EFFICIENT EMULATORS OF COMPUTER EXPERIMENTS USING COMPACTLY SUPPORTED CORRELATION FUNCTIONS, WITH AN APPLICATION TO COSMOLOGY

BY CARI G. KAUFMAN[1], DEREK BINGHAM[2], SALMAN HABIB[3], KATRIN HEITMANN[3] AND JOSHUA A. FRIEMAN

University of California, Berkeley, Simon Fraser University, Argonne National Laboratory, Los Alamos National Laboratory, and Fermi National Accelerator Laboratory and University of Chicago

Statistical emulators of computer simulators have proven to be useful in a variety of applications. The widely adopted model for emulator building, using a Gaussian process model with strictly positive correlation function, is computationally intractable when the number of simulator evaluations is large. We propose a new model that uses a combination of low-order regression terms and compactly supported correlation functions to recreate the desired predictive behavior of the emulator at a fraction of the computational cost. Following the usual approach of taking the correlation to be a product of correlations in each input dimension, we show how to impose restrictions on the ranges of the correlations, giving sparsity, while also allowing the ranges to trade off against one another, thereby giving good predictive performance. We illustrate the method using data from a computer simulator of photometric redshift with 20,000 simulator evaluations and 80,000 predictions.

Simulation of complex systems has become commonplace in scientific studies. Frequently, simulators (or computer models) are computationally demanding, and relatively few evaluations can be performed. In other cases, the computer models are fast to evaluate but are not readily available to all scientists. This arises, for example, when the simulator runs only on a supercomputer or must be run by specialists. In either case, a statistical emulator of the computer model can act as a surrogate, providing predictions of the computer model output at unsampled inputs values, with corresponding measures of uncertainty [see, e.g., Sacks et al. (1989), Santner, Williams and Notz (2003)]. The emulator can serve as a component in probabilistic model calibration [Kennedy and O'Hagan (2001)], and it

can help provide insight into the functional form of the simulator response and the importance of various inputs [Oakley and O'Hagan (2004)].

Building an emulator can be viewed as a type of nonparametric regression problem, but with a key difference. Computer experiments differ from their real-world counterparts in that they are typically deterministic. That is, running the code twice with the same set of input values will produce the same result. To deal with this difference from the usual noisy settings, Sacks et al. (1989) proposed modeling the response from a computer experiment as a realization from a Gaussian process (GP). From a Bayesian viewpoint, one can think of the GP as a prior distribution over the class of functions produced by the simulator.

The GP model is particularly attractive for emulation because of its flexibility to fit a large class of response surfaces. It is also desirable that the statistical model, and corresponding prediction intervals, reflect some type of smoothness assumption regarding the response surface, leading to zero (or very small) predictive uncertainty at the design points, small predictive uncertainty close to the design points, and larger uncertainty further away from the design points. For example, note the behavior of the confidence intervals in the illustration shown in panel (a) of Figure 1.

The main drawback of the usual GP model in our setting is that it can be computationally infeasible for large designs. The likelihood for the observations is multivariate normal, and evaluation of this density for n design points requires
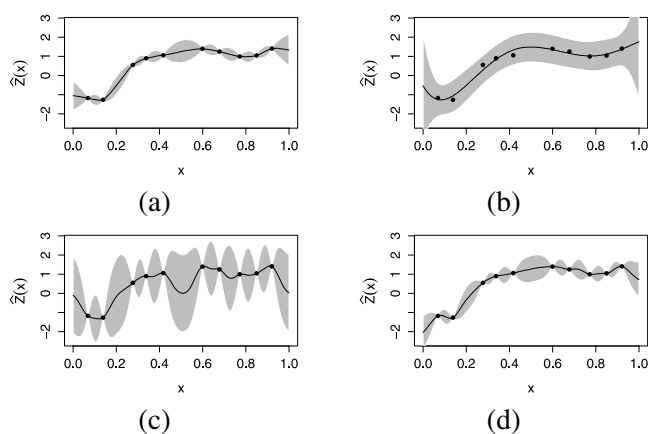

FIG. 1. *Illustrative example in which the data were drawn from a mean zero Gaussian process with covariance function* $K(x, x') = \exp\{-5|x - x'|^{1.5}\}$ *and predictions were made using one of four methods. Each plot shows the observations* (*solid dots*), *predictions* (*solid line*) *and pointwise* 95% *confidence intervals for the predictions* (*gray bands*). *The models were* (a) *zero mean GP with the correct covariance structure*, (b) *OLS using Legendre polynomials up to degree* 6, (c) *zero mean GP with the Bohman correlation function* (7), *with* $\tau = 0.1$, *and* (d) *GP with Legendre polynomials up to degree* 2 *in the mean and Bohman correlation function with* $\tau = 0.1$.

manipulations of the $n \times n$ covariance matrix that grow as $O(n^3)$. This limitation is the main motivation for this article, which was prompted by our work on just such an application in cosmology (see Section 5). Here, the basic idea is to construct a statistical model based on a set of simulated data consisting of multi-color photometry for training set galaxies, as well as the corresponding true redshifts. Given photometric information for a test galaxy, the system should produce an estimated value for the true redshift. The very large experimental design used to explore the input space, that is, the large number of galaxies used to build the training set, presents a computational challenge for the GP.

It is worth noting that the GP model is not the only approach for emulating computer simulators. Using a GP with constant mean term can be viewed as a way of forcing the covariance structure of the GP to model all the variability in the computer output. At the other extreme, one can take a regression based approach and treat the errors as white noise, as in An and Owen (2001). This approach has the benefit of being computationally efficient, as the correlation matrix is now the identity matrix. However, it does not have the attractive property of a smooth GP, namely, that the predictive distribution interpolates the observed data and that the uncertainty reflects the above properties. Instead, the white noise is introducing random error to the problem that is not actually believed to exist; it is there simply to reflect lack of fit. The implications of this for predictive uncertainty are illustrated in panels (a) and (b) of Figure 1. Panel (a) shows a set of data fit using the standard GP model, and panel (b) shows the same data fit using ordinary least squares regression with the set of Legendre polynomials up to degree six. The behavior in panel (a) is what we desire in an emulator, but the model is computationally intractable for large data sets. The model in panel (b) is very efficient from a computational standpoint, but the predictions do not reflect the determinism of the computer simulator. The approach proposed in this article can be viewed as finding an intermediate model to those in panels (a) and (b), such that the model is both fast to fit and has the appropriate behavior for prediction.

In this article, we propose new methodology for emulating computer simulators when n is large (i.e., when it is infeasible to fit the usual GP model). The approach makes the key innovation of replacing the covariance function with one that has compact support. This has the effect of introducing zeroes into the covariance matrix, so that it can be efficiently manipulated using sparse matrix algorithms. In addition, the proposed approach easily handles the anisotropy that is common in computer experiments by allowing the correlation range in each dimension to vary and also imposes a constraint on these ranges to enforce a minimum level of sparsity in the covariance matrix. We further propose a model for the mean of the GP, rather than taking it to be a scalar. The motivation for this is that the introduction of regression functions tends to decrease the estimated correlation length in the GP, thereby offsetting some of the loss of predictive efficiency introduced by using a compactly supported covariance. Last, we propose prior distributions to incorporate experimenter belief and also to make the application of regression terms and the compactly supported covariance function efficiently work together.

In the next section we introduce the GP that is commonly used for building emulators and illustrate the challenges for large data sets. In Section 3 we present new methodology for building computationally efficient emulators, and we give some details of the implementation and computational advantages in Section 4. In Section 5 we investigate the performance of the method in a simulation study. The method is then used to construct an emulator of photometric redshifts of cosmological objects in Section 6, and we conclude with some final remarks in the Appendix.

1. Gaussian process models for computer experiments. Consider a simulator that takes inputs $\mathbf{x} \in \Re^d$ and produces univariate output $Y(\mathbf{x})$. The GP model generally used in this setting treats the response as a realization of a random function:

$$(1) \qquad Y(\mathbf{x}) = \sum_{i=1}^{p} f_i(\mathbf{x})\beta_i + Z(\mathbf{x}),$$

where $f_1, \ldots, f_p$ are fixed regression functions, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$ is a vector of unknown regression coefficients, and Z is a mean zero GP. The covariance function of Z is denoted by

$$(2) \qquad \text{Cov}(Z(\mathbf{x}), Z(\mathbf{x}')) = K(\mathbf{x}, \mathbf{x}'; \sigma^2, \boldsymbol{\theta}) = \sigma^2 R(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}),$$

where σ^2 is the marginal variance of Z and $\boldsymbol{\theta}$ is a vector of parameters controlling the correlation.

We defer until the end of this section a discussion of the choice of $f_1, \ldots, f_p$ and R and first lay out some general notation. Let $\mathbf{Y} = (Y(\mathbf{x}_1), \ldots, Y(\mathbf{x}_n))'$ be the vector of observed responses. Then, ignoring a constant, the log-likelihood under this model is $l(\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2) = -\frac{1}{2}\log|\mathbf{R}(\boldsymbol{\theta})| - \frac{1}{2\sigma^2}(\mathbf{Y} - \mathbf{F}\boldsymbol{\beta})'\mathbf{R}(\boldsymbol{\theta})^{-1}(\mathbf{Y} - \mathbf{F}\boldsymbol{\beta})$, where $\mathbf{F}$ is the $n \times p$ matrix of regression functions and $\mathbf{R}(\boldsymbol{\theta})$ is the $n \times n$ matrix of correlations with $[\mathbf{R}(\boldsymbol{\theta})]_{ij} = R(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta})$. In addition, for any set of model parameters, the conditional distribution of $Y(\mathbf{x}_0)$ at a new input value, $\mathbf{x}_0$, given the observations $\mathbf{Y}$, is normal with mean and variance

$$(3) \qquad E[Y(\mathbf{x}_0)|\mathbf{Y}, \boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}] = f(\mathbf{x}_0)\boldsymbol{\beta} + \mathbf{r}_0(\boldsymbol{\theta})'\mathbf{R}(\boldsymbol{\theta})^{-1}(\mathbf{Y} - \mathbf{F}\boldsymbol{\beta}),$$

$$(4) \qquad \text{Var}[Y(\mathbf{x}_0)|\mathbf{Y}, \boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}] = \sigma^2[1 - \mathbf{r}_0(\boldsymbol{\theta})'\mathbf{R}(\boldsymbol{\theta})^{-1}\mathbf{r}_0(\boldsymbol{\theta})],$$

where $\mathbf{r}_0(\boldsymbol{\theta})$ is the n-vector of correlations between the observed responses and $Y(\mathbf{x}_0)$.

In practice, $\boldsymbol{\beta}$, σ^2 and $\boldsymbol{\theta}$ are unknown and must be estimated. This can be achieved using likelihood-based methods such as maximum likelihood (ML) or restricted maximum likelihood (REML) [see, e.g., Irvine, Gitelman and Hoeting (2007) for a comparison of ML and REML estimation]. Alternatively, one may specify a Bayesian model in which the joint posterior distribution for both parameters and predicted values of the function can be approximated via Markov Chain

Monte Carlo (MCMC). We adopt the latter approach, although most of the proposed methodology is also applicable in a frequentist setting.

Two choices must be made to complete the specification in (1) and (2), the regression functions $f_1, \ldots, f_p$ and the correlation function R. The mean structure in (1) is almost always taken to be flat over the domain of the function, with $f(x) \equiv 1$. By far the most common specification for the correlation function is a product of one-dimensional power exponential correlation functions. Specifically, writing $\boldsymbol{\theta}_k = \{\phi_k, \alpha_k\}$,

$$(5) \qquad R(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) = \prod_{k=1}^{d} R_k(|x_k - x_k'|; \boldsymbol{\theta}_k) \qquad \text{(Product form)}$$

$$(6) \qquad = \prod_{k=1}^{d} \exp\{-\phi_k |x_k - x_k'|^{\alpha_k}\} \qquad \text{(Power exponential)}$$

for $\phi_k > 0$ and $1 \leq \alpha_k \leq 2$. Since the $\boldsymbol{\phi}_k$'s are not constrained to be equal, the model can handle different signals in each input dimension of the simulator (i.e., anisotropy). This choice of constant mean term and power exponential correlation is so common in practice that we will refer to it as the "standard model." In the next section we shall diverge from the standard model and propose a more computationally efficient model with different mean and covariance structures.

No matter what choices are made for the regression terms and correlation function, inference and prediction for this model requires evaluation of the log-likelihood, typically many times. These calculations require finding the log determinant of $\mathbf{R}(\boldsymbol{\theta})$ and solving $\mathbf{R}(\boldsymbol{\theta})^{-1}(\mathbf{Y} - \mathbf{F}\boldsymbol{\beta})$. When the correlation functions are strictly positive, as in the standard model, both of these grow in complexity by order n^3, and therein lies the problem. These operations are intractable for moderate sample sizes and simply impossible for large sample sizes. It is this problem that motivates the current work.

2. Building computationally efficient emulators.
Our approach is conceptually straightforward, consisting of three main innovations:

(1) using compactly supported correlation functions to model small-scale variability,

(2) using regression functions in the mean of the GP to model large-scale variability, and

(3) specifying prior distributions for model parameters (or parameter constraints, in the frequentist case) that are flexible while still enforcing computational efficiency.

These three innovations work together to produce a flexible, fast and powerful method for computer model emulation.

2.1. *Compactly supported correlation functions.* We begin by proposing that the correlation functions in the product covariance (5) are chosen to have compact support. That is, for some $\tau_k \geq 0$, $R_k(|x_k - x'_k|; \tau_k) = 0$ when $|x_k - x'_k| \geq \tau_k$. This has the effect of introducing zeros into the covariance matrix, so computationally efficient sparse matrix techniques [Pissanetzky (1984), Barry and Pace (1997)] may be used. Compactly supported correlation functions have recently received increased attention in the literature, used either by themselves [Gneiting (2002), Stein (2008)] or multiplying another, strictly positive correlation function, which is known as tapering [Furrer, Genton and Nychka (2006), Kaufman, Schervish and Nychka (2008)]. These applications all assume that the compactly supported correlation function is isotropic, requiring a single range parameter. In contrast, anisotropic covariance functions are the norm for computer experiments because the inputs to the computer model are frequently on different scales and/or impact the response in vastly different ways. Therefore, we use correlation functions with compact support in product form.

We focus on two families of models that can be used to approximate the power exponential function (6). These functions are zero for $t \geq \tau$, and for $t < \tau$,

(7) $$\text{Bohman:} \qquad R(t; \tau) = (1 - t/\tau)\cos(\pi t/\tau) + \sin(\pi t/\tau)/\pi,$$

$$\text{Truncated power:} \quad R(t; \tau, \alpha, \nu) = [1 - (t/\tau)^\alpha]^\nu,$$
(8)
$$0 < \alpha < 2, \nu \geq \nu_d(\alpha).$$

The term $\nu_d(\alpha)$ in (8) represents a restriction so that the function is a valid correlation, with $\lim_{\alpha \to 2} \nu_d(\alpha) = \infty$ [Golubov (1981)]. Although it is difficult to calculate $\nu_d(\alpha)$ exactly, Gneiting (2001) gives upper bounds for a variety of values of α between 1.5 and 1.955. For example, upper bounds for $\nu_1(\alpha)$ when $\alpha = 3/2$ and $5/3$ are 2 and 3, respectively.

These two functions allow for some flexibility in the smoothness of the realizations of the GP, in loose analogy to the parameter α in the power exponential model (6). The truncated power function (8) is not differentiable even once at the origin, with $\alpha < 2$, corresponding to a process which is not even mean square continuous, as for (6) with $\alpha < 2$. In contrast, the Bohman function (7) is twice differentiable at the origin, corresponding to a process which is mean square differentiable. Of course, when $\alpha = 2$, the power exponential function (6) is infinitely differentiable at the origin. However, we feel this level of smoothness is often unrealistic, and in our applied work we have not found any reason to prefer a higher level of smoothness than is given by (7).

Note that the ranges play two different roles in our approach. First, they control the degree of correlation in each dimension. In this way they are similar to the range parameters ϕ_k in the power exponential function (6). However, unlike the ϕ_k, the τ_k also control the degree of sparsity of the correlation matrix. To produce computational savings, some restrictions must be applied to the τ_k. We chose to do this through the prior distribution, which we discuss in Section 2.3.

2.2. *Regression functions.* We propose using regression functions to model the mean structure of a computer model, rather than the constant mean used in the standard model. The intuition behind our approach is that if the large-scale structure of the simulator output is well captured by a linear combination of the basis functions f_i, we would naturally expect the residual process $Z(\mathbf{x})$ to have shorter-range correlations. This type of trade-off between large scale and small scale variability has been noted in the spatial statistics literature [Cressie (1993), Stein (2008)], and it was noted in the results of a simulation study of computer experiments by Welch et al. (1992). However, to our knowledge, it has not previously been exploited for computational purposes in constructing emulators, an application in which we will see the often high dimensionality of the input $\mathbf{x}$ allows this trade-off to be leveraged particularly well.

Using a richer mean structure produces more efficient predictions than the use of the compactly supported covariance alone. For example, panel (c) of Figure 1 illustrates the predictions and pointwise confidence bands under the zero mean GP model with Bohman correlation function (7) with $\tau = 0.1$. These are obviously less efficient than the results under the true correlation function in panel (a). However, this limitation can be addressed by incorporating regression terms. Panel (d) in Figure 1 illustrates the results of combining a small set of basis functions with a compactly supported correlation function. The behavior of the predictions is similar to that in panel (a), but the model in panel (d) can be applied to large data sets, whereas the model in (a) cannot.

There are a variety of basis functions among which one can choose, and detailing them is not the focus of this article. We have found a good default choice to be taking f_i to be tensor products of Legendre polynomials over [0, 1], the input variable in each dimension having been rescaled to this domain [see, e.g., An and Owen (2001)].

2.3. *Prior distributions.* The full specification of our Bayesian model requires that we choose prior distributions for the parameters τ, σ^2 and $\boldsymbol{\beta}$. We advocate the inclusion of prior information where available, but describe here a default choice of prior distributions that brings additional computational efficiency. These advantages may be had in a frequentist setting by replacing the prior distributions with certain restrictions on the parameter space, which should be obvious as we proceed.

One of the novel features of the proposed approach is use of an anisotropic compactly supported covariance and the estimation of the range parameters. Recall that the τ_k's play the role of governing the correlation in each dimension and also controlling the computational complexity of the model. Unfortunately, the mapping from a collection $\tau_1, \ldots, \tau_d$ to the sparsity of a given matrix will depend on the particular configuration of sampling points, and it is also difficult to translate a degree of sparsity (measured, say, by the percentage of off-diagonal zeroes) to computation time for a particular algorithm. However, we have found that with

a bit of trial and error, controlling the sparsity of the correlation matrix $\mathbf{R}$ can be an adequate proxy for controlling computation time. The question then is, how do we control sparsity through the prior for $\boldsymbol{\tau}$?

Throughout, we will assume that all input variables have been scaled to $[0, 1]$, so that taking $\tau_k > 1$ for all k introduces no sparsity. A simple restriction is

$$(9) \qquad \tau_j \leq B \qquad \text{for all } j, B > 0.$$

However, using this restriction in creating a prior distribution ignores an important advantage of using compactly supported correlation functions within the product correlation given in (5): that $R(x, x') \geq 0$ only if $|x_j - x'_j| < \tau_j$ for all $j = 1, \ldots, d$. That is, a pair of observations having zero correlation in *any* dimension will be independent and contribute a zero to the overall correlation matrix. Therefore, the τ_j may trade off against one another to produce the same degree of sparsity. Therefore, we restrict τ further by taking τ to be uniformly distributed over the space

$$(10) \qquad T_C = \left\{ \tau \in \Re^d : \tau_j \geq 0 \, \forall d, \sum_{j=1}^{d} \tau_j \leq C \right\}, \qquad C > 0.$$

This can be thought of as defining the prior distribution over a d-dimensional triangle rather than a cube. Because of the trade-off between the τ_j discussed above, C can generally be greater than B is in (9) and impose the same degree of sparsity. That is, some of the τ_j are allowed to be large, which they may indeed need to be to reflect a high degree of correlation in particular input dimensions. Some trial and error may be required to find a C for which calculations can actually be carried out; we return to this in the next section. This restriction may not work well for the case that certain input variables have no impact on the output variable over the range being simulated, which would correspond to $\tau_j \to \infty$. We make the assumption that such variables have been previously identified and fixed during a prior screening analysis such as described in Welch et al. (1992) or Linkletter et al. (2006).

Finally, we specify prior distributions for σ^2 and $\boldsymbol{\beta}$. We follow Berger, De Oliveira and Sansó (2001) and Paulo (2005), who proposed the form $p(\sigma^2, \beta, \tau) \propto \pi(\tau)/\sigma^2$ for Gaussian processes. As our choice of π is a proper density, it can be shown that this choice still produces a proper posterior [Berger, De Oliveira and Sansó (2001)]. One advantage of this choice is that β and σ^2 may be easily integrated out of the model, so that posterior sampling may be done only over the vector τ.

We end this section with a note relating our proposed model to existing works in the field of spatial statistics, in which estimation and prediction under large sample sizes has seen much recent development. These approaches may be characterized broadly as either approximating the likelihood for the original model, or changing the model itself to one that is computationally more convenient. Examples of the former include Stein, Chi and Welty (2004) and Kaufman, Schervish and Nychka

(2008), while a common modification to the model itself is to represent the random field in terms of a lower-dimensional random variable; models falling under this framework are reviewed by Wikle (2010). The approach we take in this paper is to change the model rather than approximate it. We did consider using compactly supported correlation functions as an approximation tool rather than using them directly, following Kaufman, Schervish and Nychka (2008). Ultimately, however, since there is no "true" random process generating the output of the computer simulator, we decided to take the conceptually simpler approach of modifying the model itself. That is, in this nonparametric regression context, the GP is simply a tool for expressing prior beliefs about the function, and this may be done effectively using compactly supported correlation functions when regression terms are also included in the model.

3. Implementation and computational considerations. Implementation of these methods for a given data set proceeds in two steps. The first step is to sample from the posterior distribution $p(\boldsymbol{\tau}|\mathbf{Y})$ using a Metropolis sampler. As noted above, our choice of prior distribution allows us to work with this marginal distribution, integrating out σ^2 and $\boldsymbol{\beta}$. This leads to additional computational savings, as we can sample the vector $\boldsymbol{\tau}$ using an efficient, adaptive Metropolis sampler, the details of which are described in the Appendix. The second step is to use these samples to generate predictions at new input values. Rather than incorporating this into the sampler, we recommend another computational trick here, which is to calculate conditional means and variances at a subset of the iterations (i.e., conditional on $\boldsymbol{\tau}^{(i)}$) and use these to reconstruct the predictive mean and variance using laws of iterated expectation. The details of this calculation are also described in the Appendix. In the remainder of this section, we outline the computational savings to be gained from our approach, compared to standard, nonsparse techniques.

The demanding aspects of the calculations all involve $\mathbf{R}(\boldsymbol{\tau})$, the correlation matrix for a particular value of $\boldsymbol{\tau}$. To efficiently calculate the quantities involved, after we propose a new $\boldsymbol{\tau}$ in the Metropolis step, we first compute a sparse representation of $\mathbf{R}(\boldsymbol{\tau})$, then compute the quantities which will be used in calculating the integrated likelihood. Here, we use the `spam` package in R [Furrer and Sain (2010)], which uses the efficient "old Yale sparse format" for representing sparse matrices, where only the nonzero elements, column indices and row pointers are stored. From a computational viewpoint, operations involving the zero elements need not be performed. The time-consuming steps in evaluating the likelihood are then as follows:

(1) Identifying pairs of input values $\mathbf{x}_i$ and $\mathbf{x}_j$ such that $|x_{ik} - x_{jk}| < \tau_k, \forall k = 1, \ldots, d$. (All other pairs will contribute a zero to the matrix.)

(2) For only these pairs, computing $\prod_{k=1}^{d} R_k(|x_{ik} - x_{jk}|; \tau_k)$ and using only these to create the sparse representation of $\mathbf{R}(\boldsymbol{\tau})$.

(3) Computing the Cholesky decomposition of the sparse matrix object, that is, $\mathbf{Q}(\boldsymbol{\tau})$ such that $\mathbf{R}(\boldsymbol{\tau}) = \mathbf{Q}(\boldsymbol{\tau})'\mathbf{Q}(\boldsymbol{\tau})$.
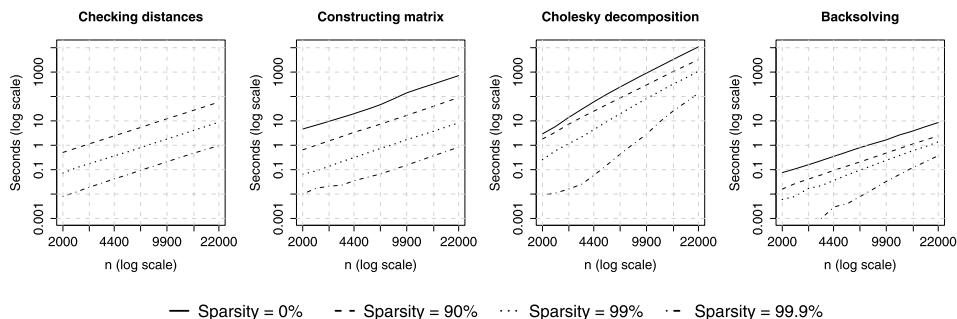
FIG. 2. *Number of seconds required to carry out each step in evaluating the likelihood, for varying degrees of sparsity and sample sizes. The case "Sparsity = 0%," shown as a solid line, corresponds to using a strictly positive correlation function, while for the other cases "Sparsity" denotes the percentage of off-diagonal elements in the matrix that are equal to zero. The "Checking distances" step only applies to the algorithm used for compactly supported correlation functions, which is why no solid line appears in this plot. All measurements are shown on the log (base 10) scale, and each gridline along the y-axis corresponds to one order of magnitude.*

(4) Backsolving to compute $(\mathbf{Q}(\boldsymbol{\tau})')^{-1}\mathbf{Y}$ and $(\mathbf{Q}(\boldsymbol{\tau})')^{-1}\mathbf{F}$.

Figure 2 shows the average number of seconds required to carry out these steps for a reference data set consisting of locations uniformly sampled over the input space $[0, 1]^4$, which is the same dimension as our example in Section 5. The calculations were carried out for various sample sizes and varying degrees of sparsity, which is imposed by the choice of the cutoff C. Each calculation was repeated 10 times, and the number of seconds required for each were averaged to produce the plot. Computations were performed on a Dell quad socket machine with four dual-core AMD Opteron 2.4 GHz CPUs and 8 GB of RAM. All axes are on the log scale. In particular, each gridline along the y-axis corresponds to one order of magnitude. Overall, we can see that using a sparse correlation function reduces most of the calculations by one to three orders of magnitude.

4. Simulation study. The method we present here is intended for use in situations in which the "standard" method is computationally infeasible. However, it is instructive to see how our method compares when *n* is small enough that the standard method can actually be implemented. In this section we compare the performance of our proposed model to that of the standard model, when the data are actually simulated under the standard model. Of course, in practice, neither model is "correct," since the object of interest is a deterministic function, not a realization of a stochastic process, but this framework allows us to compare the efficiency of predictions made under varying degrees of smoothness and for different correlation lengths.

In carrying out the simulation study, we vary both the distribution of the data and the choices made in fitting it. Regarding the data, we consider processes in

two or four dimensions, with correlation function (6) with α_k set equal (in all dimensions) to 1.5 or 1.99. (We do not consider the infinitely smooth case of $\alpha = 2$ here, to avoid dealing on a case-by-case basis with the numerical instabilities that sometimes arise.) We specify ϕ_k in (6) to be such that the effective range, defined as the distance beyond with correlations are less than 0.05, is either 0.5 or 2 in all dimensions. This gives six possible combinations.

For each combination and each of 100 replications of the simulation, we consider sample sizes of $n = 100, 150, 250, 400, 650$ and $1{,}100$. The design (choice of input settings) for each n and replication is generated using a random Latin hypercube sample (LHS) [McKay, Beckman and Conover (1979)] on $[0, 1]^d$. In addition, we generate a set of $n_{\text{pred}} = 512$ input settings at which to predict the function, using the orthogonal array-based Latin hypercube sampling (OA-LHS) algorithm by Tang (1993), with frequency $\lambda = 2$. We sampled values of $Y(x)$ over the LHS and OA-LHS jointly, corresponding to a realization from the GP. The rationale for this sampling strategy stems from our different priorities when choosing the design points and choosing the prediction points. Repeated sampling under simple LHS is our attempt to mimic the broad class of designs experimenters mights use in practice. To best evaluate the predictive accuracy of our methods over that class, however, we use OA-LHS to choose the evaluation points, due to its superiority over simple LHS in integral approximation.

In fitting each data set and making the predictions, we either use the standard model with the α_k set to the value used in simulating the data, or we use the method outlined in the previous section. For the latter, we have some choices to make. The first is the basis functions f_i to include in (1). We use fifth order Legendre polynomials in each dimension. Specifically, we include all main effects up to order five, as well as all interactions involving two dimensions, in which the sum of the maximum power of the exponent in each dimension is constrained to be less than or equal to 5. For example, in two dimensions, this implies using terms involving $x_1, \ldots, x_1^5, x_2, \ldots, x_2^5, x_1 x_2, x_1 x_2^2, \ldots, x_1^4 x_2$. We have investigated the impact of the order of the polynomial for the regression functions and have found that fourth or fifth order polynomials are sufficient for most applications. We do not try here to adapt the order of the polynomial to each particular data set. In practice, we recommend routinely using a subset of the data and performing ordinary least squares regression for increasing degrees of the polynomial. The smallest degree of the polynomial where the fit is satisfactory is chosen to be the order used in the approach (more on this in Section 5.1).

The final choice is the value of the cutoff C in (10). Here we chose C such that the maximum proportion of off-diagonal elements was either 0.02 or 0.05, both of which reflect a high degree of sparsity, as we would often be required to specify in practice. The truncated power correlation function (8) is used; the Bohman function (7) gives very similar results.

The predictions are compared using two criteria. The first, sometimes called the Nash–Sutcliffe efficiency, is equal to

$$(11) \qquad \text{NSE} = 1 - \frac{\sum_{x \in \mathcal{X}_{\text{pred}}} (\hat{Y}(x) - Y(x))^2}{\sum_{x \in \mathcal{X}_{\text{pred}}} (Y(x) - \bar{Y})^2}.$$

Here, $\hat{Y}(x)$ represents the mean of the posterior predictive distribution for $Y(x)$ given the vector of observations $\mathbf{Y}$, while $\bar{Y}$ represents the mean of $\mathbf{Y}$. Predictions are made at the set of 500 hold out points, $\mathcal{X}_{\text{pred}}$. The second term of (11) is the ratio of an estimate of the predictive mean square error to the unstandardized variance of $Y(x)$. Thus, the NSE has an interpretation similar to R^2 in linear regression, insofar as it represents an estimate of the proportion of the variability in $\mathbf{Y}$ that is explained by the model, although measured on an out-of-sample test set. We estimate the average value of (11) across repeated samples by calculating it for each iteration of the simulation study and then averaging over iterations. We do not expect NSE to be better (larger) for our method compared to the standard method, since we know we are using a different model to fit than to generate the data. However, comparing this under various conditions can help us build intuition about when the proposed method will perform well. The second criterion we consider is the empirical coverage probability of the 95% prediction intervals, measured both across the range of the input space and across repeated samples.

We begin by comparing the NSE of the sparse method to that of the standard method. Figure 3 plots NSE under the eight different combinations of dimension, power α, and effective range for which the data were generated. First examine the NSE for the standard method, shown by the solid lines in each panel. It is
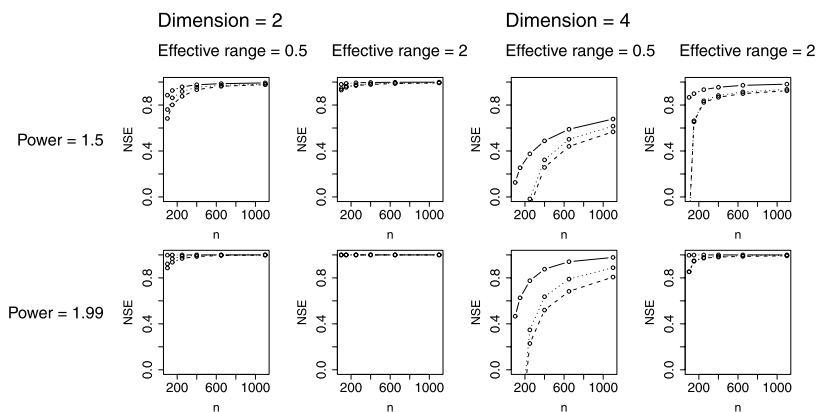

FIG. 3. *Nash–Sutcliffe efficiencies for predictions made using the posterior mean. In each panel, the solid lines corresponds to the data-generating model, while the dashed and dotted lines correspond to the proposed method with a maximum proportion of off-diagonal elements of 0.02 or 0.05, respectively.*

clear that the prediction task ranges quite a bit in difficulty across the range of eight conditions, from processes that are difficult to predict given the data (e.g., small NSE) to those which allow very high accuracy predictions (with NSE close to 1). The prediction problem is harder in four dimensions than in two, simply due to a lower data density. Also, not surprisingly, the smoother and flatter the process realizations, the easier they are to predict. That is, holding other variables constant, the standard method has higher NSE when the power is 1.99 rather than 1.5, and when the effective range is 2.0 rather than 0.5.

Now examine the NSE for the sparse method. Not surprisingly, within each panel the NSE is larger when the proportion of nonzero off-diagonal elements is allowed to be 5% (dotted line) rather than only 2% (dashed line). However, these differences are minor compared to the differences across the different processes. Another interesting trend that emerges is that, compared to the standard method, the sparse method tends to perform quite well as the sample size gets large. In most cases, by $n = 1{,}100$ the sparse method is still capturing about the same amount of the total variability as the standard method. The proposed method does perform relatively worse when the process is hard to predict, but even in these cases a large proportion of the variability is explained. In light of the fact that we have not expended any effort in determining the degree of the polynomial for each realization of the GP, the results are even more encouraging. For the large sample sizes for which the method was designed, we expect the NSE to be close to 1.

Next, consider the empirical coverage probabilities of the 95% prediction intervals. Figure 4 shows the observed coverage rates, under the same setup as in Figure 3. Not surprisingly, under the data-generating model, the coverage rates are close to the nominal rate of 95%. (The rates would be theoretically exact if we were not also estimating the range parameters.) Under the proposed model, the rates are
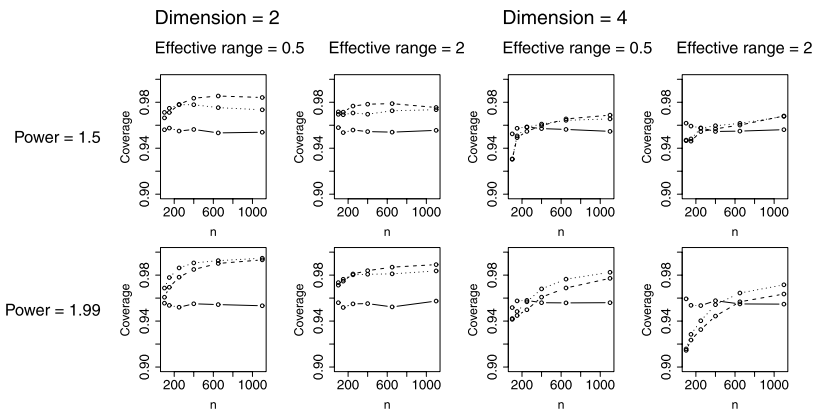

FIG. 4.   *Empirical coverage probabilities for pointwise credible intervals. In each panel, the solid line corresponds to the data-generating model, while the dashed and dotted lines correspond to the proposed method with a maximum proportion of off-diagonal elements of* 0.02 *or* 0.05, *respectively.*

often more conservative than the nominal rate, particularly when the input has only two dimensions. Although the widths of the intervals decrease with n, they do not decrease rapidly enough to maintain only 95% coverage, a fact that is attributable to the shorter correlation ranges being imposed for the sake of sparsity. This is a potential drawback to using the proposed model, although we remind the reader that the results under the data-generating model are not possible for large n; this is what motivates the approximation. These results should also not be taken to be representative of all simulators one might encounter in practice. In the application in Section 5, for example, exploratory analysis with a held-out sample suggests the coverage under the proposed model is very close to the nominal rate. In that example, posterior samples for the range parameters are well away from the boundary imposed for sparsity.

5. Application to photometric redshifts.   A major aim of upcoming cosmological surveys is to characterize the nature of dark energy, a mysterious type of energy that is driving a current epoch of acceleration in the expansion of the Universe [for a recent review, see Frieman, Turner and Huterer (2008)]. Evidence for cosmic acceleration first came from measurements of the optical luminosity from a specific class of supernovae [Riess et al. (1998), Perlmutter et al. (1999)]. In a matter-dominated Universe, the expansion rate should slow down with time, and the aim was to verify that the Universe was expanding more rapidly in the past by studying distant supernovae. Instead, observations indicated the Universe was doing exactly the opposite. So puzzling is this behavior that understanding dark energy—a hypothesized form of mass-energy accounting for the acceleration, or perhaps signaling the breakdown of general relativity on very large lengthscales—is considered to be one of the major unsolved problems in all of physical science.

Information about dark energy can be inferred in a variety of ways, some of which depend on an accurate three-dimensional representation of galaxies in a cosmological survey. It is straightforward to determine the angular location of an object in the sky, but the fundamental difficulty is determining the distance with sufficient accuracy. In large scale structure studies, the analogue of radial distance is the cosmological redshift. Due to cosmic expansion, the wavelength of light received from a distant object is stretched ("redshifted"), with more distant objects being at greater redshifts. Accurate redshift determination requires measurement of the spectrum of each galaxy at high resolution, but this is very difficult and too time-consuming for the very large numbers of very distant, hence very dim, galaxies. An alternative is to obtain galaxy photometry (flux measurements) in a restricted number of wavebands, and to attempt to reconstruct the redshift from just this information. Redshifts obtained in this way are termed "photometric" redshifts, in contrast to the more accurate "spectroscopic" redshifts. To obtain a photometric redshift estimate, however, requires estimating the functional relationship between the observations within the wavebands and the spectroscopic redshift.

The simulation we consider here models this relationship for four different wavebands. It simulates the true spectroscopic redshift and corresponding photometric measurements for the Dark Energy Survey [Abbott et al. (2005), Oyaizu et al. (2006)], which will come online in the near future. The training and validation data sets were generated to be of size 20,000 and 80,000, respectively. The design points were not chosen systematically, but were rather sampled from distributions meant to mimic what will be encountered in data from the Dark Energy Survey. The analysis in this article treats the simulation as the usual noiseless computer model case, with inputs corresponding to the flux measurements in the four wavebands and output corresponding to the spectroscopic redshift. There is some expected intrinsic scatter due to coarsening of information, potential degeneracies, and the fact that the predictors can be viewed as often having error themselves. Nonetheless, we expect that the GP will be able to efficiently predict the spectroscopic redshift from model inputs.

Our analysis proceeds in two steps. We first carry out an exploratory analysis on a small subset of the training data, so that we may explore the effects of various modeling choices before implementing these choices on the entire data set. Interestingly, we show here that the methods outlined in this paper in fact outperform the standard model in both predictive as well as computational efficiency. That is, the computationally efficient methods that allow us to scale up and work with the full data set do not come at a cost to predictive efficiency; they actually have *higher* Nash–Sutcliffe efficiency than the traditional, computationally infeasible methods. This gives us confidence in proceeding to the second step of the analysis, which is to fit the model on the set of 20,000 training points and evaluate the predictions at the 80,000 validation points.

5.1. *Preliminary analysis and model comparison.* We first normalize the input space so that each input variable lies within [0, 1]. We sample a smaller "training" and "validation" set from the full training set of 20,000; these have size 2,000 and 500, respectively, which is small enough to fit the standard model for comparison.

The first question to be addressed in this exploratory analysis is the choice of basis functions f_i in (1). As in the simulation study, we consider various tensor products of Legendre polynomials over [0, 1]. Let p represent the maximum degree, that is, the maximum power in a main effect for a single input dimension, or the maximum sum of powers in an interaction. Let m represent the maximum number of dimensions that may be involved in an interaction. Then a simple way to evaluate the choice of p and m is to find the ordinary least squares estimates $\hat{\beta}$ for each choice using the $n = 2{,}000$ training set and evaluate the Nash–Sutcliffe efficiency of predictions $X_0\hat{\beta}$ for X_0 consisting of the corresponding regression terms evaluated at the validation input set with $n = 500$. This is shown in Figure 5. The choice $p = 4$ and $m = 2$ produces the largest NSE, of 0.799, and produces relatively few regression terms ($q = 53$) compared to the sample size: here 2,000, later 20,000. Although matrix multiplication requires many fewer evaluations than
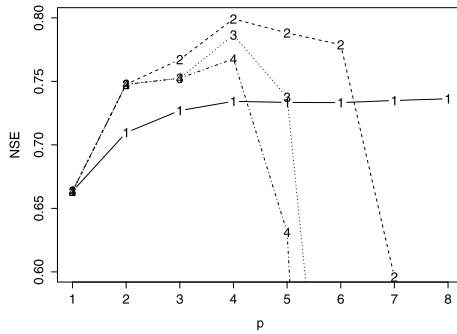
FIG. 5. *Nash–Sutcliffe efficiency* (*NSE*) *of predictions made using OLS estimates and various choices of maximum degree p and maximum interacting dimensions m* (*indicated by number on the plot*). *The choice p = 4 and m = 2 produces the highest NSE and includes relatively few regression terms* (*q = 53*) *compared to the sample size.*

the solution of a linear system of equivalent size, it can still be problematic if both q and n are large, so it is a happy coincidence that the choice of p and m that has highest NSE for this data is also computationally efficient.

We now fit the model (1) with all combinations of the following choices:

- the covariance structure R in (2) set to either the power exponential function (6) or a product of truncated power functions (8) in each dimension,
- the parameter α in (6) and (8) set either to 1, 3/2 or 5/3 in each dimension,
- the mean including an intercept only ($q = 1$) or a regression on tensor products of Legendre polynomials as described above, with $p = 4$ and $m = 2$.

In all models, we used the prior specification $p(\sigma^2, \beta) \propto 1/\sigma^2$. We took the prior for $\boldsymbol{\phi}$ in (6) to be uniform over a hyper-cube, and, as specified in (10), we took the prior for τ to be uniform over a hyper-triangle. In particular, we took the cutoff $C = 0.185$, chosen so that the proportion of off-diagonal elements in the correlation matrix that were nonzero would be at most 2%. This imposes a very high degree of sparsity, and in this initial exploratory analysis we can determine the effect of this choice on the posterior distribution for τ.

For each of the twelve modeling combinations, we sampled from the posterior distribution for model parameters and used this to generate predictions and point-wise credible intervals for the validation set. Table 1 shows the Nash–Sutcliffe efficiency for each modeling combination, and Table 2 shows the empirical coverage probabilities. First note that the largest NSE in Table 1 is for the sparse correlation structure (8) with power $\alpha = 1$ and Legendre polynomials up to degree 4, followed closely by the same model with $\alpha = 3/2$. This relationship holds across all entries in rows three and four of the table, corresponding to a model with $q = 53$ regression terms. It is interesting to note that this trend reverses in the first two rows, corresponding to the intercept only model. This provides evidence that in

TABLE 1
Nash–Sutcliffe efficiencies for predictions made using the posterior mean

		Power = 1	Power = 3/2	Power = 5/3
Degree = 0	Nonsparse	0.791	0.773	0.757
	Sparse	0.702	0.659	0.617
Degree = 4	Nonsparse	0.839	0.818	0.761
	Sparse	0.849	0.848	0.843

our method, both components—compactly supported correlation structure as well as a more structured mean term—are needed to achieve good predictive accuracy. Table 2 shows that the empirical coverage of the prediction intervals is very close to the nominal 95% level when using regression terms and a sparse correlation. This again gives us confidence in our method.

There is one final comparison we make using this preliminary test set. This is for the sparse methods only, corresponding to rows two and four of the tables. Note that the prior distribution restricts $\sum_{j=1}^{4} \tau_j \leq C$, where we chose $C = 0.185$. Figure 6 shows the trace plots of this sum over iterations of the Metropolis sampler, discarding an initial burn-in period. Note another major implication of using the Legendre polynomials: it changes the posterior distribution for τ, so that the posterior distribution of the sum moves away from this boundary. In contrast, in the intercept only model, the posterior samples of τ are varying only slightly around the upper boundary of 0.185. (Note the difference in scales between rows in Figure 6.) Another way of saying this is that a cutoff of $C = 0.185$ is too small for the intercept only GP model to capture all the variability in the data set, but it is adequate to capture variability in the residuals after introducing the Legendre polynomials. This has implications both for mixing of the sampler, which is much better in the models with the regression terms, and for computational efficiency, as we see that the cutoff C can be reduced even further, as far as 0.16 when the power $\alpha = 3/2$. The smaller this cutoff, the larger the computational savings, as it allows us to rule out more pairs of input values before the MCMC even begins to run. For this reason, as well as observing that the Nash–Sutcliffe efficiency when $\alpha = 3/2$

TABLE 2
Empirical coverage probabilities for posterior predictive intervals

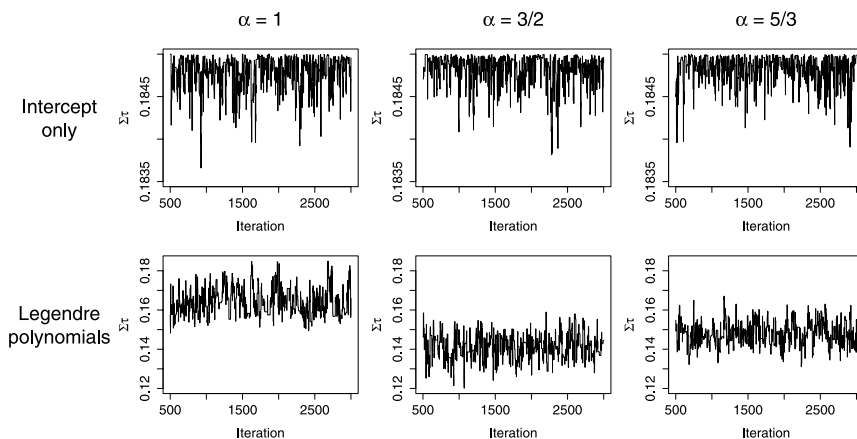|		Power = 1	Power = 3/2	Power = 5/3
Degree = 0	Nonsparse	0.936	0.928	0.908
	Sparse	0.960	0.950	0.948
Degree = 4	Nonsparse	0.952	0.926	0.906
	Sparse	0.954	0.952	0.952

FIG. 6. *Trace plots for $\sum_{j=1}^{4} \tau_j$ over iterations in the Metropolis sampler. Introducing regression terms into the mean of the Gaussian process (degree = 4) produces much better mixing, as well as the potential for increased computational efficiency.*

was only very slightly larger than the optimal one in Table 1, we choose to use this setting when working with the full data set.

5.2. *Full analysis.* In the second stage of the analysis, we fit the model using the full set of 20,000 training points and made predictions at each of the 80,000 validation points. We sampled $B = 3,000$ MCMC iterations using the Gibbs sampling algorithm described in Section 3 and the Appendix, discarding the first 500 for burn-in. For every tenth iteration thereafter, we calculated the conditional means and variances of the predictive distribution for $Y(x_0)$ given the observations and the parameter values at that iteration, from which we formed Monte Carlo approximations of the mean and variance of the posterior predictive distribution, as also described in the Appendix. The posterior means for the 80,000 new input values are shown in Figure 7, plotted against the actual spectroscopic redshift values. The Nash–Sutcliffe efficiency for the predictions is 0.831, and the empirical coverage rate for the 95% credible intervals is close to the nominal level, at 94%.

6. Discussion. In this article we have proposed new methodology for analyzing large computer experiments using a GP. The approach uses an anisotropic compactly supported covariance, as well as a regression function for mean, to emulate the computer model. With respect to the latter adaptation, we have proposed the use of a flat prior distribution for the regression parameters $\boldsymbol{\beta}$, using a preliminary study to select the set of basis functions to be used. This provides additional computational efficiency, as all parameters except $\boldsymbol{\tau}$ may be integrated out of the model. Given the model selection problem, which we solve in a rather ad-hoc way, one might also be tempted to incorporate a model selection or model averaging
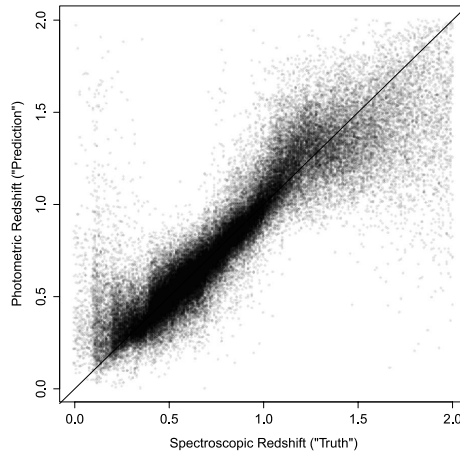
FIG. 7.   *Photometric redshift predictions and the corresponding spectroscopic redshift values from the simulator. Points are plotted with transparency, so that dark areas of the plot indicate a high density of points being over-plotted in the same region.*

approach. For example, one approach we examined was to err on the side of including more basis functions, but to use a shrinkage prior for $\boldsymbol{\beta}$, with

(12)
$$\beta_i | \xi_i \stackrel{\text{indep}}{\sim} N(0, \xi_i), \qquad i = 1, \ldots, p,$$

$$\xi_i \stackrel{\text{i.i.d.}}{\sim} IG(a, b), \qquad i = 1, \ldots, p.$$

The prior specification in (12) is akin to a generalized ridge regression, in which each coefficient receives its own shrinkage weight $w_i = \xi_i / (1 + \xi_i)$ [Denison and George (2000)]. Despite the elegance of this approach, we found that it contributed very little to the predictive efficiency of our method; it increased the Nash–Sutcliffe efficiency by only a few percentage points, and then only for small data sets, not the type that motivate our work. In our judgement, the added computational cost is not worth this small potential improvement. Instead, we advocate the simpler and computationally more efficient approach of choosing the basis functions based on a random subset of the data in a preliminary study, as described in Section 5.

We also note that the inclusion of the regression terms in the mean of the GP has implications for extrapolation beyond the range of the initial input values. Although polynomial regression can be problematic when it comes to extrapolation, it is unclear that the behavior obtained under a constant mean term is more desirable. In the standard model, when one is far from the initial input values, the GP prediction returns to the global mean and does not depend on the new inputs at all. Indeed, Bayarri et al. (2007) suggest using a more complicated mean structure to avoid this problem. The fact that neither approach is entirely satisfactory reflects the difficulty of extrapolation, and users should be aware of the implications of the structure of the mean term if it must be undertaken.

APPENDIX: POSTERIOR SAMPLING AND PREDICTION

To generate samples from the posterior distribution $p(\boldsymbol{\tau}|\mathbf{Y})$, we use an adaptive Metropolis algorithm, taking the transition density to be a multivariate normal random walk. At iteration i, we sample a candidate $\boldsymbol{\tau}^{\text{cand}}|\boldsymbol{\tau}^{(i-1)} \sim MVN(\boldsymbol{\tau}^{(i-1)}, \Sigma^{(i)})$, where $\Sigma^{(i)}$ is calculated adaptively using an algorithm to be described shortly. If the candidate value falls outside of the constrained parameter space T_C, as defined in (10), it is immediately rejected and we set $\boldsymbol{\tau}^{(i)} = \boldsymbol{\tau}^{(i-1)}$. Otherwise, we calculate the integrated likelihood $L^I(\boldsymbol{\tau}^{\text{cand}})$, where

$$L^I(\boldsymbol{\tau}) \propto |\boldsymbol{\Gamma}(\boldsymbol{\tau})|^{-1/2}|\mathbf{F}'\boldsymbol{\Gamma}(\boldsymbol{\tau})^{-1}\mathbf{F}|^{-1/2}(\hat{\sigma}^2(\boldsymbol{\tau}))^{(n-p)/2}$$

for $\hat{\sigma}^2(\boldsymbol{\tau}) = (\mathbf{Y} - \mathbf{F}\hat{\beta}(\boldsymbol{\tau}))'\boldsymbol{\Gamma}(\boldsymbol{\tau})^{-1}(\mathbf{Y} - \mathbf{F}\hat{\beta}(\boldsymbol{\tau}))$ and $\hat{\beta}(\boldsymbol{\tau}) = (\mathbf{F}'\boldsymbol{\Gamma}(\boldsymbol{\tau})^{-1}\mathbf{F})^{-1}\mathbf{F}' \times \boldsymbol{\Gamma}(\boldsymbol{\tau})^{-1}\mathbf{Y}$. The computationally demanding aspects of this calculation are described in Section 3. We then set $\boldsymbol{\tau}^{(i)} = \boldsymbol{\tau}^{\text{cand}}$ with probability $\max\{L^I(\boldsymbol{\tau}^{\text{cand}})/L^I(\boldsymbol{\tau}^{(i-1)}), 1\}$ and $\boldsymbol{\tau}^{(i-1)}$ otherwise.

We adapt the proposal covariance matrix using the Log-Adaptive Proposal algorithm of Shaby and Wells (2011), a slightly modified version of Algorithm 4 in Andrieu and Thoms (2008). The algorithm periodically updates an estimate of the posterior covariance matrix and then takes the proposal covariance matrix to be a scaled version of this, the scaling factor also being updated. This allows the sampler to target a particular acceptance rate and thereby increase efficiency. Although using previous states to generate the proposal violates the Markov property of the chain, the ergodicity of the process is maintained within a framework of "vanishing adaptation" [Roberts and Rosenthal (2009)].

After sampling from the posterior distribution for $\boldsymbol{\tau}$, the second step is to generate samples from the posterior predictive distribution for the output of the simulator at new input values. Let $\mathbf{Y}_0$ denote this new output. One could generate a sample from $p(\mathbf{Y}_0|\mathbf{Y}, \boldsymbol{\tau})$ at each iteration of the MCMC. However, since we are primarily interested in the mean and variance of the predictive distribution, we instead adopt the computationally more stable approach of calculating conditional means and variances at each iteration (i.e., conditional on $\boldsymbol{\tau}^{(i)}$) and using these to reconstruct the predictive mean and variance. Specifically, for a subset of the $\boldsymbol{\tau}$ samples (the number of which may be chosen based on an estimate of the smallest effective sample size among the elements of $\boldsymbol{\tau}$), we calculate the mean and variance of $\mathbf{Y}_0$ given $\mathbf{Y}$ and $\boldsymbol{\tau}^{(i)}$, to produce $\mathbf{m}^{(i)}$ and $\mathbf{v}^{(i)}$. As

$$\begin{pmatrix} \mathbf{Y} \\ \mathbf{Y}_0 \end{pmatrix} |\sigma^2, \boldsymbol{\beta}, \boldsymbol{\tau} \sim MVN\left(\begin{pmatrix} \mathbf{X} \\ \mathbf{X}_0 \end{pmatrix}\boldsymbol{\beta}, \sigma^2 \begin{pmatrix} \boldsymbol{\Gamma}(\boldsymbol{\tau}) & \boldsymbol{\gamma}(\boldsymbol{\tau}) \\ \boldsymbol{\gamma}'(\boldsymbol{\tau}) & \boldsymbol{\Gamma}_0(\boldsymbol{\tau}) \end{pmatrix}\right),$$

$p(\mathbf{Y}_0|\mathbf{Y}, \boldsymbol{\tau}^{(i)})$ is multivariate t, with $\mathbf{m}^{(i)} = \hat{Y}_0(\boldsymbol{\tau}^{(i)})$ and $\mathbf{v}^{(i)} = \frac{n-q}{n-q-2}\hat{\sigma}^2(\boldsymbol{\tau}^{(i)}) \times \mathbf{V}(\boldsymbol{\tau}^{(i)})$, for

$$\hat{\mathbf{Y}}_0(\boldsymbol{\tau}) = \mathbf{X}_0\hat{\beta}(\boldsymbol{\tau}) + \boldsymbol{\gamma}'(\boldsymbol{\tau})\boldsymbol{\Gamma}(\boldsymbol{\tau})^{-1}[\mathbf{Y} - \mathbf{X}\hat{\beta}(\boldsymbol{\tau})],$$

$$\hat{\beta}(\boldsymbol{\tau}) = (\mathbf{X}'\boldsymbol{\Gamma}(\boldsymbol{\tau})^{-1}\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\Gamma}(\boldsymbol{\tau})^{-1}\mathbf{Y},$$

$$\hat{\sigma}^2(\boldsymbol{\tau}) = \big(\mathbf{Y} - \mathbf{X}\hat{\beta}(\boldsymbol{\tau})\big)'\boldsymbol{\Gamma}(\boldsymbol{\tau})^{-1}\big(\mathbf{Y} - \mathbf{X}\hat{\beta}(\boldsymbol{\tau})\big)/\mathbf{n},$$

$$V(\boldsymbol{\tau}) = \boldsymbol{\Gamma}_0(\boldsymbol{\tau}) - \boldsymbol{\gamma}(\boldsymbol{\tau})'\boldsymbol{\Gamma}(\boldsymbol{\tau})^{-1}\boldsymbol{\gamma}(\boldsymbol{\tau}) + \mathbf{X}_0(\mathbf{X}'\boldsymbol{\Gamma}(\boldsymbol{\tau})^{-1}\mathbf{X})^{-1}\mathbf{X}_0.$$

Finally, we use Monte Carlo approximation to estimate $E[\mathbf{Y}(\mathbf{x}_0)|\mathbf{Y}]$ and $\mathrm{Var}[\mathbf{Y}(\mathbf{x}_0)|\mathbf{Y}]$ from the K samples according to

$$E[\mathbf{Y}(\mathbf{x}_0)|\mathbf{Y}] = E[E[\mathbf{Y}(\mathbf{x}_0)|\mathbf{Y}, \boldsymbol{\tau}]]$$

$$\approx \frac{1}{K}\sum_{i=1}^{K}\mathbf{m}^{(i)} \equiv \bar{\mathbf{m}},$$

$$\mathrm{Var}[\mathbf{Y}(\mathbf{x}_0)|\mathbf{Y}] = E[\mathrm{Var}[\mathbf{Y}(\mathbf{x}_0)|\mathbf{Y}, \boldsymbol{\tau}]] + \mathrm{Var}[E[\mathbf{Y}(\mathbf{x}_0)|\mathbf{Y}, \boldsymbol{\tau}]]$$

$$\approx \frac{1}{K}\sum_{i=1}^{K}\mathbf{v}^{(i)} + \frac{1}{K}\sum_{i=1}^{K}(\mathbf{m}^{(i)} - \bar{\mathbf{m}})^2.$$

These can be used to construct approximate pointwise credible intervals for $Y(\mathbf{x}_0)$ given $\mathbf{Y}$. If global intervals are needed, one should instead sample from the corresponding multivariate t conditional distributions at each iteration.

REFERENCES

ABBOTT, T. et al. (2005). The dark energy survey. Preprint. Available at Astro-ph/0510346.

AN, J. and OWEN, A. (2001). Quasi-regression. *J. Complexity* **17** 588–607. MR1881660

ANDRIEU, C. and THOMS, J. (2008). A tutorial on adaptive MCMC. *Stat. Comput.* **18** 343–373. MR2461882

BARRY, R. P. and PACE, R. K. (1997). Kriging with large data sets using sparse matrix techniques. *Comm. Statist. Simulation Comput.* **26** 619–629.

BAYARRI, M. J., BERGER, J. O., PAULO, R., SACKS, J., CAFEO, J. A., CAVENDISH, J., LIN, C.-H. and TU, J. (2007). A framework for validation of computer models. *Technometrics* **49** 138–154. MR2380530

BERGER, J. O., DE OLIVEIRA, V. and SANSÓ, B. (2001). Objective Bayesian analysis of spatially correlated data. *J. Amer. Statist. Assoc.* **96** 1361–1374. MR1946582

CRESSIE, N. A. C. (1993). *Statistics for Spatial Data*. Wiley, New York. MR1239641

DENISON, D. and GEORGE, E. (2000). Bayesian prediction using adaptive ridge estimators. Technical report, Dept. Mathematics, Imperial College, London, UK.

FRIEMAN, J. A., TURNER, M. S. and HUTERER, D. (2008). Dark energy and the accelerating universe. *Annual Review of Astronomy and Astrophysics* **46** 385–432.

FURRER, R., GENTON, M. G. and NYCHKA, D. (2006). Covariance tapering for interpolation of large spatial datasets. *J. Comput. Graph. Statist.* **15** 502–523. MR2291261

FURRER, R. and SAIN, S. R. (2010). spam: A sparse matrix R package with emphasis on MCMC methods for Gaussian Markov random fields. *Journal of Statistical Software* **36** 1–25.

GNEITING, T. (2001). Criteria of Pólya type for radial positive definite functions. *Proc. Amer. Math. Soc.* **129** 2309–2318 (electronic). MR1823914

GNEITING, T. (2002). Compactly supported correlation functions. *J. Multivariate Anal.* **83** 493–508. MR1945966

GOLUBOV, B. I. (1981). On Abel–Poisson type and Riesz means. *Anal. Math.* **7** 161–184. MR0635483

IRVINE, K. M., GITELMAN, A. I. and HOETING, J. A. (2007). Spatial designs and properties of spatial correlation: Effects on covariance estimation. *J. Agric. Biol. Environ. Stat.* **12** 450–469. MR2405534

KAUFMAN, C. G., SCHERVISH, M. J. and NYCHKA, D. W. (2008). Covariance tapering for likelihood-based estimation in large spatial data sets. *J. Amer. Statist. Assoc.* **103** 1545–1555. MR2504203

KENNEDY, M. C. and O'HAGAN, A. (2001). Bayesian calibration of computer models. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **63** 425–464. MR1858398

LINKLETTER, C., BINGHAM, D., HENGARTNER, N., HIGDON, D. and YE, K. Q. (2006). Variable selection for Gaussian process models in computer experiments. *Technometrics* **48** 478–490. MR2328617

MCKAY, M. D., BECKMAN, R. J. and CONOVER, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21** 239–245. MR0533252

OAKLEY, J. E. and O'HAGAN, A. (2004). Probabilistic sensitivity analysis of complex models: A Bayesian approach. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **66** 751–769. MR2088780

OYAIZU, H., CUNHA, C., LIMA, M., LIN, H. and FRIEMAN, J. (2006). Photometric redshifts for the Dark Energy Survey. In *Bulletin of the American Astronomical Society* **38** 140.

PAULO, R. (2005). Default priors for Gaussian processes. *Ann. Statist.* **33** 556–582. MR2163152

PERLMUTTER, S., ALDERING, G., GOLDHABER, G., KNOP, R., NUGENT, P., CASTRO, P., DEUSTUA, S., FABBRO, S., GOOBAR, A., GROOM, D. et al. (1999). Measurements of [Omega] and [Lambda] from 42 high-redshift supernovae. *The Astrophysical Journal* **517** 565–586.

PISSANETZKY, S. (1984). *Sparse Matrix Technology*. Academic Press, London. MR0751237

RIESS, A. G., FILIPPENKO, A. V., CHALLIS, P., CLOCCHIATTI, A., DIERCKS, A., GARNAVICH, P. M., GILLILAND, R. L., HOGAN, C. J., JHA, S., KIRSHNER, R. P. et al. (1998). Observational evidence from supernovae for an accelerating universe and a cosmological constant. *Astronomical Journal* **116** 1009–1038.

ROBERTS, G. O. and ROSENTHAL, J. S. (2009). Examples of adaptive MCMC. *J. Comput. Graph. Statist.* **18** 349–367. MR2749836

SACKS, J., WELCH, W. J., MITCHELL, T. J. and WYNN, H. P. (1989). Design and analysis of computer experiments. *Statist. Sci.* **4** 409–435. MR1041765

SANTNER, T. J., WILLIAMS, B. J. and NOTZ, W. I. (2003). *The Design and Analysis of Computer Experiments*. Springer, New York. MR2160708

SHABY, B. and WELLS, M. T. (2011). Exploring an adaptive Metropolis algorithm. Technical Report 2011-14, Dept. Statistical Science, Duke Univ., Durham, NC.

STEIN, M. L. (2008). A modeling approach for large spatial datasets. *J. Korean Statist. Soc.* **37** 3–10. MR2420389

STEIN, M. L., CHI, Z. and WELTY, L. J. (2004). Approximating likelihoods for large spatial data sets. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **66** 275–296. MR2062376

TANG, B. (1993). Orthogonal array-based Latin hypercubes. *J. Amer. Statist. Assoc.* **88** 1392–1397. MR1245375

WELCH, W. J., BUCK, R. J., SACKS, J., WYNN, H. P., MITCHELL, T. J. and MORRIS, M. D. (1992). Screening, predicting, and computer experiments. *Technometrics* **34** 15–25.

WIKLE, C. K. (2010). Low-rank representations for spatial processes. In *Handbook of Spatial Statistics* (A. E. Gelfand, P. Diggle, M. Fuentes and P. Guttorp, eds.) 107–118. CRC Press, Boca Raton, FL. MR2730946

C. G. KAUFMAN
DEPARTMENT OF STATISTICS
UNIVERSITY OF CALIFORNIA, BERKELEY
BERKELEY, CALIFORNIA 94720
USA
E-MAIL: cgk@stat.berkeley.edu

D. BINGHAM
DEPARTMENT OF STATISTICS
  AND ACTUARIAL SCIENCE
SIMON FRASER UNIVERSITY
V5A 1S6
BURNABY, BC
CANADA
E-MAIL: dbingham@stat.sfu.ca

S. HABIB
HIGH ENERGY PHYSICS DIVISION
MATHEMATICS AND COMPUTER SCIENCE DIVISION
ARGONNE NATIONAL LABORATORY
ARGONNE, ILLINOIS 60439
USA
E-MAIL: habib@lanl.gov

K. HEITMANN
ISR-1, ISR DIVISION
LOS ALAMOS NATIONAL LABORATORY
LOS ALAMOS, NEW MEXICO 87544
USA
E-MAIL: heitmann@lanl.gov

J. FRIEMAN
CENTER FOR PARTICLE ASTROPHYSICS
FERMI NATIONAL ACCELERATOR LABORATORY
BATAVIA, ILLINOIS 60510
USA
E-MAIL: frieman@fnal.gov