Supplementary materials for this article are available at 10.1007/s13253-012-0091-0.

Uncertainty Analysis for Computationally Expensive Models with Multiple Outputs

David RUPPERT, Christine A. SHOEMAKER, Yilun WANG, Yingxing LI, and Nikolay BLIZNYUK

Bayesian MCMC calibration and uncertainty analysis for computationally expensive models is implemented using the SOARS (Statistical and Optimization Analysis using Response Surfaces) methodology. SOARS uses a radial basis function interpolator as a surrogate, also known as an emulator or meta-model, for the logarithm of the posterior density. To prevent wasteful evaluations of the expensive model, the emulator is built only on a high posterior density region (HPDR), which is located by a global optimization algorithm. The set of points in the HPDR where the expensive model is evaluated is determined sequentially by the GRIMA algorithm described in detail in another paper but outlined here. Enhancements of the GRIMA algorithm were introduced to improve efficiency. A case study uses an eight-parameter `SWAT2005` (Soil and Water Assessment Tool) model where daily stream flows and phosphorus concentrations are modeled for the Town Brook watershed which is part of the New York City water supply. A Supplemental Material file available online contains additional technical details and additional analysis of the Town Brook application.

Key Words: Bayesian calibration; Computer experiments; Groundwater modeling; Inverse problems; Markov chain Monte Carlo; Radial basis functions; SOARS; Surrogate model; `SWAT` model; Town Brook watershed; Uncertainty analysis.

David Ruppert (✉) is Andrew Schultz Jr. Professor of Engineering and Professor of Statistical Science, School of Operations Research and Information Engineering and Department of Statistical Science, Cornell University, Comstock Hall, Ithaca, NY 14853, USA (E-mail: *dr24@cornell.edu*). Christine A. Shoemaker is Joseph P. Ripley Professor of Engineering, School of Civil and Environmental Engineering and School of Operations Research and Information Engineering, Cornell University, Hollister Hall, Ithaca, NY 14853, USA (E-mail: *cas12@cornell.edu*). Yilun Wang is Associate Professor, School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, China and Postdoctoral Researcher, School of Civil and Environmental Engineering, Cornell University, Ithaca, NY 14853, USA (E-mail: *yilun.wang@gmail.com*). Yingxing Li is Assistant Professor, Wang Yanan Institute for Studies in Economics, Xiamen University, Xiamen, China (E-mail: *yxli@xmu.edu.cn*). Nikolay Bliznyuk is an Assistant Professor, Department of Statistics (IFAS), University of Florida, Gainesville, FL 32611, USA (E-mail: *nbliznyuk@ufl.edu*).

Journal of Agricultural, Biological, and Environmental Statistics, Volume 17, Number 4, Pages 623–640
DOI: 10.1007/s13253-012-0091-0

1. INTRODUCTION

This paper studies the calibration and uncertainty analysis of computationally expensive computer models, often called simulators. As an example, we use the `SWAT2005` (Soil And Water Assessment Tool) simulator to model stream flows and total phosphorus in the Town Brook watershed, a component of the New York City water supply. By "calibration" we mean estimation of the parameters in the model, while by "uncertainty analysis" we mean an assessment of the accuracy of these estimates as well as of other quantities of importance.

Bayesian Markov Chain Monte Carlo (MCMC) is an excellent tool for uncertainty analysis, but often requires many, e.g., tens of thousands, of simulator runs. Depending upon the simulator, a single run may take seconds, minutes, or hours, making MCMC difficult or impossible to implement. To circumvent this problem, computationally efficient techniques using interpolation of the log-posterior have been developed. We call this methodology SOARS (Statistical and Optimization Analysis using Response Surfaces). SOARS enables uncertainty analysis both for parameters and scientifically relevant functions of the parameters. SOARS uses the GRIMA (GRow and IMprove the Approximation) algorithm of Bliznyuk, Ruppert, and Shoemaker (2012). GRIMA provides an efficient design for the computer experimentation by concentrating simulator evaluations in the high posterior density region (HPDR).

SOARS is a very general methodology and is applicable to any Bayesian analysis where the likelihood is computationally expensive; see Bliznyuk, Ruppert, and Shoemaker (2011) for another example. This paper focuses on watershed models. SOARS has already been applied to a watershed model with one constituent, that is, where the data are a univariate time series (Bliznyuk et al. 2008, 2011). This paper expands SOARS to multiple constituents (multivariate time series), and introduces an additional step using Latin hypercube sampling (LHCS) to improve SOARS's efficiency. The application here to a model with multiple outputs (flow and total dissolved phosphorus) is more complex than previous applications for one constituent, since the relationship between constituents depends on a sequence of biological and physical processes. To model phosphorus, one must also model flow, so one cannot use a simple single output analysis. Moreover, we demonstrate that SOARS, with the improvements in efficiency reported here, is computationally feasible with 8 parameters in the simulator, whereas earlier work used fewer parameters. Another novel feature here is the development of a model inadequacy function (Kennedy and O'Hagan 2001; Bayarri et al. 2007a, 2007b) for total phosphorus using a penalized spline fit to residuals.

Let $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ be the vector containing all unknown parameters in the simulator and the noise model. The noise model specifies random variation about the simulator output (see Section 3). The data consists of outcomes $\boldsymbol{Y}$ and covariates (inputs to the simulator). In our application, we use daily time series of two "outcomes" or "constituents," stream flows and dissolved phosphorus concentrations at a single location. The inputs are rainfall amounts.

The simulator and the noise model together specify the likelihood $\pi(\boldsymbol{Y}|\boldsymbol{\theta})$. The goal is to determine the set of likely values of $\boldsymbol{\theta}$. We adopt a Bayesian approach. In our application, the amount of prior information is small relative to that in the data, so we use a "noninformative" prior on $\boldsymbol{\theta}$, but more concentrated priors could be used. Given the likelihood and prior, one can in principle calculate the posterior density, $\pi(\boldsymbol{\theta}|\boldsymbol{Y})$.

Since exact calculation of the posterior is often impossible, often Markov Chain Monte Carlo (MCMC) is used. Because of the positive serial correlation usually present in an MCMC sample, to achieve a given level of accuracy, a Markov chain may need a much larger sample size than needed for an independent sample. The sample size requirement depends on the amount of serial correlation, which is difficult to know in advance. In a standard implementation of MCMC, a large Monte Carlo sample size requirement is problematic since each MCMC sample requires evaluation of the likelihood and therefore of the simulator.

A way around this computational problem is to use an "emulator" $\tilde{\ell}$ of the log-posterior. Typical emulators use either radial basis functions (RBFs) (Buhmann 2003) or Gaussian process models (kriging) (Kennedy and O'Hagan 2001; Rasmussen 2003). We used RBFs although kriging could have been used. We initially build the emulator response surface by interpolation using simulator evaluations from the optimization search for the posterior mode, plus some additional Latin hypercube samples (LHCS). The MCMC is done using only the inexpensive emulator so that lengthy MCMC runs are computationally feasible.

To construct the emulator, one must evaluate the simulator at a set of values of $\boldsymbol{\theta}$ called "evaluation points." Because the simulator is expensive, the evaluation points must be carefully chosen. Approximation over the entire parameter space is wasteful, since most of the expensive simulator evaluations will be outside the HPDR, but SOARS only approximates the log-posterior on the HPDR. The RBF emulator interpolates the log-posterior at a set of "knots" (evaluation points in the HPDR). In our applications, the HPDR is less than 1 % of the volume of the entire parameter space. Initially the location, size, and shape of the HPDR are unknown. SOARS is a methodology for locating the HPDR and then determining its size and shape with as few simulator evaluations as necessary. The design of computer experiments has a large literature; see Levy and Steinberg (2010) for a review. There exist designs to minimize the number of simulator evaluations in other contexts (Santner, Williams, and Notz 2010, Chapter 6), but these designs sample the entire parameter space. With one exception, we are not aware of any competing methods for locating and characterizing the HPDR and concentrating the sampling on that region. For example, Higdon et al. (2004) mention that the choice of the evaluation points "is an important question, but is not the focus of this paper." Other authors, e.g., Qian and Wu (2008), Qian (2009), and Cumming and Goldstein (2009), study design when there is a choice between low-cost, low-accuracy and high-cost, high-accuracy simulators. In our application, the only available simulator is the expensive SWAT program. The exception just mentioned is Rasmussen's (2003) method which requires the user supply derivatives of the posterior density and is not applicable here since SWAT output is not differentiable.

One of the main advantages of MCMC sampling is that it can be implemented using only the unnormalized posterior density. This is important, because estimation of the marginal likelihood of the data (the normalizing constant in Bayes' theorem) is often infeasible prior to MCMC sampling. SOARS retains this advantage, since all stages of SOARS, optimization, emulation, as well as MCMC, require only the unnormalized posterior density.

To test the SOARS on a realistic and difficult problem, we apply the multiple constituent version of SOARS to a widely used watershed simulator, `SWAT`. One of the challenges of uncertainty analysis of the `SWAT` model is that `SWAT` output, and therefore $\pi(\boldsymbol{\theta}|\boldsymbol{Y})$, is a very irregular function of the model parameters, especially in the region of low posterior density. The `SWAT` output is more regular in the HPDR, but finding this region is particularly difficult because of the irregular nature of the posterior density elsewhere. As a case study, we use the Town Brook watershed which flows into NYC's Cannonsville Reservoir.

The SOARS model is introduced in Section 2.1; the residual noise model that accommodates heteroscedasticity, non-Gaussian errors, and serial correlation is presented in Section 3; Section 4 discusses the watershed application; the computational requirements of SOARS and conventional MCMC are compared in Section 5; and limitations of SOARS are in Section 6. One limitation of the current implementation of SOARS is that it cannot handle multiple modes that are important (have high posterior density) and well-separated. Finally, Section 7 provides a summary and conclusions.

2. THE SOARS METHODOLOGY

2.1. Introduction to SOARS

The statistical model has two components. The first is a deterministic simulator model which, in the absence of all errors, would give the exact values of the observed data. The second is a model for the errors (noise). It is well known that fitting a model by ordinary least squares (OLS) is often not appropriate. OLS assumes that the errors are independent, normally distributed, and have a constant variance. In practice, none of these assumptions is likely to be true. In watershed modeling, we have found the observations to be right-skewed with a non-constant variance and serial correlation. The statistical noise model that we employ includes all of these features found in the data, so a Bayes' estimate using our noise model will be more efficient than OLS. The noise model is discussed in detail in Section 3.

The level-α HPDR is the set $C_R(\alpha) := \{\boldsymbol{\theta} \in \boldsymbol{\Theta} : \pi(\boldsymbol{\theta}|\boldsymbol{Y}) > c(\alpha)\}$ where $c(\alpha)$ is chosen so that $P\{C_R(\alpha)|\boldsymbol{Y}\} = 1 - \alpha$ and α is some small value, e.g., 0.01 or 0.001. As outlined in the introduction, SOARS determines $c(\alpha)$, locates and characterizes $C_R(\alpha)$, builds an emulator of log-posterior $\log\{\pi(\boldsymbol{\theta}|\boldsymbol{Y})\}$ on $C_R(\alpha)$, and uses the emulator to generate an approximate MCMC sample from $\pi(\boldsymbol{\theta}|\boldsymbol{Y})$.

SOARS has several steps which we first list and then describe in more detail in the following subsections: (1) Search for location of the posterior mode, which will be in the interior of $C_R(\alpha)$, using a global optimization algorithm such as DDS (Dynamically Dimensioned Search) developed by Tolson and Shoemaker (2007a); (2) Exploration of the region around the mode using the GRIMA algorithm (Bliznyuk, Ruppert, and Shoemaker 2012) to find the size and shape of $C_R(\alpha)$; (3) Construction of an RBF interpolant (an "emulator") of $\log\{\pi(\boldsymbol{\theta}|\boldsymbol{Y})\}$ on $C_R(\alpha)$; and (4) MCMC using the emulator in place of the computationally expensive exact log-posterior. We used an autoregressive Metropolis–Hastings (AR-MH) algorithm (Tierney 1994).

By "mode" we mean the global maximizer (assumed unique) of the posterior density. The posterior density may also have local maxima, and, to avoid misidentifying a local maximum as the mode, a global optimizer is used in Step 1. If the model is known to be unimodal, then faster results can obtained using a derivative-free local optimization method like ORBIT (Wild, Regis, and Shoemaker 2007; Wild and Shoemaker 2011).

Often the posterior has multiple local modes maxima, and this is true of the Town Brook application. Local modes that are well outside the HPDR are unimportant as they have low probability. Local modes inside the HPDR will be found by GRIMA and sampled by the MCMC after GRIMA provided that the HPDR is (topologically) connected (Bliznyuk, Ruppert, and Shoemaker 2012). When the parameter space is 8-dimensional it is challenging to discover whether the HPDR is topologically connected, but our explorations of the posterior indicate that it is.

One case where an important local mode will not be included in the posterior obtained by SOARS is when the HPDR is disconnected, e.g., where two modes of near-equal posterior density are separated by a low probability region. We have explored the Town Brook posterior density and believe that all important modes are in the HPDR. MCMC with multiple modes has been studied by, for example, Tjelmeland and Hegstad (2001), but the extension of SOARS to case of well-separated important modes is an interesting area awaiting future work. Some suggestions can be found in Section 6 of Bliznyuk, Ruppert, and Shoemaker (2012).

A referee mentioned that the Laplace approximation (Tierney and Kadane 1986) parallels Steps 1 and 2, since it requires searching for the posterior mode and then using local properties of the distribution around the approximate mode. However, the Laplace approximation computes the Hessian either analytically or numerically, whereas SOARS does not assume that the posterior is differentiable and instead uses an RBF approximation.

2.2. LOCATING THE POSTERIOR MODE

In Step 1, the objective function is $-\log\{\pi(\boldsymbol{\theta}|\boldsymbol{Y})\}$ which is minimized. Evaluations of the simulator during the global optimization step are used in the construction of the emulator. It is not necessary to locate the minimizer with great accuracy, only to locate $C_R(\alpha)$, so the design of the optimization step should focus on obtaining an "informative" set of evaluation points. Evaluation points that are close to other evaluation points are redundant and provide little additional information about $-\log\{\pi(\boldsymbol{\theta}|\boldsymbol{Y})\}$. Evaluation points outside of $C_R(\alpha)$ are also wasteful since the emulator will only be built on $C_R(\alpha)$. Therefore, an informative set of evaluation points is one that is concentrated in and evenly distributed across $C_R(\alpha)$. A space-filling design across the entire parameter space would be very inefficient since $C_R(\alpha)$ is often less than 1 % (by volume) of the parameter space. Since DDS (Tolson and Shoemaker 2007a) is a global optimization algorithm that was designed to provide a near-optimal solution with relatively few function evaluations and has worked well on watershed examples, we used DDS.

2.3. GRIMA

GRIMA expands the set of evaluation points beyond those found during optimization. A detailed description of GRIMA and an illustrative example can be found in Bliznyuk, Ruppert, and Shoemaker (2012), so here we give only a summary. After optimization, but before starting GRIMA, we found it helpful to evaluate the emulator at a moderate number of evaluation points, e.g., 500, chosen by Latin hypercube sampling (LHCS), centered at the posterior mode located during optimization. The mode, of course, is guaranteed to be inside $C_R(\alpha)$. For reasons already mentioned, the simulator was not evaluated at any point in the LHCS that was close to an existing evaluation point. The LHCS provides additional information about the shape of $-\log\{\pi(\boldsymbol{\theta}|\boldsymbol{Y})\}$ on $C_R(\alpha)$ and this information enables GRIMA to expand the set of evaluation points more efficiently. Let $\mathcal{D}_0$ be the set of evaluation points from the optimization step and the LHCS, except that evaluation points with very low probability density (e.g., outside the HPDR) are excluded. LHCS was not needed in previous applications of GRIMA where the number of parameters was four or less. Based on this study, we recommend the inclusion of LHCS after optimization and before GRIMA for higher dimensional problems.

GRIMA produces a nested sequence $\mathcal{D}_0, \mathcal{D}_1, \ldots$ of sets of evaluation points. Given the current set $\mathcal{D}_i$, let $\mathcal{C}$ be the set of parameter values whose distance from $\mathcal{D}_i$ is exactly r. Here r is a tuning parameter that varies with i (see below) and the distance from a point x to a set S is defined to be $\inf\{\|x - y\| : y \in S\}$. Let $\tilde{\ell}_i$ be the emulator of the log-posterior on $\mathcal{D}_i$.

The candidate for the next evaluation point is the point in $\mathcal{C}$ where $-\tilde{\ell}_i$ is minimized. Because this point is exactly at distance r from $\mathcal{D}_i$, it is neither redundant (e.g., too close to the current evaluation points) nor too far from the evaluation points; an evaluation point very detached from the other evaluation points should be avoided since, in our experience, it can cause an inaccurate emulator. If the candidate next evaluation point appears too far outside the current estimate of $C_R(\alpha)$, it is not accepted (so $\mathcal{D}_{i+1} = \mathcal{D}_i$). Instead r is replaced by ρr, where $0 < \rho < 1$. We used $\rho = 0.9$. The reason r is reduced is that the set of parameter values whose distance from $\mathcal{D}_i$ is at most r has grown and appears to cover much, if not all, of $C_R(\alpha)$. The next task will be to fill in gaps in the coverage of $C_R(\alpha)$ by evaluation points by reducing r.

On the other hand, if the candidate for the next evaluation point is accepted, then up to $J - 1$ additional candidate points are tried, where J is a user-selected tuning parameter. We used $J = 4$. If any of these additional candidate points are rejected, then no new candidates are tried and r is reduced as described above. If all J candidates are accepted, then r is expanded by replacing it by $\rho^{-1}r$. This expansion of r facilitates a more rapid coverage of $C_R(\alpha)$ by the evaluation points. The name "GRIMA" comes from the initial "GRowth" stage (where r tends to increase) and the subsequent "IMprove the Approximation" stage (where r is more likely to decrease), although there is not a sharp boundary between the two stages because the algorithm can oscillate between decreasing and increasing r.

Thus, up to J evaluation points can be selected during a single GRIMA iteration. The emulator, $C_R(\alpha)$, and the scaling matrix (Section 2.4) are updated at the end of any iteration such that the simulator has been run on at least M new evaluation points since the last update. These evaluations could be run in parallel; see Section 6. We used $M = 12$.

The criterion for stopping GRIMA uses the estimated marginal posterior distributions of the components of $\boldsymbol{\theta}$ at each iteration. When these distributions stop changing, as determined using the approximate total variation norm between densities at intermediate and terminal iterations, GRIMA stops. See Section 4.4.2 and Appendix A.3 of Bliznyuk, Ruppert, and Shoemaker (2012).

2.4. RADIAL BASIS FUNCTION INTERPOLATION

The emulator is constructed by RBF interpolation as discussed in detail in Appendix A.3 of Bliznyuk et al. (2008). An efficient algorithm for updating the RBF surface, which must be done repeatedly during GRIMA, is in the Appendix of Bliznyuk, Ruppert, and Shoemaker (2012). Radially symmetric interpolants such as RBFs are sensitive to the parameterization and are improved by "sphering" which, at the ith iteration of GRIMA, replaces $\boldsymbol{\theta}$ by $H_i^{-1}\boldsymbol{\theta}$ where H_i is a square root, e.g., a Cholesky factor, of the posterior covariance of $\boldsymbol{\theta}$. We call H_i the scaling matrix.

3. THE NOISE MODEL

For concreteness, we will assume that the data are a d-dimensional multivariate time series of length n, $\boldsymbol{Y}_i$, $i = 1, \dots, n$, where $\boldsymbol{Y}_i = (Y_{i,1} \dots, Y_{i,d})^{\mathsf{T}}$ is a (column) vector of observations at time i. In the application to the Town Brook watershed in Section 4, $d = 2$, $Y_{i,1}$ is the flow on day i, and $Y_{i,2}$ is the concentration of the dissolved phosphorus that day. It is assumed that, in the absence of noise and systematic errors, $\boldsymbol{Y}_i = \boldsymbol{f}_i(\boldsymbol{\beta}),\ i = 1, \dots, n$, where $\boldsymbol{f}_i(\boldsymbol{\beta}) = (f_{i,1}(\boldsymbol{\beta}), \dots, f_{i,d}(\boldsymbol{\beta}))^{\mathsf{T}}$ is the simulator output for time i and $\boldsymbol{\beta}$ is the vector of unknown parameters in the simulator. Of course, noise will be present, so one should expand the model to $\boldsymbol{Y}_i = \boldsymbol{f}_i(\boldsymbol{\beta}) + \boldsymbol{\epsilon}_i$. The term $\boldsymbol{\epsilon}_i$ represents all sources of discrepancy between the data and the model, including modeling error, measurement error in $\boldsymbol{Y}_i$, model inadequacy, and error in model inputs, e.g., rainfall in a watershed model. The expanded model is a nonlinear regression model (Bates and Watts 1988). Such models are often fit by nonlinear least-squares, but least-squares is not appropriate here because the $\boldsymbol{\epsilon}_i$ are non-normally distributed with a non-constant variance and serial correlation.

We use a variant of the Box–Cox (1964) transformation to induce approximate normality and constant variance. The Box–Cox family for $y > 0$ is $h_{\mathrm{BC}}(y, \lambda) = (y^{\lambda} - 1)/\lambda$ if $\lambda \neq 0$ and $h_{\mathrm{BC}}(y, \lambda) = \log(y) = \lim_{\lambda \to 0}(y^{\lambda} - 1)/\lambda$ if $\lambda = 0$. A technical problem is that for $\lambda \neq 0$, the transformation is bounded below by $-\lambda^{-1}$, so the transformed data cannot be Gaussian. This problem is remedied by perturbing h_{BC} slightly using the log transformation to obtain $h(y, \lambda) := (1 - \Delta) \cdot h_{\mathrm{BC}}(y, \lambda) + \Delta \log(y)$, where Δ is a small and fixed positive constant (e.g., 10^{-4}). The small term $\Delta \log(y)$ makes the transformation unbounded but close to a Box–Cox transformation. Each constituent of $\boldsymbol{Y}$ needs its own transformation parameter, so we define the multivariate transformation $h(\boldsymbol{y}, \boldsymbol{\lambda}) = \{h(y_1, \lambda_1) \cdots h(y_d, \lambda_d)\}^{\mathsf{T}}$, where $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_d)^{\mathsf{T}}$ is the vector of transformation parameters. To accommodate auto- and cross-correlations, we assume that $h(\boldsymbol{Y}_i, \boldsymbol{\lambda}) =$

$h\{\boldsymbol{f}_i(\boldsymbol{\beta}), \boldsymbol{\lambda}\} + \boldsymbol{\epsilon}_i$ where $\boldsymbol{\epsilon}_i$ is a Gaussian vector AR(1) [VAR(1)] process (Hamilton 1994); $\boldsymbol{\epsilon}_i = \boldsymbol{\Phi}\boldsymbol{\epsilon}_{i-1} + \boldsymbol{u}_i$, where $\boldsymbol{\Phi}$ is a $d \times d$ matrix and $\boldsymbol{u}_i$, $i = 1, \dots, n$, is an independent sequence of Gaussian vectors with mean 0 and covariance matrix $\boldsymbol{\Sigma}_u$. The noise parameters are integrated out of the posterior, so using a bivariate VAR(1) model, rather than simply modeling each outcome separately using scalar AR(1) processes, does not increase the dimension of the RBF emulator. This is important, since RBF approximation suffers from the curse of dimensionality.

Because the same transformation is applied to both $\boldsymbol{Y}$ and $\boldsymbol{f}_i(\boldsymbol{\beta})$, the median of $Y_{i,j}$ is $\boldsymbol{f}_{i,j}(\boldsymbol{\beta})$ regardless of the value of λ_j. Therefore, the physical meaning of the simulator $\boldsymbol{f}(\boldsymbol{\beta})$ is preserved under transformation (Carroll and Ruppert 1984, 1988).

Let $\boldsymbol{\eta} = (\boldsymbol{\lambda}, \boldsymbol{\Phi}, \boldsymbol{\Sigma}_u)$ be the noise parameters and $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\eta})$ the set of all parameters. Define $\boldsymbol{\epsilon}_i(\boldsymbol{\theta}) = h(\boldsymbol{Y}_i, \boldsymbol{\lambda}) - h\{\boldsymbol{f}_i(\boldsymbol{\beta}), \boldsymbol{\lambda}\}$ and $\boldsymbol{u}_i(\boldsymbol{\theta}) = \boldsymbol{\epsilon}_i(\boldsymbol{\theta}) - \boldsymbol{\Phi}\boldsymbol{\epsilon}_{i-1}(\boldsymbol{\theta})$, $i = 2, \dots, n$. Let $\pi(\boldsymbol{\theta})$ be the prior density. Then the log of the posterior density is

$$\begin{aligned} \log \pi(\boldsymbol{\theta}|\boldsymbol{Y}) &= \text{const} + \log \pi(\boldsymbol{\theta}) - \left(\frac{n-1}{2}\right) \log\bigl(|\boldsymbol{\Sigma}_u|\bigr) \\ &\quad - \frac{1}{2} \sum_{i=2}^{n} \boldsymbol{u}_i(\boldsymbol{\theta})^{\mathsf{T}} \boldsymbol{\Sigma}_u^{-1} \boldsymbol{u}_i(\boldsymbol{\theta}) + \sum_{i=1}^{n} \sum_{j=1}^{d} \log \frac{\partial h(Y_{i,j}, \boldsymbol{\lambda})}{\partial Y_{i,j}}. \qquad (3.1) \end{aligned}$$

The constant is the log of the normalizing factor. We drop it so $\pi(\boldsymbol{\theta}|\boldsymbol{Y})$ is an unnormalized density. The last term in (3.1) is the log of the Jacobian of the transformation of $\boldsymbol{Y}_i$.

Since our primary interest is in the simulator parameters $\boldsymbol{\beta}$, we would like to integrate the noise parameters $\boldsymbol{\eta}$ out of the posterior density to obtain the marginal posterior density of $\boldsymbol{\beta}$. There is a substantial computational advantage to working with only $\boldsymbol{\beta}$ rather than the full parameter vector $\boldsymbol{\theta}$, since the number of evaluation points needed for accurate RBF interpolation grows rapidly with the dimension. In the Town Brook example in Section 4, the dimensions of $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ are 8 and 17, respectively. However, analytic integration is not always possible and numerical integration can be computationally challenging, although in some cases it is possible to analytically integrate out a subset of the noise parameters. In our noise model, we could and did integrate out $\boldsymbol{\Sigma}_u$; see Section 6 of the Supplemental Material for the technical details. The noise parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\Phi}$ were maximized out of the posterior using `FMINCON` in `Matlab`. Maximizing over noise parameters as an approximation to integrating them out has been shown to introduce very little error (Bliznyuk et al. 2008).

As mentioned, the noise model in this section is suitable when the data are a multivariate time series. Appropriate modifications can be made for spatial or spatial-temporal data.

The initial optimization steps performed better if we assumed no serial correlation by taking $\Phi = 0$. This makes sense, since it is impossible to estimate the noise correlation well without reasonable estimates of the watershed parameters. The simulator output $\boldsymbol{f}_i(\boldsymbol{\beta})$ used initially for the objective function with no serial correlation can be reused after switching to VAR(1) noise, so there is no waste of expensive simulator evaluations.

4. APPLICATION TO THE TOWN BROOK WATERSHED

4.1. BACKGROUND

The Town Brook watershed is a 37-km^2 subwatershed of the Cannonsville watershed (1200 km^2) in New York State. There are 2192 daily observations (from October 1998 to September 2004) based on readings by the U.S. Geological Survey. The data are the multivariate time series $\boldsymbol{Y}$ of measured stream flows and total dissolved phosphorus (TDP) concentrations in water entering the West Branch of the Delaware River from the Town Brook watershed.

The water from the Town Brook and the rest of the Cannonsville watershed collects in the Cannonsville Reservoir. The reservoir water is piped hundreds of miles to New York City for drinking water. Phosphorus pollution is a concern. If the water quality is not protected, New York City might need to build a water filtration plant estimated to cost over \$8 billion.

The input information of the Town Brook simulator is discussed briefly in Tolson and Shoemaker (2007a) and in more detail in Tolson and Shoemaker (2004, 2007b).

4.2. DETAILS OF THE SWAT MODEL

The value of $\boldsymbol{f}_i(\boldsymbol{\beta})$ in (1) is the multivariate output of the SWAT watershed model on day i. We chose the SWAT model for this study because it is an example of widely used simulation models for which there is a need for computationally efficient uncertainty quantification. The SWAT model is the predominant model used for analysis of rural watersheds, and it is currently being used around the world (U.S., Africa, Canada, South America, Europe, and Asia) for determining the impacts of land use and climate on water supply and water quality. There are currently over 800 articles in the peer reviewed scientific literature related to SWAT model development (e.g., Eckhardt et al. 2002; Grizzetti et al. 2003; Shoemaker, Regis, and Fleming 2007; Tolson and Shoemaker 2007b). The initial SWAT article (Arnold et al. 1998) has been cited over 1300 times. In addition to its influence on academic research, the SWAT model has been used by over 85 government agencies and 25 companies to study alternative policies and to make regulatory decisions about water quality protection.

The `SWAT2005` Town Brook watershed model has multiple parameters. Among them, 4 flow-related parameters and 4 total dissolved phosphorous (TDP) related parameters were chosen to be estimated. They are denoted as $\beta_1, \ldots, \beta_8$; and their physical meaning, lower bounds and upper bounds, are in Table 1 of the Supplemental Material. These bounds were set in the original calibration of the model (Tolson 2005) based on physical conditions in the watershed. During optimization and GRIMA computations, each parameter was re-scaled to [0, 1] to avoid numerical scaling problems. When desired, it is easy to convert results back to the original scale. The other parameters in the model were of lesser interest and were fixed at values determined by a subject matter specialist.

4.3. Likelihood and Priors

The likelihood came from the noise model described in (3.1) of Section 3 and the SWAT model. We used uniform(0, 1) priors on $\beta_1, \ldots, \beta_8$. In this problem, the likelihood is concentrated on less than 1 % (by volume) of the parameter space. Any prior on $\beta_1, \ldots, \beta_8$ that is not sharply peaked will be nearly constant on the HPDR, and the effect of changing the prior will be quite small.

We also used uniform$(-2, 1)$ priors on λ_1 and λ_2. For $\boldsymbol{\Sigma}_u^{-1}$ we used a Wishart prior with 2 degrees of freedom and scale matrix $10^{-5}\boldsymbol{I}$ where $\boldsymbol{I}$ is the 2×2 identity matrix. As discussed in Section 6 of the Supplemental Material, this choice of prior for $\boldsymbol{\Sigma}_u^{-1}$ makes the effect of the prior upon the posterior negligible.

We used uniform$(-0.8, 0.8)$ priors on the four entries of $\boldsymbol{\Phi}$; we call this "Prior 1." We found that at the posterior mode, $\boldsymbol{\Phi}$ was on the boundary of the support of Prior 1 because both diagonal elements were 0.8. We then tried "Prior 2" which used uniform$(-2, 2)$ priors on the four entries of $\boldsymbol{\Phi}$; for all other parameters, Priors 1 and 2 are identical. The HPDR is well inside the interior of the support of Prior 2. See Section 3 of the Supplemental Material for more discussion of the priors, especially the sensitivity to Prior 1 versus Prior 2.

4.4. Results

4.4.1. Optimization

As mentioned in Section 2.1, the first step of SOARS uses a global optimization algorithm, DDS (Tolson and Shoemaker 2007a), to locate the posterior mode and roughly approximate $C_R(\alpha)$. During the optimization stage, $f(\boldsymbol{\beta})$ was computed at 1900 evaluation points (values of $\boldsymbol{\beta}$), 400 using the sum of squares (with transform-both-sides) and then 1500 using the log-posterior. The mode found by DDS will be denoted by $\hat{\boldsymbol{\beta}}_{\mathrm{OPT}}$.

To visualize the posterior surface, 800 additional simulator evaluations were run, 100 for each β, to create profile plots, which are in Figure 7 of the Supplemental Material. The profile plots are not necessary and could be omitted. Except in one case (see below), we did not make further use of the function evaluations used to produce the profile plots, because we wanted to mimic the case where the profile plots would not be generated. In the profile plots, the kth component of $\hat{\boldsymbol{\beta}}_{\mathrm{OPT}}$ is varied over a small neighborhood of $\hat{\boldsymbol{\beta}}_{\mathrm{OPT}}$ while keeping all the other components fixed. Then we plotted $-2 \times$ log-posterior versus the β_k; the "-2" converts the log-likelihood into a deviance. The kth component of $\hat{\boldsymbol{\beta}}_{\mathrm{OPT}}$, shown as a triangle, should be the minimizer in each subplot, but this is not the case for $\hat{\beta}_7$. This shows that 1900 function evaluations were not sufficient for DDS to locate the maximum of the posterior; this problem is due, at least partially, to the nonsmooth SWAT output. Fortunately, GRIMA was able to improve upon DDS; see below. Initially, we set $C_R(\alpha)$ as $\{\boldsymbol{\beta} \in C_R(\alpha) : -2l(\boldsymbol{\beta}) \le -2l(\hat{\boldsymbol{\beta}}_{\mathrm{OPT}}) + \chi^2_{0.99}(8)\}$, where $\chi^2_{0.99}(8)$ is the 0.99-quantile of the χ^2 distribution with $\dim(\beta) = 8$ degrees of freedom.

DDS needed 1900 simulator evaluations. It was noticed that β_7 was nearly constant during DDS, so we used the 100 simulator evaluations from the profile plot of β_7. This gave us a total of 2000 simulator evaluations. We recommend that in practice, any parameter that has varied little during optimization be varied after optimization in this way, with the other parameters fixed at their values in $\hat{\boldsymbol{\beta}}_{\mathrm{OPT}}$.

Table 1. Values of β that maximize the profile log likelihood of the VAR(1) model and found by optimization via DDS and after GRIMA.

Stage	$\bar{\beta}_1$	$\bar{\beta}_2$	$\bar{\beta}_3$	$\bar{\beta}_4$	$\bar{\beta}_5$	$\bar{\beta}_6$	$\bar{\beta}_7$	$\bar{\beta}_8$
DDS	0.4318	0.0747	0.0114	0.0511	0.1643	0.0019	0.0113	0.0099
GRIMA	0.3836	0.0978	0.0125	0.0459	0.3796	0.0066	0.0000	0.0002

LHCS was used to select an additional 500 evaluation points centered at and concentrated near $\hat{\boldsymbol{\beta}}_{\text{DDS}}$. As mentioned before, the LHCS provided GRIMA with information about shape of the log-posterior in the HPDR. After the optimization and LHCS, our algorithm automatically selected 264 knots within $C_R(\alpha)$ for the RBF emulator of the log-posterior.

A major benefit of the optimization and subsequent LHCS was to approximate the HPDR. This region in the Town Brook example includes less than 1 % of the volume of the full parameter domain. As a result the GRIMA algorithm could search over a much smaller region, which increases its efficiency tremendously.

4.4.2. GRIMA

After optimization and LHCS, we started GRIMA to improve our approximation of the log-posterior within $C_R(\alpha)$. In each iteration, up to $J = 4$ simulator evaluations were made and we generated an MCMC chain of length 20,000 based on the approximate posterior surface. A diagnostic MCMC chain of length 60,000 and a tuning MCMC chain of length 40,000 were generated. GRIMA went through 468 iterations and needed a total of 1017 simulator evaluations, so the number of updates of the RBF surface could be at most $1017/12 = 84.75$; in fact, the RBF surface was updated 78 times. The new estimate of the posterior mode was the maximizer of the log-posterior over the total of 3517 evaluation points from optimization, the profile plots, LHCS, and GRIMA. Table 1 compares the estimates of $\boldsymbol{\beta}$ obtained by DDS and GRIMA. Recall that DDS did not provide a good estimate of the mode of the posterior, particularly of the value of β_7 at the mode; see Figure 7 of the Supplemental Material. Table 1 shows that GRIMA is able to improve the estimate of the mode because β_7 changed from 0.0113 to 0.0000 and the latter is where the profile plot for β_7 in Figure 7 of the Supplemental Material is minimized.

The stopping criterion terminated GRIMA after 1017 simulator evaluations. Figure 1 compares the approximate total variation distances between the marginal posterior densities of β_k at termination, with 1281 knots, and earlier iterations. For each β_k, the approximation seems to improve little after the number of knots reaches 1100.

4.4.3. Checking MCMC Convergence and the Noise Model

We checked for convergence of the MCMC sampling and for goodness-of-fit of the noise model. Since these diagnostics are routine, they are not included here but can be found in Sections 2 and 4 of the Supplemental Material.

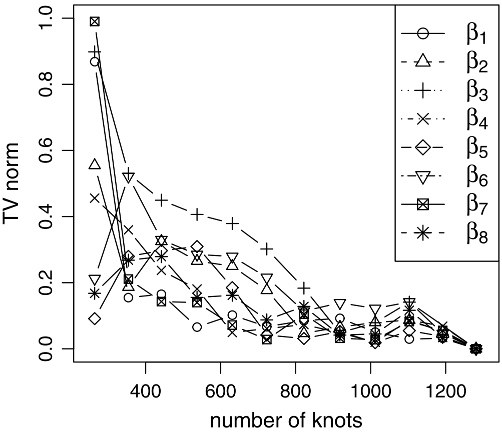

Figure 1. Approximate Total Variation norm between intermediate and terminal steps.

4.5. Model Adequacy

To check for simulator adequacy, one can plot the residuals against the predicted values or covariates. Here the only covariates are the rainfall amounts, and since these are measured with sampling error and have only short-term effects, we did not use them. A plot of the residuals versus the predicted values for TDP can be found in Figure 2(b). A penalized spline (Ruppert, Wand, and Carroll 2003) was added. All except one predicted value lie to the right of the vertical dashed line, so we will only interpret the spline to the right of this line. The simulator correctly, under, and over predicts TDP when the prediction is small, moderate, or large, respectively. The spline is a model inadequacy function (Kennedy and O'Hagan 2001) and can be added to the predicted TDP to improve the predictions; doing this, reduces the mean squared prediction error for TDP by 22 %. A similar analysis found little model inadequacy for flow, except for small predicted values; see Figure 2(a).

Other authors, e.g., Kennedy and O'Hagan (2001) use an additive model inadequacy function that is independent of the model output. For example, Bayarri et al. (2007a, 2007b) use model inadequacy functions of time. This approach is applicable where time is repeatable, as in the pedagogic example (Bayarri et al. 2007a) where time is measured from the initiation of a chemical reaction. In our example, a model inadequacy function that depended on time would not be extendable into the future and so would not be useful for predictions or management of the watershed. In a spot welding example (Bayarri et al. 2007a), the model inadequacy function of these authors in a function of load, direct current, and gauge.

4.6. Implications for Watershed Management

In this section, we show how SOARS can compute the posterior distributions, not only of the model parameters, but of interesting model outputs as well.

During the optimization and GRIMA stages, with each of the 1281 evaluations of the expensive simulator, we stored the average daily stream flow, the amount of transported phosphorous, and other outputs of interest. This information was used to generate the RBF

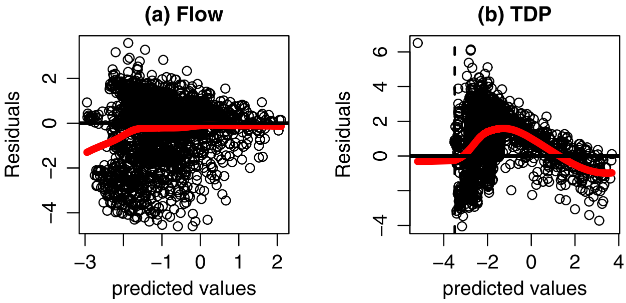

Figure 2. Residual plots for flow and TDP with model inadequacy functions as red curves.

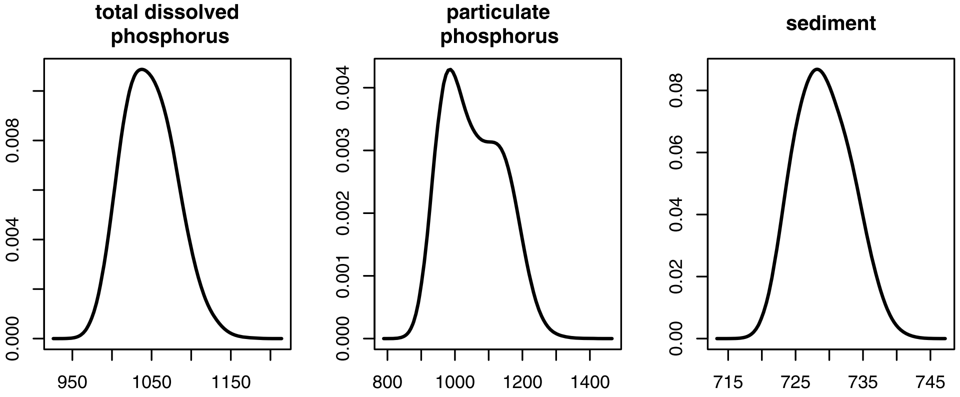

Figure 3. Posterior densities of three model outputs.

surface for each of these output variables. These RBF surfaces were evaluated at MCMC samples from the posterior of $\boldsymbol{\beta}$ to obtain samples from the posteriors of the model outputs.

Figure 3 shows the estimated posterior distributions of annual amounts of dissolved phosphorus, particulate phosphorus, and sediment transported out of the watershed. The particulate phosphorus posterior is not a symmetric function, which demonstrates that the methodology can be applied to irregularly shaped function. Note that uncertainty about particulate phosphorous is greater than for the dissolved phosphorous. This is expected since the biogeochemical and physical processes involved in the transport of particulate phosphorous are considerably more complex than for the transport of the dissolved phosphorous.

5. COMPUTATIONAL REQUIREMENTS FOR SOARS VERSUS CONVENTIONAL MCMC

SOARS was developed to enable uncertainty analysis for complex models that are too computationally intensive for conventional MCMC or related uncertainty methods. This section summarizes information in other sections on the distribution of a computational budget.

For the Town Brook problem we used 1900 exact simulations for the optimization search using DDS. We needed 500 more exact simulations to compute the LHCS, using the RBF emulator from the optimization points to locate the LHCS in the HPDR. We then needed additional 1017 exact simulations during the 468 GRIMA iterations. Hence, SOARS required a total of 3517 exact simulations. In contrast, a conventional MCMC sample would have required tens of thousands of exact simulations.

In this problem, the ratio of CPU time for conventional MCMC to SOARS is $(80,000)T/\ (3517\,T + \Phi)$, where Φ is the computational time for calculation done on the emulator. Φ is independent of the time T for the simulator. For costly simulation model, $T \gg \Phi$. So the ratio would be high.

5.1. Comparison of Posterior Densities Computed With SOARS and Conventional MCMC

As a computationally feasible alternative to SOARS, one could generate *fewer* MCMC runs using the *exact* posterior. To see how this works, we took 60,000 MCMC samples from the *exact* posterior after 20,000 MCMC samples as a burn-in period and extracted subsample of the first 3500 from the 60,000 draws. The size 3500 sample used nearly the same number of expensive simulator evaluations as SOARS, if one ignores the burn-in. (SOARS uses the emulator, not the simulator, for burn-in, so including burn-in would make the following comparisons more favorable to SOARS.) We plotted the marginal posterior densities estimates from these samples, as well from the 60,000 MCMC runs from the SOARS emulator, in Figure 4. The estimates from 60,000 runs from the exact posterior are taken as a "gold standard" and plotted as thick solid reference lines. We see that SOARS provides more accurate estimates of the marginal posterior densities of $\beta_1, \ldots, \beta_8$ than an MCMC sample from the exact posterior using the same number of expensive simulator evaluations. In particular, for the important phosphorus-related parameters, $\beta_5, \ldots, \beta_8$, especially for the last three, 3500 simulations from the exact posterior *underestimate* uncertainty. This can be seen in the density estimates which are narrower and more pointed than the estimates from either SOARS or 60,000 runs from the exact posterior.

6. LIMITATIONS OF SOARS

The main limitation of SOARS is the number of simulator evaluations needed for higher dimensional problems, especially when using a simulator such as SWAT that has nonsmooth output. For the 8-dimensional SWAT study here, thousands of simulator evaluations were needed. However, when SWAT was used for a 4-dimensional problem, the number of simulator evaluations was only in the hundreds (Bliznyuk, Ruppert, and Shoemaker 2012).

More than half of the simulator evaluations are needed for calibration alone, even using state-of-the-art optimization software such as DDS here or CONDOR (Vanden Berghen and Bersini 2005) which was used by Bliznyuk, Ruppert, and Shoemaker (2012). In addition, uncertainty analysis using only the simulator evaluations from optimization can be quite inaccurate; see Bliznyuk, Ruppert, and Shoemaker (2012) or Figure 8 of the Supplemental Material. Therefore, we see no way to reduce the number of simulator evaluations.

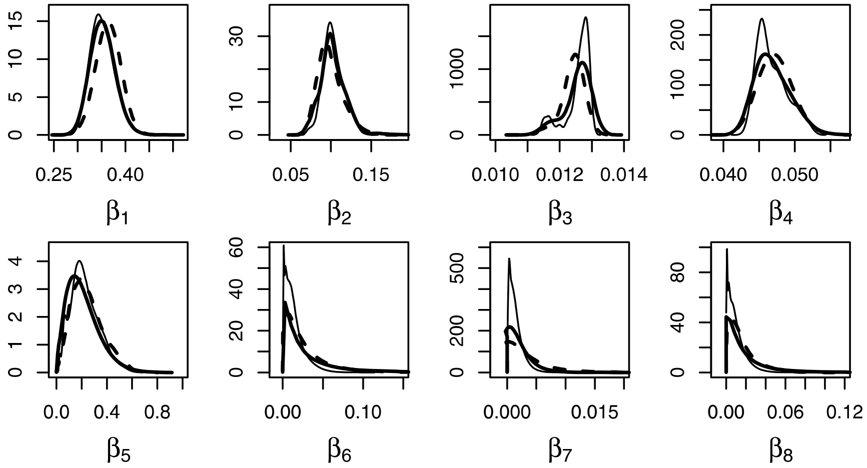

Figure 4. Kernel estimates of the marginal densities of β_k's. The SOARS results from 3517 evaluations on the exact posterior (to construct the emulator) and 60,000 evaluations of the emulator-based approximate posterior (for MCMC sampling) are given by the dashed line. The non-SOARS results from 60,000 evaluations on the exact posterior (for MCMC sampling) is in heavy solid line and from 3500 evaluations on the exact surface (again for MCMC sampling) is in thin solid line.

Fortunately, SOARS can be parallelized. DDS and Stochastic RBF (Singh 2011; Regis and Shoemaker 2009) are examples of efficient global optimization methods that can be parallelized. LHCS is easily implemented in parallel. GRIMA can also be parallelized. As mentioned in Section 2.3, the emulator is updated only after at least M new evaluation points are selected. The simulator can be run simultaneously on all of these points. GRIMA worked well on our example with $M = 12$ and it is likely that it will work adequately for larger values of M.

A larger watershed model might take one hour to run. With a moderate parallelization speedup by a factor of 12.5, the simulator could be evaluated at 300 points per day or 12 days for 3600 evaluations. A 12-day run is not convenient, but it is feasible.

In our work, we have used sufficient evaluations of the simulator so that the RBF approximation to the log-posterior is, for all intents and purposes, error-free. For higher dimensional problems and simulators that are especially computationally expensive, one might need to settle for less accuracy. Trading off between the accuracy of the emulator and the computation time is a topic well worth exploring.

7. SUMMARY AND CONCLUSIONS

We have extended the application of SOARS to watershed research beyond that presented previously and improved computational efficiency for this more difficult problem by adding an LHCS between optimization and GRIMA. SOARS performed very well for a simulator with up to eight parameters and two model outputs. A model inadequacy function where the bias is a function of the output, improved the predictive performance of the model. In other applications, the model inadequacy function could be independent of the output and depend on time, date, or other covariates. Calibration and uncertainty analysis with SOARS on the Town Brook problem required less than twice the evaluations as calibration alone, since SOARS used a total of 3517 simulator evaluations of which 1900 (or 54 %) were used for optimization. Town Brook uncertainty analysis done with MCMC on the exact SWAT simulator required more than twenty times of CPU time than with SOARS. Because of computational demands, models with many parameters would benefit from parallel processing. Fortunately, SOARS is very suitable for parallel implementations.

SUPPLEMENTAL MATERIAL

The supplemental materials file available online contains the following items:

Radial Basis Function Approximation
Checking MCMC Convergence and the Effective MCMC Sample Size
Sensitivity to the Prior
Checking the fit of the noise model
Analysis of MCMC Output
Integrating Σ_u Out of the Posterior
Supplemental Table: Table giving the physical meanings and ranges of the variable parameters of the Town Brook SWAT simulator.
Supplemental Figures: Figures giving additional information about the Town Brook study.

ACKNOWLEDGEMENTS

This research was supported by NSF Grants DMS-0434390 (Ruppert, Shoemaker), DMS-0805975 (Ruppert), EAR 0711491 (Shoemaker), CBET 075675 (Shoemaker). Assistance with the SWAT model was provided by Josh Woodbury. The authors thank the referee and the associate editor whose suggestions led to major improvements in the paper.

[Received January 2011. Accepted March 2012. Published Online May 2012.]

REFERENCES

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.

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

Bayarri, M. J., Berger, J. O., Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, C.-H., and Tu, J. (2007a), "A Framework for Validation of Computer Models," *Technometrics*, 49, 138–154.

Bayarri, M. J., Berger, J. O., Cafeo, J., Garcia-Donato, G., Liu, F., Palomo, J., Parthasarathy, R. J., Paulo, R., Sacks, J., and Walsh, D. (2007b), "Computer Model Validation with Functional Output," *The Annals of Statistics*, 35, 1874–1906.

Bliznyuk, N., Ruppert, D., and Shoemaker, C. A. (2011), "Bayesian Inference Using Efficient Interpolation of Computationally Expensive Densities With Variable Parameter Costs," *Journal of Computational and Graphical Statistics*, 20, 636–655.

——— (2012), "Local Derivative-Free Approximation of Computationally Expensive Posterior Densities," *Journal of Computational and Graphical Statistics*. doi:10.1080/10618600.2012.681255.

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 and Graphical Statistics*, 17, 270–294.

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

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

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

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

Cumming, J. A., and Goldstein, M. (2009), "Small Sample Bayesian Designs for Complex High-Dimensional Models Based on Information Gained Using Fast Approximations," *Journal of the American Statistical Association*, 51, 377–388.

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.

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.

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

Higdon, D., Kennedy, M., Cavendish, J. C., Cafeo, J. A., and Ryne, R. D. (2004), "Combining Field Data and Computer Simulations for Calibration and Prediction," *SIAM Journal of Scientific Computation*, 26, 448–466.

Kennedy, M. C., and O'Hagan, A. (2001), "Bayesian Calibration of Computer Models," *Journal of the Royal Statistical Society, Series B*, 63, 425–464.

Levy, S., and Steinberg, D. M. (2010), "Computer Experiments: A Review," *AStA Advances in Statistical Analysis*, 94, 311–324.

Qian, P. Z. G. (2009), "Nested Latin Hypercube Design," *Biometrika*, 96, 957–970.

Qian, P. Z. G., and Wu, C. F. J. (2008), "Bayesian Hierarchical Modeling for Integrating Low-Accuracy and High-Accuracy Experiments," *Technometrics*, 50, 192–204.

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.

Regis, R. G., and Shoemaker, C. A. (2009), "Parallel Stochastic Global Optimization Using Radial Basis Functions," *INFORMS Journal on Computing*, 21, 411–426.

Ruppert, D., Wand, M. P., and Carroll, R. J. (2003), *Semiparametric Regression*, Cambridge: Cambridge University Press.

Santner, T. J., Williams, B. J., and Notz, W. I. (2010), *The Design and Analysis of Computer Experiments*, New York: Springer.

Shoemaker, C. A., Regis, R., and Fleming, R. (2007), "Watershed Calibration Using Multistart Local Optimization and Evolutionary Optimization With Radial Basis Function Approximation," *Journal of Hydrologic Science*, 52, 450–465.

Singh, A. (2011), "Global Optimization of Computationally Expensive Hydrologic Simulation Models," Ph.D. Thesis in Civil and Environmental Engineering, Cornell University.

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

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

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

Tolson, B. (2005), "Automatic Calibration, Management and Uncertainty Analysis: Phosphorus Transport in the Cannonsville Watershed, " Ph.D. dissertation, School of Civil and Environmental Engineering, Cornell University.

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*.

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

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

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.

Wild, S. M., and Shoemaker, C. A. (2011), "Global Convergence of Radial Basis Functions Trust Region Derivative-Free Algorithms," *SIAM Journal on Optimization*, 20, 387–415.

Wild, S. M., Regis, R. G., and Shoemaker, C. A. (2007), "ORBIT: Optimization by Radial Basis Function Interpolation in Trust-Regions," *SIAM Journal on Scientific Computing*, 30, 3197–3219.