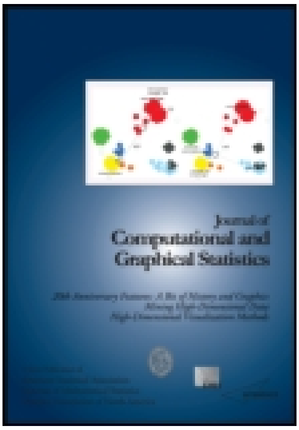

Journal of Computational and Graphical Statistics

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/ucgs20

Local Derivative-Free Approximation of Computationally Expensive Posterior Densities

Nikolay Bliznyuk [a], David Ruppert [b] & Christine A. Shoemaker [c]

[a] Department of Statistics, University of Florida, Gainesville, FL, 32611

[b] School of Operations Research and Information Engineering, Cornell University, Ithaca, NY, 14853

[c] School of Civil and Environmental Engineering, and School of Operations Research and Information Engineering, Cornell University, Ithaca, NY, 14853

Available online: 14 Jun 2012

To cite this article: Nikolay Bliznyuk, David Ruppert & Christine A. Shoemaker (2012): Local Derivative-Free Approximation of Computationally Expensive Posterior Densities, Journal of Computational and Graphical Statistics, 21:2, 476-495

To link to this article: http://dx.doi.org/10.1080/10618600.2012.681255

Local Derivative-Free Approximation of Computationally Expensive Posterior Densities

Nikolay BLIZNYUK, David RUPPERT, and Christine A. SHOEMAKER

Bayesian inference using Markov chain Monte Carlo (MCMC) is computationally prohibitive when the posterior density of interest, π, is computationally expensive to evaluate. We develop a derivative-free algorithm GRIMA to accurately approximate π by interpolation over its high-probability density (HPD) region, which is initially unknown. Our local approach reduces the waste of computational budget on approximation of π in the low-probability region, which is inherent in global experimental designs. However, estimation of the HPD region is nontrivial when derivatives of π are not available or are not informative about the shape of the HPD region. Without relying on derivatives, GRIMA iterates (a) sequential knot selection over the estimated HPD region of π to refine the surrogate posterior and (b) re-estimation of the HPD region using an MCMC sample from the updated surrogate density, which is inexpensive to obtain. GRIMA is applicable to approximation of general unnormalized posterior densities. To determine the range of tractable problem dimensions, we conduct simulation experiments on test densities with linear and nonlinear component-wise dependence, skewness, kurtosis and multimodality. Subsequently, we use GRIMA in a case study to calibrate a computationally intensive nonlinear regression model to real data from the Town Brook watershed. Supplemental materials for this article are available online.

Key Words: Bayesian calibration; Computer experiments; Groundwater modeling; Inverse problems; Markov chain Monte Carlo; Radial basis functions; Uncertainty analysis.

1. INTRODUCTION

The rapid development of Bayesian statistics over the past few decades is, undoubtedly, due to the development of Markov chain Monte Carlo (MCMC) techniques, made feasible by the speed of modern computers. Nonetheless, when the posterior density π is computationally expensive to evaluate, long MCMC runs are impractical and shorter runs cause unacceptably high estimation error. To remedy this problem, a number of authors have

Nikolay Bliznyuk is Assistant Professor, Department of Statistics, University of Florida, Gainesville, FL 32611 (E-mail: *nbliznyuk@ufl.edu*). David Ruppert is Andrew Schultz Jr. Professor of Engineering and Professor of Statistical Science, School of Operations Research and Information Engineering, Cornell University, Ithaca, NY 14853 (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, Ithaca, NY 14853 (E-mail: *cas12@cornell.edu*).

Journal of Computational and Graphical Statistics, Volume 21, Number 2, Pages 476–495
DOI: 10.1080/10618600.2012.681255

developed approximations to π: see Kennedy and O'Hagan (2001), Rasmussen (2003), Bliznyuk et al. (2008), and the references therein. An accurate approximate posterior density $\widetilde{\pi}$ can typically be constructed by interpolation using only a few hundred evaluations of π. Then, since $\widetilde{\pi}$ is computationally inexpensive, long MCMC runs can be obtained quickly using $\widetilde{\pi}$ in place of π.

In this article, π can be any posterior density, possibly unnormalized. It is assumed that π can be evaluated exactly at any point in the parameter space, but that π is expensive to compute. Thus, our setup is different from that of a density estimation problem, where a sample from a given density is available, but the density itself cannot be evaluated. We propose a new method for approximating π motivated by our work on the calibration of models for watershed management. The primary problem is to obtain an accurate approximant (interpolant) of π with a minimal number of evaluations of π, for example, only several hundreds. Global approximation of π over the whole parameter space $\mathfrak{C}$ of η is computationally wasteful since the volume of $\mathfrak{C}$ typically exceeds that of the high-probability density (HPD) region by orders of magnitude. Therefore, the key to an efficient approximation algorithm is the restriction of function evaluations of π to the HPD. This restriction is difficult because the location and shape of the HPD region are initially unknown. The problem of characterizing the HPD region is further complicated by the potential nonsmoothness of π, which makes existing approximation approaches that assume derivatives (Rasmussen 2003; Bliznyuk et al. 2008) unattractive. As we discuss later in the context of calibration problems, this roughness is often the effect of discretizations in the computer code model for the physical process (Benaman, Shoemaker, and Haith 2005; Mugunthan, Shoemaker, and Regis 2005; Mugunthan and Shoemaker 2006; Shoemaker, Regis, and Fleming 2007).

The local procedure we propose is a derivative-free extension of the approach of Bliznyuk et al. (2008), the only existing work that explicitly attempts to restrict the evaluation of π to the HPD region. Information about π is collected only at "design points"—the points in the parameter space where π is evaluated. First, the location of the HPD region is obtained using a derivative-free optimization algorithm that finds the mode of π, at least approximately. Conventional optimization routines typically produce a set of design points that is insufficient to characterize the shape of the HPD region. Therefore, to further explore the HPD region after optimization, our algorithm alternates between two steps: (a) determination of an approximate HPD region, which will be called the "design region," using a cheap-to-obtain MCMC sample from the surrogate density $\widetilde{\pi}$ and (b) selection of additional design points in the design region to improve the approximation $\widetilde{\pi}$ to π. This typically entails enlarging ("growing") the initial design region during early iterations of the algorithm and improvement of the approximation during the later iterations. Our algorithm is called GRIMA—"Grow the (design) Region and IMprove the Approximation." The algorithm is terminated when a discrepancy measure, for example, the total variation (TV) norm, between consecutive approximate densities becomes negligible.

Of particular interest to us are posterior densities that arise in computationally intensive nonlinear regression problems. In such problems, the nonlinear regression function is the output of a complex computer code f, known as *simulator*. For example, in the watershed modeling problems analyzed by Shoemaker, Regis, and Fleming (2007) and Tolson and

Shoemaker (2007a,b), for a given value of input parameter vector β, a single run of the simulator produces a vector $f(\beta)$ of time series of daily average water flows. In the simplest case, the vector of observed data Y is modeled as $Y = f(\beta) + e$, where f is used as a nonlinear regression function and e is the vector of errors whose distribution depends on additional parameters ζ. Evaluation of f can take from several seconds to a few hours depending on the application. Once f has been evaluated at a given value of β, the likelihood is obtained by substituting $Y - f(\beta)$ into the density of e. Then, π is obtained as usual by multiplying this likelihood by the prior density.

Because of its generality, our algorithm can be applied without modifications to approximation of posterior densities in contexts other than nonlinear regression. For example, in our current work, we use GRIMA to approximate the posterior density for parameters in a high-dimensional linear model with space–time dependence (Bliznyuk, Ruppert, and Shoemaker 2011). GRIMA only requires that one can evaluate the unnormalized posterior density for a given parameter value; closed-form specification of the (normalized) posterior density is not necessary. Even though we are using radial basis functions (RBFs) for approximation without assuming a particular structure on $\log(\pi)$, GRIMA is not restricted to this class of interpolants. In the online Appendix A.6, we discuss alternatives that are also applicable to inverse problems. We assume that the posterior density to be approximated is unimodal or multimodal but with a (topologically) connected high-probability region. In Section 5, we discuss how the case of disconnected modes may be addressed within our framework.

The article is structured as follows. We describe the principles behind the density approximation by interpolation and present the details of the GRIMA algorithm in Section 2. We conduct simulation experiments on test densities with varying problem dimensions, linear and nonlinear component-wise dependence, skewness, kurtosis and multimodality, and report the results in Section 3. In Section 4, we develop a statistical model for stream flows in the Town Brook watershed in upstate New York and apply GRIMA to estimate the posterior density of the simulator parameters. Possible extensions are briefly outlined in Section 5. The online Appendices provide information necessary for efficient implementation of the procedures discussed in this article, additional summaries of our simulations, and an illustration of GRIMA on a two-dimensional test problem of Rasmussen (2003), which enables the reader to visualize the progress of GRIMA without the need to follow all of the technical details of Section 2.

2. THE GRIMA ALGORITHM FOR DENSITY APPROXIMATION

Since the main ideas behind the GRIMA algorithm are simple but the details are more complicated, it is worthwhile to first present an overview of this algorithm. Therefore, prior to presenting the GRIMA algorithm in Section 2.4, we state the main ideas behind the algorithm and summarize our experience in approximating probability densities by interpolants in Sections 2.1–2.2. Precise definitions and informal justification of the procedure are provided in Section 2.3. Details of practical implementation can be found in the online Appendix A.1.

As we noted in Section 1, applicability of our approximation algorithm is not limited to nonlinear regression problems. For this reason, the exposition below treats π as an arbitrary unnormalized density.

2.1 CONSTRUCTING APPROXIMATE POSTERIOR DENSITIES BY INTERPOLATION

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

Our approach makes no smoothness assumptions about the exact posterior, except that it can be accurately approximated by a smooth RBF interpolant. This is usually a reasonable assumption. For example, simulator output often has small jump discontinuities due to discretizations in the algorithm. An RBF interpolant will smooth these out, but this may be beneficial since the jumps are only artifacts of the code.

As in Bliznyuk et al. (2008) and in Rasmussen (2003), the approximation to the logarithm of π is based on an interpolant at design points, or knots, where π has been evaluated and is a form of universal kriging with generalized covariance functions (Cressie 1991). As in the former article, however, our focus is on interpolation using RBFs, but any other class of interpolants can be used, as discussed in the online Appendix A.6.

Before describing GRIMA, it is instructive to outline the challenges of the density interpolation problem and how GRIMA addresses these difficulties:

(1) Design points should be chosen in the region of high probability under the density π. Knots chosen far away from the HPD region often cause the approximation over the HPD region to deteriorate. Our algorithm accomplishes this through an *optimization stage* prior to GRIMA, which locates the HPD region, and through GRIMA's sequential design, which restricts evaluation of π to the estimated HPD region.

(2) Since the approximation $\widetilde{\pi}$ is less reliable away from design points, the support of the approximate density must be restricted to a neighborhood of the set of design points.

(3) Our experiments indicate that the quality of interpolation by radially symmetric functions (such as RBFs) is sensitive to the parameterization of π. Accurate approximation of π typically requires a significantly greater number of design points if there are directions about the mode(s) along which $\log(\pi)$ changes much more rapidly than along other directions. For example, if $\log(\pi)$ is approximately a negative definite quadratic function, then a higher condition number of its Hessian at the mode requires a greater number of design points. To remedy this problem, GRIMA rescales the parameter vector, as discussed in the online Appendix A.1.5.

(4) Enforcing separation between design points is highly desirable because placing a new design point near an existing one provides little new information about π. In the optimization stage prior to GRIMA, we use a derivative-free algorithm, such as CONDOR (Vanden Berghen and Bersini 2005), that avoids clusters of nearby function evaluations. In GRIMA, new design points are added only if they are sufficiently far from existing design points.

2.2 Approximation of HPD Regions for Unnormalized Probability Densities

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

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

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

If $\widehat{c}(\alpha)$ is an estimator for $c(\alpha)$, then the set $\widehat{C}_R(\alpha) := \{\eta : \pi(\eta) \geq \widehat{c}(\alpha)\}$ is an approximation to $C_R(\alpha)$. For example, if $\eta^{(1)}, \ldots, \eta^{(k)}$ is a sample from π, then the αth sample quantile of $\pi(\eta^{(1)}), \ldots, \pi(\eta^{(k)})$ approximates $c(\alpha)$. Furthermore, if $\widetilde{\pi}$ is close to π, for example, with respect to the total variation norm, and $\widetilde{\eta}^{(1)}, \ldots, \widetilde{\eta}^{(k)}$ is a sample from $\widetilde{\pi}$, then it is possible to approximate $c(\alpha)$ by the αth sample quantile of $\widetilde{\pi}(\widetilde{\eta}^{(1)}), \ldots, \widetilde{\pi}(\widetilde{\eta}^{(k)})$.

As we discussed earlier, it is wasteful to evaluate π in regions of very low posterior probability. Consequently, estimation of $c(\alpha)$ using an MCMC sample from a cheap-to-evaluate surrogate density is a key step of our algorithm. It provides an estimate of the minimum height of $\log(\pi)$ over $C_R(\alpha)$, which is used to select new design points to update the surrogate density.

2.3 Notation and Definitions

This subsection collects all relevant notation in one place for easy reference.

All variables are assumed to be (column) vectors or matrices of size specified in the appropriate definition; this will *not* be emphasized by bold-face notation. The notation $\|\cdot\|$ will refer to a generalized Euclidean norm, defined as $\|x\|_A = \sqrt{x^\mathsf{T} A x}$ for a column vector x and a positive definite matrix A. To emphasize that one has a choice of the scaling matrix A to define the norm, discussed below, we omit the subscript A. The distance between a point x and a set $\mathcal{S}$ is $\operatorname{dist}(x, \mathcal{S}) = \inf_{x' \in \mathcal{S}} \|x - x'\|$. Only when applied to a vector, a single subscript extracts components, for example, x_i is the ith component of x. For sets $\mathcal{S}_1$ and $\mathcal{S}_2$, $\mathcal{S}_1 \backslash \mathcal{S}_2$ is the set of elements of $\mathcal{S}_1$ not in $\mathcal{S}_2$, and $|\mathcal{S}_1|$ is the number of elements in $\mathcal{S}_1$. $\mathbb{I}(\cdot)$ is the indicator function. As before, π is the exact unnormalized posterior density on the parameter space $\mathfrak{E} \subset \mathbb{R}^{\dim(\eta)}$.

To approximate the exact posterior density π, we let $\widetilde{l}_i$ be an RBF interpolant of $l := \log(\pi)$ at the set of design points $\mathcal{D}_i$ for which the value of π is known at the ith iteration of GRIMA. Here, the design points serve as the set of knots of the RBF interpolant, defined as

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

where ϕ is a basis function, p is a low-order polynomial, and $\{\eta^{(j)}\}$ are the knots at which $\widetilde{l}_i$ interpolates l. In the online Appendix A.1.5, we discuss the choice of the generalized Euclidean norm determining the scale for RBF fitting. For details of fitting, see the online Appendix A.3 or appendix A.3 of Bliznyuk et al. (2008).

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

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

where, for a positive coverage radius $r > 0$, discussed later in the article,

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

is a neighborhood of $\mathcal{D}_i$. Thus, the approximation is restricted to a neighborhood of design points because, as noted in Section 2.1, remarks (1) and (2), the approximation is less accurate outside of this neighborhood, and knot selection far from this region can adversely influence the quality of approximation over the HPD region. As the design region grows during the iteration of GRIMA, $\mathcal{N}_i$ should eventually cover the entire HPD region so that the restriction of $\widetilde{\pi}_i$ to $\mathcal{N}_i$ will be satisfactory.

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

The algorithm of this section, stated formally as Algorithm 1, is a derivative-free alternative to the exploratory stage of GPHMC algorithm of Rasmussen (2003) or to steps 2 and 3 of the procedure of Bliznyuk et al. (2008).

2.4.1 Initialization by Optimization. GRIMA should be started in the HPD region. Therefore, we assume that the maximum a posteriori (MAP) estimator $\widehat{\eta}$ for η has been found, at least approximately, in the *optimization stage* prior to running GRIMA. We use information from the optimization run to initialize GRIMA.

Maximization of $l := \log(\pi)$ by derivative-free optimization routines produces a set of widely separated design points $\mathcal{D}_{\text{OPT}}$. We retain a subset $\mathcal{D}_0$ of $\mathcal{D}_{\text{OPT}}$, by discarding design points at which the values of l are “extremely low,” and obtain an interpolant $\widetilde{l}_0$ of l at $\mathcal{D}_0$, as in Equation (2.2). Choosing the starting radius r in Equation (2.4) in such a way that the neighborhood $\mathcal{N}_0$ of $\mathcal{D}_0$ is connected (since the true HPD is assumed to be), we define $\widetilde{\pi}_0$ by means of Equation (2.3).

2.4.2 Sequential Design. At the ith iteration of GRIMA, an MCMC sample from the cheap-to-evaluate $\widetilde{\pi}_i$ is obtained and is used to determine the lower bound on $\widetilde{l}_i$ over the HPD region for $\widetilde{\pi}_i$ of size $(1-\alpha)$ (lines 4 and 5 of Algorithm 1). Subsequently, the surrogate log-posterior $\widetilde{l}_i$ is maximized over the boundary of $\mathcal{N}_i$ [defined by Equation (2.4)] and if the value of $\widetilde{l}_i$ is sufficiently high at the maximizer η^*, the true expensive log-posterior l is evaluated at η^* and the RBF interpolant $\widetilde{l}_i$ is updated (lines 6–16). By allowing the coverage radius r to shrink or to increase depending on the magnitude of $\widetilde{l}_i(\eta^*)$ (lines 17–21), the GRIMA algorithm attempts to choose candidate points η^* (which are on the boundary of $\mathcal{N}_i$) as far as possible from $\mathcal{D}_i$, provided that $\widetilde{l}_i(\eta^*)$ is above the threshold $\widetilde{c}_i(\alpha) - \delta$. When r grows, the approximate HPD region tends to cover the true HPD region faster, for example, with fewer design points. When r shrinks, the set of design points becomes denser and the approximation to π over the HPD region tends to improve. GRIMA automatically adjusts r (see lines 17–21 and the end of the online Appendix A.1.2). The algorithm is allowed to terminate before exhausting the computational budget if extra evaluations of π do not improve the quality of approximation significantly as judged from diagnostics (lines 22–27).

Algorithm 1 GRIMA

```
Require: Tuning parameters $r > 0, \delta > 0, \alpha \in (0,1), \rho \in (0,1), J, T$ as specified below
    (see text for recommendations)
Require: $i = 0, \mathcal{D}_0, \mathcal{N}_0, \widetilde{l}_0, \widetilde{\pi}_0$ as defined above (obtained in the *optimization stage*)
 1: while computational budget has not been exceeded do
 2:     $i \leftarrow i + 1, \mathcal{D}_i \leftarrow \mathcal{D}_{i-1}, \widetilde{l}_i \leftarrow \widetilde{l}_{i-1}$
 3:     set $\mathcal{N}_i$ as in Equation (2.4) and $\widetilde{\pi}_i$ as in Equation (2.3)
 4:     obtain an MCMC sample $\mathcal{M}_i$ from $\widetilde{\pi}_i$ of length $T$
 5:     find $\widetilde{c}_i(\alpha)$, the $\alpha$th sample quantile of the sample $\{\widetilde{l}_i(\eta) : \eta \in \mathcal{M}_i\}$
 6:     for $j = 1, \ldots, J$ do
 7:         $\mathcal{C} \leftarrow \{\eta \in \mathcal{N}_i : \mathrm{dist}(\eta, \mathcal{D}_i) = r\}$
 8:         find $\eta^* \in \mathcal{C}$ such that $\widetilde{l}_i(\eta^*) \approx \max_{\eta \in \mathcal{C}} \widetilde{l}_i(\eta)$
 9:         if $\widetilde{l}_i(\eta^*) \geq \widetilde{c}_i(\alpha) - \delta$ then
10:             $\mathcal{D}_i \leftarrow \mathcal{D}_i \cup \{\eta^*\}$
11:             evaluate $l$ at $\eta^*$
12:             update $\widetilde{l}_i$ so that it interpolates $l$ at $\mathcal{D}_i$ (in particular, at $\eta^*$)
13:         else
14:             set $j \leftarrow j - 1$ and break the for loop
15:         end if
16:     end for
17:     if $j < J$ then
18:         set $r \leftarrow \rho \cdot r$
19:     else
20:         set $r \leftarrow \rho^{-1} \cdot r$
21:     end if
22:     if diagnostics suggest that sequence $\{\widetilde{\pi}_k\}_{k=1}^{i}$ has "practically" converged then
23:         break the while loop
24:     else
25:         re-estimate scaling matrix and parameters of the MCMC sampler
26:         (optional) choose extra knots $\propto$ "probability content" of $\mathcal{N}_i$ and update $\widetilde{l}_i$
27:     end if
28: end while
29: return $\widetilde{\pi} \leftarrow \widetilde{\pi}_i$
```

2.4.3 Justification. We now justify Algorithm 1 under the simplifying assumptions below. Assume no rescaling is performed (line 25), so that the design region $\mathcal{N}$ in GRIMA is the union of balls of radius r, each centered at a point from the finite set of knots $\mathcal{D}$ lying in the bounded parameter space $\mathfrak{C}$. Let $\widetilde{l}_{\mathcal{D}}$ be the approximate log-posterior surface that interpolates at $\mathcal{D}$ the exact surface l, assumed continuous. Assume that $\|\widetilde{l}_{\mathcal{D}} - l\|_\infty$ is a nondecreasing function of the minimax distance $\sup_{\eta \in \mathfrak{C}} \mathrm{dist}(\eta, \mathcal{D})$. Under the above assumptions, the condition of line 9 implies that, for all sufficiently large values of δ, $\mathcal{N}$ will eventually contain the true connected HPD region. Every point from the true HPD region of size $(1 - \alpha)$ would be added as an element of a neighborhood of some knot in $\mathcal{D}$, for some coverage radius r.

By construction, r is the maximin distance in $\mathcal{D}$. For any fixed value of r, there are finitely many knots that can be added—those that satisfy the condition of line 9. Therefore, as the number of knots grows, r must shrink (line 18). Hence, the resulting design is space-filling (becomes dense) in the true HPD region.

3. SIMULATION EXPERIMENTS

In this section, we conduct a large simulation study using a collection of cheap-to-evaluate densities, from which a long sample can be obtained efficiently for reference purposes. Each of these test densities is treated as a "black-box" (target) posterior density for η, from which an MCMC sample is desired. Since GRIMA approximates the posterior density using interpolation, the approximation quality is adversely affected by the curse of dimensionality. Our primary interest is to determine the range of problem dimensions that are tractable within our framework. As a reference model, we consider π to be a multivariate normal (MVN) density with linearly dependent components. These densities are of interest because, in many, but, certainly, not all, inverse problems, the assumption of asymptotic normality of the regression function parameters is tenable. We investigate the robustness of GRIMA as the correlation between the components of η increases. We also examine the performance of GRIMA on heavy-tailed or skewed unimodal target densities and on bimodal densities with connected HPD regions. (The case when the HPD region is topologically disconnected is discussed in Section 5.)

For each choice of the test density, defined below, we perform 10 replications of an experiment. Each replication consists of (a) an optimization run from a point chosen uniformly at random from the d-dimensional hypercube with sides $[-10, 10]$, where d is the dimension of η, to reach the high-probability region, followed by (b) a run of GRIMA. In the experiment with multimodal densities, the optimization run is supplemented by two additional runs initialized from the neighborhood of the modes. The replications are performed automatically using default settings of CONDOR, the derivative-free algorithm of our choice, and of GRIMA. There are no guarantees that for every replication, the dominant mode of π has been reached by optimization. GRIMA is terminated whenever the TV discrepancy for every η_i is less than 0.05, which corresponds to a very accurate approximation. Because in each replication the numbers of intermediate MCMC steps in GRIMA add up to millions, and often to tens of millions of Markov chain transitions, only a modest number of replications is computationally feasible for each experimental setting. Even then, our simulation experiments took a few thousand hours of processor time.

The multivariate normal reference model is suggested by the work of Rasmussen (2003), who illustrated his procedure on a 10-dimensional MVN$(0,\Sigma)$ density, for which the variance associated with the eigenvector $(1, 1, \ldots, 1)/\sqrt{10}$ is 100 times greater than along the remaining eigenvectors. We generalize this test problem to allow the problem dimension d and the condition number of Σ, κ, to vary. Let $\Sigma(d, \kappa)$ be the correlation matrix proportional to a matrix, for which κ is the eigenvalue associated with the eigenvector $(1, \dots, 1)/\sqrt{d}$ and the remaining eigenvalues are all equal to 1. It is possible to express the ijth entry of $\Sigma(d, \kappa)$ analytically as $\{(\kappa - 1)/d + \mathbb{I}(i = j)\}/\{(\kappa - 1)/d + 1\}$, where $\mathbb{I}(\cdot)$ is the indicator function. We normalize Σ so that the marginal densities under the reference model are all standard normal, irrespective of the values of d and κ, so that we can keep the above box constraints in optimization the same for all test problems.

In our first simulation study (Experiment 1), we assess performance of GRIMA on elliptically symmetric densities from the multivariate Student's t family. Here, a logarithm of the unnormalized posterior density π equals $-0.5(\nu + d)\log(1 + \eta^{\mathsf{T}}\Sigma^{-1}\eta/\nu)$. The scale matrix $\Sigma = \Sigma(d, \kappa)$ is as defined earlier. In the experiments, $\kappa = 25, 100$ and $d = 2, 6, 10$, which all result in high positive component-wise correlations, whenever defined. For example, for $d = 10$ and $\kappa = 25, 100$, the off-diagonal entries of Σ are $0.71, 0.91$. Apart from the multivariate normal case when ν approaches infinity, we consider the important case when $\nu = 2$, so the second moment of η is infinite and, consequently, a naïve moment estimator of the scale matrix Σ from a sample from the exact posterior density is inconsistent. It is remarkable that this complication does not affect GRIMA because it uses samples from the approximate posterior densities restricted to the high-probability region for estimation of the scale matrix.

The results of the experiment are presented in Table 1. A considerable proportion of the function evaluations during the optimization stage is reused for the initialization of `GRIMA`. As expected, the computational load—as measured by the number of knots necessary for accurate approximation—increases with the problem dimension. Somewhat unexpectedly, the effect of the condition number κ appears to be negligible. We attribute this to the fact that GRIMA rescales η using the samples from the intermediate approximate densities, and it appears that a very accurate approximation to π is not necessary for this rescaling to be effective. Also, both in the optimization and in the approximation stages, the multivariate Student's t density requires roughly 50% more knots than the corresponding MVN reference density.

Our second experiment (Experiment 2) investigates the impact of skewness and nonlinear dependence. Skewed densities often arise in engineering problems when the solution of interest to an inverse problem lies at or near the boundary of the parameter space. We take π to be a multivariate skew-normal density in the "transformation method" parameterization, defined in section 2.1 of Azzalini and Dalla Valle (1996), as follows: If Z is a MVN(0,Σ) random vector and Z_0 is an independent standard normal random variable, then π is the density of η, where for $i = 1, \dots, d$, $\eta_i = D_i|Z_0| + \sqrt{1 - D_i^2}Z_i$ and the parameters $D_i \in (-1, 1)$ control the amount of skewness; see Azzalini and Dalla Valle (1996) for the expression of the density. In this experiment, we focus on the even values of the dimension $d = 2, 6, 10$ and set $D_1, \dots, D_{d/2}$ to 0 and $D_{d/2+1}, \dots, D_d$ to the same value $\xi \in \{0.5, 0.9\}$, which correspond to moderate and high skewness. The scale matrix is defined as $\Sigma = \Sigma(d, \kappa)$ for $\kappa = 25, 100$.

In Table 2, we summarize the results of the second experiment. Because no linear transformation of η removes dependence or brings the density of the transformed η closer

Table 1. Results of Experiment 1 of Section 3 (multivariate normal and *t* densities)

		Number of		Knots optimiz.		Knots retained		Knots GRIMA	
dim(η)	df	κ	reps.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev
2	∞	25	10	15.9	1.2	9.3	0.8	29.9	6.2
2	∞	100	10	15.6	2.8	5.5	1.6	33.9	8.0
2	2	25	10	22.3	2.7	12.5	3.4	37.1	13.3
2	2	100	10	29.0	7.7	13.6	6.3	45.5	13.2
6	∞	25	10	99.4	15.2	50.7	8.4	164.7	29.6
6	∞	100	10	98.4	14.2	50.4	8.3	162.7	28.3
6	2	25	10	156.4	23.9	72.3	30.1	258.3	62.8
6	2	100	10	197.6	33.9	72.5	35.3	276.5	60.2
10	∞	25	10	251.9	38.2	129.3	0.5	381.3	52.3
10	∞	100	10	245.7	28.1	129.3	0.7	377.4	45.9
10	2	25	10	370.9	23.5	184.9	23.9	552.9	65.3
10	2	100	10	517.7	86.6	144.2	76.7	628.2	103.8

NOTE: The table shows the mean and standard deviation of the number of knots (a) used in optimization, (b) retained from (a), and (c) those used for interpolation by GRIMA, including those in (b). A set of additional trials based on an earlier version of the algorithm is given in the online Appendices.

to spherical symmetry, GRIMA requires significantly more knots than in the reference MVN model of the first experiment. Increasing the parameter κ now increases the number of knots used in GRIMA. There is also a strong statistical interaction between the dimension d and the skewness parameter ξ.

Our third experiment (Experiment 3) aims at studying the impact of multimodality when the HPD region is topologically connected. Specifically, we work with a mixture of two multivariate normal densities with identity covariance matrices and means 0 and μ, with respective mixture weights equal to 0.6 and 0.4. In the experiments, we consider

Table 2. Results of Experiment 1 of Section 3 (skewed MVN densities)

			Number of	Knots optimiz.		Knots retained		Knots GRIMA	
dim(η)	ξ	κ	reps.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
2	0.5	25	10	20.5	3.2	14.1	1.7	26.1	4.4
2	0.5	100	10	25.5	4.0	17.1	3.5	30.7	10.1
2	0.9	25	10	23.4	3.7	17.2	4.0	34.0	3.3
2	0.9	100	10	27.3	2.8	21.3	3.3	41.4	7.2
6	0.5	25	10	65.0	13.5	32.0	11.4	116.0	30.2
6	0.5	100	10	117.1	22.2	69.6	15.1	165.6	35.2
6	0.9	25	10	96.0	15.1	55.1	10.7	267.5	86.5
6	0.9	100	10	137.7	13.2	86.5	11.4	491.0	150.8
10	0.5	25	10	106.1	17.5	41.4	18.7	285.3	60.4
10	0.5	100	10	224.5	47.1	128.6	30.6	416.7	91.6
10	0.9	25	10	190.9	25.7	105.0	19.7	1166.9	220.8
10	0.9	100	10	256.1	33.7	149.3	39.6	1644.8	376.9

NOTE: The table shows the mean and standard deviation of the numbers of knots (a) used in optimization, (b) retained from (a), and (c) those used for interpolation by GRIMA, including those in (b). A set of additional trials based on an earlier version of the algorithm is given in the online Appendices.

Table 3. Results of Experiment 3 of Section 3 (mixture of MVN densities)

		Number of	Knots optimiz.		Knots retained		Knots GRIMA	
dim(η)	Δ	reps.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
2	3	10	52.8	1.2	40.8	2.0	58.0	3.2
2	4	10	53.1	0.7	39.5	1.8	54.3	4.9
4	3	10	92.8	4.4	55.1	6.6	127.9	21.7
4	4	10	90.0	7.5	49.8	5.9	185.7	43.2
6	3	10	137.0	9.7	74.7	9.5	251.4	56.0
6	4	10	140.5	9.2	76.8	9.9	421.6	127.8
8	3	10	202.1	16.3	103.6	16.1	315.5	17.4
8	4	10	209.8	25.9	107.7	25.1	553.4	254.3
10	3	10	280.0	37.5	136.8	37.7	429.3	33.9
10	4	10	284.1	34.8	132.4	32.1	651.8	240.1

NOTE: The table shows the mean and standard deviation of the numbers of knots (a) used in optimization,(b) retained from (a), and (c) those used for interpolation by GRIMA, including those in (b).

dimensions $d = 2, 4, \ldots, 10$ and $\mu = \Delta u$, where $u = (1, 1, \ldots, 1)/\sqrt{d}$. The parameter Δ controls the distance between the modes; we use $\Delta = 3$ and 4, which respectively correspond to moderate and strong separation between the modes. The outcomes of the experiment, summarized in Table 3, suggest that the densities with more separated modes require a considerably greater number of function evaluations than those with modes closer together.

In Figures A.4–A.12 in the online Appendices, we report several measures of accuracy of the terminal approximations for each test problem replication. These are (a) component-wise TV norm between the exact and the approximate samples and, (b) and (c), squared errors between the exact true and the estimated approximate means and variances. As the effective sample size of the MCMC sample from the approximate posterior density is fairly large (equivalent to roughly 5000 iid variates), the major component in the squared error summaries is the squared bias. Since our main interest is in estimation of marginal densities, the termination criterion of the algorithm is based on the component-wise TV norms. However, the approximate means and variances are also very accurate.

4. CASE STUDY: TOWN BROOK

4.1 Background

The Town Brook watershed is a 37-km^2 subregion of a larger Cannonsville watershed (1200 km^2), located in the Catskills region of New York State. The time series Y of measured flow data used in the analysis consists of 1096 daily observations (from October 1997 to September 2000) of water entering the Delaware River from Town Brook watershed based on readings by the U.S. Geological Survey. The nonlinear regression function is produced by the SWAT2000 simulator (Arnold et al. 1998), which has been used by more than a thousand agencies and academic institutions worldwide in analysis of water flow and nutrient transport in watersheds (e.g., Eckhardt et al. 2002; Grizzetti et al. 2003; Shoemaker, Regis, and Fleming 2007; Tolson and Shoemaker 2007b). The water draining the Town Brook and the rest of the Cannonsville watershed collects in the Cannonsville Reservoir, from which it is piped over a hundred miles to New York City for drinking water. The quality of this drinking water is threatened by phosphorus pollution and, if not protected, could result in the need for a water filtration plant, estimated to cost over \$8 billion. For this economic reason as well as for general environmental concerns, there is great interest in quantifying the parameter uncertainty for this model. In this preliminary study, we model stream flows, which are an important component of the model for phosphorus concentrations. Bayesian inference and uncertainty analysis in a joint model for stream flows and phosphorus concentrations have been studied separately in Ruppert et al. (2012).

The input information of the SWAT2000 computer code for the Town Brook model formulation, subsequently referred to as the Town Brook simulator f, is discussed briefly in Tolson and Shoemaker (2007a), and in more detail in Tolson and Shoemaker (2004, 2007b). Earlier work (Shoemaker, Regis, and Fleming 2007) revealed that the output of f is discontinuous due to discretizations inside SWAT2000. To illustrate GRIMA, we work with a subset of four parameters for which uncertainty assessment is most critical; the rest of the input parameters were fixed at values recommended by a subject matter expert, reported in the online Appendices (Table A.3). The set of calibration variables for the SWAT2000 simulator has been recently expanded in Ruppert et al. (2012) in calibration of a model with multiple outputs (in addition to the flow, of main interest in this article) that used the DDS algorithm of Tolson and Shoemaker (2007a) for global optimization as part of the SOARS approximation framework.

4.2 Statistical Model and Analysis

In this subsection, we use a statistical model that transforms both the vector of observations Y and the environmental model (simulator) values $f(\beta)$. We start with an initial statistical model that assumes that the errors in the observed flow are independent; subsequently, the model is refined to explain temporal correlation in Y. Once an adequate model is found, we illustrate the GRIMA algorithm to approximate the posterior density of the simulator parameters β after the vector of nonsimulator parameters has been "integrated out" using heuristics.

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

$$h(Y_i, \lambda) = h\{f_i(\beta), \lambda\} + \epsilon_i, \tag{4.1}$$

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

In Equation (4.1), ϵ_i captures the error due to simulator inadequacy and the observation error, but only the sum of these two errors is identifiable. Other sources of variability are discussed in section 2.1 of Kennedy and O'Hagan (2001) and could be incorporated into our statistical model. For example, it is possible to introduce the simulator inadequacy function via a hierarchical linear model, while the residual variability can be addressed using a measurement error model for the covariates (such as weather inputs) of the simulator f.

However, as discussed in the online Appendix A.7, doing so is a challenge in its own right and is beyond the scope of this article. Since GRIMA does not assume a particular form of the posterior density, any of these extensions could be handled so long as the posterior density of interest can be evaluated.

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

$$h_{\mathrm{BC}}(y,\lambda) := \begin{cases} (y^{\lambda}-1)/\lambda & \text{if } \lambda \neq 0 \\ \log(y) = \lim_{\lambda\to 0}(y^{\lambda}-1)/\lambda & \text{if } \lambda = 0. \end{cases} \tag{4.2}$$

For right-skewed data, such as those in our application, λ does not exceed 1.

However, since the support of Gaussian ϵ_i in Equation (4.1) is $\mathbb{R}$, $h(\cdot,\lambda)$ must map the positive reals $\mathbb{R}^{+}$ onto the entire real line $\mathbb{R}$ for every value of the parameter λ (otherwise, the inverse of $h(\cdot,\lambda)$ is undefined). In the Box–Cox family, only the log transformation maps $\mathbb{R}^{+}$ onto $\mathbb{R}$. We "repair" this defect of the Box–Cox family by perturbing h_{BC} using the logarithmic transformation

$$h(y,\lambda) := (1-\Delta)\cdot h_{\mathrm{BC}}(y,\lambda) + \Delta\log(y), \tag{4.3}$$

where Δ is a small positive constant (e.g., 10^{-4}). An advantage of this transformation over an earlier one by Bliznyuk et al. (2008) is that the parameter λ retains its conventional interpretation.

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

$$L(\beta,\zeta\,|\,Y) := -\frac{1}{2}\log\det\Sigma(\theta) - \frac{1}{2}\|h(Y,\lambda) - h\{f(\beta),\lambda\}\|^{2}_{\Sigma(\theta)^{-1}} + \sum_{i=1}^{n}\log\frac{\partial h(Y_i,\lambda)}{\partial Y_i}, \tag{4.4}$$

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

$$\widehat{L}(\beta) := \sup_{\zeta} L(\beta,\zeta\,|\,Y). \tag{4.5}$$

Since derivatives of L with respect to the vector ζ of nonsimulator parameters are available analytically, the value $\widehat{L}(\beta)$ typically is cheap to compute numerically once $f(\beta)$ has been computed. (We compute the supremum in Equation (4.5) by the quadratic programming routine FMINCON in MATLAB.)

We fit the initial model that assumes that the errors ϵ_i are iid $N(0,\theta_1^2)$. Following the suggestions in Bliznyuk et al. (2008), we estimate β by maximization of $\widehat{L}$ to obtain the MLE (maximum likelihood estimate) $\widehat{\beta}$ using the minimization routine CONDOR and recover the MLE $\widehat{\zeta}$ for ζ by maximizing $L(\widehat{\beta},\cdot)$ with respect to ζ. In the absence of information about the location of $\widehat{\beta}$, CONDOR is initialized at the center of the parameter space $\mathfrak{B}$. The optimization required 91 runs of the simulator to converge. (The starting and terminal estimates for β for the models we consider, as well as parameter spaces for model parameters, are reported in Table 4.)

Table 4. Values of β and ζ (with appropriate parameter spaces) that maximize the log-likelihood L found by optimization by CONDOR for models with iid and AR(1) errors, and in the course of GRIMA for AR(1) model

	β				ζ				# of extra runs
Stage	$\widehat{\beta}_1$	$\widehat{\beta}_2$	$\widehat{\beta}_3$	$\widehat{\beta}_4$	$\widehat{\lambda}$	$\widehat{\theta}_1$	$\widehat{\theta}_2$	$-2L(\widehat{\beta},\widehat{\zeta})$	of f
CONDOR iid ϵ_i	6.12	0.646	34	0.892	−0.039	0.647	0	−112.6	91
CONDOR AR (1) ϵ_i	6.97	0.66	27.11	0.75	−0.152	0.454	0.814	−1135.7	65
GRIMA AR (1) ϵ_i	7.24	0.998	25.83	0.751	−0.147	0.45	0.817	−1145.5	88
Lower bound	0.001	0.001	0.001	0.75	−10	0	−1	–	–
Upper bound	500	1	180	1.25	1	100	1	–	–

Even though the simulated flow $f(\widehat{\beta})$ predicts the observed flow Y well, the residuals

$$e_i := h(Y_i, \widehat{\lambda}) - h\{f_i(\widehat{\beta}), \widehat{\lambda}\} \tag{4.6}$$

exhibit serial correlation: the plot of the autocorrelation function (ACF) shows (roughly) exponential decay, and the plot of the partial autocorrelation function (PACF) has a spike of height close to 0.8 at lag 1. One likely cause of residual autocorrelation is sampling error in rain flow, an input to the simulator. Rain flow over the entire watershed is estimated using data from a single weather station. Underestimation of a rainfall event, for example, will cause underestimation of stream flows that persists, though with decay, until the next rainfall event.

Consequently, correlation was incorporated into the statistical model of Equation (4.1) by modeling ϵ_i as an AR(1) process. Even though

$$\mathrm{Cov}(\epsilon_i, \epsilon_j) = \theta_1^2 \cdot \theta_2^{|i-j|} \tag{4.7}$$

under the AR(1) model implies that $\Sigma(\theta)$ is a dense matrix, the inverse of $\Sigma(\theta)$ is tridiagonal (e.g., Hamilton 1994, chap. 5). Hence, for a known $f(\beta)$, $L(\beta, \cdot)$ can be evaluated in time $\mathcal{O}(n)$, and the overhead to maximize $L(\beta, \cdot)$ to compute $\widehat{L}(\beta)$ is insignificant.

The second run of CONDOR to maximize $\widehat{L}$ under the AR(1) model for errors was initialized at the MLE $\widehat{\beta}$ under the iid model for errors. This second stage of maximization took 65 runs of the simulator. (During the exploratory stage of this research, additional optimization runs suggested that the profile log-likelihood has other local modes, but they are negligible, which was confirmed by MCMC sampling from the exact density, below.)

Examination of the ACF and PACF plots of the AR(1)-corrected residuals, obtained from the residuals e_i of Equation (4.6) as

$$u_i := e_{i+1} - \widehat{\theta}_2 \cdot e_i \tag{4.8}$$

for $i = 1, \ldots, n-1$, reveals a small (less than 0.2) correlation at lag 2. The starting and terminal estimates for β and estimates for ζ for the two models that we consider are reported in Table 4. We believe that the AR(1) model is adequate for assessing uncertainty in the stream flow parameters, so we did not try to refine the model further.

4.2.2 Approximation. Having settled on the statistical model, we move on to posterior density approximation using the GRIMA algorithm.

After putting a proper uniform prior distribution on β over the bounded parameter space $\mathfrak{B}$, we define the *profile posterior* as

$$\pi(\beta) := \exp\{\widehat{L}(\beta)\} \cdot \mathbb{I}(\beta \in \mathfrak{B}). \tag{4.9}$$

We used the profile posterior as an approximation to the marginal posterior density of β to avoid the difficulty of numerically integrating out the nonsimulator parameters ζ. We also considered holding ζ fixed at its MAP (maximum a posteriori) $\widehat{\zeta}$, which produced results (not shown) very similar to those reported later in this section.

We associate η with β and $\mathfrak{C}$ with $\mathfrak{B}$ of Section 2. The rest of the definitions for the application of the GRIMA algorithm have already been presented in Sections 2.3 and 2.4.

We use the approximation of Equation (A.1) in the online Appendices and choose q to be 0.99th quantile of the chi-squared distribution with $\dim(\beta) = 4$ degrees of freedom, which allows us to reuse 29 points from the preceding optimization runs to create an initial approximation $\widetilde{l}_0$ to l.

We initialize $T = 2 \cdot 10^4$, $J = 4$, and the rest of the parameters as in Section A.4. Every 12–14 evaluations of the exact posterior π, we do a long MCMC run of length $6 \cdot 10^4$ to assess the quality of approximation and to re-estimate the scaling matrix H_i for the MCMC sampler and to refit the RBF surface; we reset r after this new linear change of variables. The Markov chain mixes reasonably well, with lag 1 autocorrelation being less than 0.9 for each β_i and the overall acceptance rate between 0.2 and 0.3.

Figure 1 compares estimates of the TV norm between terminal (i.e., "most recent") and preceding approximate marginal distributions of β_k. (More precisely, we compare samples from marginal distributions of β_k based on $\widetilde{\pi}_j$ with $29, 41, \ldots, 104$ knots against those from $\widetilde{\pi}_i$ with 117 knots.) Examination of this plot and of plots of estimates of preceding densities suggest that for each β_k, the approximation improves little after the number of design points used in interpolation grows beyond 89. (Recall that 29 of these points come from the CONDOR run.) Consequently, we terminate the algorithm at 117 knots.

For the sake of comparison, we did an MCMC run of length $2 \cdot 10^4$ using the exact posterior density π, which took over 15 hr to complete on a modern workstation. (On the other hand, GRIMA produced a very accurate approximation in less than 1 hr.) In Figure 2, we overlay plots of estimates (from respective MCMC runs) of marginal densities of β_k's using the initial approximate posterior density $\widetilde{\pi}_0$, the terminal approximate posterior density $\widetilde{\pi}$, and the exact posterior density π. The plots in Figure 2 and Table 4 suggest that π is maximized when β_2 is at its upper boundary, but CONDOR terminated prematurely. Remarkably, GRIMA was able to recover from this deficiency and produced a very accurate approximation to π in 88 (or fewer) *extra* runs of the simulator f. (The value of β at which π is highest, reached within 29 extra evaluations of π by GRIMA, is reported in the third row of Table 4.)

The time required for MCMC with the exact posterior is long but tolerable. However, the use of the exact likelihood would be infeasible for larger problems, such as the simultaneous modeling of stream flows and several water quality variables over the entire Cannonsville watershed (recall Section 4.1), for which one run of the simulator takes minutes rather than

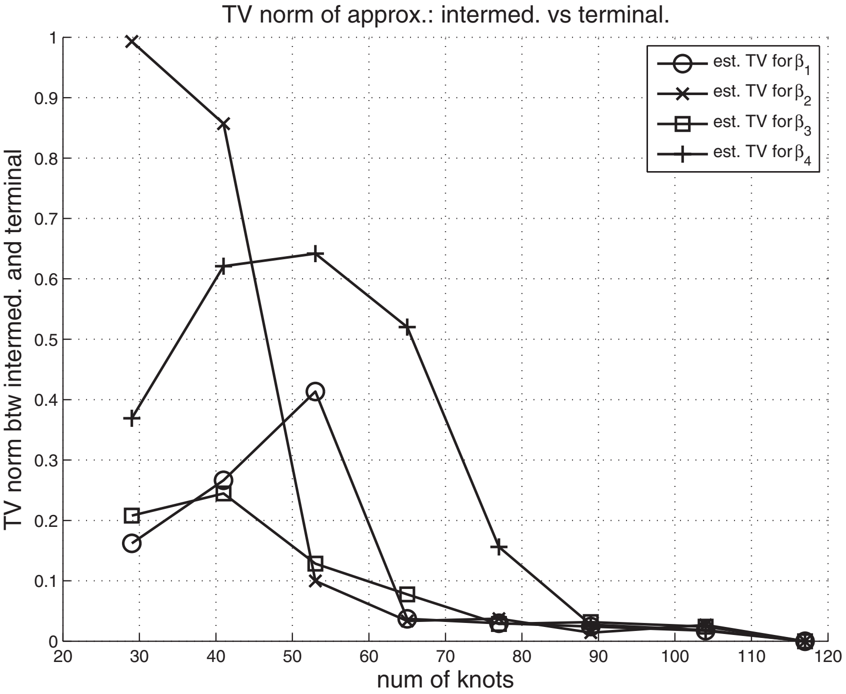

Figure 1. Estimated TV norm between intermediate and terminal (based on 117 knots for the Town Brook posterior density with AR(1) errors)approximate densities for β_i, $i = 1, \dots, 4$, as a function of the number of knots used to obtain intermediate approximate densities.

seconds; so, in the case of larger problems, the time for an MCMC run using the exact posterior could be on the order of $(15)(60) = 900$ hr or over five weeks.

From Figure 2, one can also appreciate the virtues of local (done over an HPD region) rather than global approximation to π. It is seen that nearly all of the mass of the posterior is contained in the hyper-rectangle with the lower bound [4, 0.7, 22, 0.75] and the upper bound [12, 1, 32, 0.79] of volume 0.96, which constitutes about 0.002% of the volume of the parameter space $\mathfrak{B}$. Therefore, the naïve approach that approximates π over the whole $\mathfrak{B}$ would waste nearly all of the computational budget on the unimportant low-probability region.

5. DISCUSSION AND CONCLUSIONS

In this article, we presented GRIMA, a major extension of the procedure of Bliznyuk et al. (2008) for sampling from posterior densities, such as those arising in the Bayesian treatment of computationally intensive nonlinear regression problems. However, because GRIMA does not assume a particular form of the posterior density, its applicability is not limited to inverse problems. The method proposed herein uses interpolation of the logarithm l of the exact posterior density using RBFs to construct a gradually improving sequence $\{\widetilde{l}_i\}$, from which the approximate posterior densities $\widetilde{\pi}_i$ are obtained. As the approximate posterior densities become more accurate, so do the respective approximate

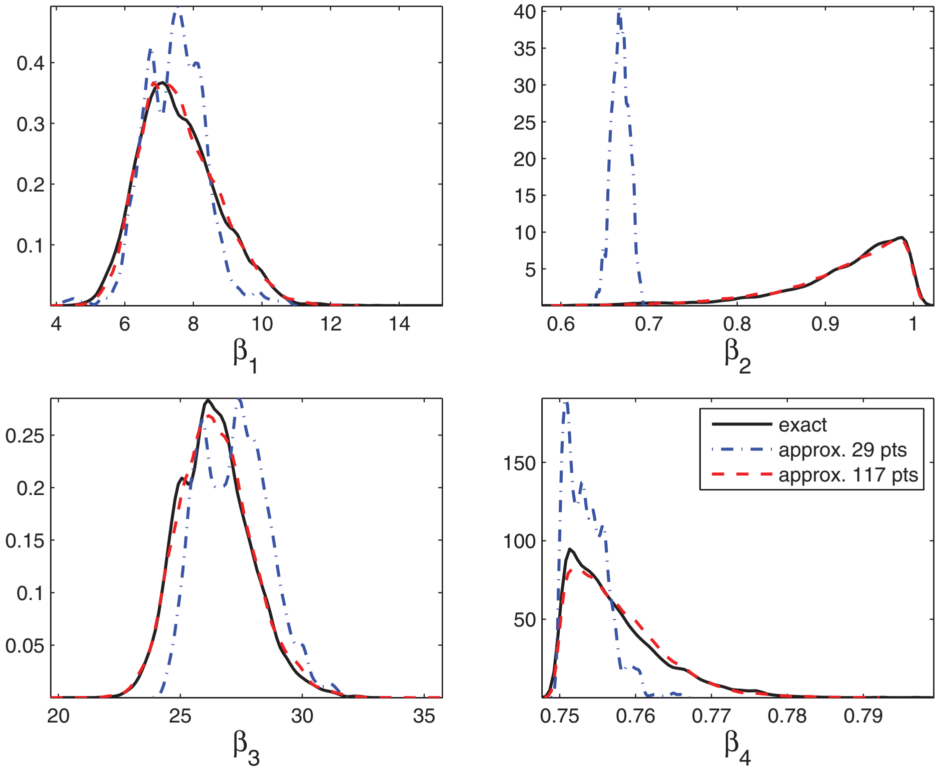

Figure 2. Kernel-smoothed estimates of marginal densities of β_i, $i = 1, \dots, 4$, using MCMC samples from exact (solid line) and approximate initial (dash-and-dot line, 29 knots), and terminal (dashed line, 117 knots) multivariate posterior densities (for the Town Brook statistical model with AR(1) errors). For the exact density, the sample size is $2 \cdot 10^4$; for approximate, $6 \cdot 10^4$ (drawn using random-walk MCMC in all cases). The online version of this figure is in color.

HPD regions. As a consequence, the knots for RBF fitting, selected to satisfy a maximum separation criterion, are chosen on a true HPD region for π. Our local approach has a considerable advantage over the approximation over the whole parameter space $\mathfrak{C}$ (Kennedy and O'Hagan 2001) in keeping low the proportion of knots chosen on the unimportant low-probability region. Unlike in the earlier articles, GRIMA does not require derivatives of π and is applicable to the problems with connected HPD regions of arbitrary shapes. Our stopping criterion, which is based on the estimated TV norm (online Appendix A.2) between the "most recent" and all of the preceding approximate densities, allows one to terminate the algorithm when the approximate densities become sufficiently accurate, thereby reducing the waste of computational budget in expensive evaluations. Of course, our stopping criterion can be replaced by a more general loss function depending on the goals of application.

We illustrated the progress and robustness of GRIMA on a synthetic test problem of Rasmussen (2003) in Section A.4 of the online Appendices. Subsequently, our algorithm was applied to solve the Bayesian parameter calibration problem for the Town Brook watershed. Our results indicate that our algorithm is capable of reducing the computational load (relative to MCMC sampling from the exact posterior density) by an order of magnitude or more. Despite the curse of dimensionality inherent in all interpolation-based approximation approaches such as GRIMA, our simulation results of Section 3 suggest that a computational budget of several hundred function evaluations is sufficient for accurate approximation of the wide range of unimodal and bimodal densities of dimension up to 10.

In the applications considered in this article, we attribute the success of GRIMA to the linear change of variables before RBF fitting (discussed in the end of Section 2.1 and in the end of the online Appendix A.1) and, on doing the approximation, only over the approximate HPD region. The merits of updating the RBF surface (online Appendix A.3) over refitting it from scratch, although not manifested in these applications, will be realized when the number of knots for interpolation is on the order of thousands.

In this article, we assumed that the HPD region is connected, which is crucial for ensuring that the random-walk MCMC sampler traverses the support of the target distribution easily. However, it is possible to extend the algorithm to deal with multimodal posterior densities with disconnected modes: one would apply GRIMA locally around each high-probability mode, represent the approximate posterior density as a mixture density, and then proceed with sampling from a multimodal density, as discussed in Tjelmeland and Hegstad (2001). Thus, one can view GRIMA as a procedure for local parameter uncertainty analysis.

It is worth noting that with only minor modifications, GRIMA can be parallelized. Indeed, one just needs to replace the **for** loop of Algorithm 1 (lines 6–16) with an assignment of (at most) J candidate points to (at most) J processors so that the exact posterior density π can be evaluated in parallel at the candidate points. It is also easy to spread the computational load of the MCMC simulation over multiple processors, for example, by running several shorter Markov chains in parallel.

SUPPLEMENTAL MATERIALS

Supplemental materials include (a) online Appendices with details of implementation, additional summaries of our simulations and an illustration of GRIMA on a 2-dimensional test problem of Rasmussen (2003) (appendix.pdf) and (b) a zip. file with MATLAB implementation of the proposed procedures (see the !readme.txt for details).

ACKNOWLEDGMENTS

The research of Bliznyuk, Ruppert, and Shoemaker was supported by the National Science Foundation (NSF) under grant DMS-0434390 (Principal Investigators: Ruppert and Shoemaker). Bliznyuk was also supported by the National Institute of Environmental Health Sciences (NIEHS) grant ES-07142, Environmental Protection Agency (EPA) grant R-832416, and the postdoctoral training grant CA090301 from the National Cancer Institute. Ruppert was also supported by the NSF grant DMS-0805975. Shoemaker was also supported by NSF grants 0711491, 0756575, and 1116298. The model for Town Brook data was developed under the NSF BES-0229176 grant. We thank Dillon Cowan, Yingxing Li, Rommel Regis, and Benjamin Shaby for sharing their experience and providing advice about various aspects of this article.

[Received September 2009. Revised June 2011.]

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. [486]

Azzalini, A., and Dalla Valle, A. (1996), "The Multivariate Skew-Normal Distribution," *Biometrika*, 83, 715–726. [484]

Benaman, J., Shoemaker, C. A., and Haith, D. A. (2005), "Calibration and Validation of Soil and Water Assessment Tool on an Agricultural Watershed in Upstate New York," *ASCE Journal of Hydrologic Engineering*, 10, 363–374. [477]

Bliznyuk, N., Ruppert, D., and Shoemaker, C. A. (2011), "Efficient Interpolation of Computationally Expensive Posterior Densities With Variable Parameter Costs," *Journal of Computational & Graphical Statistics*, 20, 636–655. [478]

Bliznyuk, N., Ruppert, D., Shoemaker, C. A., Regis, R., Wild, S., and Mugunthan, P. (2008), "Bayesian Calibration and Uncertainty Analysis for Computationally Expensive Models Using Optimization and Radial Basis Function Approximation," *Journal of Computational & Graphical Statistics*, 17, 270–294. [477,477,479,481,481,487,488,491]

Box, G. E. P., and Cox, D. R. (1964), "An Analysis of Transformations," *Journal of the Royal Statistical Society*, Series B, 26, 211–246. [488]

Carroll, R. J., and Ruppert, D. (1984), "Power Transformation When Fitting Theoretical Models to Data," *Journal of the American Statistical Association*, 79, 321–328. [487]

———, (1988), *Transformation and Weighting in Regression*, New York: Chapman & Hall. [487]

Cressie, N. (1991), *Statistics for Spatial Data*, New York: Wiley. [479]

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. [486]

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. [486]

Hamilton, J. D. (1994), *Time Series Analysis*, Princeton, NJ: Princeton University Press. [489]

Kennedy, M. C., and O'Hagan, A. (2001), "Bayesian Calibration of Computer Models," *Journal of the Royal Statistical Society*, Series B, 63, 425–464. [477,487,492]

Mugunthan, P., and Shoemaker, C. A. (2006), "Assessing the Impacts of Parameter Uncertainty for Computationally Expensive Groundwater Models," *Water Resources Research*, 42, W10428. [477]

Mugunthan, P., Shoemaker, C. A., and Regis, R. G. (2005), "Comparison of Function Approximation, Heuristic and Derivative-Based Methods for Automatic Calibration of Computationally Expensive Groundwater Bioremediation Models," *Water Resources Research*, 41, W11427. [477]

Rasmussen, C. E. (2003), "Gaussian Processes to Speed Up Hybrid Monte Carlo for Expensive Bayesian Integrals," in *Bayesian Statistics 7: Proceedings of the 7th Valencia International Meeting*, eds. J. M. Bernardo, J. O. Berger, A. P. Berger, and A. F. M. Smith, pp. 651–659. [477,477,478,479,481,483,492,493]

Ruppert, D., Shoemaker, C. A., Wang, Y., Li, Y., and Bliznyuk, N. (2012), "Uncertainty Analysis for Computationally Expensive Models with Multiple Outputs," *Journal of Agricultural, Biological and Environmental Statistics*. doi: 10.1007/s13253–012–0091–0 [487,487]

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. [477,479,486,487]

Tjelmeland, H., and Hegstad, B. K. (2001), "Mode Jumping Proposals in MCMC," *Scandinavian Journal of Statistics*, 28, 205–223. [493]

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* [487]

——— (2007a), “The Dynamically Dimensioned Search Algorithm for Computationally Efficient Automatic Calibration of Environmental Simulation Models,” *Water Resources Research*, 43, W01413. [478,479,487]

——— (2007b), “Cannonsville Reservoir Watershed SWAT2000 Model Development, Calibration and Validation,” *Journal of Hydrology*, 337, 68–89. [478,479,486,487]

Vanden Berghen, F., 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. [480]