Modularization in Bayesian Analysis, with Emphasis on Analysis of Computer Models*

Fei Liu, M.J. Bayarri, and James Berger
University of Missouri, Universitat de València, Duke University

September 28, 2008

Abstract

Bayesian analysis incorporates different sources of information into a single analysis through Bayes theorem. When one or more of the sources of information are suspect (e.g., if the model assumed for the information is viewed as quite possibly being significantly flawed), there can be a concern that Bayes theorem allows this suspect information to overly influence the other sources of information. We consider a variety of situations in which this arises, and give methodological suggestions for dealing with the problem.

After consideration of some pedagogical examples of the phenomenon, we focus on the interface of statistics and the development of complex computer models of processes. Three testbed computer models are considered, in which this type of issue arises.

Keywords and phrases: Complex computer models; Confounding; Emulators; Identifiability; MCMC mixing; Partial likelihood; Random effects.

*This research was supported in part by the National Science Foundation (GRANTS AST-0507481, DMS-0103265, DMS-0757549, and DMS-0757527), by the Spanish Ministry of Education and Science (Grant MTM2007-61554) and by a Research Board grant at the University of Missouri. The research began under the auspices and support of the Statistical and Applied Mathematical Sciences Institute (SAMSI) 2006-7 research program on the Development, Assessment and Utilization of Complex Computer Models. Our thanks to David Higdon, Marc Kennedy, Rui Paulo, and Jerry Sacks for many useful discussions concerning these issues.

1 Introduction

1.1 Background

Two of the strengths of Bayesian analysis are that it allows simultaneous incorporation of all relevant data into the analysis, and simultaneously deals with all uncertainties in the model. These strengths can be weaknesses, however, when either some of the data or some parts of the model are suspect.

We consider the situation where the overall model consists of a number of distinct components, that we will call *modules*. We will focus on discussion of why it may be beneficial to keep modules at least partly separate in the Bayesian computation, a technique that we will call *modularization*.

A version of modularization that is well known in statistics is *partial likelihood*. The partial likelihood approach essentially results in ignoring one or more factors of the likelihood function in the analysis. We briefly review some of the motivations for this in Section 2.1.

Within Bayesian analysis, there is increasing use of modifications to posterior distributions that do not strictly flow from Bayes theorem. One common type of manipulation is to represent the posterior distribution in a formally correct way involving a sequence of marginal and conditional distributions, but then to simplify one or more of the terms in the expression. Here are some of the reasons this has been done:

1. A 'good' module might be kept separate from a 'suspect' module to prevent it from being unreasonably influenced (which we will term 'contamination').

2. Keeping modules separate might be viewed as important for scientific understanding and future scientific development of the modules.

3. There may be a lack of identifiability in the problem, with unknown parameters in one module being confounded with unknown parameters in another.

4. A poorly mixing MCMC analysis can start mixing quickly if certain (probabilistically invalid) modifications to the posterior are made.

5. One might be in a problem in which computational complexity prevents analysis, yet (incoherent) manipulation of the posterior will yield an answer.

We are not questioning Bayesian analysis here; if one is comfortable with all the modeling and prior assessments that go into an analysis, and if it is possible to carry out

the ensuing Bayesian computation, then certainly we would not argue for altering the posterior distribution. Uncertainties in modeling and practical computational realities, however, may suggest certain types of modifications of the posterior, and our goal is to try to understand which modifications are reasonable. Note that one does not have the coherency of Bayesian analysis as an automatic support for the modified analysis, so supplementary justifications are often needed.

Ideally, uncertainty in modeling would be best addressed formally, through diagnostic checking (see, e.g., Evans and Moshonov (2006)). One can, indeed, imagine comparing the original model and the 'modularized model' through such methods, formally ascertaining when use of the modularized model is better.

Modularization arose, however, in settings that are too complex for such formal analysis (at least with the current state-of-the-art). Indeed, in typical applications of modularization, one identifies that there is a problem (often, a non-mixing MCMC for some parameters) and the problem itself suggests an easy modularization 'fix', yet it can be extremely difficult to actually identify the modeling flaw directly. A real example in which this occurred – and which provided the main motivation for this paper – will be discussed in Section 3.3.

Another name given to this idea of preventing the information from 'uncertain' modules to 'contaminate' good modules is *cutting feedback* (Spiegelhalter *et al.* (2003)), implemented in the Bayesian software WinBugs. Often this is also used to facilitate the MCMC computation; by not allowing full information flow across the steps of an MCMC, much more rapid mixing can be achieved. (We delay, until the end, a discussion of the merits of such adjustments when done only for computational reasons.)

Certain versions of this idea have also been implemented and formally studied in particular application-contexts. For instance, *inconsistent dependency networks* (Heckerman *et al.* (2000)) is an approach to dealing with graphical models with very complicated dependency structures: in the posterior, the dependency structure is vastly simplified to allow computation, but in a way that is inconsistent with a true joint distribution. Another alteration made for computational simplicity is *inconsistent Gibbs* (Gelman and Raghunathan (2001), Raghunathan *et al.* (2001)), where the likelihood is the product of separately assessed full conditional distributions, which might or might not define a joint posterior. Still another method seeking an easily computable approximation to the posterior is the *weighted likelihood bootstrap* (Newton and Raftery (1994)) where factors in the likelihood function are raised to weights that have a joint distribution.

1.2 Application to Computer Models

For pedagogical reasons, we will first illustrate some of these issues in Section 2, on rather simple artificial examples. We then turn to the practical domain in which we encountered these issues, namely the interface of statistics and complex computer modeling.

Complex computer models are increasingly being used to simulate natural or engineering processes. Statistical analysis involving such computer models (following, e.g., Sacks *et al.* (1989), Kennedy and O'Hagan (2001), Craig *et al.* (2001), Santner *et al.* (2003), Higdon *et al.* (2004), and Qian and Wu (2008)) typically involves three modules:

Module 1. The computer model itself, which may have unknown parameters, and is typically very expensive to run, often necessitating use of a response surface approximation called an *emulator*.

Module 2. The field data, which are measurements of the real process with some modeled error structure.

Module 3. The bias or discrepancy between the computer model and the real process, which is typically an unknown function, often also represented by a response surface model.

In Section 3, we will discuss the various uses that have been made of modularization in analysis of computer models, and argue for regular use of certain modularizations.

2 Pedagogical Examples

This section considers a series of relatively simple examples that are designed to illustrate different types of modularization, and to show how modularization can fix 'flaws' in the modeling. Because the examples are relatively simple, a number of more formal (and better) alternatives exist for analyzing and fixing the revealed problems. We are simply using the examples pedagogically, to show how modularization works; later examples in the paper will demonstrate the use of modularization in problems that are too complex for the more formal methods.

2.1 Partial Likelihood

Although statistical analysis should, in principle, depend on the entire likelihood function, there are a number of reasons that it is relatively common to ignore factors of the likelihood

in the analysis. The most common situation in which this arises is when the likelihood for data $\boldsymbol{Y}$ depends on the unknown parameters of interest $\boldsymbol{\theta}$ and unknown nuisance parameters $\boldsymbol{\eta}$ and is of the form

$$f(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\eta}) = g(\boldsymbol{s}_1 \mid \boldsymbol{\theta}, \boldsymbol{s}_2) h(\boldsymbol{s}_2 \mid \boldsymbol{\theta}, \boldsymbol{\eta}) \,,$$

where $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ are two statistics. It is then tempting to ignore the second term involving the nuisance parameter, and base the analysis only on the first term. We briefly discuss two reasons in this section for employing partial likelihood.

2.1.1 A Suspect Module

Suppose we have two sources of data concerning an unknown parameter μ. The first is a reliable sample $\boldsymbol{y}$ (of size m) arising, say, from a $N(\mu, 1)$ distribution; the second is a possibly biased sample $\boldsymbol{y}^b$ (of size n) arising from a $N(\mu+b, 1)$ distribution, with unknown bias b. The full likelihood is thus

$$f(\boldsymbol{y}, \boldsymbol{y}^b \mid \mu, b) \propto \exp\{-\frac{m}{2}\,(\bar{y} - \mu)^2\}\ \exp\{-\frac{n}{2}\,[\bar{y}^b - (\mu + b)]^2\} \,,$$

where $\bar{y}$ and $\bar{y}^b$ are the sample means. Intuitively, it is reasonable to ignore the biased data, and just base the analysis on the first component of the likelihood, unless one feels quite certain that b is very small.

To see the possible danger in using the full likelihood, suppose μ is assigned a constant objective prior, while b is subjectively assessed to have a $N(0, \sigma_b^2)$ prior distribution. The posterior mean for μ is then

$$E(\mu \mid \boldsymbol{y}, \boldsymbol{y}^b) = \frac{m(\sigma_b^2 + n^{-1})}{1 + m(\sigma_b^2 + n^{-1})}\ \bar{y} + \frac{1}{1 + m(\sigma_b^2 + n^{-1})}\ \bar{y}^b \,, \tag{2.1}$$

and the posterior variance is $(m + 1/(\sigma_b^2 + n^{-1}))^{-1}$.

Note first that, even if the sample size n for the biased data is huge, the posterior variance does not drop below $(m + 1/\sigma_b^2)^{-1}$, which will typically not be much of an improvement over the posterior variance m^{-1} of the partial likelihood estimate $\bar{y}$ when m is moderate or large, unless σ_b^2 is quite small. The main issue, however, is that it can be dangerous to use (2.1) if the prior assessment was an inaccurate reflection of real beliefs. For instance, if one evaluates robustness by overall mean squared error of the estimate (expected squared error over the data and the prior distribution of b), it can be shown that

(2.1) is worse than the partial likelihood estimate $\bar{y}$ if the true prior variance is larger than $2\sigma_b^2 + m^{-1} + n^{-1}$. Note that folklore says that prior variances are typically underestimated by a factor of 3, in which case the use of the full likelihood here would be detrimental unless σ_b^2 were very small. If the error in the assessment were in the distributional form of the prior, the result could be far worse. For instance, if the true prior for b were actually Cauchy, the overall mean squared error of the estimate in (2.1) would be infinite. In conclusion, there is typically little to gain and much to lose in attempting to incorporate the biased data into the analysis, unless the bias is known to be very small.

2.1.2 Ease of Analysis

Another common reason for ignoring a component of the likelihood function is that the analysis is much easier if one does so. Usually, of course, arguments are also made that the ignored likelihood component ($h(\boldsymbol{s}_2 \mid \boldsymbol{\theta}, \boldsymbol{\eta})$ in (2.1)) is not overly informative, so that the analysis remains reasonable. For examples involving likelihood methodology, see Cox (1972), Cox (1975), Møller and Sorensen (1994), and Diggle (2006).

Note that, in Bayesian analysis, it is less common to utilize partial likelihood solely for computational reasons, because MCMC computational techniques can generally deal with the full likelihood rather easily. In the situation of (2.1) for instance, one might either be able to incorporate the second factor in the likelihood into the analysis by Gibbs sampling, or by a Metropolis step (utilizing the first factor of the likelihood to construct a proposal distribution).

2.2 Modularization as a Modeling Surrogate

2.2.1 A Random Effects Example

This section considers a simple example of the issues involving modularization. Interesting and somewhat disturbing issues are also raised concerning standard random effects analysis or hierarchical modeling.

Consider a simple random effects model in which we have n independent observations on each of N groups:

$$\begin{aligned} y_{ij} \mid b_i &= b_i + \epsilon_{ij}\,, \qquad j = 1, \ldots n; \quad i = 1 \ldots N\,, \\ \epsilon_{ij} \mid \sigma_i^2 &\sim N(0, \sigma_i^2), \\ b_i \mid \tau^2 &\sim N(0, \tau^2), \end{aligned} \tag{2.2}$$

where the σ_i^2's and τ^2 are unknown. This model has two modules: Module 1 is the distribution of the observables y_{ij}, while Module 2 is the distribution of the random effects b_i.

With objective priors $\pi(\sigma_i^2) \propto (\sigma_i^2)^{-1}$ and $\pi(\tau^2 \mid \boldsymbol{\sigma}^2) \propto (\tau^2 + \bar{\sigma}^2/n)^{-1}$, where $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_N^2)$ and $\bar{\sigma}^2 = \sum \sigma_i^2/N$, the marginal posterior for τ^2 and $\boldsymbol{\sigma}^2$ is (see Appendix A for details)

$$\pi(\tau^2, \boldsymbol{\sigma}^2 \mid \bar{\boldsymbol{y}}, \boldsymbol{s}^2) \propto \frac{1}{\tau^2 + \bar{\sigma}^2/n} \prod_{i=1}^{N} (\sigma_i^2)^{-\frac{n+1}{2}} \exp\{-\frac{n s_i^2}{2\,\sigma_i^2}\} \; \frac{1}{(\tau^2 + \sigma_i^2/n)^{1/2}} \; \exp\left(-\frac{\bar{y}_i^2}{2(\tau^2 + \sigma_i^2/n)}\right); \quad (2.3)$$

here $\bar{\boldsymbol{y}}$ is the vector of N sample means and $\boldsymbol{s}^2$ is the N-vector with components $s_i^2 = \sum_{j=1}^{n} (y_{ij} - \bar{y}_i)^2$. The conditional posterior distribution for $\boldsymbol{b} = (b_1, \ldots, b_N)$, given τ^2 and $\boldsymbol{\sigma}^2$, is

$$\pi(\boldsymbol{b} \mid \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2, \bar{\boldsymbol{y}}, \boldsymbol{s}_i^2) \;\; \sim \;\; \prod_{i=1}^{N} \mathrm{N}\left(\frac{\tau^2}{\tau^2 + \sigma_i^2/n} \bar{y}_i \,, \quad \frac{\tau^2 \sigma_i^2/n}{\tau^2 + \sigma_i^2/n}\right). \quad (2.4)$$

Consider the following scenario:

- The number of groups, N, is large, while the number of replications, n, is relatively small but large enough for reasonably accurate estimation of the σ_i^2; also suppose that the σ_i^2 are near 1.
- We are very confident in the normality of the errors ϵ_{ij}, but are *not* confident about the normality assumption for the random effects, b_i.

Suppose now that one of the random effects, say b_k, happens to be large (e.g., 10), while the others are small, say near one. (Of course this is incompatible with the normality assumption for the b_i but, for the moment, we are imagining that this cannot be directly ascertained.) The common belief in hierarchical Bayesian analysis is that this situation would result in a posterior under which τ^2, the variance of the random effects, is large, and so there would be little shrinkage of the usual estimates $\bar{y}_i$ of the b_i.

Something quite different happens in the posterior in (2.3), however, under the above scenario. The terms to focus on from this posterior are

$$\exp\left(-\frac{\bar{y}_k^2}{2(\tau^2 + \sigma_k^2/n)}\right) (\sigma_k^2)^{-\frac{n+1}{2}} \frac{1}{(\tau^2 + \bar{\sigma}^2/n)} \prod_{i=1}^{N} \frac{1}{(\tau^2 + \sigma_i^2/n)^{1/2}}. \quad (2.5)$$

Note that $\bar{y}_k^2$ will be large because b_k is large, so the first term in (2.5) (and hence the posterior density) can be kept away from zero only by choosing either τ^2 or σ_k^2/n to be correspondingly large. The second term in (2.5) tries to prevent σ_k^2 from being large, while the last terms try to prevent τ^2 from being large. The last terms win out, since the power for τ^2 is effectively $-(N+2)/2$ (and N is large) while the power for σ_k^2 is $-(n+1)/2$ (and n is relatively small). Hence the Bayesian analysis will let σ_k^2/n get large, while keeping τ^2 small.

The consequence of this is revealed from the posterior for the bias in (2.4): having σ_k^2/n large and τ^2 small will result in the posterior for b_k being approximately $\mathrm{N}(0\,,\tau^2)$, a dramatic – and very incorrect – shrinkage of $\bar{y}_k$ towards 0.

Note also the 'contamination' from one module to another that is causing the problem. Module 1 contains the replication data, which contains almost all of the real information about σ_k^2 (and which would say that σ_k^2 is not large). However, the suspect Module 2 is allowed to influence Module 1 through Bayes theorem, and its influence happens to be dramatic (and harmful).

2.2.2 The Modular Approach in the Example

The modular approach in this example is simply to insist that the posterior for $\boldsymbol{\sigma}^2$ in Module 1 be based only on the replicate observations. To formally see what is being proposed, write the joint posterior of τ^2 and $\boldsymbol{\sigma}^2$ as

$$\pi(\tau^2, \boldsymbol{\sigma}^2 \mid \bar{\boldsymbol{y}}, \boldsymbol{s}^2) \;=\; \pi(\tau^2 \mid \boldsymbol{\sigma}^2, \bar{\boldsymbol{y}}, \boldsymbol{s}^2)\pi(\boldsymbol{\sigma}^2 \mid \bar{\boldsymbol{y}}, \boldsymbol{s}^2)\,, \tag{2.6}$$

where

$$\begin{aligned} &\pi(\boldsymbol{\sigma}^2 \mid \bar{\boldsymbol{y}}, \boldsymbol{s}^2) \\ \propto\; &\left[\prod_{i=1}^{N} \sigma_i^{-n-1} \exp\{-\frac{n s_i^2}{2\sigma_i^2}\}\right] \int \frac{1}{\tau^2 + \bar{\sigma}^2/n} \prod_{i=1}^{N} \frac{1}{(\tau^2 + \sigma_i^2/n)^{1/2}} \;\exp\left(-\frac{\bar{y}_i^2}{2(\tau^2 + \sigma_i^2/n)}\right) d\tau^2\,. \end{aligned}$$

The modular posterior distribution for $\boldsymbol{\sigma}^2$ is that arising from the first expression (in square brackets) above, simply ignoring the integral. In other words, (2.6) is replaced with

$$\pi(\tau^2, \boldsymbol{\sigma}^2 \mid \bar{\boldsymbol{y}}, \boldsymbol{s}^2) \approx \pi(\tau^2 \mid \boldsymbol{\sigma}^2, \bar{\boldsymbol{y}}, \boldsymbol{s}^2)\pi(\boldsymbol{\sigma}^2 \mid \boldsymbol{s}^2)\,. \tag{2.7}$$

Note that the conditional posterior distributions for τ^2 and $\boldsymbol{b}$ are unchanged in terms of their mathematical expressions, but will change very considerably in terms of their

location: with σ_k^2 no longer being able to accommodate the outlier, τ^2 will become large, and the posterior for b_k will remain near $\bar{y}_k$.

To implement this modularization in the MCMC, first sample σ_i^2 from the $Gamma^{-1}\big((n-1)/2, n\, s_i^2/2\big)$ posteriors given only the s_i^2, and then draw τ^2 and $\boldsymbol{b}$ from their true posterior conditional distributions in (2.3) (with $\boldsymbol{\sigma}^2$ viewed as given) and (2.4).

2.2.3 The Modeling Flaw in the Example

The problem with the random effects model here is, of course, that the normality assumption for the b_i is bad: the actual random effects were all moderately valued, except for one large outlier. This is not a scenario in which using a normal distribution is expected to work.

The nature of the failure, however, is disturbing, and carries ramifications for random effects analyses in general. Folklore suggests that the random effects or hierarchical modeling assumption is rather risk free in that, if the effects vary widely, then τ^2 will be large and the analysis essentially collapses to a fixed-effects analysis; no harm done (although the potential gain from reasonable shrinkage is not realized). Here, however, we see that the analysis instead collapses to the disastrous conclusion that b_k is essentially zero when, in fact, it is large. Furthermore, simple diagnostics, such as seeing if the posterior means of the b_i are compatible with a normal distribution, would not have revealed a problem at all. While we are focusing on this example only from the perspective of modularization, the analysis reveals a considerable potential danger in routine application of standard random effects analysis when the number of random effects is larger than the replication size.

The correct Bayesian solution to the situation is, of course, to use a better model for the b_i, such as a t-distribution with small degrees of freedom (which can accommodate outliers well). This will be illustrated in an application in Section 3.3.

The appeal of modularization is that it can be much easier to identify a useful restriction of the Bayesian analysis, than to develop a better model for suspect modules. In the random effects problem, for instance, deciding that the σ_i^2 will be determined only from the replications is much simpler than attempting to infer a good model for the random effects (which are not directly observed). In more complicated scenarios, this difference in difficulty can be even more pronounced.

2.2.4 Computational Considerations in the Example

From a computational perspective, modularization is also much simpler. Indeed, it typically leads to an easier MCMC than even the original analysis; thus it is much easier to sample from $\pi(\boldsymbol{\sigma}^2 \mid \boldsymbol{s}^2)$ in (2.7) than from $\pi(\boldsymbol{\sigma}^2 \mid \bar{\boldsymbol{y}}, \boldsymbol{s}^2)$ in (2.6). See, also, Section 3.3, where other computational issues arise in the full Bayesian analysis.

3 Modularization in Analysis of Computer Models

3.1 Background on Computer Models

Our motivation for consideration of modularization arose in the analysis of complex computer models which, as discussed in the introduction, typically have three distinct modules: field observations, the computer model itself, and computer model bias or discrepancy. In such analyses, we have routinely encountered situations in which modularization is needed or, at least, useful.

Computer experiments typically have data of two types: runs of the computer model itself (which we here assume to be deterministic) at various inputs, and runs of physical experiments (field data). A typical computer model will have two types of inputs, denoted by $\boldsymbol{x}$ and $\boldsymbol{u}$. Inputs $\boldsymbol{x}$ occur in both the computer model and the field runs, whereas $\boldsymbol{u}$ are calibration/tuning parameters that are only needed to run the computer model. We represent the (unknown) true value of the calibration parameter by $\boldsymbol{u}^*$; note that this is often just conceptual, in that it will often be impossible to determine $\boldsymbol{u}^*$ from data, for reasons discussed below.

Given input vectors $\boldsymbol{x}$ and $\boldsymbol{u}$, we represent the corresponding computer model output by $y^M(\boldsymbol{x}, \boldsymbol{u})$, and the j^{th} field replicated run by $y_j^F(\boldsymbol{x})$, respectively. The goal is to combine these two sets of observations, to facilitate better understanding of the real processes, to perform calibration or tuning for unknown parameters of the computer model, and to evaluate the computer model in terms of its accuracy in representing the real process.

Following Kennedy and O'Hagan (2001), the third key module in computer modeling arises from representing the *real* process as the sum of the computer model $y^M(\boldsymbol{x}, \boldsymbol{u}^*)$ at $\boldsymbol{u}^*$ and the model discrepancy $b(\boldsymbol{x})$,

$$y^R(\boldsymbol{x}) = y^M(\boldsymbol{x}, \boldsymbol{u}^*) + b(\boldsymbol{x})\,;$$

modeling of $b(\boldsymbol{x})$ then becomes the third module of the problem. Reality is connected

with the field observations by viewing the field runs as realizations from

$$y_j^F(\boldsymbol{x}) = y^R(\boldsymbol{x}) + \epsilon_{xj} = y^M(\boldsymbol{x}, \boldsymbol{u}^*) + b(\boldsymbol{x}) + \epsilon_{xj}\,, \tag{3.8}$$

where ϵ_{xj} are measurement errors, $\epsilon_{xj} \sim \mathrm{N}(0, 1/\lambda_e)$.

Many computer models are time-consuming to run. It is thus often useful to construct a fast surrogate – usually called an *emulator* – to the expensive-to-run computer code. Gaussian process response-surface methodology (GASP) has been consistently effective for constructing emulators, since its introduction for this purpose in Sacks *et al.* (1989), Currin *et al.* (1991), Welch *et al.* (1992), and Morris *et al.* (1993). The idea is to assign a Gaussian stochastic process as the prior distribution of the computer model,

$$y^M(\cdot) \sim \mathrm{GP}\left(\mu_M, \frac{1}{\lambda_M}\mathrm{corr}(\cdot,\cdot)\right)\,, \tag{3.9}$$

where μ_M, λ_M and $\mathrm{corr}(\cdot,\cdot)$ are the mean, precision, and correlation function that characterize the Gaussian process. In the computer modeling literature, the most commonly used correlation function is the exponential correlation function, which takes the form

$$\mathrm{corr}\left((\boldsymbol{x},\boldsymbol{u}),(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{u}})\right) = \exp(-\sum_k \beta_k \mid \boldsymbol{x}_k - \tilde{\boldsymbol{x}}_k \mid^{\alpha_k}) \times \exp(-\sum_l \beta_l^* \mid \boldsymbol{u}_l - \tilde{\boldsymbol{u}}_l \mid^{\alpha_l^{(*)}})\,. \tag{3.10}$$

At any (untried) input $(\boldsymbol{x}, \boldsymbol{u})$, the emulator predicts the corresponding output using the posterior predictive distribution of $y^M(\cdot)$, which will be denoted $\pi(y^M(\boldsymbol{x},\boldsymbol{u}) \mid \mathrm{Data})$. We similarly use a GASP prior to represent our uncertainty about the discrepancy function $b(\cdot)$. Completing the modeling with priors on the unknown GASP mean, precision and correlation parameters, along with any needed priors for the inputs, allows Bayesian analysis both to assess the accuracy of the computer model and to provide predictions of the real process utilizing all the information.

Extensions of this validation framework include that in Bayarri *et al.* (2007a) and Liu *et al.* (2008), which use a hierarchical structure to deal with smooth functional data and allow for uncertainty in the inputs, and that in Bayarri *et al.* (2007a), which uses wavelets to incorporate irregular functional outputs. See also Higdon *et al.* (2007) for approaches using other basis functions. For nonstationary processes, dynamic emulators have been considered by Conti *et al.* (2005), Liu (2007), and Reichert *et al.* (2008). Another related approach is the treed Gaussian process method of Gramacy and Lee (2008).

3.2 Modularizing the Emulator of the Computer Model

Recall that we denote the (possibly unknown and deterministic) computer model by $y^M(\cdot)$, and the runs of the computer model and the field experiments by $\boldsymbol{y}^M$ and $\boldsymbol{y}^F$, respectively. As is clear from (3.8), analysis of $y^M(\cdot)$ will involve both $\boldsymbol{y}^M$ and $\boldsymbol{y}^F$ in a full Bayesian analysis. The corresponding posterior predictive distribution, $\pi(y^M(\cdot) \mid \boldsymbol{y}^M, \boldsymbol{y}^F)$, is the emulator under the full Bayesian analysis.

When this is presented to computer modelers, it is viewed as strange that data from the field runs is allowed to affect the emulator for the computer model; much more natural, from their perspective, is to build an emulator of the computer model using only runs of the computer model itself, i.e., using only $\boldsymbol{y}^M$ together with the model (3.9). If only $\boldsymbol{y}^M$ is used, we will call this *modularizing the emulator*, and denote the resulting posterior predictive by $\pi(y^M(\cdot) \mid \boldsymbol{y}^M)$. Note that we are not talking about calibration (i.e., estimation of $\boldsymbol{u}$) here; calibration certainly requires use of both the computer run and field run data. We are, rather, talking only about emulating the computer model response to the variables $\boldsymbol{x}$ and $\boldsymbol{u}$.

Part of the concern of computer modelers with full Bayesian analysis is that computer models are typically in an ongoing state of development and, during this development, it is typically important to clearly separate computer model uncertainty at untried inputs from the discrepancy $b(\cdot)$, in order to provide guidance for improvement of the computer model. The modular approach essentially ensures that this separation happens (to the extent that is possible), while the full Bayesian approach is much more likely to confound the two.

To see this, consider the pedagogical example of a one-dimensional dampened cosine function, as considered in Santner *et al.* (2003) and Joseph (2006).

- The true computer model is $y^M(x) = \exp(-1.4x)\cos(7\pi x/2), x \in (0,1)$, but is only observed at the $m = 7$ inputs, $x_i^M = (i - 0.5)/7$, for $i = 1, \ldots, m$.
- The real process is $y^R(x) = \exp(-1.4x - 0.05/x)\cos(7\pi x/2)$.
- $n = 7$ replicate field observations are made at each of the field design points in $D^F = (x_1^F, \ldots, x_4^F) = (1/28, 3/14, 7/14, 11/14)$, with measurement errors distributed as $\mathrm{N}(0, \frac{1}{4000})$. Note that the last 3 points also occurred in the computer model runs, but 1/28 was not a design point in those runs.

The resulting data is shown in in Figure 1.

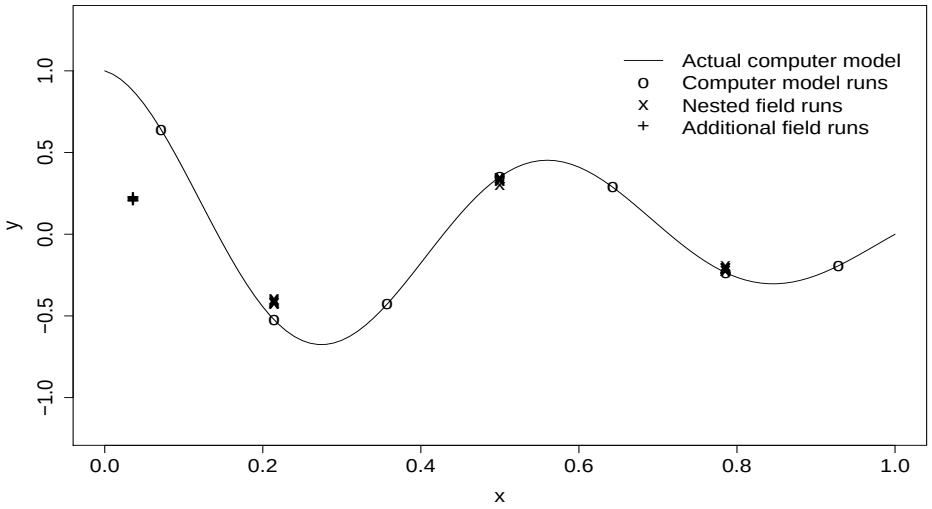

Figure 1: The simulated data from the pedagogical example.

Appendix B gives the details of the prior choices for the GASP parameters (for the computer model and the discrepancy function) and details of the ensuing Bayesian computation. The resulting emulators from both the modular approach and the full Bayes analysis are given in Figure 2. The emulators are described in this figure by their posterior means and 95% confidence bands. The emulators necessarily pass through the computer model design points, and both emulators are in reasonable agreement in the regions between these design points.

Of main interest is the design point $x = 1/28$, since this was used only in the field data, and not for the computer model runs. The modular approach provides a reasonable answer at this design point, in the sense that the 95% confidence band for the computer model from the emulator does contain the actual computer model value at that point.

In the full Bayesian analysis, however, something quite different happens; the Bayesian analysis knows that reality is near the field data, and chooses to estimate the computer model as being comparatively near this reality. The modular approach, in contrast, will ascribe the difference between the estimate of the computer model and the field data to the discrepancy function.

The problem with the full Bayesian analysis is again due to the fact that there is a suspect module, namely the model for the discrepancy $b(\cdot)$. Formally fixing the problem by improving the model for $b(\cdot)$ is very difficult, however, since essentially nothing is known about the discrepancy apriori. Fixing the problem by modularization is, in contrast,

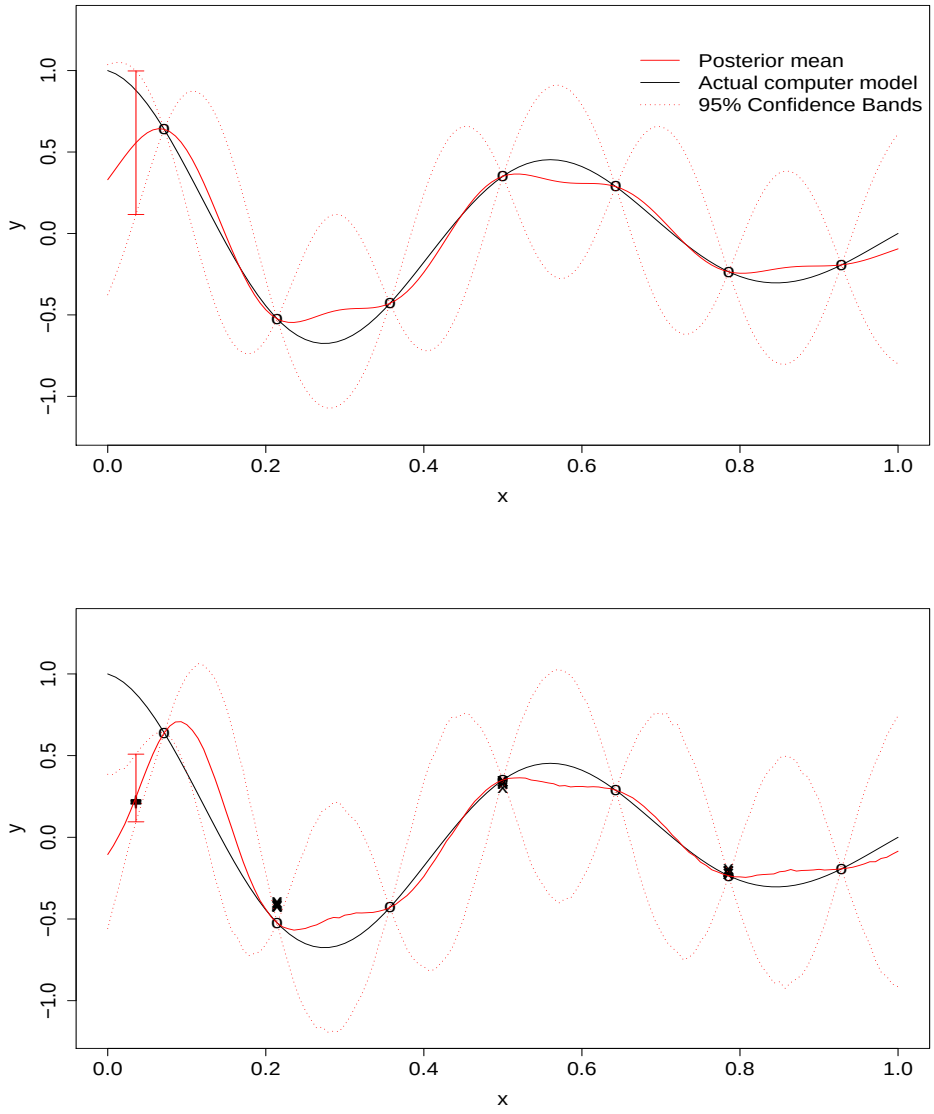

Figure 2: Emulators – as described by the posterior mean and 95% confidence bands – of the actual computer model from the modular approach (top) and from full Bayesian analysis (bottom). Vertical bars in the graphs highlight the 95% confidence bands at $x = 1/28$.

simple and natural.

In the above example, the modularized emulator was a true Bayesian emulator based on the model-run data $\boldsymbol{y}^M$. For the correlation parameters, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, that occur in the emulator GASP, an even more radical simplification is often employed: replacing

the parameters with their maximum likelihood estimates, based on $\boldsymbol{y}^M$. The rationale for doing so is that (i) there are often a large number of such parameters; (ii) the data about the parameters is typically weak; and (iii) they are typically the least important parameters in the problem, in that similar predictions often result from many reasonable choices of these parameters. See Bayarri *et al.* (2007b) for a more extensive discussion of the use of maximum likelihood to estimate the correlation parameters. Use of maximum likelihood to deal with correlation parameters also sometimes arises for computational reasons, as in (Bayarri *et al.*, 2007a), where there were hundreds of such parameters.

3.3 Modularization of the Field Data Module

3.3.1 The Motivating Example

Bayarri *et al.* (2007a) considered validation of a computer model which predicted loads resulting from stressful events on a vehicle suspension system over time. The key idea was to represent the functional data (over time t) by wavelet basis functions $\Psi_i(t)$, for i in an index set I, of size 289. This reduced the problem to analysis of the resulting wavelet coefficients $w(\cdot)$, to which the validation methodology described in Section 3.1 was applied. The ensuing model to analyze was

$$w_i^R(\boldsymbol{x}) = w_i^M(\boldsymbol{x}, \boldsymbol{u}^*) + w_i^b(\boldsymbol{x})\,, \quad w_{ir}^F(\boldsymbol{x}) = w_i^R(\boldsymbol{x}) + \varepsilon_{ir}\,, \quad \forall i \in I,\, r = 1, \ldots, 7\,, \quad (3.11)$$

where $\boldsymbol{x} = (x_1, \ldots, x_7)$ and $\boldsymbol{u} = (u_1, u_2)$ are input vectors (with $\boldsymbol{u}^*$ the true value for $\boldsymbol{u}$); $w_i^M(\boldsymbol{x}, \boldsymbol{u})$, $w_i^b(\boldsymbol{x})$, $w_i^R(\boldsymbol{x})$, $w_{ir}^F(\boldsymbol{x})$, ε_{ir} are wavelet coefficients for the model run, for the discrepancy, for reality, for the r^{th} field run, and for the error process, respectively. An additional complication in this example was that the inputs $\boldsymbol{x}$ for the field runs are unknown; only their distributions are known (specified by the engineers in the study).

We used the following distributions to model w_i^b and ε_{ir}:

$$\pi(w_i^b \mid \tau_{j(i)}^2) \sim N\left(0, \tau_{j(i)}^2\right)\,, \quad \varepsilon_{ir} \sim \mathrm{N}(0, \sigma_i^2)\,,$$

where $j(i)$ is the resolution level of the i^{th} wavelet basis. The σ_i^2 are given the usual noninformative priors $\pi(\sigma_i^2) \propto 1/\sigma_i^2$. The prior for τ_j^2 takes the form $\pi(\tau_j^2 \mid \{\sigma_i^2\}) \propto (\tau_j^2 + \frac{1}{7}\bar{\sigma}_j^2)^{-1}$, where $\bar{\sigma}_j^2$ is the average of σ_i^2 for i at level j.

3.3.2 The Motivation for Modularization

As discussed in Section 3.1, it was necessary to also model the emulators of the computer model coefficients $w_i^M(\boldsymbol{x}, \boldsymbol{u}^*)$, using GASPs. In this analysis there were 289 such coefficients, and including all 289 vectors of correlation coefficients in the full Bayesian analysis is not feasible. Hence, and also for the reasons indicated in Section 3.2, we used modularization; indeed, we simply plugged in the maximum likelihood estimates of the correlation parameters based on only the model run data.

An MCMC was then run with this partially modularized model, but proved to have a problem. A few (of the many hundreds) of parameters were not mixing well. For instance, Figure 3 shows the trace plot for σ_{170}; it is clear that this parameter is not mixing. (Note that virtually all of the parameters were mixing well.)

The cause of the problem for σ_{170} was clear: it was stuck at large values and we had been using, as a proposal distribution, the Inverse Gamma distribution arising from the replicate data for the variance, and this was concentrated on small values. We realized, however, that the problem was not really with the proposal distribution, but was with the posterior itself; the modeling was faulty if it encouraged such large values of σ_{170}.

The modularization solution to this problem was easy: simply generate the MCMC samples of the variances only from the Inverse Gamma distributions arising from the replications (as in the simplified setting of Section 2.2). Reasonable mixing of the MCMC was then achieved, as observed from the traceplot of σ_{170} in Figure 3. The point of the modularization was not, however, to achieve mixing of the MCMC but, rather, to overcome the identified problem of the model suggesting inappropriately large values of σ_{170}. This will be a recurring theme: poor mixing in an MCMC may lead to a conclusion that the modeling is inadequate and can be improved through modularization, but is not itself a reason for modularization.

3.3.3 Improved Modeling

From a purist Bayesian position, the modularization solution is unsatisfying; one has identified that the modeling is doing something wrong but, instead of identifying the modeling flaw and changing the model, one applies a rather adhoc patch.

To further study this issue for the current paper, we investigated and eventually identified the modeling flaw: the assumption of normality of the wavelet coefficients was flawed (at least at the wavelet level that contained σ_{170}). It was not easy to discover this, since the distribution of the estimated wavelet coefficients seemed quite compatible with nor-

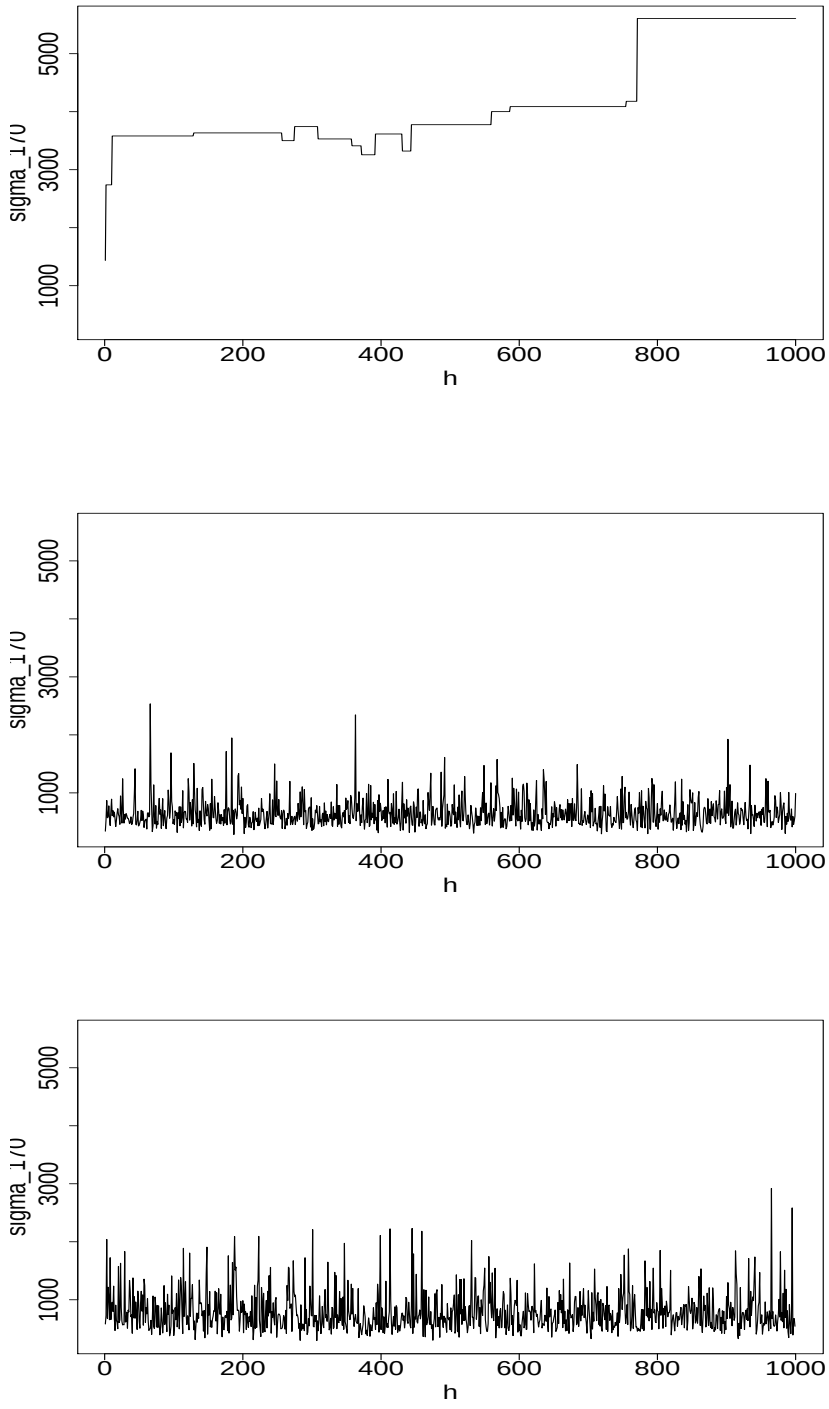

Figure 3: Trace plots for σ_{170} under full Bayesian analysis with the normal assumption (top), modularized Bayesian analysis with the normal assumption (middle), and full Bayesian analysis with the Cauchy assumption (bottom).

mality, and we had no reason to think that an error in this hyper-distribution could so strongly affect first-level model variances (for which there were a number of replications). Indeed, it was not until construction of the simplified example in Section 2.2 that we actually understood the modeling flaw and could see how to correct it, namely use more flat-tailed distributions for the wavelet coefficients. (If we had not been investigating modularization itself here, we would not likely have found this modeling error, and would have had to content ourselves with the modularized answers.)

To verify that this was the modeling problem, we redid the analysis using what is known to be a robust alternative to normality: we assumed that the w_i^b at level j follow a Cauchy distribution with scale parameter $\tau^2_{j(i)}$. Perhaps surprisingly, this model enhancement does not greatly complicate the analysis, as discussed in Appendix C. And, indeed, implementation of this robust model seemed to fix the problem; from Figure 3, we see that σ_{170} now remains small, which also causes the mixing problem to disappear.

One might wonder if the modularization or improved modeling makes a difference in the overall analysis of the study. Figure 4 gives the discrepancy functions estimated under the three approaches, while Figure 5 gives the corresponding overall predictions of reality. With full Bayesian analysis under the normal assumption, the discrepancy is clearly shrunk too much towards 0 (see the related discussion in Section 2.2) and, consequently, the computer model is not corrected appropriately, as evidenced in the top panel of Figure 5. Under both modularization and the Cauchy model, the discrepancy is estimated as being considerably larger and the prediction of reality is much more realistic.

The interesting difference between the modularized approach and use of the Cauchy model is that the confidence bands for the latter are typically wider, for both the discrepancy and the prediction of reality. More investigation would be required to clarify the reason for this difference but it does, at least, provide a warning that modularization may lead to an underestimate of variance.

3.4 Modularization of the Discrepancy Module

Dealing with the discrepancy function in computer modeling is quite challenging, in part because there typically are not direct observations from the discrepancy process and, in part, because the discrepancy is almost always seriously confounded with other unknowns in the model. Modularization is thus routinely needed to deal with the discrepancy process. This section explores several possible ways to institute modularization and illustrates the possibilities with an example.

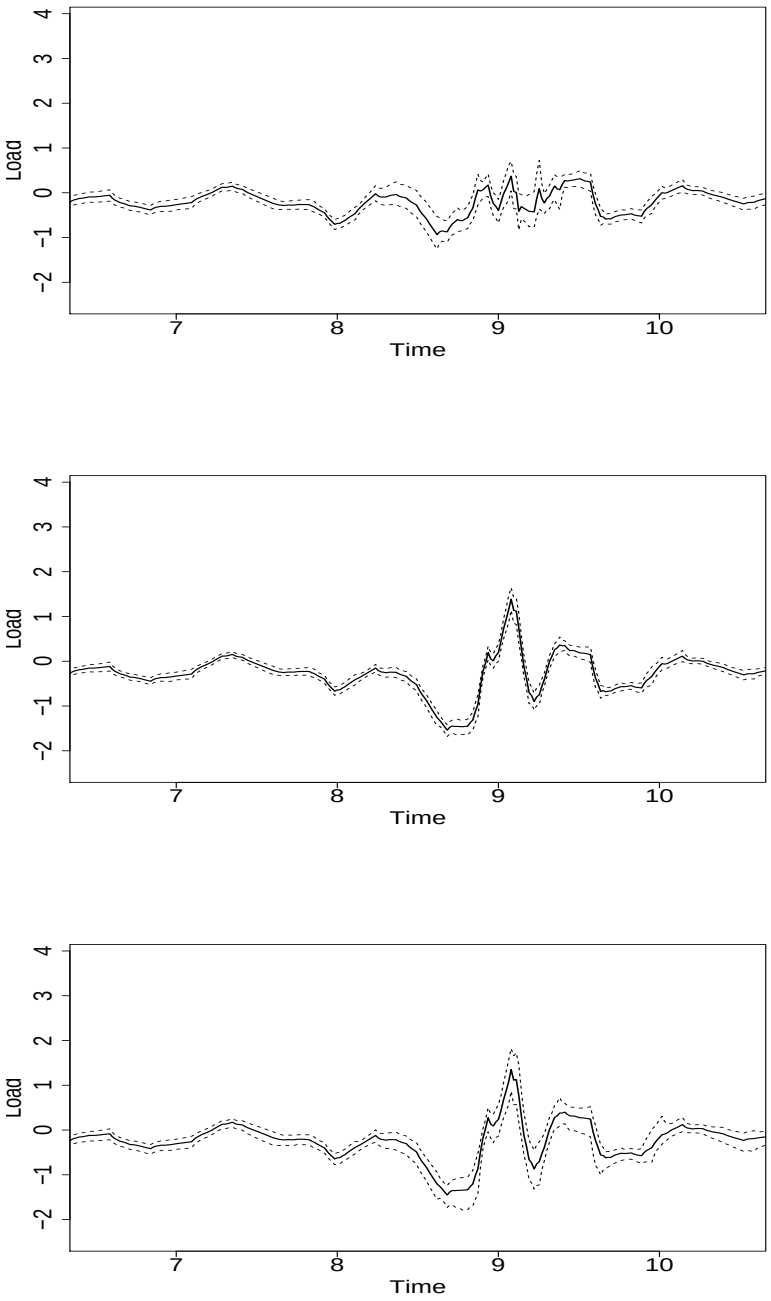

Figure 4: Posterior mean of the discrepancy function with 90% confidence bands under full Bayesian analysis with the normal assumption (top), modularized Bayesian analysis with the normal assumption (middle), and full Bayesian analysis with the Cauchy assumption (bottom). The solid black lines are the posterior mean and dashed black lines are the 90% confidence bands.

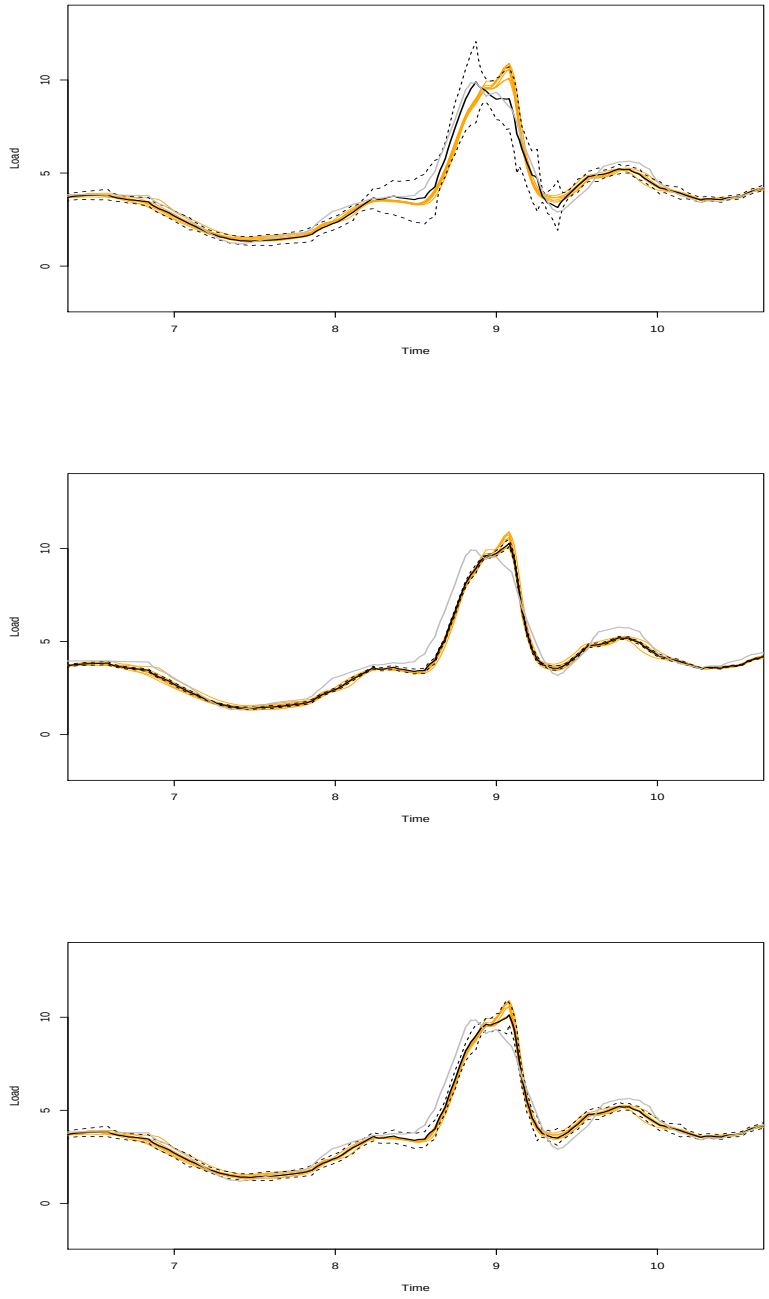

Figure 5: Posterior mean of the prediction for reality with 90% confidence bands under full Bayesian analysis with the normal assumption (top), modularized Bayesian analysis with the normal assumption (middle), and full Bayesian analysis with the Cauchy assumption (bottom). The solid black lines are the posterior means, the dashed black lines are the 90% confidence bands, the light grey lines are the 7 field runs, and the dark grey lines are the posterior prediction at the nominal input.

3.4.1 The Testbed Example and Original Analysis

The *thermal challenge problem* was studied in Liu *et al.* (2008). The computer model itself is very simple: given an input vector $\boldsymbol{x} = (\kappa, \rho, q, L)$, it predicts the temperature of a particular device at time t ($t = 100, 200, \ldots, 1000$) as

$$y^M(\kappa, \rho, L, q, t) = 25 + \frac{qL}{\kappa}\left[\frac{\kappa t/\rho}{L^2} + \frac{1}{3} - \sum_{N=1}^{6} \frac{2}{\pi^2 n^2} \exp\left(-\frac{n^2\pi^2\kappa t}{L^2\rho}\right)\right] .$$

Here, (κ, ρ) are physical properties varying from device to device; they are unknown in physical experiments, and are assumed to have a common (informative) prior distribution. The inputs (L, q) are controllable and are varied in both computer model runs and physical experiments.

In this example, field observations, denoted by $y_i^F(L, q, t)$, are taken of a device at configuration (L, q) and with the associated (unknown) physical properties equal to κ_i and ρ_i. These observations have essentially zero measurement error. The computer model discrepancy function should ideally also be a function of (L, q) and (κ_i, ρ_i), but having an unknown discrepancy that itself depends on unknown parameters (κ_i, ρ_i) is a rather severe overparameterization. Hence we make the simplifying assumption that the computer model discrepancy function corresponding to the i^{th} observation is of the form $b_i(L, q, t) = b(L, q, t) + e_i(t)$, where $e_i(t)$ is a "nugget" introduced to account for the unmodeled extra variation. The overall statistical model being used to relate field observations to the computer model is thus

$$y_i^F(L, q, t) = y^M(L, q, \kappa_i, \rho_i, t) + b(L, q, t) + e_i(t) . \tag{3.12}$$

As before, the prior distributions of $b(L, q, t)$ and $e_i(t)$ are specified to be Gaussian processes; the prior for $b(L, q, t)$ is

$$b(\cdot) \sim \mathrm{GP}\left(\mu_b, \tau^2 \mathrm{corr}\left(\cdot, \cdot\right)\right) ,$$

and the prior for $e_i(t)$ is $\mathrm{GP}\left(0, \sigma^2 c_t(\cdot, \cdot)\right)$, which is independent of $b(\cdot)$ and $e_j(\cdot)(j \neq i)$. The correlation function for the discrepancy is assumed to be separable,

$$\mathrm{corr}\left(b(L, q, t), b(L_*, q_*, t)\right) = c\left((L, q), (L_*, q_*)\right) \times c_t\left(t, t_*\right) ,$$

with

$$c\left((L,q),(L_*,q_*)\right) = \exp\left(-\beta_1|L-L_*|^2 - \beta_2|q-q_*|^2\right) \ , \quad c_t\left(t,t_*\right) = \exp\left(-\beta^{(t)}|t-t_*|^{\alpha^{(t)}}\right) \ .$$

Note that we impose the same correlation structure for the time component of e_i and of b, allowing fast computation of the inverse of the correlation matrix through use of Kronecker product simplifications (Bayarri *et al.*, 2005b).

We first considered a full Bayesian analysis, which make draws of the discrepancy correlation parameters $\boldsymbol{\theta}^{(GP)} = \left(\beta_1, \beta_2, \alpha^{(t)}, \beta^{(t)}\right)$ within an overall MCMC loop. The Markov chain was not mixing well, however, as indicated by the trace plots and autocorrelation plots for $\boldsymbol{\kappa}$ in Figure 6. The slow convergence rate was likely due to the confounding between the unknown discrepancy and the unknown (κ_i, ρ_i) in (3.12).

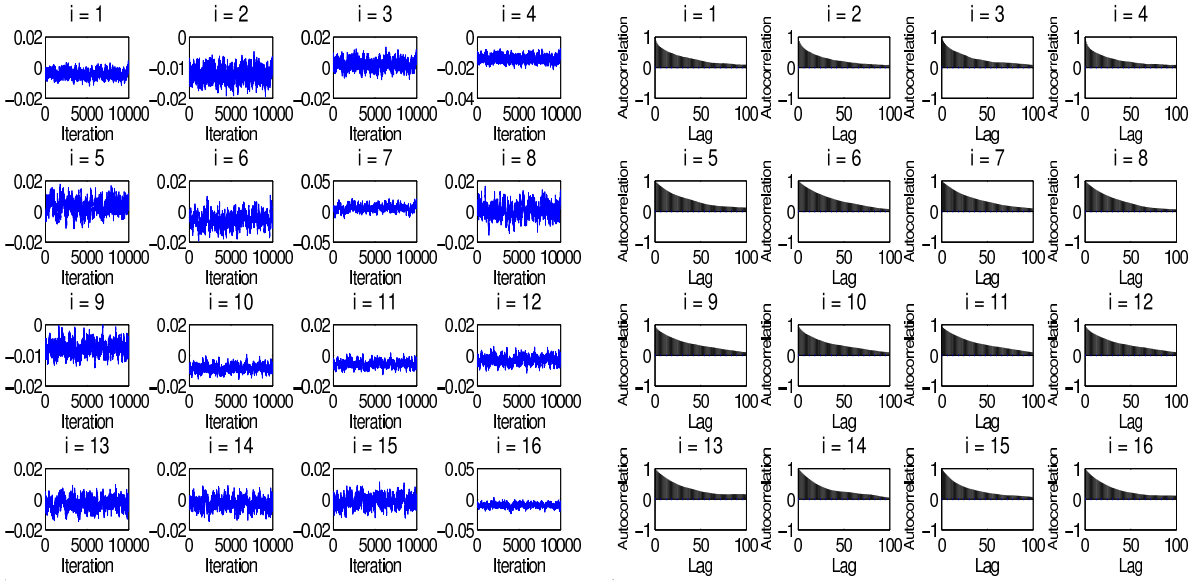

Figure 6: Trace plots (left) and acf plots (right) for $\boldsymbol{\kappa}$ under the full Bayesian approach.

To try to reduce this confounding, we take the modular approach of trying to fix the correlation parameters $\boldsymbol{\theta}^{(GP)}$ at some reasonable values before running the MCMC over the rest of the model. Again, the goal is not just to improve the computational mixing, but to hopefully deal with model inadequacies through modularization. Confounding of parameters is not itself a sign of model inadequacy, but confounding between modules of the analysis can greatly increase the detrimental effects of model inadequacies (since the confounding ensures that the different modules exert a strong influence on each other in a full Bayesian analysis).

3.4.2 Modularization Approaches

There are a variety of possible modularization schemes to fix the values of $\boldsymbol{\theta}^{(GP)}$, and it is not obvious which is best. Here we explore three intuitively natural modularization schemes.

Modular approach 1. As suggested in Bayarri *et al.* (2007b), one can estimate the discrepancy function by subtracting from the field data an estimate of the computer model output,

$$\hat{b}_i(L, q, t) = y_i^F(L, q, t) - y^M(\hat{\kappa}, \hat{\rho}, L, q, t)\,,$$

where $\hat{\kappa}$ and $\hat{\rho}$ are the prior means of the unknown device properties. One can then treat the $\hat{b}_i(L, q, t)$ as realizations from the Gaussian process with a nugget, and conduct an initial Bayesian analysis to fix $\boldsymbol{\theta}^{(GP)}$ at its posterior mean.

Modular approach 2. Sample the (κ_i, ρ_i) from their prior distributions, not their posterior distributions. Then sample the remaining parameters (including the $\boldsymbol{\theta}^{(GP)}$) from their posteriors, conditional on the prior-generated values of the (κ_i, ρ_i). Finally, $\boldsymbol{\theta}^{(GP)}$ is fixed at its resulting posterior mean.

Modular approach 3. Initially assume that the discrepancy is zero, and obtain the resulting posterior distribution for the (κ_i, ρ_i). (This would correspond to a standard 'model-tuning' operation under the assumption that the model is correct.) Next, fix (κ_i, ρ_i) at the resulting posterior means and operate as in in Approach 1 to obtain an estimate of $\boldsymbol{\theta}^{(GP)}$.

Figure 7 gives the trace plots and auto-correlation functions for $\boldsymbol{\kappa}$ when Modular Approach 1 is used; the mixing has very much improved. The estimates of the $\boldsymbol{\theta}^{(GP)}$ are actually more or less the same as they were under the full Bayesian analysis (although we were not sure it had converged), so there there may not have been any real modeling problems with the full Bayesian analysis. In any case, the modularization has greatly simplified the analysis and seems to provide essentially the same answers.

With Modular Approach 2, we observed that the the $\boldsymbol{\theta}^{(GP)}$ were much more variable than under the full Bayesian analysis or under Modular Approach 1. For example, the 95% credible interval for β_1 under Modular Approach 2 is $(1.20, 115.49)$, while the interval is $(1.56, 17.56)$ and $(1.29, 36.86)$, respectively, for the full Bayesian analysis and Modular Approach 1.

Modular Approach 3 appeared to overtune the parameters $\boldsymbol{\kappa}$ and $\boldsymbol{\rho}$ to best fit the data in the initial step, leading to unrealistic estimates of the $\boldsymbol{\theta}^{(GP)}$. This also resulted in bad

estimates of the other parameters in the model. Finally, the mixing of the MCMC under Modular Approach 3 did not seem to improve. Modular Approach 1 thus appears to be the clear winner for practical use.

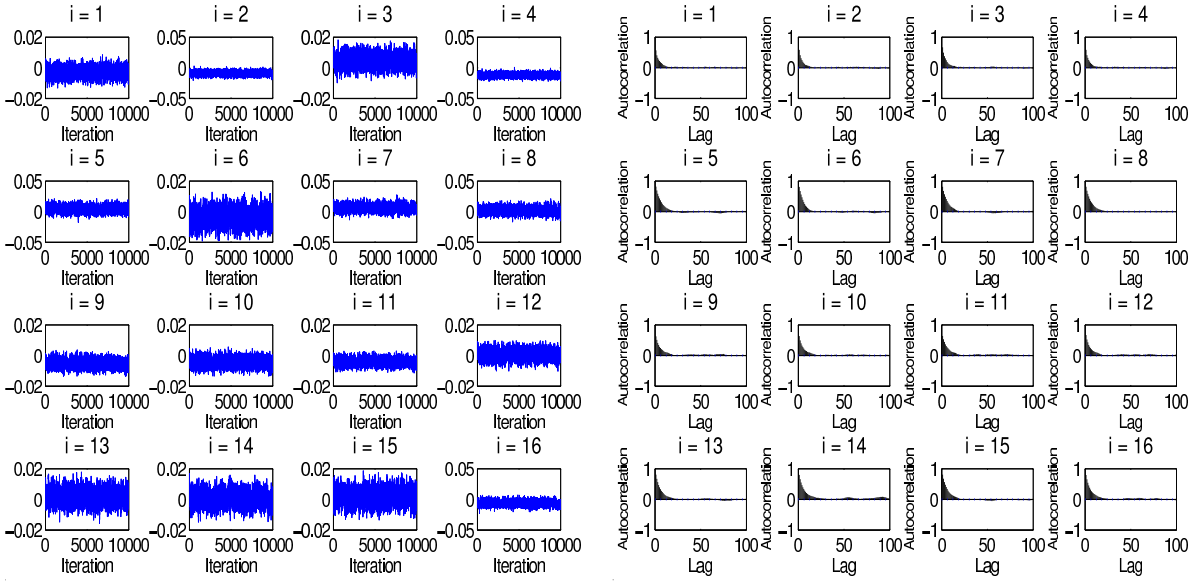

Figure 7: Left: trace plots for $\boldsymbol{\kappa}$ under Modular Approach 1; Right: acf plots for $\boldsymbol{\kappa}$.

4 Discussion

4.1 Rationales for Modularization

Five possible reasons for modularization were given in the introduction. Here is our current perspective on each (slightly rephrased), based on the examples considered in the paper.

1. *Keep a good module separate from a suspect module to avoid contamination:* This notion was encountered in several of the examples, and generally seems to be a useful idea. Of course, improving the modeling of the suspect module would be optimal, but there are many situations where this is not directly feasible, either because the suspect module is hard to identify, or because it is unknown how to improve the module.

2. *Scientific understanding and development can require modularization:* This is not necessarily distinct from the above rationale, but it's focus is different. Modularization

can allow better individual development of modules, as in computer modeling when there are ongoing efforts to improve the computer model (or, for that matter, to improve the field data); the confounding that can occur from a poorly understood discrepancy function can make improvement of these modules more difficult.

3. *Identifiability concerns or confounding might require modularization:* This is an issue of a somewhat different character. The modeling in all modules can be perfectly fine, and yet there may be unavoidable confounding of unknowns because of a lack of identifiability. Ideally, one deals with such a situation either by obtaining data of a new type that eliminates the confounding or specifying subjective prior distributions that do so but, if such additional information is not available, there is no principled solution.

If confounding occurs between parameters in different modules, there is increased danger of having a poor module 'contaminate' a good module. Furthermore, if there is confounding and the goal is overall prediction (as in overall prediction of reality in computer modeling), rather than determination of the confounded parameters themselves, it is often the case that fixing some of the confounded parameters at reasonable values will not significantly affect the predictions. So, in situations where there is significant confounding and some model inadequacy is possible, employing modularization is arguably a reasonable practical strategy. Note that fixing some parameters to eliminate confounding is not always wise; see Gustafson (2005).

4. *Mixing of MCMC analyses can greatly improve under modularization:* We have not considered using modularization simply to fix a poorly mixing MCMC. Poorly mixing MCMC's are a possible sign of a modeling problem, but they also could simply be a poorly constructed MCMC (or an unavoidably hard computational problem). Ideally, any modularization designed to improve mixing should also be justified from a modeling perspective.

5. *The computation is not otherwise possible:* This is the extreme of Reason 4 and, indeed, deviations from ideal practice are unavoidable in such situations; modularization is one of many simplifications that one might need to employ to obtain an answer. Again, however, it is preferable to have auxiliary reasons for thinking that the modularization might be reasonable, since the guarantees that come with a coherent Bayesian analysis will no longer apply.

4.2 Recommendations for Modularization in Computer Modeling

- Typically determine unknown parameters of an emulator of a computer model by modularization, utilizing only the computer model runs to construct the emulator.
- When replications of field observations are available, modularize inference about the error structure of field observations to the extent possible, by using only the residuals in the inference.
- Modularizing the discrepancy function to the extent possible is important; there are rarely direct observations on the discrepancy and the assumptions made about the discrepancy are usually quite uncertain and yet can have a profound effect. The modularization technique we have found most effective is to predetermine the discrepancy GASP correlation parameters by
  - utilizing 'fake' discrepancy observations obtained as the difference of field runs and computer model runs (or emulator means) at the same input values, with any unknown inputs in the computer model replaced by their posterior means;
  - running a Bayesian (or maximum likelihood) analysis on these fake discrepancy observations to obtain estimates of the correlation parameters.

  Note that we do not predetermine the discrepancy function variance parameters and, of course, in the overall analysis the unknown inputs are again allowed to vary. Considerable confounding remains, therefore, and care must be taken to ensure that the MCMC is constructed efficiently but, through the above modularization techniques, enough of the less important parameters can be fixed to allow for a feasible MCMC in spite of the remaining confounding.

References

Bayarri, M., Berger, J., Garcia-Donato, G., Liu, F., Palomo, J., Paulo, R., Sacks, J., Walsh, D., Cafeo, J., and Parthasarathy, R. (2007a). Computer model validation with functional outputs. *Annals of Statistics* **35**, 1874–1906.

Bayarri, M. J., Berger, J. O., Kennedy, M., Kottas, A., Paulo, R., Sacks, J., Cafeo, J. A., Lin, C. H., and Tu, J. (2005b). Validation of a computer model for vehicle collision. Tech. rep., National Institute of Statistical Sciences.

Bayarri, M. J., Berger, J. O., Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, J., Lin, C.-H., and Tu, J. (2007b). A framework for validation of computer models. *Technometrics* **49**, 2, 138–154.

Conti, S., Anderson, C., O'Hagan, A., and Kennedy, M. (2005). Bayesian analysis of complex dynamic computer models. In K. Hanson and F. Hemez, eds., *Sensitivity Analysis of Model Output*, 147–156. Loas Alamos National Laboratory. available at http://library.lanl.gov/ccw/samo2004/.

Cox, D. (1972). Regression models and life tables (with discussion). *Journal of the Royal Statistical Society B* **34**, 187–220.

Cox, D. (1975). Partial likelihood. *Biometrika* **62**, 269–275.

Craig, P. S., Goldstein, M., Rougier, J. C., and Seheult, A. H. (2001). Bayesian forecasting for complex systems using computer simulators. *Journal of the American Statistical Association* **96**, 454, 717–729.

Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991). Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association* **86**, 953–963.

Diggle, P. (2006). Spatio-temporal point processes, partial likelihood, foot and mouth disease. *Statistical methods in medical research* **15**, 325–336.

Evans, M. and Moshonov, H. (2006). Checking for prior-data conflict. *Bayesian Analysis* **1**, 4, 893–914.

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling based approaches to calculating marginal densities. *Journal of the American Statistics Association* **85**, 398–409.

Gelman, A. and Raghunathan, T. E. (2001). Conditionally specified distributions: An introduction: Comment. *Statistical Science* **16**, 268–269.

Gramacy, R. and Lee, H. (2008). Bayesian treed gaussian process models with an application to computer modeling. *Journal of the American Statistical Association* .

Gustafson, P. (2005). On model expansion, model contraction, identifiability and prior information: two illustrative scenarios involving mismeasured variables. *Statistical Science* **20**, 111–140.

Heckerman, D., Chickering, D. M., Meek, C., Rounthwaite, R., and Kadie, C. M. (2000). Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research* **1**, 49–75.

Higdon, D., Gattiker, J., Williams, B., and M., R. (2007). Computer model validation using high dimensional outputs. In J. Bernardo, M. J. Bayarri, A. P. Dawid, J. O. Berger, D. Heckerman, A. F. M. Smith, and M. West, eds., *Bayesian Statistics 8*. Oxford University Press, London. (in press).

Higdon, D., Kennedy, M. C., Cavendish, J., Cafeo, J., and Ryne, R. D. (2004). Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing* **26**, 448–466.

Joseph, V. R. (2006). Limit kriging. *Technometrics* **48**, 4, 458–466.

Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models (with discussion). *Journal of the Royal Statistical Society B* **63**, 425–464.

Liu, F. (2007). *Bayesian Functional Data Analysis for Computer Model Validation*. Ph.D. thesis, Duke University.

Liu, F., Bayarri, M. J., Berger, J. O., Paulo, R., and Sacks, J. (2008). A bayesian analysis of the thermal challenge problem. *Computer Methods in Applied Mechanics and Engineering (CMAME)* **197**, 2457–2466.

Møller, J. and Sorensen, M. (1994). Statistical analysis of a spatial birth and death process model with a view to modelling linear dune fields. *Scandinavian Journal of Statistics* **21**, 1–19.

Morris, M. D., Mitchell, T. J., and Ylvisaker, D. (1993). Bayesian design and analysis of computer experiments: Use of derivatives in surface prediction. *Technometrics* **35**, 243–255.

Newton, M. and Raftery, A. (1994). Approximate bayesian inference by the weighted likelihood bootstrap (with discussion). *Journal of the Royal Statistical Society, series B* **56**, 3–48.

Qian, P. Z. and Wu, J. C. (2008). Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments. *Technometrics* **50**, 192–204.

Raghunathan, T. E., Lepkowski, J. M., Van Hoewyk, J., and Solenberger, P. (2001). A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey Methodology* **27**, 85–95.

Reichert, P., White, G., Bayarri, M., Pitman, E., and Santer, T. (2008). Mechanism-based emulation of dynamic simulators: Concept and application in hydrology. Tech. rep., SAMSI.

Robert, C. P. and Casella, G. (2002). *Monte Carlo Statistical Methods*. Springer.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments (C/R: p423-435). *Statistical Science* **4**, 409–423.

Santner, T., Williams, B., and Notz, W. (2003). *The Design and Analysis of Computer Experiments*. Springer-Verlag.

Spiegelhalter, D., Thomas, A., Best, N., and Lunn, D. (2003). *WinBUGS User Manual*. http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/manual14.pdf.

Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J., and Morris, M. D. (1992). Screening, predicting, and computer experiments. *Technometrics* **34**, 15–25.

A Details from Section 2.2

For each i,

$$y_{ij} \mid b_i, \sigma_i^2 \;\sim\; N(b_i, \sigma_i^2)\;, \quad \text{i.i.d. for } \; j = 1, \ldots n\,.$$

Reduction by sufficiency gives

$$\begin{aligned} \bar{y}_i \mid b_i, \sigma_i^2 \;&\sim\; \mathrm{N}(b_i,\; \frac{\sigma_i^2}{n}), \quad \text{independent}, \\ s_i^2 \mid \sigma_i^2 \;&\sim\; \mathrm{Ga}(\frac{n-1}{2},\; \frac{1}{2\sigma_i^2}), \quad \text{independent}, \end{aligned}$$

where $\bar{y}_i = \sum_{j=1}^n y_{ij}/n$, $s_i^2 = \sum_{j=1}^n (y_{ij} - \bar{y}_i)^2$, and Ga is the Gamma density: $\mathrm{Ga}(x \mid a, b) \propto x^{a-1} \exp\{-bx\}$. Hence,

$$f(\bar{\boldsymbol{y}}, \boldsymbol{b} \mid \tau^2, \boldsymbol{\sigma}^2) = \prod_{i=1}^N f(\bar{y}_i \mid b_i, \sigma_i^2) \;\prod_{i=1}^N \pi(b_i \mid \tau^2)\,,$$

and the (integrated) likelihood function for $(\tau^2, \boldsymbol{\sigma}^2)$ is:

$$\begin{aligned} \mathrm{L}(\tau^2, \boldsymbol{\sigma}^2; \bar{\boldsymbol{y}}, \boldsymbol{s}^2) &= \left(\int f(\bar{\boldsymbol{y}}, \boldsymbol{b} \mid \tau^2, \boldsymbol{\sigma}^2)\; d\,\boldsymbol{b} \right) \prod_{i=1}^N f(s_i^2 \mid \sigma_i^2) \\ &= \prod_{i=1}^N N(\bar{y}_i \mid \mu_i\,, \tau^2 + \frac{\sigma_i^2}{n}) \;\prod_{i=1}^N Ga(s_i^2 \mid \frac{n-1}{2},\; \frac{n}{2\sigma_i^2}\;) \\ &\propto \prod_{i=1}^N (\sigma_i^2)^{-(n-1)/2} \exp\{-\frac{n s_i^2}{2\sigma_i^2}\} \prod_{i=1}^N \frac{1}{(\tau^2 + \sigma_i^2/n)^{1/2}} \;\exp\left(-\frac{\bar{y}_i^2}{2(\tau^2 + \sigma_i^2/n)} \right). \end{aligned}$$

The posterior distribution in (2.3) is obtained as the product of the likelihood and the priors $\pi(\boldsymbol{\sigma}^2)$ and $\pi(\tau^2 \mid \boldsymbol{\sigma}^2)$.

B Details from Section 3.2

B.1 Prior Distributions

We use y_{ij}^F to represent the j^{th} field run at x_i^F, and $D_2^F = D^F \backslash \left\{x_1^F\right\}$ to represent the nested field design with $n_2 = 3$ being the number of nested field design points. Finally, let $\boldsymbol{y}_1^M = \{y^M(x) : x \in D^M \backslash D_2^F\}$; $\boldsymbol{y}_2^M = \{y^M(x) : x \in D_2^F\}$; $\boldsymbol{y}_1^F = \{y_{ij}^F : i = 1; j = 1, \ldots, 7\}$; and $\boldsymbol{y}_2^F = \{y_{ij}^F : x_i \in D_2^F \text{ and } j = 1, \ldots, 7\}$.

We assign the Gaussian process in (3.9) as the prior for the computer model. Following Santner *et al.* (2003) and Joseph (2006), we specify the correlation function as $\text{corr}(x, x') = \exp(-136.1(x - x')^2)$. Let $\boldsymbol{\theta}^M$ denote the mean and precision parameter in the prior for the computer model. Because of the limited design space for the field data (and only three overlapping with model run design points), the discrepancies $b(x_i)$ at each design point are simply modeled as i.i.d. $\text{N}(0, 1/\lambda_b)$, rather than following a GASP. Note that the true discrepancy function here is $b(x) = \exp(-1.4x)\cos(7\pi x/2)[\exp(-0.05/x) - 1]$.

To complete the Bayesian model, we must specify prior distributions for the unknown parameters $\boldsymbol{\theta} = (\mu_M, \lambda_M, \lambda_e, \lambda_b)$. We use non-informative priors $\pi(\mu_M) \propto 1$, $\pi(\lambda_M) \propto \lambda_M^{-1}$, $\pi(\lambda_e) \propto \lambda_e^{-1}$, and $\pi(\lambda_b \mid \lambda_e) \propto \left(\lambda_b^{-1} + (n\lambda_e)^{-1}\right)^{-1}$.

B.2 Posterior distributions

B.2.1 Posterior distribution under full Bayesian analysis

Write the likelihood as a product of the following three factors,

$$\text{L}(\boldsymbol{\theta}; \text{Data}) = f(\boldsymbol{y}^M \mid \boldsymbol{\theta}^M) f(\boldsymbol{y}_2^F \mid \boldsymbol{\theta}, \boldsymbol{y}^M) f(\boldsymbol{y}_1^F \mid \boldsymbol{\theta}, \boldsymbol{y}^M, \boldsymbol{y}_2^F)\,,$$

where the first factor is

$$f(\boldsymbol{y}^M \mid \boldsymbol{\theta}^M) = \frac{\lambda_M^{m/2}}{(2\pi)^{m/2} \mid \Sigma \mid^{m/2}} \exp\left(-\frac{\lambda_M}{2}(\boldsymbol{y}^M - \mu_M \mathbf{1}_m)' \boldsymbol{\Sigma}^{-1} (\boldsymbol{y}^M - \mu_M \mathbf{1}_m)\right)\,,$$

with $\boldsymbol{\Sigma}$ being the correlation matrix of the computer model runs, $(\boldsymbol{\Sigma})_{ij} = \exp(-136.1(x_i^M - x_j^M)^2)$.

The sufficient statistics, $\bar{y}_i^F = \frac{1}{r}\sum_{j=1}^r y_{ij}^F$ and $s_i^F = \frac{1}{n}\sum_{j=1}^n (y_{ij}^F - \bar{y}_i^F)^2$, have distributions (after integrating out the discrepancies)

$$(\bar{y}_i^F \mid y_i^M, \boldsymbol{\theta}) \sim \text{N}(y_i^M, \frac{1}{\lambda_b} + \frac{1}{n\lambda_e})\,, \qquad s_i^F \mid \boldsymbol{\theta} \sim \text{Ga}(\frac{n-1}{2}, \frac{n\lambda_e}{2})\,.$$

Given $\boldsymbol{y}_2^M$, $\boldsymbol{y}_2^F$ is conditionally independent of $\boldsymbol{y}_1^M$, and given $\boldsymbol{y}^M$, y_1^F is conditionally independent of $\boldsymbol{y}_2^F$. Therefore, the second factor in the likelihood is proportional to

$$\prod_{i: x_i^F \in D_2^F} \left[\left(\frac{1}{\lambda_b} + \frac{1}{n\lambda_e}\right)^{-1/2} \exp\left(-\frac{(\bar{y}_i^F - y^M(x_i^F))^2}{2(\frac{1}{\lambda_b} + \frac{1}{n\lambda_e})}\right)\right] \left[(\lambda_e s_i^2)^{(r-1)/2-1} \exp(-\frac{\lambda_e n s_i^2}{2})\right]\,,$$

and the third factor can be written as

$$\int f(y_1^F \mid \boldsymbol{\theta}, y^M(\boldsymbol{x}_1^F)) f(y^M(\boldsymbol{x}_1^F) \mid \boldsymbol{\theta}, \boldsymbol{y}^M) \mathrm{d}y^M(\boldsymbol{x}_1^F) \propto$$

$$(\lambda_e s_1^2)^{(r-1)/2-1} \exp(-\frac{\lambda_e n s_1^2}{2}) \left(\frac{1}{\lambda_b} + \frac{1}{n\lambda_e} + \frac{\hat{V}(x_1^F)}{\lambda_M} \right)^{-1/2} \exp\left(-\frac{(\bar{y}_1^F - \hat{\mu}(x_1^F))^2}{2\left(\frac{1}{\lambda_b} + \frac{1}{n\lambda_e} + \frac{\hat{V}(x_1^F)}{\lambda_M} \right)} \right),$$

where $\hat{\mu}(x)$ and $\hat{V}(x)$ are the mean and variance functions

$$\hat{\mu}(x) = \mu_M + \boldsymbol{\rho}' \boldsymbol{\Sigma}^{-1} (\boldsymbol{y}^M - \mu_M \mathbf{1}), \quad \hat{V}(x) = \frac{1}{\lambda_M} \left(1 - \boldsymbol{\rho}' \boldsymbol{\Sigma}^{-1} \boldsymbol{\rho} \right), \tag{2.13}$$

and $\boldsymbol{\rho} = \left(\mathrm{corr}(x, x_1^M), \ldots, \mathrm{corr}(x, x_m^M) \right)'$ is the column vector of correlation functions between x_1^F and D^M.

Combining the prior with the likelihood $\mathrm{L}(\boldsymbol{\theta}; \mathrm{Data})$ yields the posterior distribution $\pi(\boldsymbol{\theta} \mid \mathrm{Data}) \propto \mathrm{L}(\boldsymbol{\theta}; \mathrm{Data}) \pi(\boldsymbol{\theta})$. MCMC techniques (cf. Robert and Casella (2002)) are used to make draws from this posterior distribution; see Appendix B.3 for details. The result of the MCMC is draws $\boldsymbol{\theta}^{(i)} = (\mu_M^{(i)}, \lambda_M^{(i)}, \lambda_e^{(i)}, \lambda_b^{(i)})$, for $i = 1, \ldots, N$. Additionally, at iteration i, we make a draw from the emulator given under a full Bayesian analysis. This can be done as follows. Given $\boldsymbol{\theta}^{(i)}$,

$$\begin{pmatrix} y^M(x) \\ \boldsymbol{y}^M \\ \bar{y}_1^F \end{pmatrix} \sim \mathrm{N}\left(\mu_M^{(i)} \mathbf{1}_{m+2}, \tilde{\Sigma}^{(i)} \right),$$

where

$$\tilde{\boldsymbol{\Sigma}}^{(i)} = \begin{pmatrix} \frac{1}{\lambda_M^{(i)}} & \frac{1}{\lambda_M^{(i)}} \boldsymbol{\rho}'(x, D^M) & \frac{1}{\lambda_M^{(i)}} \rho(x, x_1^F) \\ \frac{1}{\lambda_M^{(i)}} \boldsymbol{\rho}'(x, D^M) & \frac{1}{\lambda_M^{(i)}} \boldsymbol{\Sigma} & \frac{1}{\lambda_M^{(i)}} \boldsymbol{\rho}(x_1^F, D^M) \\ \frac{1}{\lambda_M^{(i)}} \rho(x, x_1^F) & \frac{1}{\lambda_M^{(i)}} \boldsymbol{\rho}(x_1^F, D^M) & \frac{1}{\lambda_M^{(i)}} + \frac{1}{\lambda_b^{(i)}} + \frac{1}{n\lambda_e^{(i)}} \end{pmatrix}$$

and $\boldsymbol{\rho}(x, D^M)$ is a column vector with the i^{th} element equal to $\mathrm{corr}(x, x_i^M)$ and $\rho(x, x_1^F)$ equal to $\mathrm{corr}(x, x_1^F)$. As a result, we have $\pi(y^M(x) \mid \boldsymbol{\theta}^{(i)}, \boldsymbol{y}^M, \boldsymbol{y}^F) \sim \mathrm{N}(E_x^{(i)}, V_x^{(i)})$, with

$$\begin{aligned} E_x^{(i)} &= \mu_M^{(i)} + \frac{1}{\lambda_M^{(i)}} [\boldsymbol{\rho}'(x, D^M), \rho(x, x_1^F)] (\tilde{\boldsymbol{\Sigma}}^{(i)})^{-1} \left(\begin{pmatrix} \boldsymbol{y}^M \\ \bar{y}_1^F \end{pmatrix} - \mu_M^{(i)} \mathbf{1} \right) \\ V_x^{(i)} &= \frac{1}{\lambda_M^{(i)}} - \frac{1}{\lambda_M^{(i)}} [\boldsymbol{\rho}'(x, D^M), \rho(x, x_1^F)] (\tilde{\boldsymbol{\Sigma}}^{(i)})^{-1} [\boldsymbol{\rho}'(x, D^M), \rho(x, x_1^F)]'. \end{aligned} \tag{2.14}$$

B.2.2 Posterior distribution under modularization

Under modularization, the likelihood depends only on the computer model runs $\boldsymbol{y}^M$ and $\boldsymbol{\theta}^M$, and is thus simply $f(\boldsymbol{y}^M \mid \boldsymbol{\theta}^M)$, the first factor arising in the likelihood under the full Bayesian analysis. The posterior distribution for $\boldsymbol{\theta}^M$ is $\pi(\boldsymbol{\theta}^M \mid \boldsymbol{y}^M) \propto f(\boldsymbol{y}^M \mid \boldsymbol{\theta}^M)\pi(\boldsymbol{\theta}^M)$.

Draws can be made from $\pi(\boldsymbol{\theta}^M \mid \boldsymbol{y}^M)$ by Gibbs sampling (Gelfand and Smith, 1990). Conditioning on λ_M,

$$\mu_M \mid \lambda_M, \boldsymbol{y}^M \sim \mathrm{N}\left(V_\mu^{-1}\mathbf{1}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{y}^M, V_\mu = [\lambda_M\mathbf{1}'\boldsymbol{\Sigma}^{-1}\mathbf{1}]^{-1}\right),$$

and conditioning on μ_M,

$$\lambda_M \mid \mu_M, \boldsymbol{y}^M \sim \mathrm{IG}\left(m/2, (\boldsymbol{y}^M - \mu_M\mathbf{1})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}^M - \mu_M\mathbf{1})/2\right).$$

Alternating draws from these two conditionals results in $\boldsymbol{\theta}^{M(i)} = (\mu_M^{(i)}, \lambda_M^{(i)})$, for $i = 1, \ldots, N$. Additionally, at iteration i, we make a draw from the emulator given under modularization, $\pi(y^M(x) \mid \boldsymbol{\theta}^{M(i)}, \boldsymbol{y}^M)$, utilizing (2.13).

B.3 MCMC algorithm under the full Bayesian analysis

In this section, we describe a Gibbs sampling algorithm to make draws from $\pi(\boldsymbol{\theta} \mid \text{Data})$. The full conditional distribution for μ_M is $\mathrm{N}(E_\mu, V_\mu)$ with

$$\begin{aligned} V_\mu^{-1} &= \left[\lambda_M\mathbf{1}'\boldsymbol{\Sigma}^{-1}\mathbf{1} + \left(\lambda_M^{-1}\hat{V}(x_1^F) + \lambda_b^{-1} + (r\lambda_e)^{-1}\right)^{-1}(1 - \boldsymbol{\rho}'\boldsymbol{\Sigma}^{-1}\mathbf{1})^2\right], \\ E_\mu &= V_\mu\left[\lambda_M\mathbf{1}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{y}^M + \left(\lambda_M^{-1}\hat{V}(x_1^F) + \lambda_b^{-1} + (r\lambda_e)^{-1}\right)^{-1}(1 - \boldsymbol{\rho}'\boldsymbol{\Sigma}^{-1}\mathbf{1})\left(\bar{y}_1^F - \boldsymbol{\rho}'\Sigma^{-1}\boldsymbol{y}^M\right)\right], \end{aligned}$$

where $\hat{\mu}(x)$, $\hat{V}(x)$ are the mean and variance functions given by (2.13).

The full conditional distribution for λ_M, $\pi(\lambda_M \mid \mu_M, \lambda_b, \lambda_e, \text{Data})$ can be written as

$$\pi(\lambda_M \mid \mu_M, \lambda_b, \lambda_e, \text{Data}) \propto \lambda_M^{m/2-1}\exp\left(-\frac{\lambda_M}{2}(\boldsymbol{y}^M - \mu_M\mathbf{1})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}^M - \mu_M\mathbf{1})\right) \times f(\bar{y}_1^F \mid \boldsymbol{\theta}, \boldsymbol{y}^M),$$

where $\bar{y}_1^F \mid \boldsymbol{\theta}, \boldsymbol{y}^M \sim \mathrm{N}(\hat{\mu}(x_1^F), \lambda^{-1}\hat{V}(x_1^F) + \lambda_b^{-1} + (r\lambda_e)^{-1})$. Similarly, the full conditional

distribution for λ_e is proportional to

$$\lambda_e^{(n_2+1)(r/2-3/2)-1} \exp\left(-\frac{\lambda_e}{2}\sum_{i=1}^{n} r s_i^2\right) \times \prod_{i:x_i \in D_2^F} f(\bar{y}_i^F - Y^M(x_i^F) \mid \boldsymbol{\theta}),$$

where $\bar{y}_i^F - Y^M(x_i^F) \mid \boldsymbol{\theta} \sim \mathrm{N}(0, \lambda_b^{-1} + (r\lambda_e)^{-1})$, for $x_i^F \in D_2^F$. Finally, the full conditional distribution for λ_b is proportional to

$$\left(\lambda_b^{-1} + (r\lambda_e)^{-1}\right)^{-n_2+1} \exp\left(-\frac{1}{2(\lambda_b^{-1} + (r\lambda_e)^{-1})} \sum_{i:x_i^F \in D_2^F} (\bar{y}_i^F - Y^M(x_i^F))^2\right) \times f(\bar{y}_1^F \mid \boldsymbol{\theta}, \boldsymbol{y}^M).$$

At the i^{th} iteration, the Gibbs sampling algorithm proceeds as follows.

Step 1. Given $\lambda_M^{(i)}$, $\lambda_b^{(i)}$ and $\lambda_e^{(i)}$, draw $\mu_M^{(i+1)}$ from $\mathrm{N}(E_\mu^{(i)}, V_\mu^{(i)})$.

Step 2. Given $\mu_M^{(i+1)}$, $\lambda_b^{(i)}$ and $\lambda_e^{(i)}$, propose λ_M from

$$\mathrm{Ga}\left(m/2, (\boldsymbol{y}^M - \mu^{(i+1)}\mathbf{1})^t \boldsymbol{\Sigma}^{-1} (\boldsymbol{y}^M - \mu_M^{(i+1)}\mathbf{1})/2\right).$$

Calculate the acceptance ratio,

$$\rho = \min\left\{1, \frac{f(\bar{y}_1^F \mid \mu_M^{(i+1)}, \lambda_b^{(i)}, \lambda_e^{(i)}, \lambda_M, \boldsymbol{y}^F, \boldsymbol{y}^M)}{f(\bar{y}_1^F \mid \mu_M^{(i+1)}, \lambda_b^{(i)}, \lambda_e^{(i)}, \lambda_M^{(i)}, \boldsymbol{y}^F, \boldsymbol{y}^M)}\right\}.$$

Set $\lambda_M^{(i+1)}$ equal to λ_M with probability ρ, and to $\lambda_M^{(i)}$ with probability $1-\rho$.

Step 3. Given $\mu_M^{(i+1)}$, $\lambda_M^{(i+1)}$ and $\lambda_b^{(i)}$, propose a new value λ_e from

$$\mathrm{Ga}\left((n_2+1)(r/2-3/2), \frac{1}{2}\sum_{i:x_i^F \in D_2^F} r s_i^2\right).$$

Calculate the acceptance ratio,

$$\rho = \min\left\{1, \frac{\prod_{i:x_i \in D_2^F} f(\bar{y}_i^F - y^M(x_i^F) \mid \mu_M^{(i+1)}, \lambda_M^{(i+1)}, \lambda_e, \lambda_b^{(i)})}{\prod_{i:x_i \in D_2^F} f(\bar{y}_i^F - y^M(x_i^F) \mid \mu_M^{(i+1)}, \lambda_M^{(i+1)}, \lambda_e^{(i)}, \lambda_b^{(i)})}\right\}.$$

Set $\lambda_e^{(i+1)}$ equal to λ_e with probability ρ, and to $\lambda_e^{(i)}$ with probability $1-\rho$.

Step 4. Given $\mu_M^{(i+1)}$, $\lambda_M^{(i+1)}$ and $\lambda_e^{(i+1)}$, propose a new value of λ_b by first making a draw from

$$\text{IG}\left(n_2/2, \frac{1}{2}\sum_{i:x_i^F \in D_2^F}(\bar{y}_i^F - y^M(x_i^F))^2\right).$$

Denoting this resulting draw by ig, we obtained a proposed value for λ_b by $(ig - (r\lambda_e^{(i+1)})^{-1})^{-1}$. If this value is less than 0, propose another value for λ_b by repeating the above process. Otherwise, calculate the acceptance ratio

$$\rho = \min\left\{1, \frac{f(\bar{y}_1^F \mid \mu_M^{(i+1)}, \lambda_b, \lambda_e^{(i+1)}, \lambda_M^{(i+1)}, \boldsymbol{y}^F, \boldsymbol{y}^M)}{f(\bar{y}_1^F \mid \mu_M^{(i+1)}, \lambda_b^{(i)}, \lambda_e^{(i+1)}, \lambda_M^{(i+1)}, \boldsymbol{y}^F, \boldsymbol{y}^M)}\right\}.$$

Set $\lambda_b^{(i+1)}$ to λ_b with probability ρ, to $\lambda_b^{(i)}$ with probability $1-\rho$.

To avoid highly correlated samples, we cycle within each step of Step 2-4 for 200 iterations before proceed to the next step.

C Analysis under the Cauchy model in Section 3.3

The computation for this example under the assumption of normality for the discrepancies at each wavelet level and using modularization, was given in Bayarri *et al.* (2007a). This algorithm can easily be modified to provide a full Bayesian analysis under the normality assumption, and also a full Bayesian analysis under the assumption that the discrepancies are Cauchy.

For full Bayesian analysis under normality, simply replace Step 1 in Appendix B of Bayarri *et al.* (2007a) by the following Step 1b.

Step 1b: Propose σ_i^2 from the following distribution:

$$\text{InverseGamma}\left(3, \frac{2}{s_i^2}\right) \left(shape = 3, scale = \frac{2}{s_i^2}\right).$$

Calculate the acceptance ratio by

$$\rho = \frac{\pi_{post}(\boldsymbol{\delta}^h, \boldsymbol{u}^h, \boldsymbol{\tau}^{2h}, \boldsymbol{\sigma}^2 \mid \boldsymbol{D})}{\pi_{post}(\boldsymbol{\delta}^h, \boldsymbol{u}^h, \boldsymbol{\tau}^{2h}, \boldsymbol{\sigma}^{2h} \mid \boldsymbol{D})}$$

and define $\boldsymbol{\sigma}^{2(h+1)} = \boldsymbol{\sigma}^2$ with probability $\min(1, \rho)$; $\boldsymbol{\sigma}^{2(h+1)} = \boldsymbol{\sigma}^{2h}$ otherwise.

To make draws under the full Bayesian analysis with the Cauchy assumption, we first represent the Cauchy distributions as mixtures of normal and Gamma distributions,

$$\pi(w_i^b \mid \tau_{j(i)}^2) \sim \mathrm{N}\left(0, \tau_{j(i)}^2/\lambda_i\right) , \quad \pi(\lambda_i) \sim \mathrm{Gamma}(1/2, 2) .$$

Then use Gibbs sampling to make draws from the posterior distribution. This results in the following changes to the algorithm:

- Replace $\tau_{j(i)}^h$ by $\tau_{j(i)}^h/\lambda_i^h$ in (14), (17), and (18) of Bayarri *et al.* (2007a), and condition on $\boldsymbol{\lambda}_i^h = \{\lambda_i^h\}$ in Steps 1-4 in Appendix B of that paper.

- Add one additional step within each iteration of the Gibbs sampling, to update $\boldsymbol{\lambda}$ according to its full conditional posterior distribution

$$\pi_{post}\left(\lambda_i \mid \boldsymbol{w}^b, \boldsymbol{\delta}^*, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2, \boldsymbol{D}\right) \propto \exp\left(-\frac{\tau_{j(i)}^2 + (w_i^b)^2}{2\tau_{j(i)}^2}\lambda_i\right) .$$