813/2710`

[49] ———— (2007a), "The Dynamically Dimensioned Search Algorithm for Computationally Efficient Automatic Calibration of Environmental Simulation Models," *Water Resources Research*, 43, W01413, doi: 10.1029/2005WR004723.

[50] ———— (2007b), Cannonsville Reservoir Watershed SWAT2000 Model Development, Calibration And Validation, *Journal of Hydrology*, 337,68–89, doi:10.1016/j.jhydrol.2007.01.017.

[51] Trosset, M. W. (1999), "Approximate Maximin Distance Designs," in *American Statistical Association Proceedings of the Physical and Engineering Sciences Section*, pp. 223–227.

[52] Vanden Berghen, F., and Bersini, H. (2005), "CONDOR, a New Parallel, Constrained Extension of Powell's UOBYQA Algorithm: Experimental Results and Comparison With the DFO Algorithm," *Journal of Computational and Applied Mathematics*, 181, 157–175.

[53] Wu, C. F. J. (1981), "Asymptotic Theory of Nonlinear Least Squares Estimation", *Annals of Statistics*, 9, 501–513.