Coupling Computer Models through Linking their Statistical Emulators*

Ksenia N. Kyzyurova†, James O. Berger‡, and Robert L. Wolpert‡

Abstract. Direct coupling of computer models is often difficult for computational and logistical reasons. We propose coupling computer models by linking independently developed Gaussian process emulators (GaSPs) of these models. Linked emulators are developed that are closed form, namely normally distributed with closed form predictive mean and variance functions. These are compared with a more direct emulation strategy, namely running the coupled computer models and directly emulating the system; perhaps surprisingly, this direct emulator was inferior in all illustrations. Pedagogical examples are given as well as an application to coupling of real computer models.

Key words. Gaussian process emulator (GaSP), Coupling, Computer model, System of simulators

AMS subject classifications. 60G15, 62F15, 62M20, 62P35, 86A04

1. Introduction.

Gaussian processes (GaSPs) have become a common tool for emulating (approximating) complex computer models. An example is [1], where an objective Bayesian implementation of a GaSP is used to approximate a computer model of a pyroclastic flow on a volcano, with the ultimate goal of identifying conditions which lead to hazardous events.

Sometimes more than one computer model needs to be utilized for the predictive goal. For instance, to model the true danger of a pyroclastic flow, one might need to combine the flow model (which can produce the flow size and force at a location) with a computer model that provides an assessment of structural damage, for a given flow size and force. Or to predict the danger from volcanic ash, one needs to combine a plume model that gives the magnitude and height of an eruption, together with a wind model that will predict its dispersion. Coupling of a system of models is used in many other important applications, e.g. climate modeling [22], oil fracturing simulation [19], and seismic activity modeling [13].

The specific context we consider is that of having a computer model $g(z_1, \ldots, z_d)$, the z_i being model inputs, at least some of which themselves arise from computer models, i.e., $z_i = f_i(\cdot)$, where $f_i(\cdot)$ is a computer model with its own inputs.

Direct coupling of computer models is often difficult, for both computational reasons and logistical reasons (e.g., the outputs of one model may not be completely compatible with the inputs of the other). Thus, in this work, we propose coupling computer models by first developing separate Gaussian process emulators for each model, and then linking the emulators through analytic methods; we will call this the *linked emulator*.

Another possible approach is to sequentially exercise the coupled computer models, obtaining input/output pairs, the inputs from the first model and the outputs from the coupled

*Submitted to the editors DATE.

Funding: This research was performed as part of the first author's PhD thesis at Duke University, and supported by NSF grants DMS-12-28317, EAR-1331353, DMS-14-07775 and DMS-16-22467.

†CEMSE division, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia (ksenia.kyzyurova@kaust.edu.sa, http://www2.stat.duke.edu/~kk194/).

‡Department of Statistical Science, Duke University, Durham, NC, 27708 (berger@stat.duke.edu, wolpert@stat.duke.edu).

model. A *composite emulator* is an emulator developed directly from this input/output data. While seemingly the natural way to emulate a coupled system, this approach cannot always be implemented; it might not be feasible to sequentially exercise the computer models, or one might only have previous separate runs of the models to deal with. This provided the original motivation for developing the linked emulator.

Perhaps surprisingly, we found in all our illustrations that the linked emulator actually performed better than the composite emulator (and often dramatically so) according to all evaluation measures used. The reason appears to be that coupled models are typically less smooth than the component models, making their emulation more difficult. This, of course, need not always be the case, but the fact that we encountered this in all our examples (many not shown in the paper) is revealing.

The approach to the problem of linking statistical emulators that we have taken originates from the work on sensitivity analysis of the output due to uncertain inputs [4, 16]. The straightforward approach to linking would be simply to do so by simulation [8]: for a given input to the first emulator, draw a sample from the GaSP emulator output, and then run this sample through the second emulator to obtain a sample from the linked emulator. This can become computationally expensive, however, especially because one often needs to perform an optimization or MCMC analysis involving randomness in the original emulator input. Alternatively, variational Bayesian methods [7] may be applied for finding a good approximation to the system. Other papers also work with individual models of coupled systems. For instance, [21] provides an excellent review of the uses of experiments on individual models in the overall task of verification and assessment of a coupled system; the paper does not consider emulators, however.

In this work, we seek a closed form expression for the linked emulator and its uncertainty. For certain GaSPs, one can give closed form expressions for the overall mean and variance of the linked emulator [6], and we generalize those results to the more complex situations considered herein. Unfortunately, the linked emulator itself does not have a simple distribution, so we simply approximate it by a normal distribution with the closed form mean and variance; this forms our recommended closed form linked emulator. The accuracy of the normal approximation is studied, empirically and with limited theoretical results, and seems to be very good.

Illustrations given in the paper include several pedagogical examples and an application to coupling of real computer models: coupling of *bent* – a computer model of volcanic ash plumes arising from a vent – and *puff* – a computer model of ash dispersion.

The model *bent* has four inputs: vent radius, vent source velocity, and the mean and standard deviation of ejected volcanic particles. The model solves for characteristics of the ensuing volcanic eruption column, in particular, giving the minimum and maximum height of the column, its width, and the size characteristics of ash particles in the plume (in terms of their means and standard deviations); denote these $d = 5$ outputs $f_1(\cdot), \ldots, f_5(\cdot)$. The outputs of *bent* act as inputs to the model *puff*, denoted as $g(f_1(\cdot), \ldots, f_5(\cdot))$, which solves for the ensuing ash cloud height at various space-time locations, based on a specified wind-field that disperses the ash. The schematic diagram of inputs and outputs of the coupled model is provided in Figure 1.

The outline of the paper is as follows. Section 2 gives a general description of the GaSP

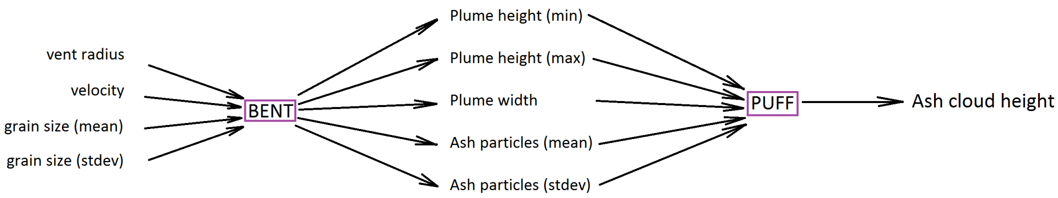

Figure 1. *The diagram of the composite coupled model of volcano eruption, bent-puff.*

emulator methodology. We present the linked emulator in section 3 and its linear approximation in section 4. An illustration of the approach is given in subsection 4.1. In section 5 we compare the linked emulator to the alternative composite emulator, directly constructed from the coupled computer models.

For computational reasons, uncertainty in the parameters of Gaussian process emulators is often ignored. Objective Bayes methodology provides a framework to overcome this problem, by providing analytically tractable full Bayesian inference [2]. Section 6 provides a description of the GaSP emulator within this framework, and the corresponding linked emulators are given. We present the bent-puff case study in section 7. We conclude with a discussion in section 8.

2. Preliminaries.

2.1. GaSP emulator of a computer model. Suppose a computer model g represents a smooth function $g(z)$, which takes input $z \in D \subseteq R^d$ (possibly, multidimensional, $d \geq 1$) and produces an output $g(z) \in R$. Suppose we observe m computer model outputs $(g(z_1), \dots, g(z_m))$, evaluated at corresponding inputs $\mathbf{z} = (z_1, \dots, z_m)$. From this set of inputs and outputs, assuming a Gaussian process prior on computer model data, one finds a probabilistic representation of an output of a computer model g at a new input z'.

A Gaussian stochastic process, $g^M(\cdot)$, is fully specified by its mean and covariance function. Given parameters, $\theta_{\mathbf{g}}$, of the GaSP, for any finite set $\mathbf{z} = \{\mathbf{z_1}, \dots, \mathbf{z_m}\}$ of d-dimensional inputs $\{\mathbf{z_i} = (z_{i1}, z_{i2}, \dots, z_{id})\}_{i=1}^m$, $\mathbf{g^M}(\mathbf{z}) = \{g^M(\mathbf{z_1}), \dots, g^M(\mathbf{z_m})\} = (g(\mathbf{z_1}), \dots, g(\mathbf{z_m}))$ has a multivariate normal distribution. That is,

$$\mathbf{g^M}(\mathbf{z}) \sim \mathrm{N}(\mu(\mathbf{z}), \sigma_g^2 K_z)\,,$$

where $\mu(\mathbf{z}) = (\tilde{\mu}(\mathbf{z_1}), \dots, \tilde{\mu}(\mathbf{z_m}))$ and $\tilde{\mu}(\cdot)$ is the mean function of the GaSP, σ_g^2 is the unknown variance and K_z is the correlation matrix whose (k, l) element is a correlation function $c(\mathbf{z_k}, \mathbf{z_l})$.

Sometimes the GaSP model has to be augmented with iid mean-zero Gaussian white noise ϵ to provide a more appropriate emulator [11] or for numerical stability of a GaSP [9]. Then, for any $\mathbf{z}$, $\mathbf{g^M}(\mathbf{z}) \sim \mathrm{N}(\mu(\mathbf{z}), \sigma^2 K_z + \tau^2 I)$. For convenience we reparametrize the model as

$$\mathbf{g^M}(\mathbf{z}) \sim \mathrm{N}(\mu(\mathbf{z}), \sigma_g^2 C_z)\,, \tag{1}$$

where $C_z = K_z + \eta I$, with K_z being a correlation matrix defined as before and η determining the ratio of the nugget variance τ^2 to σ_g^2.

We present the methodology for this general case of an emulator augmented with a nugget; note, however, that the results will also apply if the nugget effect is initially assumed to be zero (simply set $\eta = 0$ in the expressions).

Note that GaSP computer model run outputs, $\mathbf{g^M}(\mathbf{z})$, at a set of inputs $\mathbf{z}$, together with computer model run outputs, $\mathbf{g^M}(\mathbf{z}')$, at another set of inputs $\mathbf{z}'$, follow a joint multivariate normal distribution

$$\begin{pmatrix} \mathbf{g^M}(\mathbf{z}) \\ \mathbf{g^M}(\mathbf{z}') \end{pmatrix} \sim \mathrm{N}\left(\begin{pmatrix} \mu(\mathbf{z}) \\ \mu(\mathbf{z}') \end{pmatrix}, \sigma^2 \begin{pmatrix} C_z & c(\mathbf{z}, \mathbf{z}') \\ c(\mathbf{z}', \mathbf{z}) & C_{z'} \end{pmatrix} \right),$$

where $C_{z'}$ is the correlation matrix whose (k, l)th element is a correlation function $c(\cdot, \cdot)$ and a nugget component for diagonal elements, that is $c(\mathbf{z'_k}, \mathbf{z'_l}) + \eta \mathbb{1}_{k=l}$.

It follows that, conditional on the observed computer model evaluations $\mathbf{g^M}(\mathbf{z})$, the posterior predictive GaSP at any input $\mathbf{z}'$ (given GaSP parameters $\theta_\mathbf{g}$) follows a normal distribution with mean $\mu^*(\mathbf{z}')$ and variance $\sigma^{*2}(\mathbf{z}')$ given by

$$\mu^*(\mathbf{z}') = \mu(\mathbf{z}') + c(\mathbf{z}', \mathbf{z}) C_z^{-1} (\mathbf{g^M}(\mathbf{z}) - \mu(\mathbf{z})), \tag{2}$$

$$\sigma^{*2}(\mathbf{z}') = \sigma^2 (C_{z'} - c(\mathbf{z}', \mathbf{z}) C_z^{-1} c(\mathbf{z}, \mathbf{z}')). \tag{3}$$

Traditionally the GaSP mean function $\tilde{\mu}(\cdot)$ is chosen to be a linear model $\mathbf{h}(\cdot)\beta$, where $\mathbf{h}(\cdot)^T$ is a vector of regression functions [18] and $\beta \in R^n$ is a vector of unknown regression coefficients, i.e.,

$$\mathbf{h}(\cdot)\beta = \beta_0 h_0(\cdot) + \beta_1 h_1(\cdot) + \ldots + \beta_n h_n(\cdot).$$

The GaSP correlation function $c(\cdot, \cdot)$ is typically assumed to be in the form of a product of one-dimensional correlation functions along each dimension of the d-dimensional inputs. The correlation between outputs at two inputs $\mathbf{z_k}$ and $\mathbf{z_l}$ equals

$$c(\mathbf{z_k}, \mathbf{z_l}) = \prod_{j=1}^{d} c(z_{kj}, z_{lj}).$$

For the jth coordinate, the correlation is often assumed to be of the power exponential form

$$c(z_{kj}, z_{lj}) = \exp\left\{ - \left(\frac{|z_{kj} - z_{lj}|}{\delta_j} \right)^{\alpha_j} \right\},$$

with a range parameter $\delta_j \in (0, \infty)$ and a smoothness parameter $\alpha_j \in (0, 2]$ along each coordinate.

The correlation $c(\mathbf{z}, \mathbf{z}')^T = c(\mathbf{z}', \mathbf{z}) = (c(\mathbf{z}', \mathbf{z_1}), \ldots, c(\mathbf{z}', \mathbf{z_m}))$. For any two inputs $\mathbf{z_i}$ and $\mathbf{z}'$, the resulting power exponential correlation is $c(\mathbf{z}', \mathbf{z_i}) = \exp\left(- \sum_{j=1}^{d} \left(\frac{|z'_j - z_{ij}|}{\delta_j} \right)^{\alpha_j} \right)$, where d is the number of coordinates in input $\mathbf{z}$ and $j = 1, \ldots, d$ denotes one of the d coordinates in each $\mathbf{z_i} = (z_{i1}, \ldots, z_{id})$; $i = 1, \ldots, m$ denotes one of m inputs $\mathbf{z_1}, \ldots, \mathbf{z_m}$.

We will primarily consider the case $\alpha_j = 2$, $j = 1, \ldots, m$, as this is the most important scenario in which closed form expressions for the mean and variance of the linked emulator are available. This will be discussed in section 3.

Once all the parameters of the GaSP, $\theta_{\mathbf{g}}$, are specified, the conditional posterior predictive distribution is used for emulation of the computer model g.

2.2. Estimating parameters in the GaSP. It is common to just use maximum likelihood to estimate the GaSP parameters. However, these can be very problematical [12], and a better method is to develop estimates as posterior modes using an objective Bayesian implementation, as initially done in [3]. Follow up work in [10], [12] and [17] has led to the following recommendations for estimating the highly confounded parameters σ^2, η and $\{\delta_j\}_{j=1,\ldots,d}$ in the covariance function. First, transform to $\tilde{\delta_j} = -\alpha_j \log \delta_j$, for all $j = 1, \ldots, d$, and $\tilde{\eta} = \log \frac{\eta}{1-\eta}$. Then estimate these as the marginal posterior modes found by objective Bayesian analysis, using the reference priors that are available in the above references. Finally, transform back to obtain estimates of η and $\{\delta_j\}_{j=1,\ldots,d}$. All our analyses will be based on using these estimates.

For the parameters β and σ^2, however, there are several possibilities. One is to just use their maximum likelihood estimates, which are readily available; we will give resulting emulators the label *ML*. The second possibility is to perform a full objective Bayesian analysis with the mean parameters β, but use the maximum likelihood estimate of σ^2; such emulators we assign the label *POB*, for 'partial objective Bayes.' We discuss this choice in subsection 6.1. The third possibility is to perform a full objective Bayesian analysis for both β and σ^2; such emulators will be given the label OB, and will be discussed in subsection 6.2.

2.3. Predictive evaluations. Although some theoretical evaluations of studied emulators will be possible, for most of the paper the evaluations will be empirical. We will utilize three standard predictive criteria: empirical frequency coverage (EFC) of a function by credible intervals from the emulator, root-mean-square predictive error (RMSPE) and average length ($\mathrm{L_{CI}}$) of the credible intervals.

Let $\mathbf{u} = (u_1, \ldots, u_n)$ be n test points, for which the true value of a simulator $f(u_i)$ is known for each $i = 1, \ldots, n$. For each test point we find a predictive distribution $p_i \sim \mathrm{N}(\mu_i, \sigma_i^2)$ or $p_i \sim \mathrm{T_{df}}(\mu_i, \sigma_i^2)$ (needed for evaluation of later emulators), and form the 95% credible interval $\mathrm{CI_i} = (q_i^{0.025}, q_i^{0.975})$, where $q_i^{0.025}$ and $q_i^{0.975}$ are, respectively, the 2.5% and 97.5% quantiles of the predictive distribution p_i. Then the predictive criteria are defined as follows:

$$\mathrm{EFC} = \sum_{i=1}^{n} I_{Y_i \in \mathrm{CI_i}}/n,$$

$$\mathrm{RMSPE} = \sqrt{\sum_{i=1}^{n} (f(u_i) - \mu_i)^2/n},$$

$$\mathrm{L_{CI}} = \frac{\sum_{i=1}^{n} (q_i^{0.975} - q_i^{0.025})}{n}.$$

EFC, with the nominal value 95%, is the propotion of times the true function falls within the 95% credible intervals. RMSPE assesses the discrepancy between a simulator and an emulator's mean. $\mathrm{L_{CI}}$ is a measure of the stated accuracy of an emulator.

It is sometimes helpful to compare RMSPE with the reference quantity

$$\mathrm{RMSPE}_{\mathrm{base}} = \sqrt{\sum_{i=1}^{m}(f(u_i) - \overline{\mathbf{f}(\mathbf{z})})^2/m},$$

where $\overline{\mathbf{f}(\mathbf{z})}$ is the sample mean of the observed computer model outputs over the design points $\mathbf{z} = (z_1, \ldots, z_m)$ used to construct the emulator. $\overline{\mathbf{f}(\mathbf{z})}$ is, in some sense, the crudest possible emulator, so the ratio RMSPE/RMSPE$_{\mathrm{base}}$ measures the quality of the emulator being studied.

3. Linked emulator.

Suppose that we have two computer models, g and f, and have constructed their corresponding GaSP emulators, g^M and f^M, as described in subsection 2.1. Thus the GaSP emulator $g^M(\cdot)$, of the model g at any new input, given pairs $\{\mathbf{z}, \mathbf{g}(\mathbf{z})\}$ of model runs and GaSP parameters θ_g, is

$$g^M(\cdot) \mid \mathbf{g^M}(\mathbf{z}), \theta_{\mathbf{g}} \sim \mathrm{GaSP}(\mu_g^*(\cdot), \sigma_g^{*2}(\cdot,\cdot)). \tag{4}$$

Likewise, the GaSP emulator $f^M(\cdot)$, of the model f, given pairs $\{\mathbf{x}, \mathbf{f}(\mathbf{x})\}$ of model runs and its parameters θ_f, is

$$f^M(\cdot) \mid \mathbf{f^M}(\mathbf{x}), \theta_{\mathbf{f}} \sim \mathrm{GaSP}(\mu_f^*(\cdot), \sigma_f^{*2}(\cdot,\cdot)).$$

In this section, the GaSP parameters are assumed known. In practice, they will either be specified (in the case of the shape parameters) or estimated, following subsection 2.2. The expressions in this section then just apply with the estimates plugged in.

Suppose first that input z to g arises from the computer model f, so that we have the composite computer model $g \circ f$. Assuming we have the above emulators for each model, we can then define the associated emulator $g^M \circ f^M$. Actually, we are primarily interested only in the marginal distribution of this emulator, namely

$$\begin{aligned} &p((g\circ f)^M(\mathbf{u}) \mid \mathbf{g^M}(\mathbf{z}), \mathbf{f^M}(\mathbf{x}), \theta_{\mathbf{f}}, \theta_{\mathbf{g}}, \mathbf{u}) = \\ &\qquad \int p(g^M(f^M(\mathbf{u})) \mid \mathbf{g^M}(\mathbf{z}), f^M(\mathbf{u}), \theta_{\mathbf{g}}) p(f^M(\mathbf{u}) \mid \mathbf{f^M}(\mathbf{x}), \theta_{\mathbf{f}}) df^M(\mathbf{u}). \end{aligned} \tag{5}$$

Definition 3.1. *The variable* $\xi = (g\circ f)^M(\mathbf{u}) \mid \mathbf{g^M}(\mathbf{z}), \mathbf{f^M}(\mathbf{x}), \theta_{\mathbf{f}}, \theta_{\mathbf{g}}, \mathbf{u}$ *with the distribution* (5) *is called the linked emulator.*

We will sometimes use the shortcut notation for the linked emulator

$$p((g\circ f)^M(\cdot) = \int p(g^M(f^M(\cdot)))p(f^M(\cdot))df^M(\cdot). \tag{6}$$

More generally, as defined in subsection 2.1, g will have a d-dimensional input. Suppose that the first $b-1$ inputs do not arise from other computer models and hence do not need to be linked. (But they will still be part of the emulation of g.) The remaining inputs will result from computer models, f_j, for coordinates $j \in b, \ldots, d$. Assume, for each $j \in b, \ldots, d$, that we then construct a GaSP emulator, $f_j^M(\cdot)$, of the model f_j, given pairs $\{\mathbf{x^j}, \mathbf{f}(\mathbf{x^j})\}$ of model runs and parameters $\theta_{\mathbf{f_j}}$, as

$$f_j^M(\cdot) \mid \mathbf{f^M}(\mathbf{x^j}), \theta_{\mathbf{f_j}} \sim \mathrm{GaSP}(\mu^*_{f_j}(\cdot), \sigma^{*2}_{f_j}(\cdot,\cdot)). \tag{7}$$

Assuming independence of the $f_j^M(\cdot)$, the marginal distribution of the emulator of the composite computer model $g \circ (f_b, \dots, f_d)$, at a new input $\mathbf{u}$, is then

$$\begin{aligned} p((g \circ (f_b, \dots, f_d))^M(\mathbf{u}) \mid \mathbf{g^M}(\mathbf{z}), \mathbf{f_j^M}(\mathbf{x^j})_{j\in b,\dots,d}, \theta_{\mathbf{f_j}\, j\in b,\dots,d}, \theta_{\mathbf{g}}, \mathbf{u}) = \\ \int p(g^M(\mathbf{u^0}, f_b^M(\mathbf{u^b}), \dots, f_d^M(\mathbf{u^d})) \mid \mathbf{g^M}(\mathbf{z}), \mathbf{f_j^M}(\mathbf{x^j})_{j\in b,\dots,d}, \theta_{\mathbf{g}}) \\ \prod_{j=b}^{d} p(f_j^M(\mathbf{u^j}) \mid \mathbf{f_j^M}(\mathbf{x^j}), \theta_{\mathbf{f_j}}) df_b^M(\mathbf{u^b}), \dots, df_d^M(\mathbf{u^d}), \end{aligned} \tag{8}$$

where the new input $\mathbf{u}$ consists of the first $b-1$ inputs of the model g and the new inputs $\mathbf{u^j}$ are the inputs to the models f_j $\forall j \in b, \dots, d$, i.e. $\mathbf{u} = \cup(\mathbf{u^0}, \{\mathbf{u^j}\}_{j=b}^d)$ with $\mathbf{u^0} = (z_1, \dots, z_{b-1})$.

Definition 3.2. *The variable* $\xi = (g \circ (f_b, \dots, f_d))^M(\mathbf{u}) \mid \mathbf{g^M}(\mathbf{z}), \mathbf{f_j^M}(\mathbf{x^j})_{j\in b,\dots,d}$, $\theta_{\mathbf{f_j}\, j\in b,\dots,d}, \theta_{\mathbf{g}}, \mathbf{u}$, *with the distribution (8), is called the multivariate-input linked emulator.*

For brevity, in the rest of the paper we will simply write $g^M(\cdot)$ and $f^M(\cdot)$, without the conditioning, implicitly assuming the conditioning on relevant model run data and model parameters.

The linked emulator ξ does not have a closed form distribution. The key fact in developing an approximation is that, if the smoothness parameter of the power exponential correlation function is $\alpha = 2$ or $\alpha = 1$ and if $p(z)$ is one of three distributions – Normal, Laplace or Exponential – one can give closed form expressions for the overall mean and variance of the linked emulator, providing the regression functions in the GaSP mean $h(z)$ (and $h(z)^2$) have closed form expectations when $z \sim p(z)$. Typically $h(z)$ is either zero, constant, or linear in the inputs, in which case these expectations will be available in closed form.

The choice $\alpha = 1$ corresponds to the exponential covariance function, which yields GaSP sample paths that are not mean-square differentiable. In most applications, the computer model is a smooth function of the inputs, so $\alpha = 1$ is not usually a reasonable choice. On the other hand, $\alpha = 2$ corresponds to the squared exponential covariance function, which results in a GaSP with infinitely differentiable sample paths, and so is a better reflection of the smoothness of the computer model in applications. In the spatial statistics literature, $\alpha = 2$ is often criticized for producing too smooth sample functions, but the situation with computer models is different than that for most spatial models, in that the input points for the computer model runs are usually quite distant. GaSPs with squared exponential covariance functions have also proven to be useful for incorporating shape constraints such as monotonicity and convexity [23].

Theorem 3.3. *Suppose* g^M *has the linear mean function* $\mathbf{h}(\mathbf{z}')\beta = \beta_0 + \beta_1 z'_b$, *and a product power correlation function with* $\alpha_j = 2$ *for coordinates* $j \in b, \dots, d$. *For each* $j \in b, \dots, d$, *let*

f_j^M, as in (7), be an independent emulator of f_j, the function which gives rise to the value of input j for $g(\cdot)$. Then the mean $\mathrm{E}\xi$ and variance $\mathrm{V}\xi$ of the linked emulator ξ of the coupled simulator $(g \circ (f_b, \dots, f_d))(\mathbf{u})$, as defined in Definition 3.2, are

$$\mathrm{E}\xi = \beta_0 + \beta_1 \mu^*_{f_b}(\mathbf{u^b}) + \sum_{i=1}^{m} a_i \prod_{j=1}^{b-1} \exp\left(-\left(\frac{|u_j - z_{ij}|}{\delta_j}\right)^{\alpha_j}\right) \prod_{j=b}^{d} I_j^i,$$

$$\begin{aligned}\mathrm{V}\xi = {} & \sigma^2(1+\eta) + \beta_0^2 + 2\beta_0\beta_1\mu^*_{f_b}(\mathbf{u^b}) + \beta_1^2(\sigma^{*2}_{f_b}(\mathbf{u^b}) + (\mu^*_{f_b}(\mathbf{u^b}))^2) - (\mathrm{E}\xi)^2 + \\ & \left(\sum_{k,l=1}^{m} (a_l a_k - \sigma^2\{C_z{}^{-1}\}_{k,l}) \prod_{j=1}^{b-1} e^{-\left(\left(\frac{|u_j - z_{kj}|}{\delta_j}\right)^{\alpha_j} + \left(\frac{|u_j - z_{lj}|}{\delta_j}\right)^{\alpha_j}\right)} \prod_{j=b}^{d} I1_j^{k,l}\right) + \\ & 2\sum_{i=1}^{m} a_i \prod_{j=1}^{b-1} \exp\left(-\left(\frac{|u_j - z_{ij}|}{\delta_j}\right)^{\alpha_j}\right)\left(\beta_0 I_b^i + \beta_1 I{+}_b^i\right) \prod_{j=b+1}^{d} I_j^i,\end{aligned}$$

where $a = (a_1, \dots, a_m)^T = C_z^{-1}(\mathbf{g^M}(\mathbf{z}) - \mathbf{h}(\mathbf{z})\beta)$ and

$$I_j^i = \frac{1}{\sqrt{1 + 2\frac{\sigma^{*2}_{f_j}(\mathbf{u^j})}{\delta_j^2}}} \exp\left(-\frac{(z_{ij} - \mu^*_{f_j}(\mathbf{u^j}))^2}{\delta_j^2 + 2\sigma^{*2}_{f_j}(\mathbf{u^j})}\right)$$

$$I1_j^{k,l} = \frac{1}{\sqrt{1 + 4\frac{\sigma^{*2}_{f_j}(\mathbf{u^j})}{\delta_j^2}}} e^{-\frac{\left(\frac{z_{kj}+z_{lj}}{2} - \mu^*_{f_j}(\mathbf{u^j})\right)^2}{\frac{\delta_j^2}{2} + 2\sigma^{*2}_{f_j}(\mathbf{u^j})}} e^{-\frac{(z_{kj}-z_{lj})^2}{2\delta_j^2}}$$

$$I{+}_b^i = \frac{2\frac{\sigma^{*2}_{f_b}(\mathbf{u^b})}{\delta_b^2} z_{ib} + \mu^*_{f_b}(\mathbf{u^b})}{\sqrt{\left(1 + 2\frac{\sigma^{*2}_{f_b}(\mathbf{u^b})}{\delta_b^2}\right)^3}} \exp\left(-\frac{(z_{ib} - \mu^*_{f_b}(\mathbf{u^b}))^2}{\delta_b^2 + 2\sigma^{*2}_{f_b}(\mathbf{u^b})}\right).$$

The proof of a more general theorem, having mean $\mathbf{h}(\cdot)\beta$, is given in the appendix.

4. Linked GaSP as a normal approximation to the linked emulator. In this section we consider the normal approximation to the linked emulator, using its analytical mean and variance. After the definition, we present a numerical example and then some theoretical results.

Definition 4.1. *Whereas ξ in Definition Definition 3.2 was called the linked emulator, the variable $\zeta \sim \mathrm{N}(\mathrm{E}\xi, \mathrm{V}\xi)$ will be called the* linked GaSP.

4.1. Illustration 1. To illustrate the developed methodology of the linked GaSP, two functions are considered as simulators: $f(x) = 3x + \cos(5x)$, $x \in [-1, 1]$ and $g(z) = \cos(7z/5) - z$, $z \in [-4, 4]$, and we are interested in coupled model

$$g \circ f(x) = \cos(7[3x + \cos(5x)]/5) - [3x + \cos(5x)]\,.$$

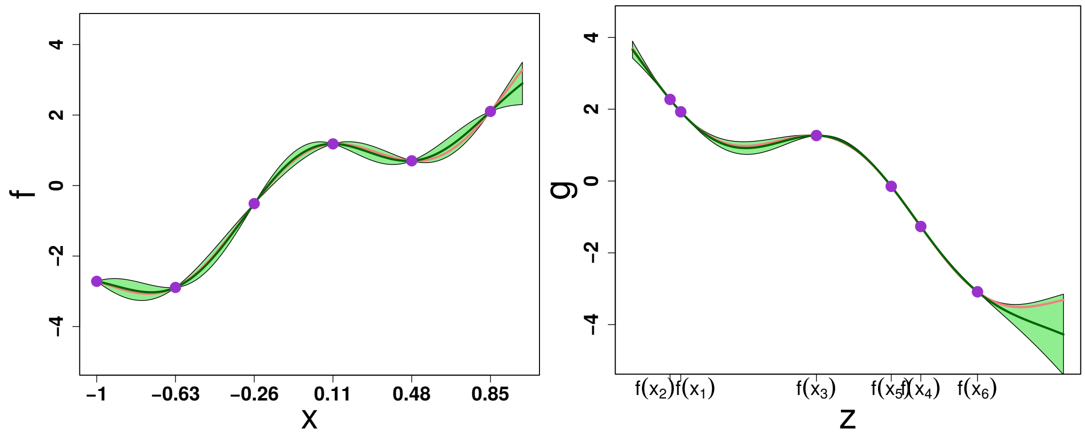

Figure 2. *Independent emulators constructed for two test functions: $f(x)$ on the left panel and $g(z)$ on the right. Each emulator is an interpolator at its design points. The pink lines are the true functions. The dark green lines are the emulator means. The green shaded regions are the regions enclosed by the 2.5% and 97.5% quantiles of the emulators. The circles on the plots correspond to the design points which were used to fit the emulators.*

Model f is evaluated at 6 equally spaced training input points $\mathbf{x} = (-1, -0.63, -0.26, 0.11, 0.48, 0.85)$, resulting in $\mathbf{z} = \mathbf{f}(\mathbf{x}) = (f(x_1), \ldots, f(x_6)) = (z_1, \ldots, z_6)$. An emulator $f^M(\cdot)$ of the model f is constructed, based on these observations $\{\mathbf{x}, \mathbf{z}\}$, as described in subsection 2.1.

The output points $\mathbf{z}$ are then used as design input points to the model g. (This utilization of the output of one model as input to the other was done so that this example can be used later to compare the linked emulator strategies with traditional coupling strategies; of course, the linked emulator strategies do not require using the outputs of one model as the design points for the other, one of the their big advantages.) Model g is evaluated at $\mathbf{z}$, resulting in $\mathbf{g}(\mathbf{z}) = (g(z_1), \ldots, g(z_6))$. We then use $\{\mathbf{z}, \mathbf{g}(\mathbf{z})\}$ to construct the emulator $g^M(\cdot)$ of the simulator g.

Parameter estimates of each of the GaSPs were obtained, using the methodology described in subsection 2.2, with the ML approach used for the mean and variance parameters. (We refrain from attaching the ML label to the emulators, until we later encounter emulators arising from other estimation methods.) The resulting emulators are shown in Figure 2.

Utilizing the individual function emulators, the linked GaSP (ζ) and linked emulator (ξ) were then determined, the latter through simulation. (After constructing the emulator $f^M(\cdot)$ of model f and the emulator $g^M(\cdot)$ of model g, one simply generates a realization from the emulator $f^M(\cdot)$ and then a realization from $g^M(\cdot)$ conditional on the realization from $f^M(\cdot)$; the result is a realization from the true linked emulator $(g \circ f)^M(\cdot)$.) Repeating this procedure many times results in a Monte Carlo representation of the true linked emulator.The results are presented in Figure 3. The linked GaSP is doing exceptionally well, acting as an interpolator at the design points to the simulator f and capturing the entire composite function $g \circ f(x)$ on $x \in [-1, 1]$ within its 95% credible area. Furthermore, it is indistinguishable from the linked emulator, providing support for the use of the normal approximation based on the known mean and variance.

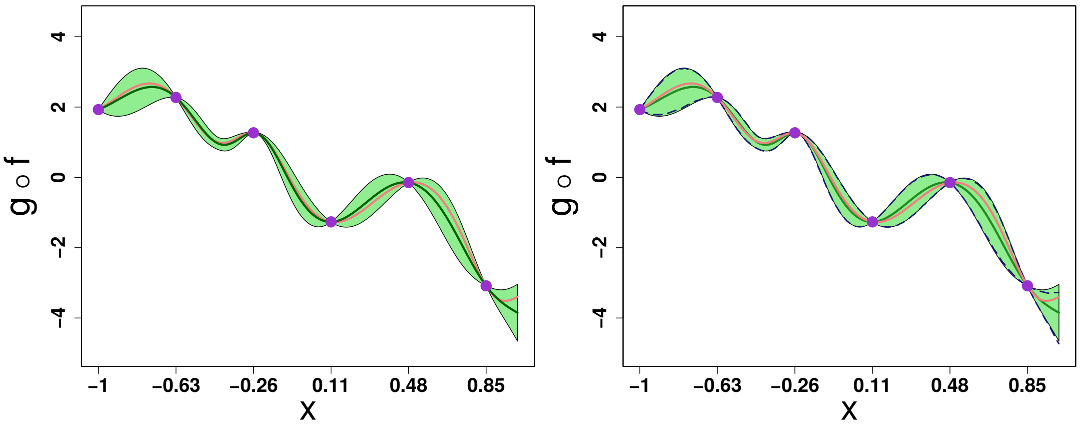

Figure 3. *The left panel is the linked GaSP constructed from $f^M(\cdot)$ and $g^M(\cdot)$, and the right panel is the linked emulator, estimated by simulation (10^4 samples). The pink lines are the true functions. The dark green lines are the emulator means. The green shaded regions are the regions enclosed by the 2.5% and 97.5% quantiles of the emulators. The dashed lines correspond to quantiles of the linked emulator. The circles on the plots correspond to the design points $\{\mathbf{x}, (\mathbf{g}\circ\mathbf{f})(\mathbf{x})\}$.*

4.2. Theoretical results. We consider the situation in which the variances of the $f_j^M(\cdot)$ are small, establishing theoretically (as expected) that the linked GaSP is then an excellent approximation to the linked emulator.

First, it is useful to find a first order approximation to the linked emulator.

Lemma 4.2. *If, for each $j \in b, \dots, d$, the emulators $f_j^M(\cdot)$ are as in (7) and $g^M(\cdot)$ is an emulator as in (4), then, for any new input $\mathbf{u}$,*

$$\mathrm{E}\left|\xi - \left(g^M(\mathbf{u^0}, \mu^*_{f_b}(\mathbf{u^b}), \dots, \mu^*_{f_d}(\mathbf{u^d})) + \sum_{j=b}^{d} \mu^{*'}_{g_{z_j}}(\mathbf{u^0}, \mu^*_{f_b}(\mathbf{u^b}), \dots, \mu^*_{f_d}(\mathbf{u^d}))\sigma^*_{f_j}(\mathbf{u^j})\mathrm{N}_{0,1}\right)\right|^2$$

$$= \sum_{j=b, k_j=1}^{d} \sigma^{*2''}_{g\ z_j z_j}(\mathbf{u^0}, \mu^*_{f_b}(\mathbf{u^b}), \dots, \mu^*_{f_d}(\mathbf{u^d}))\sigma^{*2}_{f_j} + O\left(\sum_{\substack{|K|=3,|L|=1;\\ |K|=|L|=2;\\ |K|=1,|L|=3}} \sigma^*_{f_j}(\mathbf{u^j})^{k_j+l_j}\sigma^*_{f_i}(\mathbf{u_i})^{k_i+l_i}\right),$$

where ξ is the linked emulator, $\mu^{*'}_{g_{z_j}}$ is the partial derivative of the function $\mu^*_g(\cdot,\dots,\cdot)$ with respect to the jth coordinate, $\sigma^{*2''}_{g\ z_j z_j}$ is the second-order partial derivative of the function $\sigma^{*2}_g(\cdot,\dots,\cdot)$ with respect to the jth coordinate and $\mathrm{N}_{0,1}$ is a standard normal random variable.

The proof of the lemma is given in the appendix.

The following are immediate consequences.

Theorem 4.3. *Under the same conditions and using Lemma 4.2 $\xi - \left(g^M(\mathbf{u^0}, \mu^*_{f_b}(\mathbf{u^b}), \dots, \mu^*_{f_d}(\mathbf{u^d})) + \sum_{j=b}^{d} \mu^{*'}_{g_{z_j}}(\mathbf{u^0}, \mu^*_{f_b}(\mathbf{u^b}), \dots, \mu^*_{f_d}(\mathbf{u^d}))\sigma^*_{f_j}(\mathbf{u^j})\mathrm{N}_{0,1}\right)$ converges in L_2-norm to zero when all $\sigma^{*2}_{f_j}(\mathbf{u^j})$ go to zero.*

Corollary 4.4. *Under the same conditions as in Lemma 4.2, it follows from Theorem 4.3 that* ξ *and* ζ *both converge in* L_2*-norm to* $g^M(\mathbf{u^0}, \mu^*_{f_b}(\mathbf{u^b}), \ldots, \mu^*_{f_d}(\mathbf{u^d}))$ *as all the* $\sigma^{*2}_{f_j}(\mathbf{u^j})$ *go to zero.*

Theorem 4.5. *Under the same conditions as in Lemma 4.2 and if, for each* $j \in b, \ldots, d$, $\mathbf{u^j}$ *is such that* $\sigma^{*2}_{f_j}(\mathbf{u^j}) = 0$*, then* $\xi = g^M(\mathbf{u^0}, \mu^*_{f_b}(\mathbf{u^b}), \ldots, \mu^*_{f_d}(\mathbf{u^d})) = \zeta$.

This last theorem states that, under the indicated conditions, the linked GaSP is exactly the linked estimator.

5. Comparison of the linked GaSP to the composite emulator.

The other natural emulator that we mentioned is the *composite emulator*, formed by sequentially running the simulators f and g, and then developing an emulator based only on the inputs to f and outputs of g. More formally, suppose we have m d-dimensional inputs $\mathbf{z} = \{\mathbf{z_1}, \ldots, \mathbf{z_m}\}$ to a composite simulator $g \circ (f_b, \ldots, f_d)$. In order to evaluate a composite computer model at these inputs, we first evaluate each model f_j, for $j \in b, \ldots, d$, at corresponding inputs $\mathbf{z^j_{1:m}} = (\mathbf{z^j_1}, \ldots, \mathbf{z^j_m})$ where $\mathbf{z^j_i} = \{z_{ik}\}_{k \in I_j}$. That is, for each input $\mathbf{z^j_i}$ we obtain output $f_j(\mathbf{z^j_i})$. Then, using the outputs from all models $f_j, j \in b, \ldots, d$, we evaluate g at each of the ith d-dimensional inputs $(z_{i1}, \ldots, z_{i\,(b-1)}, f_b(\mathbf{z^b_i}), \ldots, f_d(\mathbf{z^d_i}))$. Thus, we obtain a set of training inputs-outputs of the composite simulator $\mathbf{z} = \{\mathbf{z_i}\}_{i=1}^m$ and $\{(g \circ (f_b, \ldots, f_d))(\mathbf{z_i})\}_{i=1}^m$. Then the emulator of $(g \circ (f_b, \ldots, f_d))^M(\cdot)$ may be constructed, using these inputs-outputs from the coupled system, as described in subsection 2.1.

Definition 5.1. *The GaSP emulator* $(g \circ (f_b, \ldots, f_d))^M(\cdot)$ *of the composite model* $g \circ (f_b, \ldots, f_d)$*, given GaSP parameters* $\theta_{\mathbf{g \circ (f_b, \ldots, f_d)}}$*, namely the emulator* $(g \circ (f_b, \ldots, f_d))^M(\cdot)|\cdot, (\mathbf{g} \circ (\mathbf{f_b}, \ldots, \mathbf{f_d}))^{\mathbf{M}}(\mathbf{z}), \theta_{\mathbf{g \circ (f_b, \ldots, f_d)}}$ *described above, will be called the composite emulator.*

It may not always be possible to construct a composite emulator, in that one might not have control over running the models $f(\cdot)$ or $g(\cdot)$, and instead just have available collections of previous runs. Thus there will always be times in which only the linked emulator (or linked GaSP) is available.

Perhaps surprisingly, it seems that utilization of the composite emulator may not be desirable, even when it can be constructed. As a first indication of this, consider the illustration in subsection 4.1. Figure 4 shows the composite emulator for $(g \circ f)(x)$ in the domain $x \in [-1, 1]$, using the same design points $\mathbf{x}$ as in subsection 4.1, and with parameters again estimated through the ML approach from subsection 2.2. Surprisingly, the composite emulator does a much worse job of emulation (compare to Figure 3). It has a much bigger variance but, even worse, the confidence bands miss the true composite function over part of the domain.

This comparatively poor behavior of the the composite emulator is quite common. It seems to arise because, while the functions f and g might be quite smooth – which allows for their accurate emulation with a small number of design points, the composite function $(g \circ f)(x)$ can be considerably more 'wiggly', and hence much harder to emulate directly. Additional evidence for this will be seen later.

Note that the computational costs in training the linked emulator and the composite emulator were identical in this example; each required six runs of each model.

We also assessed the linked GaSP and the composite emulator of the coupled simulator

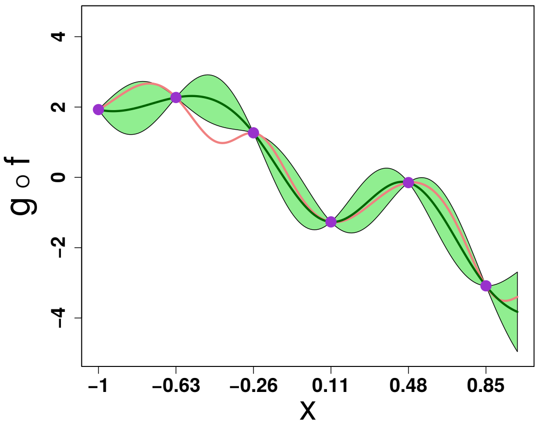

Figure 4. *Composite emulator of a composite test function. The pink line is the true function. The dark green line is the emulator mean. The green shaded region is the region enclosed by the* 2.5% *and* 97.5% *quantiles of the emulator. The circles on the plot correspond to the design points* $\{\mathbf{x}, (\mathbf{g} \circ \mathbf{f})(\mathbf{x})\}$.

Table 1

Predictive evaluations for the linked GaSP and composite emulators in the illustration.

Emulator	EFC	RMSPE	L_{CI}
Linked	1.00	0.13	0.62
Composite	0.76	0.43	0.92

$g \circ f$, using the predictive measures from subsection 2.3. 201 test points, $\mathbf{u}$, equally spaced in $[-1, 1]$, were used for the assessment. Numerical results are presented in Table 1. The performance of the linked GaSP is much better in terms of the predictive measures than the performance of the composite emulator. The RMSPE of the linked GaSP is more than 3 times smaller than that of the composite emulator. While the linked GaSP is capturing the composite simulator on the whole domain $[-1, 1]$ in its 95% credible intervals, the composite emulator intervals miss the truth about 24% of the time. The length of the credible intervals of the linked GaSP are about two thirds of those of the composite emulator.

6. The POB and OB linked emulators. Previously, we have only considered emulators of a function g when all parameters of the emulator are given; in the illustrations, we simply replaced parameters by their estimates, discussed in subsection 2.2 as the ML approach. Here we develop linked emulators for the POB approach (full objective Bayesian treatment of the mean parameters, but utilizing an estimate for the variance) and the OB approach (full objective Bayesian treatment of both mean and variance parameters). Emulators that account for the uncertainty in the mean or mean and variance parameters give more accurate assessments of the emulator predictive variance, and hopefully this will carry through when they are linked.

6.1. POB linked emulator. Suppose that the GaSP mean is a linear function. We perform an objective Bayesian analysis with the parameters β in the mean (using a constant prior $\pi(\beta) \propto 1$), but use the marginal maximum likelihood estimate of σ^2. The corresponding

GaSP (with β integrated out, the MLE estimate of σ^2 plugged in, and the reference posterior mode estimates of ζ plugged in), conditional on the observed computer model evaluations $\mathbf{g^M}(\mathbf{z})$, follows a normal distribution with mean $\mu^*(\mathbf{z}')$ and variance $\sigma^*(\mathbf{z}')$ given by

$$
\begin{aligned}
\mu^*(\mathbf{z}') &= \mathbf{h}(\mathbf{z}')\beta + c(\mathbf{z}',\mathbf{z})C_z^{-1}(\mathbf{g^M}(\mathbf{z}) - \mathbf{h}(\mathbf{z})\beta),\\
\sigma^*(\mathbf{z}') &= \sigma^2 \big(C_{z'} - c(\mathbf{z}',\mathbf{z})C_z^{-1}c(\mathbf{z},\mathbf{z}') + (\mathbf{h}(\mathbf{z}') - c(\mathbf{z}',\mathbf{z})C_z^{-1}\mathbf{h}(\mathbf{z}))\\
&\qquad (\mathbf{h}(\mathbf{z})^T C_z^{-1}\mathbf{h}(\mathbf{z}))^{-1}(\mathbf{h}(\mathbf{z}') - c(\mathbf{z},\mathbf{z}')C_z^{-1}\mathbf{h}(\mathbf{z}))^T\Big)\ ;
\end{aligned}
$$

this is the POB GaSP emulator. Note that it differs from the ML GaSP emulator only in having additional (positive) terms in the predictive variance.

6.1.1. Development of the POB linked emulator.

Theorem 6.1. *Let g^M, with given parameters $\theta_{\mathbf{g}} = (\sigma^2, \eta, \{\delta_j\}_{j=1,\ldots,d})$, be a POB GaSP emulator of a simulator g exercized at training input points $\mathbf{z}$. Suppose the mean is linear in the bth cooordinate of an input $\mathbf{z}'$, so that the mean is $\mathbf{h}(\mathbf{z}')\beta = \beta_0 + \beta_1 z_b'$. Let the $g^M(\cdot)$ GaSP correlation function smoothness parameters α_j of coordinates $j \in b,\ldots,d$ be equal to 2. For each $j \in b,\ldots,d$ let f_j^M be an independent emulator of a simulator f_j, corresponding to the coordinate j of the input to the simulator g, i.e. $f_j^M(\cdot)$ is any GaSP with predictive mean and variance at any input $\cdot$ denoted as $\mu^*_{f_j}(\cdot)$ and $\sigma^{*2}_{f_j}(\cdot)$ respectively. Then the mean $\mathrm{E}\xi$ and variance $\mathrm{V}\xi$ of the linked emulator ξ of the coupled simulator $(g \circ (f_b,\ldots,f_d))(\mathbf{u})$ are*

$$
\begin{aligned}
\mathrm{E}\xi &= \hat{\beta}_0 + \hat{\beta}_1\mu^*_{f_b} + \sum_{i=1}^m a_i \prod_{j=1}^{b-1} \exp\left(-\left(\frac{|u_j - z_{ij}|}{\delta_j}\right)^{\alpha_j}\right)\prod_{j=b}^d I_j^i,\\
\mathrm{V}\xi &= \hat{\sigma^2}(1+\eta) + \hat{\beta_0}^2 + 2\hat{\beta}_0\hat{\beta}_1\mu^*_{f_b}(\mathbf{u^b}) + \hat{\beta_1}^2(\sigma^{*2}_{f_b}(\mathbf{u^b}) + (\mu^*_{f_b}(\mathbf{u^b}))^2) - (\mathrm{E}\xi)^2 +\\
&\quad \sum_{k,l=1}^m (a_l a_k + \hat{\sigma^2} q_{kl}) \prod_{j=1}^{b-1} e^{-\left(\left(\frac{|u_j - z_{kj}|}{\delta_j}\right)^{\alpha_j} + \left(\frac{|u_j - z_{lj}|}{\delta_j}\right)^{\alpha_j}\right)} \prod_{j=b}^d {I1}_j^{k,l} +\\
&\quad 2\sum_{i=1}^m a_i \prod_{j=1}^{b-1}\exp\left(-\left(\frac{|u_j - z_{ij}|}{\delta_j}\right)^{\alpha_j}\right)\left(\hat{\beta}_0 I_b^i + \hat{\beta}_1 {I^+}_b^i\right)\prod_{j=b+1}^d I_j^i +\\
&\quad \hat{\sigma^2}(T_{11} + (T_{12} + T_{21})\mu^*_{f_b}(\mathbf{u^b}) + T_{22}(\sigma^{*2}_{f_b}(\mathbf{u^b}) + (\mu^*_{f_b}(\mathbf{u^b}))^2)) +\\
&\quad \hat{\sigma^2}\sum_{i=1}^m \left(A_{2i} {I_2}_b^{ui} - 2A_{1i} {I_1}_b^{ui}\right),
\end{aligned}
$$

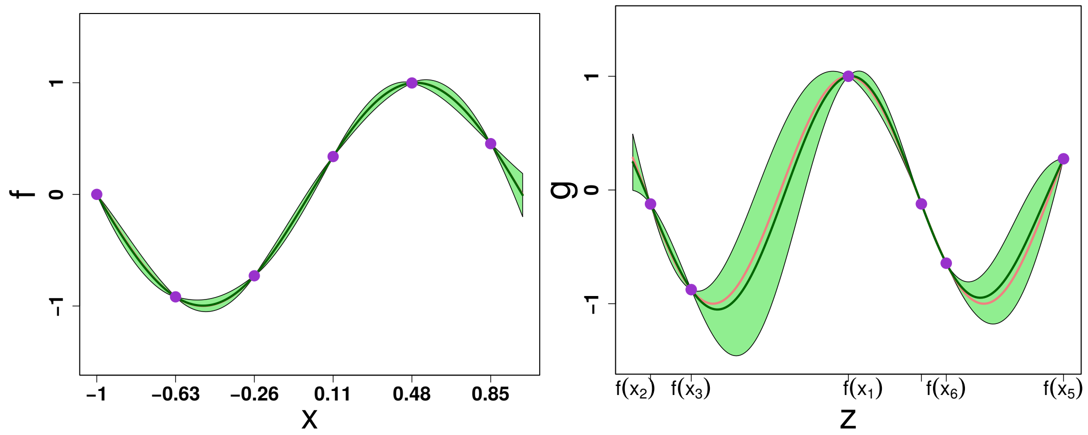

Figure 5. *Independent POB emulators constructed for two test functions: $f(x)$ on the left panel and $g(x)$ on the right. Each emulator is an interpolator at its design points. The pink lines are the true functions. The dark green lines are the emulator means. The green shaded regions are the regions enclosed by the 2.5% and 97.5% quantiles of the emulators. The circles on the plots correspond to the design points which were used to fit the emulators.*

where

$$\begin{aligned}
\begin{pmatrix}\hat{\beta}_0\\ \hat{\beta}_1\end{pmatrix} &= (\mathbf{h}(\mathbf{z})^T C_z^{-1}\mathbf{h}(\mathbf{z}))^{-1}(\mathbf{h}(\mathbf{z})^T C_z^{-1}\mathbf{g}^{\mathbf{M}}(\mathbf{z}),\\
q_{kl} &= (C_z^{-1}\mathbf{h}(\mathbf{z})(\mathbf{h}(\mathbf{z})^T C_z^{-1}\mathbf{h}(\mathbf{z}))^{-1}\mathbf{h}(\mathbf{z})^T C_z^{-1} - C_z^{-1})_{k,l},\\
T &= \begin{pmatrix}T_{11} & T_{12}\\ T_{21} & T_{22}\end{pmatrix} = (\mathbf{h}(\mathbf{z})^T C_z^{-1}\mathbf{h}(\mathbf{z}))^{-1},\\
A &= (\mathbf{h}(\mathbf{z})^T C_z^{-1}\mathbf{h}(\mathbf{z}))^{-1}\mathbf{h}(\mathbf{z})^T C_z^{-1},\\
\hat{\sigma^2} &= \frac{1}{m}\,\mathbf{g}^{\mathbf{M}}(\mathbf{z})^T(C_z^{-1} - C_z^{-1}\mathbf{h}(\mathbf{z})(\mathbf{h}(\mathbf{z})^T C_z^{-1}\mathbf{h}(\mathbf{z}))^{-1}\mathbf{h}(\mathbf{z})^T C_z^{-1})\mathbf{g}^{\mathbf{M}}(\mathbf{z})\,,
\end{aligned}$$

and

$$I_{1\,b}^{ui} = \frac{\delta_b}{\sqrt{\delta_b^2 + 2\sigma_{f_b}^{*2}(\mathbf{u}^{\mathbf{b}})}} e^{-\frac{\left(u_i - \mu_{f_b}^*(\mathbf{u}^{\mathbf{b}})\right)^2}{\delta_b^2 + 2\sigma_{f_b}^{*2}(\mathbf{u}^{\mathbf{b}})}}, \quad I_{2\,b}^{ui} = \frac{2\delta_b\sigma_{f_b}^{*2}(\mathbf{u}^{\mathbf{b}})u_i + \delta_b^3\mu_{f_b}^*(\mathbf{u}^{\mathbf{b}})}{\sqrt{\left(\delta_b^2 + 2\sigma_{f_b}^{*2}(\mathbf{u}^{\mathbf{b}})\right)^3}} e^{-\frac{\left(u_i - \mu_{f_b}^*(\mathbf{u}^{\mathbf{b}})\right)^2}{\delta_b^2 + 2\sigma_{f_b}^{*2}(\mathbf{u}^{\mathbf{b}})}}.$$

Definition 6.2. $\zeta \sim \mathrm{N}(\mathrm{E}\xi, \mathrm{V}\xi)$ *will be called the* POB linked GaSP*.*

6.1.2. Illustration 2. Two functions are considered as simulators: $f(x) = \sin(\pi x)$ in the domain $x \in [-1, 1]$ and $g(z) = \cos(5z)$ in the domain $z \in [-1, 1]$. Model $f(x)$ was evaluated at 6 equally spaced training input points $\mathbf{x}$, resulting in $\mathbf{z} = \mathbf{f}(\mathbf{x})$. Model g was then evaluated at these output points, $\mathbf{z}$. POB emulators, $f^M(\cdot)$ and $g^M(\cdot)$, of the functions were developed using these input, and shown in Figure 5,

The POB linked emulator was then constructed using Theorem 6.1 and is shown in the left panel of Figure 6. The linked GaSP is a good emulator, acting as an interpolator at the

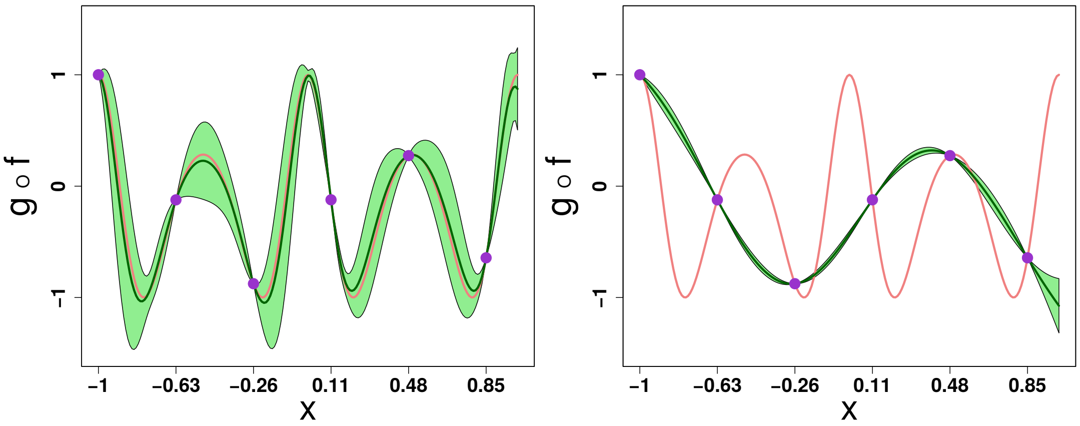

Figure 6. *The left panel is the POB linked GaSP of $g \circ f$. The right panel is the POB composite emulator of the composite model. The pink lines are the true functions. The dark green lines are the emulator means. The green shaded regions are the regions enclosed by the 2.5% and 97.5% quantiles of the emulators. Circles on the plot correspond to sequentially obtained design points $\{\mathbf{x}, (\mathbf{g} \circ \mathbf{f})(\mathbf{x})\}$.*

design points of the emulator $f^M(\cdot)$ and providing 100% coverage, better than the nominal coverage.

The first example in the supplementary materials demonstrates that if we take two functions and run them on separately developed designs, then we can still construct a good approximation to the coupled model without ever observing the coupled system. In our previous examples we get design for two functions sequentially, not independently. This may not be desirable in practice, since this brings restrictions on possible experimental designs and may be detrimental for individual emulators. The example in the supplementary materials highlights that there is no need for running computer models sequentially in order to apply the methodology of the linked emulator.

Important though is that we can not construct the composite emulator if the simulators are ran independently (not sequentially), so then there is no any benchmark to compare our linked emulator to. Thus, we left the example with the sequential designs in the manuscript, and the additional example with independent designs (using the same functions) is given in the supplementary materials.

6.1.3. Comparison of the POB linked GaSP and POB composite emulator. The POB composite emulator of $g \circ f$ was also constructed, and is represented in the right panel of Figure 6. The emulator completely misses the behavior of the function, and the reason is that mentioned earlier: $g \circ f$ is much more wiggly than either g or f, and so cannot be captured with only 6 design points.

The predictive criterion of subsection 2.3 were also applied for this comparison, but there is no point in reporting the results here, since the POB composite emulator was so bad. These results can be found in the supplementary materials.

6.2. OB linked emulator. The OB emulator of g utilizes the usual objective prior $\pi(\beta, \sigma^2) \propto 1/\sigma^2$ for the mean parameters and variance of the GaSP, and treats these parameters in a full Bayesian fashion. The remaining parameters are estimated as discussed in subsection 2.2, but here will just be considered given. The resulting emulator (see [11]), g^M, conditional on the observed computer model evaluations $\mathbf{g^M}(\mathbf{z})$, follows Student's t-distribution with $m-q$ degrees of freedom and mean $\mu^*(\mathbf{z}')$ and variance $\sigma^*(\mathbf{z}')$ given by

$$\mu^*(\mathbf{z}') = \mathbf{h}(\mathbf{z}')\beta + c(\mathbf{z}',\mathbf{z})C_z^{-1}(\mathbf{g^M}(\mathbf{z}) - \mathbf{h}(\mathbf{z})\beta),$$

$$\sigma^*(\mathbf{z}') = \frac{\mathbf{g^M}(\mathbf{z})^T(C_z^{-1} - C_z^{-1}\mathbf{h}(\mathbf{z})(\mathbf{h}(\mathbf{z})^T C_z^{-1}\mathbf{h}(\mathbf{z}))^{-1}\mathbf{h}(\mathbf{z})^T C_z^{-1})\mathbf{g^M}(\mathbf{z})}{m-q}$$
$$\Big(C_{z'} - c(\mathbf{z}',\mathbf{z})C_z^{-1}c(\mathbf{z},\mathbf{z}') + (\mathbf{h}(\mathbf{z}') - c(\mathbf{z}',\mathbf{z})C_z^{-1}\mathbf{h}(\mathbf{z}))$$
$$(\mathbf{h}(\mathbf{z})^T C_z^{-1}\mathbf{h}(\mathbf{z}))^{-1}(\mathbf{h}(\mathbf{z}') - c(\mathbf{z},\mathbf{z}')C_z^{-1}\mathbf{h}(\mathbf{z}))^T\Big),$$

where q is the number of terms in the linear mean function.

For linking with f^M, it is not possible to compute the predictive mean and variance in closed form if an OB emulator is used for f. Thus we will assume that f^M is a POB emulator. The resulting linked emulator is given in the next theorem.

Theorem 6.3. *Let g^M, with given parameters $\theta_{\mathbf{g}} = (\eta, \{\delta_j\}_{j=1,\dots,d})$, be an OB GaSP emulator of a simulator g that was exercized at training input points $\mathbf{z}$. Suppose the mean is linear in the bth cooordinate of an input $\mathbf{z}'$, so that the mean is $\mathbf{h}(\mathbf{z}')\beta = \beta_0 + \beta_1 z_b'$. Let the $g^M(\cdot)$ GaSP correlation function smoothness parameters α_j of coordinates $j \in b,\dots,d$ be equal to 2. For each $j \in b,\dots,d$ let f_j^M be an independent emulator of a simulator f_j, corresponding to the coordinate j of the input to the simulator g, i.e. $f_j^M(\cdot)$ is any GaSP with predictive mean and variance at any input $\cdot$ denoted as $\mu^*_{f_j}(\cdot)$ and ${\sigma^*}^2_{f_j}(\cdot)$ respectively. Then the mean* $\mathrm{E}\xi$ *of the linked emulator ξ of the coupled simulator $(g\circ(f_b,\dots,f_d))(\mathbf{u})$ is the same as that of POB linked emulator. The variance* $\mathrm{V}\xi$ *differs from that of the POB linked emulator by the expression for $\hat{\sigma^2}$, which instead is*

$$\hat{\sigma^2} = \frac{1}{m-2}\,\mathbf{g^M}(\mathbf{z})^T(C_z^{-1} - C_z^{-1}\mathbf{h}(\mathbf{z})(\mathbf{h}(\mathbf{z})^T C_z^{-1}\mathbf{h}(\mathbf{z}))^{-1}\mathbf{h}(\mathbf{z})^T C_z^{-1})\mathbf{g^M}(\mathbf{z})\,.$$

Definition 6.4. $\zeta \sim \mathrm{N}(\mathrm{E}\xi, \mathrm{V}\xi)$ *will be called the* OB linked GaSP.

Note that the predictive means of the ML linked emulator, POB linked emulator, and OB linked emulator are all the same. Thus the linked emulator only differ in their predicted variances. The difference between the predictive variances of the ML linked emulator and POB linked emulator can be quite substantial, but the different between those of the POB and OB linked emulators is usually modest, since the only difference is normalizing the variance estimate by m instead of $m-2$. For small m this could be an appreciable difference, but not for typical training sample sizes.

7. Case study.

We present an example of coupling two real computer models.

7.1. Volcano ash cloud system of computer models.

The two models that are to be coupled are the *bent* model of a volcanic ash plume and the *puff* model describing how the ash cloud disperses; see [5, 15, 20] for discussion. A direct coupling of bent and puff (not emulation) was used for analysis of the 14 April 2010 paroxysmal phase of the Eyjafjallajökull eruption, Iceland, based on observations of Eyjafjallajökull volcano and information from

Table 2
Predictive evaluations for the bent output emulators.

Bent output	EFC	RMSPE/RMSPE$_{\text{base}}$	L_{CI}
plumeMax (m)	0.980	0.017	58.84
plumeMin (m)	0.978	0.016	55.10
plumeHwidth (km)	0.949	0.030	0.196
AshLogMean, $\left(\log_2 \frac{1}{mm}\right)$	0.991	0.007	0.009
AshLogSdev, $\left(\log_2 \frac{1}{mm}\right)$	0.978	0.021	0.021

Table 3
Predictive evaluations for the emulator of puff.

EFC	RMSPE/RMSPE$_{\text{base}}$	L_{CI}
0.95	0.16	18.46

other similar eruptions of the past. The goal in this section is to develop a linked emulator of these computer models.

7.1.1. Bent simulator. Bent is a volcanic eruption column model. The inputs to this model are the source conditions for an eruption. Most of the parameters of the models are fixed at particular values, with only four parameters – vent radius, vent velocity, mean grain size, and grain size standard deviation – being variable. These four parameters, $\mathbf{x} = (x_1, x_2, x_3, x_4)$, are thus the inputs to the bent model.

Bent produces 5 output variables: plumeMax, plumeMin, plumeHwidth, ashLogMean and ashLogStdev. We model each output variable, $j = 1, \ldots, 5$, as an individual POB GaSP, depending on input $\mathbf{x}^*$, so that $f_j^M(\mathbf{x}^*) \sim \mathrm{N}(\mu^*_{f_j}(\mathbf{x}^*), \sigma^{*2}_{f_j}(\mathbf{x}^*))$.

7.1.2. Puff simulator. The puff computer model takes the output of bent, the 5-dimensional vector $\mathbf{z}$, and produces positions of representative numerical particles of the ash cloud, as they are affected by wind, turbulence and gravity. These outputs are post-processed to extract quantities of interest at a given geographical location and time point. The puff output that we emulate here is the maximum height of the ash (at a given space-time location) $g^M(\cdot) \sim \mathrm{GaSP}(\mu^*_g(\cdot), \sigma^{*2}_g(\cdot,\cdot))$. The puff simulator produces different random output values for the same input, so that we choose an OB GaSP emulator, with a nugget, to model puff.

7.1.3. Construction and evaluation of the individual GaSPs. The emulator of bent was trained on 400 randomly chosen design inputs and validated on 1000 randomly chosen held-out points, both chosen out of 5454 initial points from a Latin Hypercube Design. The puff emulator was trained on a total of 739 outputs and tested on 1000 held-out data points. The 739 outputs were obtained by, using as inputs, the 400 outputs of bent, i.e., the models were run sequentially. (Again, this would not be necessary to construct the OB linked GaSP, but is necessary to construct the composite emulator for later comparison.) Since puff is not a deterministic model, it was rerun at 339 of these 400 inputs, resulting in the total of 739 outputs.

Tables 2 and 3 give the predictive evaluations of the bent and puff emulators, respectively,

Table 4
Predictive evaluations for the OB linked GaSP and composite emulator of a coupled system of puff and bent.

Emulator	EFC	RMSPE/$\text{RMSPE}_{\text{base}}$	L_{CI}
Linked GaSP	0.951	0.17	18.55
Composite	0.960	0.17	19.52

for each of the output variables for which the emulators were constructed. The relative ratios of RMSPEs are very low, and the CIs are small, indicating the emulators are giving excellent approximations to the simulators. The empirical coverages are either close to or greater than the 95% nominal values, indicating that the uncertainties given by the emulators are also good.

7.2. OB linked GaSP and OB composite emulator.

We constructed the OB linked GaSP $(g \circ (f_1, \ldots, f_5))^M(\cdot)$ from the individual GaSPs, utilizing Theorem 6.3. The emulator was evaluated at the same held-out test points as before; the resulting predictive evaluations are shown in Table 4. The performance of the emulator is excellent, with rather small credible intervals having empirical coverage very close to the nominal value.

The OB composite emulator was constructed from the 739 outputs obtained by sequentially running bent and puff. The emulator was evaluated at the same held-out test points as before; the resulting predictive evaluations are shown in Table 4. The composite emulator performance is good here: small CIs with empirical coverage just above the nominal value. The credible intervals are slightly longer than those for the OB linked GaSP (indeed, 989 out of 1000 test points had linked emulator credible intervals smaller than that of the composite emulator), but are still fine.

8. Conclusions and generalizations.

The problem of coupling computer models was tackled by developing a closed form linked emulator, from GaSP emulators of each computer model separately. In particular, multiple real-valued computer models were allowed as inputs to another computer model. Of the various linked emulators developed, we would recommend utilizing the OB linked GaSP, as it is closed form and incorporates the uncertainties in the mean and variance parameters of the component GaSPs (as well as the uncertainties in the individual GaSPs).

The approach was based on utilization of separately developed emulators for each computer model, since these are available even when the computer models to be coupled cannot be run sequentially. The illustrations in the paper were constructed in a sequential fashion, with the outputs of one model being the inputs to the other; this also allowed construction of the composite emulator, based solely on the inputs to the first model and resulting outputs of the second. Perhaps surprisingly, the linked emulator performed better in the illustrations than the composite emulator, by all predictive measures considered.

This also bodes well for the possibility of coupling emulators for more complex systems of computer models than considered here. Separately developing emulators for each computer model in the system, and then linking the emulators, is an attractive divide-and-conquer strategy. Of course, one would have to be careful in choosing the design spaces for each emulator development, to ensure the the emulator is being developed over the region of important outputs from the preceding coupled model. Further discussion of this can be found in [14].

The generalization of linking a GaSP emulator of computer model g with a GaSP emulator of f, having multivariate output, is presented in the supplementary materials to this paper. Closed form expressions for the resulting mean and variance functions are provided. We did not highlight these results in this paper because it was subsequently found that multivariate modeling does not bring significant advantages over individual modeling of each univariate output of f [14]; the results from multivariate modeling are almost the same as those from individual modeling of each output.

Appendix A. Proof of 3.3.

Proof. The mean and variance of the linked emulator can be expressed through the law of iterated expectation and the law of total variance respectively.

For general mean of the GaSP $h(\cdot)$, the expressions are

$$\mathrm{E}\xi = \mathrm{E}\mathbf{h}(\mathbf{u^0}, f_b^M(\mathbf{u^b}), \ldots, f_d^M(\mathbf{u^d}))\beta + \sum_{i=1}^{m} a_i \prod_{j=1}^{b-1} \exp\left(-\left(\frac{|u_j - z_{ij}|}{\delta_j}\right)^{\alpha_j}\right) \prod_{j=b}^{d} I_j^i,$$

$$\begin{aligned}\mathrm{V}\xi = {} & \sigma^2(1+\eta) - (\mathrm{E}\xi)^2 + \\ & \sum_{k,l=1}^{m} (a_l a_k - \sigma^2\{C_z{}^{-1}\}_{k,l}) \prod_{j=1}^{b-1} e^{-\left(\left(\frac{|u_j - z_{kj}|}{\delta_j}\right)^{\alpha_j} + \left(\frac{|u_j - z_{lj}|}{\delta_j}\right)^{\alpha_j}\right)} \prod_{j=b}^{d} I1_j^{k,l} + \\ & \int \left(\left(\mathbf{h}(\mathbf{u^0}, f_b^M(\mathbf{u^b}), \ldots, f_d^M(\mathbf{u^d}))\beta\right)^2 + 2\left(\mathbf{h}(\mathbf{u^0}, f_b^M(\mathbf{u^b}), \ldots, f_d^M(\mathbf{u^d}))\beta\right)\right. \\ & \left.\sum_{i=1}^{m} a_i \prod_{j=1}^{d} \exp\left(-\left(\frac{|u_j - z_{ij}|}{\delta_j}\right)^{\alpha_j}\right)\right) \prod_{j=b}^{d} p(f_j^M(\mathbf{u^j})) df_j^M(\mathbf{u^j}).\end{aligned}$$

∎

Appendix B. Proof of lemma 4.2.

Proof. Taylor expansion of $g^M(u_1, \ldots, u_{b-1}, f_b^M(\cdot), \ldots, f_d^M(\cdot))$ is

$$\begin{aligned}g^M(u_1, \ldots, u_{b-1}, f_b^M(\mathbf{u^b}), \ldots, f_d^M(\mathbf{u^d})) = {} & g^M(u_1, \ldots, u_{b-1}, \mu^*_{f_b}(\mathbf{u^b}), \ldots, \mu^*_{f_d}(\mathbf{u^d})) + \\ & \sum_{|K|=1}^{\infty} \frac{D^{k_b,\ldots,k_d} g^M(u_1, \ldots, u_{b-1}, \mu^*_{f_b}(\mathbf{u^b}), \ldots, \mu^*_{f_d}(\mathbf{u^d}))}{k_b! \ldots k_d!} \prod_{j=b}^{d} (f_j^M(\mathbf{u^j}) - \mu^*_{f_j}(\mathbf{u^j}))^{k_j},\end{aligned}$$

where the sum is taken over all combinations of $k_b, \ldots, k_d$ such that $k_b + \ldots + k_d = |K|$.

The convergence in L_2-norm is established as

$$\mathrm{E}\Bigg| g^M(u_1,\dots,u_{b-1},f_b^M(\mathbf{u^b}),\dots,f_d^M(\mathbf{u^d}))-$$
$$\left(g^M(u_1,\dots,u_{b-1},\mu^*_{f_b}(\mathbf{u^b}),\dots,\mu^*_{f_d}(\mathbf{u^d}))+\sum_{j=b}^{d}\mu^{*\prime}_{g_{z_j}}(\mu^*_{f_j}(\mathbf{u^j}))\sigma^*_j(\mathbf{u^j})\mathrm{N}_{0,1}\right)\Bigg|^2=$$
$$\mathrm{E}\Bigg|\sum_{j=b}^{d}\Big(D'_{z_j}g^M(u_1,\dots,u_{b-1},\mu^*_{f_b}(\mathbf{u^b}),\dots,\mu^*_{f_d}(\mathbf{u^d}))-\mu^{*\prime}_{g_{z_j}}(\mu^*_{f_j}(\mathbf{u^j}))\Big)$$
$$\sigma^*_j(\mathbf{u^j})\left(\frac{f_j^M(\mathbf{u^j})-\mu^*_{f_j}(\mathbf{u^j})}{\sigma^*_j(\mathbf{u^j})}\right)+$$
$$\sum_{|K|=2}^{\infty}\frac{D^{k_b,\dots,k_d}g^M(u_1,\dots,u_{b-1},\mu^*_{f_b}(\mathbf{u^b}),\dots,\mu^*_{f_d}(\mathbf{u^d}))}{k_b!\dots k_d!}$$
$$\prod_{j=b}^{d}\sigma_j^{*k_j}(\mathbf{u^j})\left(\frac{f_j^M(\mathbf{u^j})-\mu^*_{f_j}(\mathbf{u^j})}{\sigma^*_j(\mathbf{u^j})}\right)^{k_j}\Bigg|^2. \qquad \blacksquare$$

Let $V_j=\frac{f_j^M(\mathbf{u^j})-\mu^*_{f_j}(\mathbf{u^j})}{\sigma^*_j(\mathbf{u^j})}\sim\mathrm{N}(0,1)$, then $V_b,\dots,V_d$ are iid. The statement of the lemma follows.

Appendix C. Proof of theorem 4.5.

Proof. Since $\sigma^{*2}_{f_j}(\mathbf{u^j})=0$, $f_j^M(\mathbf{u^j})$ has a degenerate distribution with $Pr(f_j^M(\mathbf{u^j})=\mu^*_{f_j}(\mathbf{u^j}))=1$.

$$p((g\circ(f_b,\dots,f_d))^M(\mathbf{u})|\mathbf{g^M}(\mathbf{z}),\mathbf{f_j^M}(\mathbf{x^j})_{j\in b,\dots,d},\theta_{\mathbf{f_j}\,j\in b,\dots,d},\theta_{\mathbf{g}},\mathbf{u})$$
$$=\int p(g^M(\mathbf{u^0},f_b^M(\mathbf{u^b}),\dots,f_d^M(\mathbf{u^d}))|\mathbf{g^M}(\mathbf{z}),\mathbf{f_j^M}(\mathbf{x^j})_{j\in b,\dots,d},\theta_{\mathbf{g}})$$
$$\prod_{j=b}^{d}\delta(f_j^M(\mathbf{u^j})-\mu^*_{f_j}(\mathbf{u^j}))df_b^M(\mathbf{u^b}),\dots,df_d^M(\mathbf{u^d}). \qquad \blacksquare$$

Acknowledgments. We thank Marcus Bursik and Abani K. Patra for generous discussions of this work and also for providing access to the simulators bent and puff. We thank University at Buffalo Center for Computational Research, and in particular Matthew D. Jones for assistance with obtaining computer model data. We also thank E. Bruce Pitman and Elaine T. Spiller for helpful considerations of the work.

REFERENCES

[1] M. Bayarri, J. O. Berger, E. S. Calder, K. Dalbey, S. Lunagomez, A. K. Patra, E. B. Pitman, E. T. Spiller, and R. L. Wolpert, *Using statistical and computer models to quantify volcanic hazards*, Technometrics, 51 (2009), pp. 402–413.

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

[3] J. O. Berger, V. De Oliveira, and B. Sansó, *Objective Bayesian analysis of spatially correlated data*, Journal of the American Statistical Association, 96 (2001), pp. 1361–1374.

[4] E. Borgonovo and E. Plischke, *Sensitivity analysis: a review of recent advances*, European Journal of Operational Research, (2015).

[5] M. Bursik, M. Jones, S. Carn, K. Dean, A. Patra, M. Pavolonis, E. B. Pitman, T. Singh, P. Singla, P. Webley, et al., *Estimation and propagation of volcanic source parameter uncertainty in an ash transport and dispersal model: application to the eyjafjallajokull plume of 14–16 april 2010*, Bulletin of volcanology, 74 (2012), pp. 2321–2338.

[6] J. Q. Candela, A. Girard, J. Larsen, and C. E. Rasmussen, *Propagation of uncertainty in Bayesian kernel models-application to multiple-step ahead forecasting*, in Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on, vol. 2, IEEE, 2003, pp. II–701.

[7] A. Damianou and N. Lawrence, *Uncertainty propagation in Gaussian process pipelines*, in NIPS workshop on modern non-parametrics, 2014.

[8] P. W. Goldberg, C. K. Williams, and C. M. Bishop, *Regression with input-dependent noise: A Gaussian process treatment*, Advances in neural information processing systems, 10 (1997), pp. 493–499.

[9] R. B. Gramacy and H. K. Lee, *Cases for the nugget in modeling computer experiments*, Statistics and Computing, 22 (2012), pp. 713–722.

[10] M. Gu, *Robust Uncertainty Quantification and Scalable Computation for Computer Models with Massive Output*, PhD thesis, Duke University, 2016.

[11] M. Gu and J. O. Berger, *Parallel partial Gaussian process emulation for computer models with massive output*, The Annals of Applied Statistics, 10 (2016), pp. 1317–1347.

[12] M. Gu, J. Palomo, and J. Berger, *RobustGaSP: Robust Gaussian Stochastic Process Emulation*, 2016, https://CRAN.R-project.org/package=RobustGaSP. R package version 0.5.

[13] B. Jha and R. Juanes, *Coupled multiphase flow and poromechanics: A computational model of pore pressure effects on fault slip and earthquake triggering*, Water Resources Research, 50 (2014), pp. 3776–3808.

[14] K. N. Kyzyurova, *On Uncertainty Quantification for Systems of Computer Models*, PhD thesis, Duke University, 2017.

[15] R. Madankan, S. Pouget, P. Singla, M. Bursik, J. Dehn, M. Jones, A. Patra, M. Pavolonis, E. Pitman, T. Singh, et al., *Computation of probabilistic hazard maps and source parameter estimation for volcanic ash transport and dispersion*, Journal of Computational Physics, 271 (2014), pp. 39–59.

[16] J. E. Oakley and A. O'Hagan, *Probabilistic sensitivity analysis of complex models: a Bayesian approach*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 66 (2004), pp. 751–769.

[17] R. Paulo, *Default priors for Gaussian processes*, Annals of Statistics, (2005), pp. 556–582.

[18] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, *Design and analysis of computer experiments*, Statistical science, (1989), pp. 409–423.

[19] R. R. Settgast, P. Fu, S. D. Walsh, J. A. White, C. Annavarapu, and F. J. Ryerson, *A fully coupled method for massively parallel simulation of hydraulically driven fractures in 3-dimensions*, International Journal for Numerical and Analytical Methods in Geomechanics, 41 (2017), pp. 627–653.

[20] E. Stefanescu, A. Patra, M. Bursik, R. Madankan, S. Pouget, M. Jones, P. Singla, T. Singh, E. Pitman, M. Pavolonis, et al., *Temporal, probabilistic mapping of ash clouds using wind field stochastic variability and uncertain eruption source parameters: Example of the 14 april 2010 eyjafjallajökull eruption*, Journal of Advances in Modeling Earth Systems, 6 (2014), pp. 1173–1184.

[21] G. Stevens and S. Atamturktur, *Mitigating error and uncertainty in partitioned analysis: A review*

of verification, calibration and validation methods for coupled simulations, Archives of Computational Methods in Engineering, (2016), pp. 1–15.

[22] K. E. Taylor, R. J. Stouffer, and G. A. Meehl, *An overview of cmip5 and the experiment design*, Bulletin of the American Meteorological Society, 93 (2012), pp. 485–498.

[23] X. Wang and J. O. Berger, *Estimating shape constrained functions using a new class of Gaussian processes*, SIAM/ASA Journal on Uncertainty Quantification, 4 (2016), pp. 1–25.