TECHNOMETRICS
2017, VOL. 59, NO. 2, 153–164
http://dx.doi.org/10.1080/00401706.2016.1164078

Emulation of Numerical Models With Over-Specified Basis Functions

Avishek Chakraborty[a], Derek Bingham[b], Soma S. Dhavala[c], Carolyn C. Kuranz[d], R. Paul Drake[e], Michael J. Grosskopf[b], Erica M. Rutter[f], Ben R. Torralva[e], James P. Holloway[g], Ryan G. McClarren[h], and Bani K. Mallick[i]

[a]Department of Mathematical Sciences, University of Arkansas, Fayetteville, Arkansas; [b]Department of Statistics and Actuarial Science, Simon Fraser University, Burnaby, Canada; [c]Perceptron Learning Solution Pvt. Ltd., Bengaluru, India; [d]Center for Radiative Shock Hydrodynamics, University of Michigan, Ann Arbor, Michigan; [e]Department of Atmospheric, Oceanic, and Space Sciences, University of Michigan, Ann Arbor, Michigan; [f]School of Mathematical and Statistical Sciences, Arizona State University, Tempe, Arizona; [g]Nuclear Engineering & Radiological Sciences, University of Michigan, Ann Arbor, Michigan; [h]Department of Nuclear Engineering Texas A&M University, College Station, Texas; [i]Department of Statistics, Texas A&M University, College Station, Texas

ABSTRACT
Mathematical models are frequently used to explore physical systems, but can be computationally expensive to evaluate. In such settings, an emulator is used as a surrogate. In this work, we propose a basis-function approach for computer model emulation. To combine field observations with a collection of runs from the numerical model, we use the proposed emulator within the Kennedy-O'Hagan framework of model calibration. A novel feature of the approach is the use of an over-specified set of basis functions where number of bases used and their inclusion probabilities are treated as unknown quantities. The new approach is found to have smaller predictive uncertainty and computational efficiency than the standard Gaussian process approach to emulation and calibration. Along with several simulation examples focusing on different model characteristics, we also use the method to analyze a dataset on laboratory experiments related to astrophysics.

ARTICLE HISTORY
Received January 2014
Revised February 2016

KEYWORDS
Calibration; Computer experiments; Emulators; Generalized polynomial chaos; Random inputs; Reversible jump Markov chain Monte Carlo

1. Introduction

Using deterministic computer models (or simulators) to explore physical systems is common in many scientific disciplines. Simulators are often used in *model calibration* (Kennedy and O'Hagan 2001) endeavors where realizations of the computer model and field observations are combined to estimate parameters governing the system and to build a predictive model.

Computer models are often computationally expensive and an emulator is used in its place. The traditional approach for computer model emulation uses a Gaussian process (*GP*) prior (Sacks, Schiller, and Welch 1989). For example, suppose that the computer model is exercised at n input locations $(x_1, x_2, \ldots, x_n)$, giving outputs $y_1, y_2, \ldots, y_n$. At any unsampled input x_0, the emulator's role is to predict the model output y_0. Sacks, Schiller, and Welch (1989) proposed viewing the computer model as a realization of a random function, f, so that $y = f(x)$, and the covariance of f is related to smoothness of the response. This is achieved by using a GP prior on f where, for input configurations x and x', the covariance between the responses is specified using a stationary correlation function ρ, scale σ_δ, and correlation parameter ν_δ such that covariance function looks like: $C(f(x), f(x')) = \sigma_\delta^2 \rho(x, x'; \nu_\delta)$. Typically, the function ρ is chosen to be the product of of univariate correlation functions—one for each input. In this case, ρ has a separable stationary form, that is, $\rho(x, x'; \nu_\delta) = \prod_{j=1}^{p} \rho_i(x_j - x'_j; \nu_\delta(i))$ for p-dimensional inputs. The one-dimensional correlation functions $\{\rho_i(d; \nu_\delta(i)) : i = 1, 2, \ldots, p, \}$ are usually chosen of the form $\exp(-\nu d^\alpha)$, where $\alpha = 1$ and $\alpha = 2$ correspond to the exponential and Gaussian correlation functions, respectively. More flexible choice for ρ involves nonstationary models where the correlation depends separately on x_j and $x_{j'}$ instead of their distance; see Paciorek and Schervish (2004). The main reasons for using a GP are that the estimated response surface interpolates the observed outputs and provides a foundation for inference in a deterministic setting, the latter point being most relevant to our work. The uncertainty at unsampled inputs reflects the different sample paths the GP can take—a type of model uncertainty.

The seminal article of Kennedy and O'Hagan (2001) outlined a hierarchical model for combining field observations and computer model output to estimate the values of unknown inputs (i.e., calibration parameters) and construct a predictive model for the physical system. Their approach (hereafter the KOH model) is centered on a GP emulator for the simulator and, potentially, another for the discrepancy between the computer model and field response surfaces. Conditional on the knowledge of the response surface(s), likely values of the calibration parameters are proposed. The hierarchical nature of the model, along with Bayesian posterior inference, allows these steps to be iterated sequentially so that the uncertainty in learning one part of their model is always factored into the other.

CONTACT Avishek Chakraborty ✉ ac032@uark.edu 🖃 Department of Mathematical Sciences, University of Arkansas, Fayetteville, AR 72701, USA.

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/r/TECH.
Supplementary materials for this article are available online. Please go to http://www.tandfonline.com/r/TECH.

A related problem is that of propagating the distribution of uncertainty in inputs across a computation model. The object of interest is the distribution of outputs. The most common approach is that of the use of polynomial chaos (Xiu and Karniadakis 2002) expansions. The so-called nonintrusive approach is most related to the problems we consider insofar as the numerical model is treated as a black box. This method relies on the simulator output at a set inputs that can be used to determine the expansion coefficients; see Xiu (2007).

Here, we replace the GP in the KOH model with a basis function expansion. In the context of computer model calibration, Higdon et al. (2008) used an expansion of univariate GPs to model the multivariate computer output. In the proposed approach, we explicitly use prior distributions in the inputs to select the appropriate family of functions. This can be achieved with the generalized polynomial chaos expansions (gPC; Xiu and Karniadakis 2002) that were originally used for variance propagation. The form of the polynomials is chosen to create an orthogonal family with respect to the input distribution, which implies faster convergence to the true simulator.

The model we propose in Section 2 specifies a stochastic model for the response surface and employs Markov chain Monte Carlo (MCMC) to explore the posterior predictive distribution of the calibration parameters. Bayesian inference with gPC has been discussed in the context of collocation (Marzouk and Xiu 2009). However, the approach therein evaluates the posterior distribution of the unknown input in the field data after approximating the basis coefficients with a set of plug-in estimates. This results in only a partial uncertainty quantification and underestimates uncertainty in determining the coefficients and sparsity of the expansion. Additionally, previous work on gPC-based uncertainty quantification was mostly based on use of a fixed-dimensional, truncated version of the series but, in Section 3, we present a reversible jump Markov chain Monte Carlo (RJMCMC; Richardson and Green 1997) sampler to adaptively select the constituent polynomials. This eliminates the need for subjective truncation, allowing the data to control sparsity and order of the expansion. Altogether, the proposed hierarchical model, coupled with the adaptive estimation scheme from Section 3, allows for a comprehensive uncertainty quantification. Finally, we incorporate a noise term in our model as well. This results in a noninterpolating emulator capturing the main, important features of the simulation function. In Section 2.2, we discuss foundational reasons for adding the noise specific to the context of this problem.

Comparing the traditional GP prior in KOH model, the proposed approach adds to the flexibility of the computer model framework in multiple ways: it can be used to model non-Gaussian output processes, can accommodate continuous and discrete inputs, and uses the information on the nature of randomness of the input to construct the functional form of the emulator. From a computational point of view, basis function-based emulators have a natural advantage over a GP because the former uses a linear mean structure to capture the input–output relationship in the data whereas the latter uses its covariance matrix—much more difficult to handle with large sample sizes (Kaufman et al. 2011). Furthermore, for computer model emulation, the GP acts as a prior on the function space for the simulator response surface. The proposed approach can be viewed in the same way. By proposing an oversized set of basis functions, we can control the complexity of the emulator by using a stochastic selection algorithm that can include or exclude a particular basis function based on the its contribution to the likelihood of the observed data. The uncertainty in our model arises from the possible set of basis functions that represent the simulator, conditional on the observed data. We have found the proposed approach particularly efficient, compared to GPs, when a significant number of runs from the simulator are available so that there is abundant information to identify the best subset of functions with high precision.

The article is outlined as follows. In Section 2, we propose a fixed-dimensional basis function representation of the emulator and discuss the appropriateness of gPCs. In Section 3, the benefit of a variable dimension emulator is explained along with a description of the MCMC scheme. We demonstrate the performance of the proposed method on simulated examples (Section 4) and on a real-world application (Section 5). We conclude the article with comments and future directions in Section 6. All runtimes mentioned in this article are based on an R (*http://cran.r-project.org/*) implementation on a single processor machine.

2. Emulating and Calibrating Computer Models Using Basis Functions

In this section, new methodology for emulating and calibrating simulators is proposed. We use a set of over-specified basis functions to emulate the computer model. The approach focuses on only a subset of the basis functions and predictions are made by averaging over the posterior distribution of the collections of basis functions and associated model parameters. Adaptive selection of the basis functions is discussed in Section 3.

The aim of model calibration is to combine field observations and simulator trials to estimate parameters that impact the physical process and also to build a predictive model for the system. There are two types of inputs to the simulator in a calibration experiment: (i) *calibration parameters* that are unknown in the field, but where a value must be specified to run the computational model; and (ii) *design variables* that are adjustable or measurable in the physical system. For example, in the real-world application in Section 5, gas pressure and laser energy represent the design variables because they are both measurable in the field. On the other hand, there are three calibration parameters (electron flux limiter, Be gamma, and wall opacity) corresponding to intrinsic material properties in the system whose values must be specified to run the simulator, but their values in the field are unknown and have to be estimated (i.e., a type of inverse problem).

Begin by considering the case where the input is only a vector of calibration parameters $\boldsymbol{\zeta} = (\zeta_1, \zeta_2, \dots, \zeta_q)^T$ (i.e., no design variables). Denote the physical system mean response by $G_R(\boldsymbol{\zeta})$ and the computer model response as $G(\boldsymbol{\zeta})$. Assume for the moment that the numerical model exactly describes the system dynamics and thus $G_R \equiv G$ (this assumption will be relaxed later in this section). In principle, if the value of $\boldsymbol{\zeta}$ were known in the physical system, the computer model could be run at this input value to acquire the process mean.

Suppose the simulator is evaluated at inputs $\boldsymbol{\zeta}_i^{(c)}, i = 1, \ldots, n$ (e.g., chosen by a Latin hypercube design (McKay, Beckman, and Conover 1979)). To estimate the response at unsampled inputs, we propose to write G in terms of a collection of basis functions over the support of $\boldsymbol{\zeta}$. The dependence between the simulator inputs and output is specified as

$$y_i^{(c)} = \beta_0 + \sum_j \beta_j \Psi_j \left(\boldsymbol{\zeta}_i^{(c)} \right).$$

This specification implies that the true process G_R belongs to the span of the chosen collection of functions $\{\Psi_j(\boldsymbol{\zeta})\}$, with certain regularity assumptions (i.e., we can write G_R as a linear combination of Ψ_j's). In principle, infinitely many of the Ψ_j's could be specified to model the response process; however, in practice this is not possible. Instead, we use only a finite number, k, of such functions. For example, if a response surface is exactly linear in its inputs, choosing Ψ_j as a first-order polynomial in $\boldsymbol{\zeta}$ would suffice. In more complicated settings, using a larger set of basis functions is necessary to satisfactorily approximate the true response. We use white-noise ϵ to represent any high-frequency variation that could be actually present in the response, but is unlikely to be detected by the model and is not accounted for by the k basis functions (more on this in Section 2.2). The proposed emulation model is

$$y_i^{(c)} = \beta_{k0} + \sum_{j=1}^{k} \beta_{kj} \Psi_j \left(\boldsymbol{\zeta}_i^{(c)} \right) + \epsilon_i^{(c)}, \quad \epsilon_i^{(c)} \sim N\left(0, \sigma_c^2\right), \\ i = 1, 2, \ldots, n, \tag{1}$$

where $\boldsymbol{\beta}_k = (\beta_{k0}, \beta_{k1}, \ldots, \beta_{kk})^T$ is the vector of (k+1) regression coefficients. In particular, β_{kj}, for $j \le k$, stands for the coefficient of jth basis function when k basis functions are used in the model. The variability of the unaccounted for dependence is represented by σ_c^2.

For model calibration, we also have a vector of physical observations $\boldsymbol{y}^{(r)}$, and the real-world value of $\boldsymbol{\zeta}$ needs to be estimated. Once we define prior distributions for $\boldsymbol{\zeta}$, $\boldsymbol{\beta}_k$, and σ_c^2 (more on this later), this can be done through the posterior distribution of $\boldsymbol{\zeta}$ given by

$$\pi\left(\boldsymbol{\zeta} | \boldsymbol{y}^{(r)}, y_{1:n}^{(c)}, k\right) \propto \pi\left(\boldsymbol{y}^{(r)} | \boldsymbol{\zeta}, y_{1:n}^{(c)}, k\right) \pi(\boldsymbol{\zeta}) \\ \propto \left[\int_{\boldsymbol{\beta}_k, \sigma_c^2} \pi(\boldsymbol{y}^{(r)} | \boldsymbol{\zeta}, \boldsymbol{\beta}_k, \sigma_c^2, k) \pi(\boldsymbol{\beta}_k, \sigma_c^2 | y_{1:n}^{(c)}, k) \, d\boldsymbol{\beta}_k d\sigma_c^2 \right] \\ \times \pi(\boldsymbol{\zeta}), \tag{2}$$

where $\pi(\boldsymbol{y}^{(r)} | \boldsymbol{\zeta}, \boldsymbol{\beta}_k, \sigma_c^2, k)$ denotes the likelihood of $y^{(r)}$ given the parameters and $\pi(\boldsymbol{\beta}_k, \sigma_c^2 | y_{1:n}^{(c)}, k)$ denotes the posterior distribution of $(\boldsymbol{\beta}_k, \sigma_c^2)$ conditional on the simulator runs.

We now relax the assumptions in the beginning of this section. First, the computer model is often not a perfect representation of reality and thus the discrepancy between the experiment and the simulator should be taken into account. This can be done by embedding the proposed model within the well-established KOH framework. Second, we also consider the more common scenario where there are both design and calibration inputs.

Let $\mathbf{x}$ be the p-dimensional design variables with input region $\mathcal{X}$. If $\mathcal{F}_j$ denotes the range of ζ_j, $1 \le j \le q$, and $\mathcal{F} = \otimes \mathcal{F}_j$, then $\mathcal{X} \times \mathcal{F}$ denotes the support for the inputs $(\mathbf{x}, \boldsymbol{\zeta})$. We use $\mathbf{x}_i^{(r)}$ and $\mathbf{x}_i^{(c)}$ to denote values of $\mathbf{x}$ at ith experiment and simulation, respectively. If m field observations are available, the joint model for $y^{(c)}$ and $y^{(r)}$ is specified as

$$y_i^{(c)} = f\left(\mathbf{x}_i^{(c)}, \boldsymbol{\zeta}_i^{(c)}\right) + \epsilon_i^{(c)}, \; \epsilon_i^{(c)} \sim N(0, \sigma_c^2), \; i = 1, 2, \ldots, n, \\ y_i^{(r)} = f\left(\mathbf{x}_i^{(r)}, \boldsymbol{\zeta}\right) + \delta(\mathbf{x}_i^{(r)}) + \epsilon_i^{(r)}, \; \epsilon_i^{(r)} \sim N(0, \sigma_r^2), \; i = 1, 2, \ldots, m. \tag{3}$$

In (3), f is the emulator mean, $\delta(\cdot)$ is the discrepancy function accounting for the systematic difference between the emulator and field observation response surfaces, and $\epsilon^{(r)}$ is observation error. KOH defined δ only over the space of design variables (i.e., to attempt to separately identify δ and its input $\boldsymbol{\zeta}$). This implies, $G_R(\mathbf{x}, \boldsymbol{\zeta}) - G(\mathbf{x}, \boldsymbol{\zeta})$ can only be a function of $\mathbf{x}$ and does not explicitly depend on $\boldsymbol{\zeta}$. The joint signal shared by the field observations and simulator outputs, f, is modeled with a basis expansion

$$f = \beta_0 + \sum_{j=1}^{k} \beta_j \Psi_j, \tag{4}$$

where β_0 is a constant and $\{\Psi_j\}$ is a family of basis functions in x and $\boldsymbol{\zeta}$. An important feature of our approach is the selection of the $\{\Psi_j\}$'s, and is described in the next subsection.

We assign conjugate prior distributions—Gaussian and Inverse-Gamma (not independent), respectively—to the location ($\boldsymbol{\beta}_k$) and scale (σ_c^2, σ_r^2) parameters, respectively. The prior distribution for $\boldsymbol{\zeta}$ is chosen based on the scientific knowledge (and previous studies, if available). A priori $\{\zeta_i\}$ are assumed to be independent variables, so the prior for $\boldsymbol{\zeta}$, defined over the $\mathcal{F}$, is of the form $\pi(\boldsymbol{\zeta}) = \prod \pi_i(\zeta_i)$. A zero-mean GP prior is used for $\boldsymbol{\delta}_m = \delta(\mathbf{x}_{1:m}^{(r)})$ with covariance function $C(\sigma_\delta^2, \nu_\delta)$ as in Section 1.

2.1 Construction of an Optimal Collection of Basis Functions

Consider, for the moment, a system with only a single calibration input ζ and output y. Let $\pi(\cdot)$ be the prior distribution for ζ over its support $\mathcal{F}$. If π has finite moments, there exists a sequence of orthonormal polynomials $\{\psi_i\}_{i=0}^{\infty}$ such that $\psi_i \in \mathcal{L}^2(\mathcal{F}, \pi)$ and $\int_{\mathcal{F}} \psi_i(\zeta)\psi_{i'}(\zeta)\pi(\zeta)\, d\zeta = 1$ if $i = i'$, 0 otherwise (Kubo, Kuo, and Namli 2007).

The computer model response can be written as a function of the random input: $y = G(\zeta)$. If G is assumed to have finite variance with respect to $\pi(\cdot)$, then it follows (Ernst et al. 2012) that G can be approximated in the L_2 sense as a linear combination of polynomial functionals $\{\psi_i(\zeta)\}$, that is, there exists $\boldsymbol{\beta}^0 = \{\beta_i^0\}$ so that for $G_k(\zeta) = \sum_{i=0}^{k} \beta_i^0 \psi_i(\zeta)$,

$$\int_{\mathcal{F}} |G(\zeta) - G_k(\zeta)|^2 \pi(\zeta)\, d\zeta \to 0 \text{ as } k \to \infty.$$

G_k is the kth degree gPC expansion of G, and $\boldsymbol{\beta}^0$ is the sequence of polynomial coefficients in the expansion. Using the orthonormality of ψ_i's we have $\beta_i^0 = \int G(\zeta)G_i(\zeta)\pi(\zeta)\, d\zeta$.

Orthonormality of polynomials with respect to the input measure implies (Marzouk and Xiu 2009) that the approximation converges at a rate depending on the smoothness α of G so that for some constant C, we have

$$\int_{\mathcal{F}} |G(\zeta) - G_k(\zeta)|^2 \pi(\zeta)\, d\zeta \; < Ck^{-\alpha}. \tag{5}$$

For $\alpha > 1$, we have established almost sure convergence of G_k toward G (see Appendix A of the online supplementary materials). In other words, when convergence in second moment occurs at a sufficient fast rate, almost sure convergence also holds. For some commonly used continuous and discrete distributions of random inputs, optimal choices for orthonormal sequence of polynomials are mentioned in table 4.1 in Xiu and Karniadakis (2002).

For a q-dimensional input $\boldsymbol{\zeta}$, a tensor product of univariate polynomials—$\Psi_i(\boldsymbol{\zeta}) = \prod_{j=1}^{q} \psi_{i_j}(\zeta_j)$, $i = 0, 1, 2, \ldots$—is used, where $\{\psi_{i_j}\}$ represents the sequence of orthonormal polynomials with respect to the prior distribution $\pi_j(\cdot)$ of ζ_j. When components of $\boldsymbol{\zeta}$ are assumed to be independent, orthonormality of $\{\psi_{i_j}\}$ with respect to $\pi_j(\zeta_j)$ implies orthonormality for $\{\Psi_i\}$ with respect to the product measure π. Eldred, Webster, and Constantine (2008) argued that this assumption can be justified if one thinks of $\boldsymbol{\zeta}$ as a transformation of the original input such that each coordinate of $\boldsymbol{\zeta}$ represents a distinct source of randomness. The assumption of priori independence among components of $\boldsymbol{\zeta}$ is also used in this article.

2.2 Further Discussion of the Model

The GP is used as an emulator largely because of its ability to interpolate and because it provides a foundation for statistical inference for deterministic computer models. The interpolation property can be relaxed by adding a pure error to the model, referred to as the *nugget*. The addition of a nugget term allows for smoothing that is useful when the data exhibit deviation from standard GP assumptions (Gramacy and Lee 2012) and also accounts for numerical jitter seen in many simulators. In our method, $\epsilon^{(c)}$ plays a similar role. We also argue that we do not necessarily need the model to interpolate since we already know the computer model response for the n computer model trials. Instead, our priority is to ensure that the emulator does a good job of prediction at unsampled inputs.

The case of statistical inference is more crucial. For the GP, prediction uncertainty at new inputs comes from the possible sample paths the random function can follow. The proposed emulator also provides a foundation for uncertainty quantification. Here, the sources of predictive uncertainty, conditional on the training data, arise from (i) selecting an appropriate set of k polynomials; (ii) estimating the coefficients of the polynomials; and (iii) high-frequency variation that we are unlikely to observe due to the sparseness of the sampling of the input space. Note that we can also view the Gaussian process model as an infinite dimensional basis-function expansion. For example, if we take the Karhunen-Loéve expansion (Loéve 1978) of a GP f with input x then, with known covariance parameters, it can be written as $f(x) = \sum_{k=1}^{\infty} w_k e_k(x)$, where the the basis functions $e_k(x)$ are orthogonal and the coefficients $\{w_k\}$ are independent, mean-zero normal random variables with variance of w_k equal to the kth largest eigenvalue λ_k (assume they are arranged in decreasing order). Here, the variability in the emulator comes from the uncertainty in the coefficients after conditioning on the data. That is, the source of variability at unsampled inputs comes from model uncertainty. It is a similar argument—for (i) and (ii) above—that we use to provide a foundation for inference from the proposed methodology. Moreover, if the Karhunen-Loéve expansion is truncated to exclude the high-frequency terms, then the sum of the excluded terms will be normally distributed (the errors will not be independent and identically distributed). In our case, the nugget in (iii) is analogous to the high-frequency variation that is excluded from the Karhunen-Loéve expansion via truncation.

On another note, the proposed gPC emulator is different from a GP with polynomial mean function since, for the former, the form of the polynomial is chosen depending on the type of stochasticity in the input to achieve optimality, whereas in the latter it is user-driven. Also, if we marginalize out $\{\beta_j\}$, this results in a nonstationary input-dependent covariance model for $y^{(c)}$ as follows:

$$\operatorname{cov}(y_i^{(c)}, y_{i'}^{(c)}) = \sigma_c^2 1_{i=i'} + \sigma_\beta^2 \sum_{j=1}^{k} \Psi_j(\mathbf{x}_i^{(c)}, \boldsymbol{\zeta}_i^{(c)}) \Psi_j(\mathbf{x}_{i'}^{(c)}, \boldsymbol{\zeta}_{i'}^{(c)}).$$

This type of nonstationarity is different from other nonstationary approaches such as specifying correlation models (Paciorek and Schervish 2004) or partitioning the input space (Gramacy and Lee 2008). Finally, we can make it more flexible if we let the data choose the component polynomials Ψ_j in an adaptive way that allows for differential treatment for different inputs based on their influence on the response. We do exactly that in Section 3.

3. Adaptive Selection of Chaos Terms

The proposed model lets the number of polynomial terms, k, be a parameter to be estimated. To do so, an adaptive scheme is proposed that has two objectives: (i) reducing the need to use a truncated model; (ii) allowing the model to include/delete terms similar to a variable selection method, encouraging sparsity whenever possible. To achieve this, the RJMCMC sampler of Richardson and Green (1997) is used to switch between different sets of polynomials. It follows from (ii) that, instead of selecting a truncation bound and considering all polynomials within that bound, it is more efficient to let the model choose the most important terms from a much larger pool of terms. A priori, one needs to make two choices: *which* subset of polynomials to choose from and *how many* of them to be allowed to enter the emulator expansion at a time.

3.1 Prior Support for the Hierarchical Model

There is an immediate theoretical advantage of treating k as a parameter. When we define a hierarchical model, we specify a probability distribution for the observations using parameters and/or processes and also assign prior distributions to the parameters. However, the true system need not resemble the structure of the model. So, it is desirable that the true system response surface lies within the *prior support* for the model, that is, there exists a set of realizations of these parameters and processes such that the probability distribution of the observations under the hierarchical model at those values is "close" to the probability distribution under the true data-generating mechanism.

Let (G_0, σ_{c0}^2) and $(\boldsymbol{\zeta}, G_{R0}, \sigma_{r0}^2)$ be the true processes and parameters for the simulation and experiment, respectively, such that $y^{(c)} \sim N(G_0(\mathbf{x}^{(c)}, \boldsymbol{\zeta}^{(c)}), \sigma_{c0}^2)$ and $y^{(r)} \sim N(G_{R0}(\mathbf{x}^{(r)}, \boldsymbol{\zeta}), \sigma_{r0}^2)$. Then, the full set of outputs, denoted by $Y^{(f)} = [\begin{smallmatrix} y_{1:m}^{(r)} \\ y_{1:n}^{(c)} \end{smallmatrix}]$, has a multivariate distribution that is actually determined by $\Delta_0 = (G_0, \sigma_{c0}^2, \boldsymbol{\zeta}, G_{R0}, \sigma_{r0}^2)$. Under the proposed model in (3), $\Theta = (\sigma_c^2, \sigma_r^2, \delta, k, \boldsymbol{\zeta}, \boldsymbol{\beta}_k)$ governs the distribution of $Y^{(f)}$; so it is important to investigate whether for any specific realization of Δ_0, there exists a subset in the domain of Θ such that $[Y^{(f)}|\Delta_0]$ and $[Y^{(f)}|\Theta]$ are sufficiently close. Theorem 1 asserts that they are indeed close under some assumptions on the true functions (the proof is included in Appendix B of the online supplementary materials).

Theorem 1 (Prior Support). Assume that

(A1) G_0 has finite variance and is of smoothness $\alpha > 1$.

(A2) $\mathcal{X}$ is a bounded subset of $\mathbb{R}^p$, $F_0(\mathbf{x}) = G_{R0}(\mathbf{x}, \boldsymbol{\zeta}) - G_0(\mathbf{x}, \boldsymbol{\zeta})$ is a continuous function of $\mathbf{x}$ only and free of $\boldsymbol{\zeta}$.

(A3) The correlation function of discrepancy process is chosen as $\rho(\mathbf{x}_1, \mathbf{x}_2; \nu_\delta) = \prod_{i=1}^{p} \rho_i(x_{1i} - x_{2i}; \nu_\delta(i))$, where ρ_i is a nowhere zero continuous, symmetric density function on $\mathbb{R}$.

Then, given $\epsilon > 0$, $\exists$ a set D such that $\Theta \in D \Rightarrow \mathrm{KL}([Y^{(f)}|\Delta_0] \,||\, [Y^{(f)}|\Theta]) < \epsilon$, where KL denotes the Kullback-Leibler distance metric between two probability distributions.

3.2 Parameter Estimation

Let us introduce some notations for specifying the sampling procedure from the posterior distributions. Let $\mathbf{x}_f = [\begin{smallmatrix} \mathbf{x}_{1:m}^{(r)} & \boldsymbol{\zeta}^T \otimes \mathbf{1}_m \\ \mathbf{x}_{1:n}^{(c)} & \boldsymbol{\zeta}_{1:n}^{(c)} \end{smallmatrix}]$. We reparameterize $\sigma_\beta^2 = \sigma_c^2 \sigma_\beta^2$ and $\sigma_r^2 = \sigma_c^2 \sigma_r^2$ for convenience in computing the integrals. Also define $D = [\begin{smallmatrix} \sigma_r^2 \mathbf{I}_m & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_n \end{smallmatrix}]$, $D_\delta = C(\sigma_\delta^2, \nu_\delta)$. It follows that the joint distribution of $Y^{(f)}$, conditional on $\boldsymbol{\beta}_k$ and $\boldsymbol{\delta}_m$, is (m+n)-dimensional multivariate normal (MVN) with parameters

$$E[Y^{(f)}] = \sum_{h}^{k} \beta_h \Psi_h[\mathbf{x}_f] + \begin{bmatrix} \boldsymbol{\delta}_m \\ \mathbf{0}_n \end{bmatrix}, \quad D[Y^{(f)}] = \sigma_c^2 D, \qquad (6)$$

where $\boldsymbol{\delta}_m \sim \mathrm{MVN}(\mathbf{0}, D_\delta)$ and $\Psi_h[A]$ represent the vector obtained by applying the polynomial Ψ_h on each row of the matrix A.

Marginalizing out $\boldsymbol{\beta}_k$ and σ_c^2, the probability distribution, $p(Y^{(f)}|k, \alpha_k, \ldots)$, can be written in closed form (see Appendix C in the online supplementary materials). This has two advantages. First, the form of marginal likelihood is computationally convenient as it does not involve $n \times n$ matrix inversion like the GP. Instead, as Appendix C shows, it only involves computation of determinant and inverse of a $k \times k$ matrix formed by the polynomial functions with a $\mathcal{O}(k^3)$ computational complexity. Second, as we can integrate out $\boldsymbol{\beta}_k$ from $p[Y^{(f)}| \ldots]$, this enables RJMCMC to move across different values of k without involving the k-dimensional coefficients. Being able to easily compare the marginal likelihood of the data for two choices of basis functions considerably increases the applicability of this method.

One key step in parameter estimation is defining a subset of candidate functions for constructing the gPC emulator. We propose to create a pool of functions $V_{r,s,t}$ as follows:

$$V_{r,s,t} = \{\{h_{i_j}\}_{j=1}^{p+q} : 0 \le h_{i_j} \le r, \\ \sum_j h_{i_j} \le t, \text{at most } s \text{ many } h_{i_j} \neq 0\}. \qquad (7)$$

The idea behind specifying $V_{r,s,t}$ is to make the space of available functions simpler and to prevent overfitting. Here, r is the maximum order of any ψ_i coming from a single input, s controls the number of factors in an interaction, and t puts a bound on the combined order of any Ψ_i. Hence, $r \le t$ and $s \le p + q$. When we have a single input ζ, $r = t$. As an example, a choice of $r = 4$, $s = 3$, and $t = 8$ implies: (i) main effects have degree 4 or less, (ii) there cannot be any interaction term that is a tensor product of polynomials of more than three inputs, and (iii) any interaction term cannot have a combined degree larger than 8.

Once $V_{r,s,t}$ is selected, a prior distribution on its members and a proposal distribution to move from one member to another is required. The movements can be of three types: adding a polynomial term (birth), deleting a polynomial term (death), or modifying a polynomial term. Subsequently, we compute the acceptance ratio and decide to accept/reject the proposed move. Given the functional form of the emulator, posterior sampling for other model parameters is relatively standard. We mention the entire MCMC scheme in detail in Appendix D of the online supplementary materials.

A key feature of the proposed approach is the ability to focus on simpler models. The reversible jump step for selecting polynomials involves a prior distribution on members of $V_{r,s,t}$. Three parameters control the overfitting: the Poisson prior for k, the upper bound for that prior k_0, and the order restriction on the polynomials induced by r, s, and t in (7). Conditional on k terms in the expansion, we use a uniform prior on every possible selection of k terms from $V_{r,s,t}$. Alternative prior specifications to emphasize even simpler models can be considered (see Appendix F of the online supplementary materials).

4. Simulation Studies

In this section, some of the properties of the proposed approach are investigated via simulation. The real-world application that motivated this work is presented in Section 5.

4.1 Effect of Basis-Set Size on Predictive Uncertainty

The change in predictive uncertainty as a result of an increase in the number of candidate basis functions is now investigated. For one-dimensional inputs, this amounts to increasing the value of r. As r increases, there are likely more choices of basis functions that can adequately fit the data, thereby increasing the model uncertainty. To see this, consider the following numerical model:

$$y(\zeta) = \zeta^3 + 4\sin(3\pi\zeta)/(1 + \zeta^2),$$

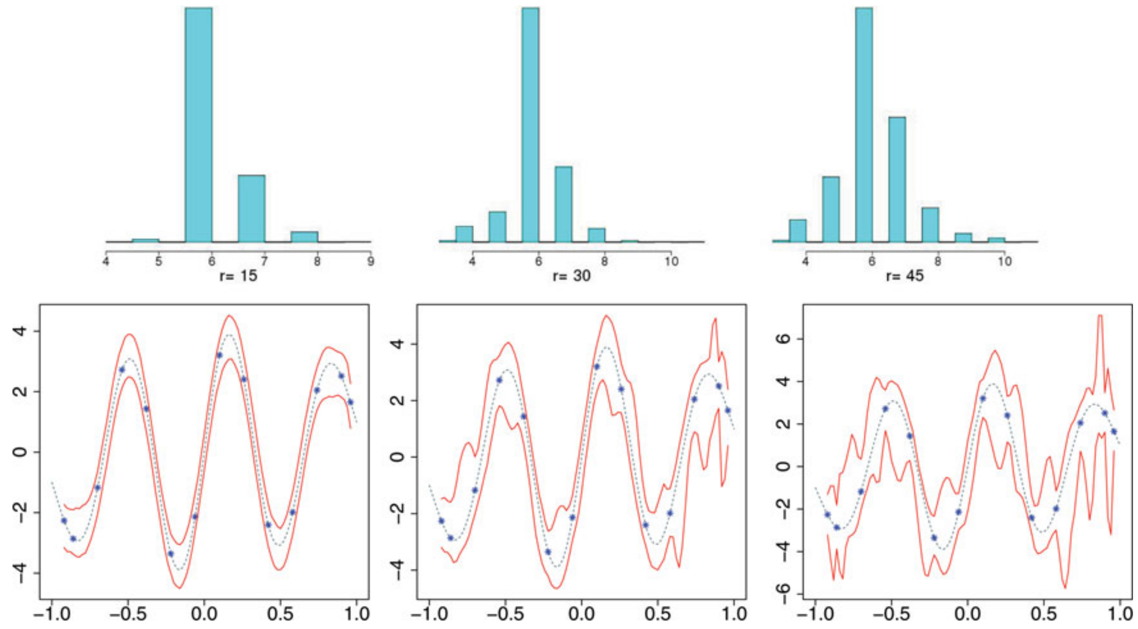

Figure 1. Uncertainty analysis for the adaptive emulator chosen from of the first $r =$ (left) 15, (middle) 30, and (right) 45 Legendre polynomials: (top row) the sampling variability in number of selected nonconstant functions k for each choice of r; (bottom row) posterior predictive uncertainty (90% credible intervals) for $y(\zeta)$ over the range of ζ, the blue dashed curves indicate the function $y(\zeta)$ and the blue stars represent the training data.

where the ζ is chosen uniformly over $[-1, 1]$. The function is evaluated at 14 different inputs that are used to emulate the model and compare the predictive uncertainty over a grid. We choose to work with a small training set to easily visualize any change in uncertainty due to changes in r. The proposed methodology is deployed with $r = 15, 30,$ and 45 Legendre polynomials. To explore the effect of r, we set $k_0 = r$, that is, we let the model choose all basis functions if required. For each value of r, we also look at the posterior sampling distribution of k.

Simulation results are shown in Figure 1. Comparing the diagrams in the top panels, the peak of $k = 6$ is similarly attained for each choice of r. In the bottom panel, we observe a tighter predictive interval with $r = 15$ than $r = 30$. Since there are fewer collections of basis functions, there is less model uncertainty. Setting $r = 45$ gives a wider interval due to increased in model uncertainty. As we keep increasing r, the prediction intervals for the proposed model begin to mimic the shape of uncertainty intervals for GP-based models.

So, when is working with a large number of basis functions beneficial? If we have knowledge about the complexity and shape of the computer model, this information will be useful to determine the necessary number of basis functions. In absence of specific knowledge, it does make sense that one should first attempt to determine some notion about the complexity of the response surface. This should be done in any case to choose the sample size for the computer experiment. A benefit of our approach, from a design and analysis perspective, is one explicitly considers the model complexity.

4.2 Predictive Performance of the Emulator

We move to comparative studies based on synthetic datasets from four numerical models. The first two models F1 and F2 have a two-dimensional input $\boldsymbol{\zeta} = (\zeta_1, \zeta_2)^T \in [-1, 1]^2$. The next two models F3 and F4 use a three-dimensional input $\boldsymbol{\zeta} = (\zeta_1, \zeta_2, \zeta_3)^T \in [-1, 1]^3$. The response functions are presented in Table 1. These four models cover a wide class of functions including logarithmic, exponential, trigonometric, irrational, as well as interactions between inputs. They are also fairly smooth and thus GP models should do relatively well.

For each of these, we simulate between 400 and 700 observations and leave out 100 randomly as *test* sets. Placing uniform prior distributions for the inputs, the appropriate orthogonal polynomial family is the Legendre polynomials. We fit our model with the training data and predict the value of the response over the test set. For the set of candidate polynomials, we use $V_{r,s,t}$ with $r = 10$, $s = 3$, and $t = 10$. The proposed gPC model is compared to a stationary GP emulator and a nonstationary alternative (treed Gaussian process (tGP; Gramacy and Lee 2008)). The GP and tGP methods were implemented using the "tGP" R package (Gramacy 2007) with constant mean, a separable power exponential correlation function, and a nugget. Each MCMC was run for 12,500 iterations, rejecting the first 20% of draws and thinning the rest at every fifth draw. The emulation approaches are compared using the following criteria: (i) absolute predictive bias: average absolute difference between the actual response and its posterior predictive estimate for each point in the test set; (ii) predictive uncertainty: width of 90% posterior credible set (between 5% and 95% quantiles of posterior samples) for each point in the test set; (iii) empirical coverage: the proportion of the test responses are contained inside their 90% credible sets; and (iv) runtime. Randomizing the selection of the test points, we replicate the simulation 50

Table 1. Response functions used for simulation studies.

Function	$y(\boldsymbol{\zeta})$	Function	$y(\boldsymbol{\zeta})$
F1	$\exp(\zeta_1 + \zeta_2)$	F3	$4\zeta_1^2\zeta_2^2\zeta_3^2 + \log(1 + \zeta_1\zeta_3)$
F2	$3\cos(2\pi\zeta_1) + 7\zeta_1^2 +$ $2\zeta_1 \log(1 + \zeta_2)$	F4	$2\zeta_1^2 + \sqrt{1 + \zeta_3}$

Table 2. Comparison of predictive performance and runtimes.

Simulation details						
Model	Data size	Method	Abs. predictive bias	Predictive uncertainty	Empirical coverage by 90% credible sets	Runtime
F1	300	gPC	1.15 $\times 10^{-2}$	0.065	0.982	$\sim$ 35 sec
		GP	10.83 $\times 10^{-2}$	0.180	1	$\sim$ 2.5 min
		tGP	1.45 $\times 10^{-2}$	0.185	1	$\sim$ 2.5 min
F2	400	gPC	1.25 $\times 10^{-2}$	0.068	0.958	$\sim$ 40 sec
		GP	4.07 $\times 10^{-2}$	0.562	1	$\sim$ 5 min
		tGP	3.06 $\times 10^{-2}$	0.456	0.999	$\sim$ 5 min
F3	500	gPC	1.76 $\times 10^{-2}$	0.093	0.935	$\sim$ 40 sec
		GP	8.97 $\times 10^{-2}$	1.143	1	$\sim$ 9 min
		tGP	6.69 $\times 10^{-2}$	0.996	0.999	$\sim$ 9 min
F4	600	gPC	1.41 $\times 10^{-2}$	0.094	0.947	$\sim$ 40 sec
		GP	2.82 $\times 10^{-2}$	0.470	1	$\sim$ 16 min
		tGP	2.94 $\times 10^{-2}$	0.443	1	$\sim$ 16 min

times. Table 2 provides a summary of the performance averaged over the replications.

Table 2 reveals generally better results for the proposed approach. Specifically, the average credible set width reduces by 75%–90% under gPC compared to the other approaches. Additionally, in spite of the significantly shorter prediction intervals, the 90% credible sets have always covered more than 90% of the test responses. From a computational perspective, the gPC-based emulator is relatively fast—about 40%–90% faster than a GP-based emulator. A closer look at those numbers reveals that, with an increase in size of the training dataset, the relative savings in runtime goes up rapidly. When large numbers of runs from a simulator are available, the computational advantage of the proposed emulator over GP will be beneficial. While computer models can be computationally challenging, experimenters may have moderate to large samples because of supercomputing resources (Kaufman et al. 2011; Goh et al. 2013).

The choice of k_0 (the maximum number of bases used at any iteration) is now investigated. As an illustration of what happens in practice, we simulate 500 observations from the model (F4) and fit it with four different values of $k_0 = 5, 20, 30, 45$, keeping everything else exactly same. Posterior samples of k, the number of nonconstant polynomial functions included in the model, are obtained under each value of k_0. The corresponding posterior histograms are shown in Figure 2.

Observe that with $k_0 = 5$, the proposed method mostly chooses models with five basis functions. This indicates that a larger number of basis functions is likely desirable. With $k_0 = 20$, the decaying right-tail behavior of the posterior distribution indicates that enough basis functions are present to sufficiently capture most of the variability. We decided to increase k_0 further to see if this pattern changes when more terms can be included. With k_0 set at 30 and 45, the behavior of the posterior distribution of k is essentially unchanged from $k_0 = 20$. That is, the model continues to put maximum probability in the range of four to eight basis functions. In general, we recommend going through the above exercise to determine a reasonable choice for k_0 for any dataset. Conservatively, we have found that choosing k_0 to be roughly twice the maximum number of functions selected in the MCMC iterations is a good default choice. This generally allows good fit to the data and also includes enough basis functions to incorporate a sufficient amount of model uncertainty.

All of the above models had inputs that impacted the output. However, in many computer experiments, there are a large number of redundant simulator inputs. We carried out another study using the Zhou (1998) function after adding inert inputs to the data analysis. We found that the tGP results in a smaller predictive bias compared to our method, but gPC achieves a significant reduction in predictive uncertainty—the 90% credible intervals, on an average, are approximately one-third to one-half as wide as those from tGP. We present the entire simulation study in Appendix E of the online supplementary materials.

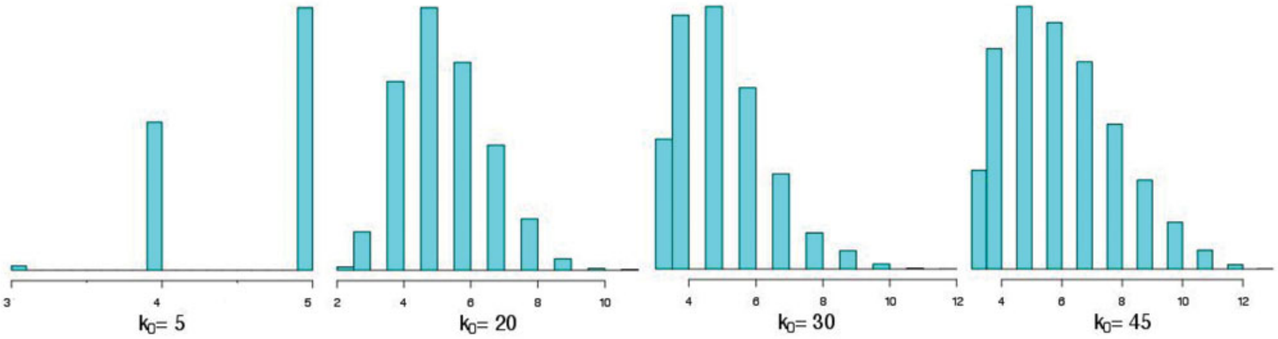

Figure 2. Posterior variability of number of nonconstant functions in the emulator under different choices of the bound k_0. The dataset has been simulated using the model in (F4).

4.3 Simulation Studies for KOH Setting

We extend our simulations to the model described in (3) that uses a KOH framework for combining simulation output with field observations. First, we look at predictive diagnostics for examples where the field experiment resembles the computer model without any discrepancy. Next, the proposed approach is investigated in simulations in presence of a discrepancy between the simulator and the mean of the field process. The latter example also illustrates the proposed model's ability to include inputs that are not continuous.

4.3.1 Models Without Discrepancy

We compare the predictive accuracy of the proposed methodology with the original GP-based KOH model when we have a combination of model outputs and field observations. Here, we assume no discrepancy between the two processes so $\delta = 0$ in (3). For the numerical models, we choose (F2) and (F3) from Section 4.2. For (F2), we select ζ_1 as the controlled input and ζ_2 as the calibration parameter. For (F3), we consider two variations: (i) (ζ_1, ζ_2) are used as design variables and ζ_3 as the calibration parameter; and (ii) ζ_1 is the design variable and (ζ_2, ζ_3) are chosen to be calibration parameters. The objective is to investigate how increasing the number of calibration parameters (that need to be estimated by the model) affects the uncertainty in prediction of field observations at new inputs.

We first simulate 125 observations from the computer model. Then, we fix a value of the calibration parameter and simulate 20 training points and 70 test points for field data by varying the design variable. The MCMC procedure involving Legendre polynomials from Section 3.2 is employed on the computer model and training data (without using the fixed value of calibration parameter for the "field" data) to predict the response at 70 test points. Same has been done with gPC replaced by the GP function, as in original KOH setting. We replicate the entire simulation procedure 30 times and, in Table 3, we provide a summary of the predictive performance averaged over the replications:

As observed in Table 3, use of gPC results in considerable gain over GP with respect to predictive uncertainty in all cases without affecting the desired 90% coverage rate. The bias is comparable for F3 but is significantly smaller for F2. An interesting outcome of this analysis is the change in predictive performance of the model between the two cases of F3. There is significant increase in predictive uncertainty and bias when we change one of the controlled inputs ζ_2 to a calibration parameter. So, uncertainty in learning ζ_2 for the latter propagates to the prediction of responses at test points resulting in reduced empirical coverage proportion for 90% credible intervals. Another significant part of the simulation is comparison of the runtimes for both methods. In GP settings, every candidate for $\boldsymbol{\zeta}$ induces a (partial) change in the correlation matrix requiring additional computations. We can expedite the computation by using a component wise random-walk MH sampler instead of a multinomial draw, as discussed in Section 3.2. If we use the component-wise multinomial update of $\boldsymbol{\zeta}$ for GP as we have done for gPC, the runtime will increase by a large factor from what is observed above. Hence, gPC provides a computationally faster KOH implementation, with better prediction quality compared to GP.

4.3.2 Model With Discrepancy and Discrete Input

Now, we consider an example where the "field" model is different from the numerical model. For the choice of inputs, we specifically consider a scenario where GP-based emulator can *not* be employed. We start with the following numerical model:

$$y = 2.0 + 0.20x^2 + 1.5\log(1 + \zeta^2),$$

where we draw x using a uniform distribution on $[-1, 1]$ and ζ from a Poisson(4) distribution. We simulate 170 observations from this model using different combinations of x and ζ. Next, 20 training and 100 test observations are generated with a "real life" discrepancy $\delta(x) = 1.80(1 + \sin(2\pi x))^2$ impacting the above model. A measurement error with scale parameter 0.05 is also added to these "real" observations to match the setup of the shock experiment dataset in Section 5. Because ζ takes discrete values, a Gaussian process prior on the input cannot be used. (See Qian, Wu, and Wu (2009) and Zhou, Qian, and Zhou (2011) for a direction on how to define a GP prior when there are qualitative outputs in the computer model.) However, we can still use the gPC representation in (4). The gPC is constructed using Legendre polynomials for x and Charlier polynomials for ζ. We use two different models—with and without a model for discrepancy function as in (3). We replicate the entire simulation procedure 50 times and, in Table 4, we provide a summary of the performance averaged over the replications.

The model fit without the discrepancy adjustment results in larger bias and wider credible sets for test responses. This is expected because, in the simulation step, the discrepancy function was added to the mean of the response. Since we have not used a model for $\delta(x)$, the scale parameter σ_r of the error term

Table 3. Predictive diagnostics for "field" observations at new input configurations.

		Input		Prediction criteria			
Simulation model	Method	Design variables	Calibration parameters	Absolute bias	Uncertainty	Empirical coverage by 90% credible set	Runtime
F2	gPC	ζ_1	ζ_2	0.029	0.391	0.999	$\sim$ 5 min
	GP			0.157	4.083	0.999	$\sim$ 17 min
F3	gPC	ζ_1, ζ_2	ζ_3	0.071	0.459	0.964	$\sim$ 5 min
	GP			0.074	1.012	0.981	$\sim$ 17 min
F3	gPC	ζ_1	ζ_2, ζ_3	0.223	1.149	0.939	$\sim$ 8 min
	GP			0.217	1.435	0.959	$>$ 30 min

Table 4. Effect of accounting for discrepancy on predictive diagnostics.

Method	Prediction criteria		
	Absolute bias	Uncertainty	Empirical coverage by 90% credible sets
gPC with discrepancy	0.451	3.319	0.980
gPC without discrepancy	1.668	5.775	0.836

$\epsilon^{(r)}$ gets large to account for the additional signal not accounted for by the statistical model. As a result, the prediction intervals get wider. In Figure 3, we illustrate this decrease in predictive accuracy due to exclusion of discrepancy term for one of those 50 replications. It is clear that for a considerable number of test points, the credible interval for model without discrepancy underestimated the true responses. This happens because the simulation model for $\delta(x)$ is a nonnegative function of x.

5. Analysis of Radiative Shock Experimental Dataset

We turn to a real application based on an experimental dataset on radiative shocks. The experiment is connected to the investigation of how radiative shocks influence the observed emissions from supernova and how they affect the evolution of young supernova remnants (Chevalier 1997). It is possible to generate and observe radiating shock waves (whose behavior is relevant to that of astrophysical shock waves) in the laboratory by using high-energy lasers to shock and then accelerate a Be disk down a plastic tube filled with Xe gas; see Reighard et al. (2006) and Kuranz et al. (2013) for details. The quantity of interest is the time when the laser-driven shock wave first emerges from the Be disk, designated as the “shock breakout time.” This time ranged from about 200 to 500 picoseconds. These experiments were the subject of extended research using computer simulations (McClarren et al. 2011; Van der Holst et al. 2013). Some of these simulations used a two-dimensional Lagrangian radiation hydrodynamics code, referred to here as H2D. Our analysis used the results of one specific H2D run set, defined as follows. There are five inputs to the computational model: $p = 2$ design variables ($\mathbf{x}$): Be disk thickness (ranging from 10 to 21 microns), laser energy (ranging from 3403 to 3946 Joules); and $q = 3$ calibration inputs ($\boldsymbol{\zeta}$): electron flux limiter, Be gamma constant, and the wall opacity of the plastic tube. The first four of these inputs were found, in McClarren et al. (2011), to be important in a previous analysis that explored the relative importance of 15 factors. The wall opacity parameter was added as an additional source of uncertainty specific to our two-dimensional code. A set of 104 runs of the computer model was conducted to cover the five-dimensional input space. Stripling et al. (2013) described the design of this run set in detail. Additionally, we work with results from eight of the laboratory experiments.

The range of measurement inaccuracy (±10 picoseconds), added with another error of ±50 picoseconds arising from the experimental settings, is converted into a zero mean Gaussian error, with scale determined using the 3σ criterion. All the five inputs in this problem are known to lie within bounded intervals. Independent uniform prior distributions are placed over the intervals for the calibration parameters. Location-scale adjustments were applied to the input data to shift their values within $[-1, 1]$. To implement the proposed methodology, Legendre polynomials were used for all inputs.

We first evaluate how the proposed gPC methodology compares to the conventional GP-based model for emulating the computer model (we move on to calibration shortly). For all subsequent analysis, we choose $V_{r,s,t}$ with $r = 4$, $s = 3$, and $t = 8$. The GP model is chosen to have a constant mean and a stationary, anisotropic covariance function with an exponential form separable in its arguments, that is, $C(a, b) = \sigma^2 \exp(-\sum_{i=1}^{p+q} \nu_i |a_i - b_i|)$.

The inference proceeds as follows: we leave out 11 randomly chosen simulator runs as our *test* sample, then we build the

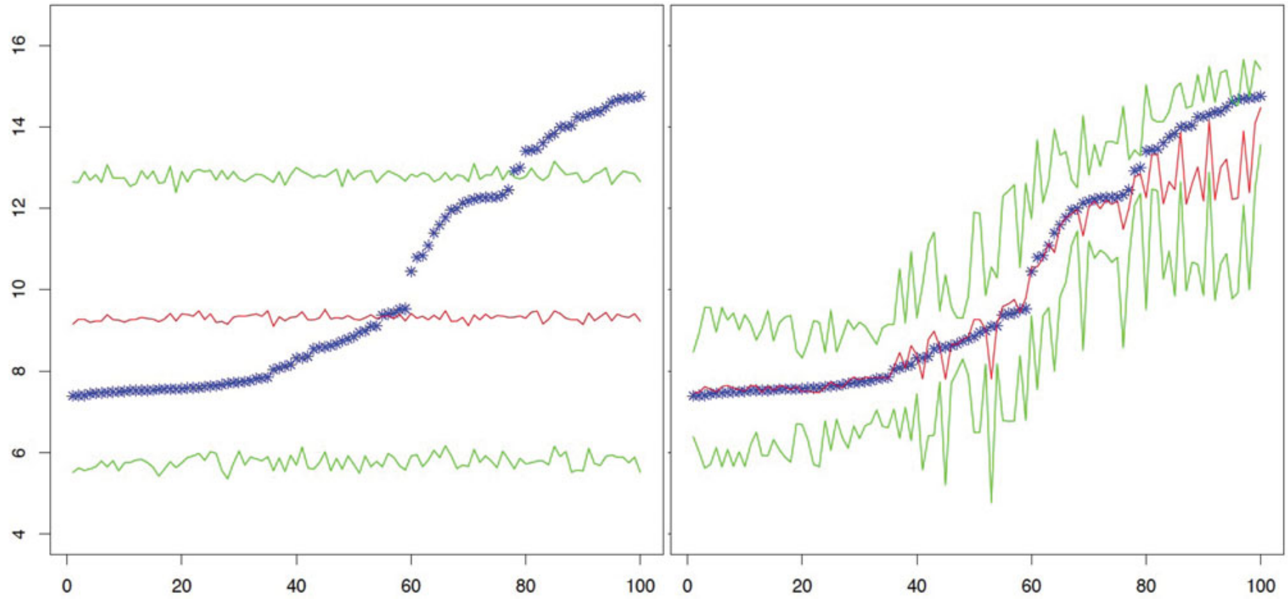

Figure 3. Comparison of predictive accuracy under gPC-based emulators (left) without discrepancy and (right) with discrepancy for one of the 50 replications. The blue stars indicate the 100 response values from the test set, the red line connects the posterior mean response at the test inputs, and the green lines represent the middle 90% posterior credible sets. The points are arranged in the increasing order of response values.

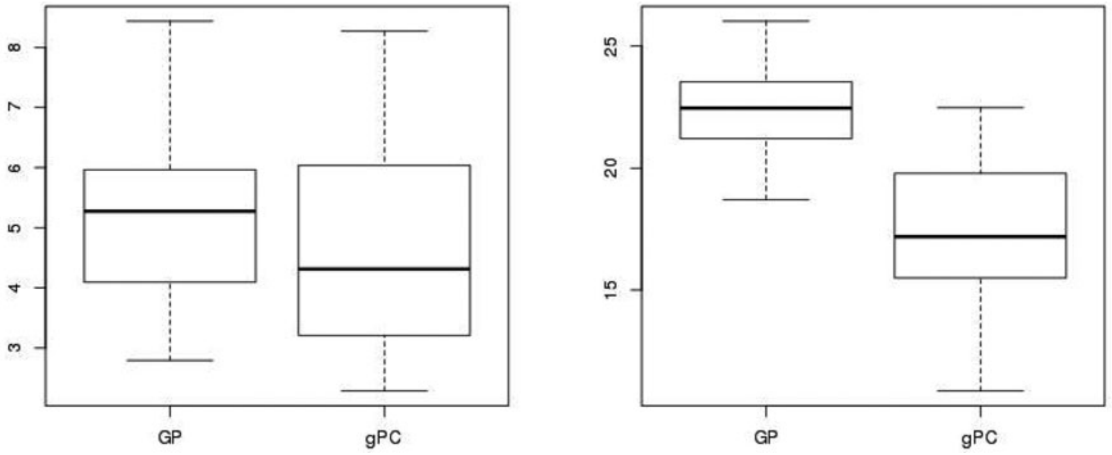

Figure 4. Model performance on (left) average absolute bias and (right) width of 90% credible interval for prediction (in picoseconds) of test samples for the H2D output.

emulator using the gPC or the GP model with the remaining 93 points as training data and obtain point estimate and 90% predictive interval for each point in the test sample. This completes one trial. Randomizing selection of the test samples, we perform 40 such trials. In Figure 4 and Table 5, we provide the summary measures of predictive performance for both the models—average absolute bias, predictive uncertainty (width of 90% posterior credible set), and empirical coverage rates. These diagnostics indicate that use of the gPC contributes to smaller bias as well as shorter predictive intervals without sacrificing the expected coverage rate for future predictions.

The results from the simulations and from the laboratory experiments can be simultaneously analyzed using the full hierarchical model from Section 3. First, we want to know if the H2D numerical model is a good substitute for the mean of the physical system. In Section 2, we have noted that the discrepancy process $\delta(\mathbf{x}^{(r)})$ captures any potential difference between the two systems. So, we have decided to fit the model with and without including the δ function. Here, we follow a leave-one-out validation procedure. Models are fit to remaining data (simulator and experimental outcomes together); point- and interval-estimates for the leave-one-out data are obtained from posterior draws, which were converted to estimates of absolute bias and uncertainty as before. Table 6 displays a summary for the results.

Table 5. Improvement in predictive performance (in picoseconds) for gPC over GP.

	Prediction criteria		
Method	Absolute bias	Uncertainty	Empirical coverage within 90% credible set
GP	5.120	22.174	0.902
gPC	4.736	17.403	0.900

Table 6. Summary of discrepancy analysis between numerical and experimental outputs.

	Prediction criteria (measured in picoseconds)	
Method	Absolute bias	Uncertainty
gPC with discrepancy	18.586	63.565
gPC without discrepancy	18.785	63.544

Table 7. Posterior summaries for components of ζ.

Summary statistics	Be gamma	Wall opacity	Flux limiter
Mean	1.432	0.995	0.060
90% credible interval	(1.402,1.469)	(0.725,1.262)	(0.050,0.072)

Adding discrepancy term does not cause any significant change in bias or prediction uncertainty. It is evident that the H2D code adequately represents the dynamics of an actual shock breakout experiment, so in future it can be relied upon for estimating the shock breakout times in the explored experimental ranges. Using the hierarchical model without discrepancy, the calibration parameters were also investigated (ζ). The empirical summaries of the posterior distributions of the calibration parameters are reported in Table 7. The corresponding posterior histograms are presented in Figure 5.

The posterior distributions of Be gamma in the above model looks to be most informative, as it is concentrated only over a small subregion of its prior range. Flux limiter is reasonably well constrained, but little is learned about the wall opacity. The posterior histograms also suggest that future studies should consider a larger range of Be gamma and flux limiter to see of the potentially smaller values of the parameters are appropriate.

6. Discussion

We have presented a full Bayesian approach to emulate and calibrate a computational model using chaos-based expansions. This specification uses the stochastic information about the input to capture a wide range of output processes outside the Gaussian family. With an adaptive basis function expansion, integrating out the expansion coefficients results in an input-dependent marginal nonstationary covariance pattern in the emulator. Use of RJMCMC with conjugate hierarchical specification enables us to efficiently perform a model search over the space of candidate functions with varying complexity. Prediction at a new input configuration is computed as weighted combination of predictions from the selected models. This approach differs from the idea of choosing the "best" model to use for prediction. We recommend it over the latter for two reasons: (i) with many candidate functions, there are often multiple competing models with comparable performances (choosing only one

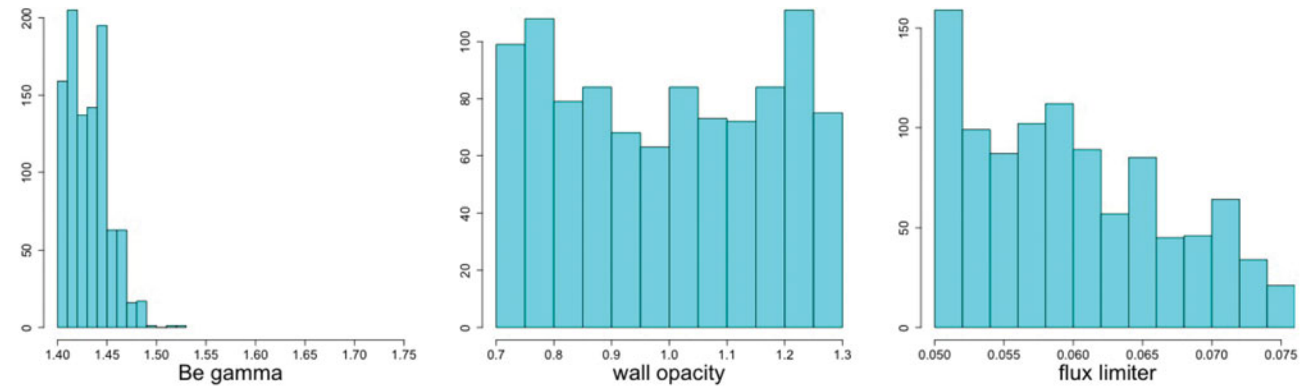

Figure 5. Posterior density estimates for components of ζ in the H2D dataset.

of them as the best and leaving everything else out amounts to over-relying on just one specification); and (ii) like GPs, including model uncertainty allows for complete uncertainty quantification.

A point of interest would be to know in what situations the use of the proposed methodology will be beneficial. Our method employs a selection approach from a large pool of functions. When the data size is small, as in Section 4.1, there is not sufficient information to identify the best subset of functions. As a result, the uncertainty increases. In that situation, use of an interpolator such as GP is recommended. However, with moderate or large number of data points, our method can provide a much improved predictive performance at a significantly faster runtime compared to any GP-based method and we strongly promote use of this methodology in similar situations. Another situation of interest is when the response is a function of several inputs but only a few of them are actually consequential. As our method finds an "effective" set of covariates $\{\Psi_j\}$ from the actual covariates by adaptively choosing important patterns, it will offer a sparse representation of the predictive model in those situations. There is scope for further research in this direction.

Supplementary Materials

Additional details: All appendices mentioned in this article (.pdf file)

R-code and dataset: The CRASH-UQ datasets used in Section 5 and R codes to fit gPC-based model (one uses only the computer model output and another uses the computer output and experimental outcomes simultaneously) (.zip file)

Acknowledgments

This work was funded by the Predictive Sciences Academic Alliances Program in DOE/NNSA-ASC via grant DEFC52-08NA28616. Research of Dr. Bani K. Mallick is supported by U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under Award Number DE-FG02-13ER26165.

References

Chevalier, R. A. (1997), "Type II Supernovae SN 1987A and SN 1993J," *Science*, 276, 1374–1378. [161]

Eldred, M. S., Webster, C. G., and Constantine, P. (2008), "Evaluation of Non-Intrusive Approaches for Wiener-Askey Generalized Polynomial Chaos," in *Proceedings of the 10th AIAA Non-Deterministic Approaches Conference*, Paper No. AIAA 2008-1892. [156]

Ernst, O. G., Mugler, A., Starkloff, H. J., and Ullmann, E. (2012), "On the Convergence of Generalized Polynomial Chaos Expansions," *ESAIM: Mathematical Modelling and Numerical Analysis*, 46, 317–339. [155]

Goh, J., Bingham, D., Holloway, J. P., Grosskopf, M. J., Kuranz, C. C., and Rutter, E. (2013), "Prediction and Computer Model Calibration Using Outputs From Multifidelity Simulators," *Technometrics*, 55, 501–512. [159]

Gramacy, R. (2007), "tgp: An R Package for Bayesian Nonstationary, Semiparametric Nonlinear Regression and Design by Treed Gaussian Process Models," *Journal of Statistical Software*, 19, 1–46. [158]

Gramacy, R. B., and Lee, H. K. H. (2008), "Bayesian Treed Gaussian Process Models With an Application to Computer Modeling," *Journal of the American Statistical Association*, 103, 1119–1130. [156,158]

——— (2012), "Cases for the Nugget in Modeling Computer Experiments," *Statistics and Computing*, 22, 713–722. [156]

Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008), "Computer Model Calibration Using High-Dimensional Output," *Journal of the American Statistical Association*, 103, 570–583. [154]

Kaufman, C. G., Bingham, D., Habib, S., Heitmann, K., and Frieman, J. A. (2011), "Efficient Emulators of Computer Experiments Using Compactly Supported Correlation Functions, With an Application to Cosmology," *The Annals of Applied Statistics*, 5, 2470–2492. [154,159]

Kennedy, M. C., and O'Hagan, A. (2001), "Bayesian Calibration of Computer Models," *Journal of the Royal Statistical Society*, Series B, 63, 425–464. [153]

Kubo, I., Kuo, H. H., and Namli, S. (2007), "The Characterization of a Class of Probability Measures by Multiplicative Renormalization," *Communications on Stochastic Analysis*, 1, 455–472. [155]

Kuranz, C. C., Drake, R. P., Huntington, C. M., Krauland, C. M., Stefano, C. A., Trantham, M., Grosskopf, M. J., Klein, S. R., and Marion, D. C. (2013), "Early-Time Evolution of a Radiative Shock," *High Energy Density Physics*, 9, 315–318. [161]

Loéve, M. (1978), *Probability Theory* (4th ed.), New York: Springer-Verlag. [156]

Marzouk, Y. M., and Xiu, D. (2009), "A Stochastic Collocation Approach to Bayesian Inference in Inverse Problems," *Communications in Computational Physics*, 6, 826–847. [154,156]

McClarren, R. G., Ryu, D., Drake, P., Grosskopf, M., Bingham, D., Chou, C., Fryxell, B., Van der Holst, B., Holloway, J. P., Kuranz, C. C., Mallick, B., Rutter, E., and Torralva, B. R. (2011), "A Physics Informed Emulator for Laser-Driven Radiating Shock Simulations," *Reliability Engineering and System Safety*, 96, 1194–1207. [161]

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979), "Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code," *Technometrics*, 21, 239–245. [155]

Paciorek, C., and Schervish, M. (2004), "Nonstationary Covariance Functions for Gaussian Process Regression," *Advances in Neural Information Processing Systems*, 16, 273–280. [153,156]

Qian, P. Z. G., Wu, H., and Wu, C. F. J. (2009), "Gaussian Process Models for Computer Experiments With Qualitative and Quantitative Factors," *Technometrics*, 50, 383–396. [160]

Reighard, A. B., Drake, R. P., Dannenberg, K. K., Kremer, D. J., Grosskopf, M. J., Harding, E. C., Leibrandt, D. R., Glendinning, S. G., Perry, T. S., Remington, B. A., Greenough, J., Knauer, J., Boehly, T., Bouquet, S., Boireau, L., Koenig, M., and Vinci, T. (2006), "Observation of Collapsing Radiative Shocks in Laboratory Experiments," *Physics of Plasmas*, 13, 082901–082901–5. [161]

Richardson, S., and Green, P. J. (1997), "On Bayesian Analysis of Mixtures With an Unknown Number of Components," *Journal of the Royal Statistical Society*, Series B, 0, 731–792. [154,156]

Sacks, J., Schiller, S., and Welch, W. (1989), "Designs for Computer Experiments," *Technometrics*, 31, 41–47. [153]

Stripling, H. F., McClarren, R. G., Kuranz, C. C., Grosskopf, M. J., Rutter, E., and Torralva, B. R. (2013), "A Calibration and Data Assimilation Method Using the Bayesian MARS Emulator," *Annals of Nuclear Energy*, 52, 103–112. [161]

Van der Holst, B., Tóth, G., Sokolov, I. V., Torralva, B. R., Powell, K. G., Drake, R. P., Klapisch, M., Busquet, M., Fryxell, B., and Myra, E. S. (2013), "Simulating Radiative Shocks With the CRASH Laser Package," *High Energy Density Physics*, 9, 8–16. [161]

Xiu, D. (2007), "Efficient Collocational Approach for Parametric Uncertainty Analysis," *Communications in Computational Physics*, 2, 293–309. [154]

Xiu, D., and Karniadakis, G. E. (2002), "The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations," *SIAM Journal of Scientific Computing*, 24, 619–644. [154,156]

Zhou, Q., Qian, P. Z. G., and Zhou, S. (2011), "A Simple Approach to Emulation for Computer Models With Qualitative and Quantitative Factors," *Technometrics*, 53, 266–273. [160]