The Annals of Applied Statistics
2016, Vol. 10, No. 3, 1317–1347
DOI: 10.1214/16-AOAS934

PARALLEL PARTIAL GAUSSIAN PROCESS EMULATION FOR COMPUTER MODELS WITH MASSIVE OUTPUT[1]

BY MENGYANG GU* AND JAMES O. BERGER*,†

*Duke University** *and King Abdulaziz University, Jeddah, Saudi Arabia*†

We consider the problem of emulating (approximating) computer models (simulators) that produce massive output. The specific simulator we study is a computer model of volcanic pyroclastic flow, a single run of which produces up to 10^9 outputs over a space–time grid of coordinates. An emulator (essentially a statistical model of the simulator—we use a Gaussian Process) that is computationally suitable for such massive output is developed and studied from practical and theoretical perspectives. On the practical side, the emulator does unexpectedly well in predicting what the simulator would produce, even better than much more flexible and computationally intensive alternatives. This allows the attainment of the scientific goal of this work, accurate assessment of the hazards from pyroclastic flows over wide spatial domains. Theoretical results are also developed that provide insight into the unexpected success of the massive emulator. Generalizations of the emulator are introduced that allow for a nugget, which is useful for the application to hazard assessment.

1. Introduction. Computer models—henceforth *simulators*—are used to generate data to reproduce the behavior of physical, engineering or human processes. We will be working with the testbed simulator TITAN2D [Patra et al. (2005), Pitman et al. (2003)], which simulates the volcanic pyroclastic flow that surges down a volcano after an eruption, based on inputs such as the initiating volume of flow. A key issue with such simulators is that they are typically very computationally expensive to run; TITAN2D requires up to 2 hours for a single run.

To be useful, simulators typically need a host of interactions with data and statistics, a process which has come to be called *uncertainty quantification*. We use this term herein without precisely defining its meaning. The point, however, is that these interactions often require a very large number of evaluations of the simulator at settings of the inputs which have not been run and the expense of running the simulator becomes a prohibitive barrier.

The key to progress is the development of an *emulator* (approximation) of the simulator that is accurate and which can be run very quickly; the uncertainty quan-

Received January 2015; revised April 2016.

[1]Supported in part by NSF Grants DMS-10-07773, DMS-12-28317, EAR-1331353 and DMS-14-07775.

Key words and phrases. Gaussian process, computer model emulation, space–time coordinate, objective Bayesian analysis.

tification tasks are then carried out with the emulator. The task here is the prediction of the simulator output at a new input (e.g., a pyroclastic flow at volume $10^{7.32}$ cubic meters, which has never been run in TITAN2D). A key feature of statistical emulators is that they have an internal assessment of their approximation accuracy, which makes possible a realistic assessment of uncertainty in predictions.

This is a very well-studied paradigm [Bayarri et al. (2007a, 2009), Sacks et al. (1989)]. This paper focuses on a challenging aspect of the problem, namely, emulating a simulator that produces massive output over a coordinate space. For instance, TITAN2D produces flow information at approximately 10^9 space–time coordinates. While there is a vast body of research concerning emulating the simulator at one of a small number of simulator outputs, simultaneous emulation of the output over many coordinates is less studied. Some papers that do so are Higdon et al. (2008), Marrel et al. (2011), Rougier (2008), Rougier et al. (2009), Xiao et al. (2010); these are further discussed in Section 5.2.1, and representative methods will subsequently be compared with the methodology introduced here.

The scientific motivation for this work is to determine hazard probabilities for future volcanic eruptions. In previous studies of volcanic hazard [Lopes (2011), Spiller et al. (2014)], the hazard probability at a specific location is the probability of a catastrophic event happening at least once during next T years; a catastrophic event is typically characterized by a maximum flow height larger than 1 meter during the flow event. In Bayarri et al. (2009), the estimation of the hazard probability at two locations (Plymouth and Bramble Airport) on Montserrat Island were given. In Lopes (2011) and Spiller et al. (2014), this was extended to a number of locations in Belham Valley, an at-risk area on Montserrat. One of the main scientific goals of this work is to enable computation of these hazard probabilities, not at individual locations, but simultaneously over a large spatial region. Furthermore, policymakers might be interested in events other than just maximum flow height exceeding a meter; they could want to use a lower threshold or some other measure entirely, such as damage to structures by the force of the flow. To achieve the flexibility to answer any such posed question, an emulator is needed that can quickly predict the entirety of the output of TITAN2D.

The inputs to the simulator will be denoted $(\mathbf{x}, \mathbf{s})$, where $\mathbf{x}$ describes the driving inputs for the simulator (e.g., the volume of the pyroclastic flow) and $\mathbf{s}$ denotes a coordinate (e.g., the space–time coordinate) at which the simulator evaluates pyroclastic flow; this notation is not convenient for the later technical development, but is useful in this introduction.

The main idea of this development is that $\mathbf{x}$ *must* be accurately involved in the emulation (there is no chance in predicting a pyroclastic flow without adjusting for the volume of the flow, and we must consider volumes that have not yet been observed), but it is often just fine to perform the predictions of flow on just the space of $\mathbf{s}$, which will typically be a fixed grid of space–time coordinates; it is not typically necessary to interpolate into new space–time locations if the original grid is detailed enough.

The straightforward approach to emulating the simulator simultaneously at many locations is discussed in Conti and O'Hagan (2010), Lee et al. (2011, 2012) and utilized for TITAN2D in Spiller et al. (2014). This approach, which is called the Many Single (MS) emulator approach, is simply to fit separate emulators at each coordinate. We will be using Gaussian process (GaSP) emulators, which are characterized by an unknown mean function, an unknown variance and unknown correlation parameters. In the MS emulator approach, these are all determined separately at each location, resulting in a highly computationally intensive process.

This paper provides a computationally feasible alternative to the MS approach, which we call the *parallel partial* (*PP*) emulator approach. This approach has the following features:

- There are independent emulators at each of k coordinates $\mathbf{s}_1, \ldots, \mathbf{s}_k$ (with k being up to 10^9 for TITAN2D).
- Each coordinate emulator is allowed a different mean function and variance because pyroclastic flows behave very differently at different locations on the mountain (e.g., the height of the flow at locations near the initiation of the flow event will be much larger than at locations far from this point).
- All coordinates share common Gaussian process correlation parameters, and these are estimated from the joint likelihood of all emulators.

The name "parallel partial" is used to reflect the fact that the locations have probabilistically independent parallel emulators, but they are only partially independent in the sense that they share common correlation parameters, estimated from the overall likelihood (as will be discussed in Sections 3 and 7). The PP emulator is computationally feasible because it is linear in k; more precisely, after some precomputation steps, computation of the emulator predictions for a new input $\mathbf{x}^*$ at all k locations requires $O(n^2 + nk)$ numerical operations, where n is the number of simulator runs upon which the emulator is based. Such computational details are discussed in Section 5.1.

One natural concern with this approach is that the simulator is (usually) very tightly constrained at nearby locations, while the PP emulator provides independent predictions at each location. A related concern is the use of common Gaussian process correlation parameters at all locations, as opposed to the more flexible modeling of the MS emulator. The surprising reality is that the PP emulator not only is accurate in emulation over the k coordinate points, but also usually seems to be substantially better than alternatives such as the MS emulator, which do allow for differing correlation parameters. Both theoretical reasons and numerical evidence for this will be presented.

The paper is organized as followed. In Section 2, we review the GaSP emulator for simulators with real-valued output, and introduce the TITAN2D testbed simulator. In Section 3, we define and motivate the PP GaSP emulator and a generalization involving a nugget. Section 4 addresses the major scientific question

of prediction of hazard probabilities over a wide region. Section 5 studies the performance of the PP emulator and competitors. Section 6 presents theoretical justification for the PP emulator. Section 7 discusses the problem of estimation of the PP-emulator correlation parameters; reference priors and composite likelihood methods are utilized.

2. Background.

2.1. *GaSP emulator at a given space–time location.* The GaSP emulator is a frequently used surrogate for expensive simulators [see, e.g., Bayarri et al. (2007b), Kennedy and O'Hagan (2001), Kennedy et al. (2008), Li and Sudjianto (2005)]. To set notation, let $\mathbf{x} \in \mathscr{X}$ denote the p-dimensional vector of inputs to the simulator, and let $y(\mathbf{x})$ denote the resulting simulator output, assumed in this section to be real-valued. The simulator $y(\mathbf{x})$ is viewed as an unknown function [because the simulator is expensive to run, we will at most be able to evaluate $y(\mathbf{x})$ at a few points] modeled via a Gaussian Process,

$$y(\cdot) \sim \mathrm{GaSP}\big(\mu(\cdot), C(\cdot,\cdot)\big), \tag{2.1}$$

having mean function $\mu(\cdot)$ and covariance $C(\cdot,\cdot) = \sigma^2 c(\cdot,\cdot)$ with variance σ^2, and correlation function $c(\cdot,\cdot)$. For any inputs $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ from $\mathscr{X}$, the likelihood is a multivariate normal,

$$\big(y(\mathbf{x}_1), \ldots, y(\mathbf{x}_m)\big)^T | \boldsymbol{\mu}, \sigma^2, \mathbf{R} \sim \mathrm{MVN}\big(\big(\mu(\mathbf{x}_1), \ldots, \mu(\mathbf{x}_m)\big)^T, \sigma^2 \mathbf{R}\big),$$

where σ^2 is the unknown variance and $\mathbf{R}$ is the correlation matrix (or Gram matrix [Rasmussen and Williams (2006)]) with (i, j) element $c(\mathbf{x}_i, \mathbf{x}_j)$. It is common to model the mean function via regression,

$$\mu(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\theta} = \sum_{t=1}^{q} h_t(\mathbf{x})\theta_t,$$

where $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_q(\mathbf{x}))$ is a vector of specified basis functions and θ_t is the unknown regression parameter for basis function h_t. A commonly used correlation function for inputs $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$ and $\mathbf{x}_j = (x_{j1}, \ldots, x_{jp})$ is the exponential family correlation of the form [Rasmussen and Williams (2006)]

$$c(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{-\sum_{t=1}^{p} \left(\frac{|x_{it} - x_{jt}|}{\gamma_t}\right)^{\alpha_t}\right\}, \tag{2.2}$$

with $\gamma_t \in (0, \infty)$ and $\alpha_t \in [1, 2]$; the resulting Gaussian process is thus stationary. The emulator is developed from runs of the simulator at a set of n chosen inputs $\mathbf{x}^{\mathscr{D}} = \{\mathbf{x}_1^{\mathscr{D}}, \ldots, \mathbf{x}_n^{\mathscr{D}}\}$, often selected using a Latin Hypercube Design (LHD) over the input space $\mathscr{X}$ [Forrester, Sobester and Keane (2008), Sacks et al.

(1989)]; let $\mathbf{y}^{\mathscr{D}} = (y(\mathbf{x}_1^{\mathscr{D}}), \ldots, y(\mathbf{x}_n^{\mathscr{D}}))^T$ denote the corresponding simulator outputs. The unknown parameters of the emulator will be handled via a mixed objective Bayesian/likelihood approach.

To deal with the unknown mean and variance, we simply utilize the standard reference prior for a location-scale parameter, namely,

$$\pi^R(\boldsymbol{\theta}, \sigma^2) \propto \frac{1}{\sigma^2}.$$

The points in $\mathbf{x}^{\mathscr{D}}$ are typically chosen as far apart as possible in order to sample the simulator at as many diverse points as possible. This means that the parameters α_t are not highly influential and typically have quite flat likelihood surfaces. They also are typically highly confounded with the γ_t and σ^2, causing computational and inferential difficulties if left in the model [Gelfand et al. (2010), Zhang (2004)]. It is thus common to fix them to a constant value—often 1.9 (which we adopt herein), to reflect a typical desire for smoothness of the emulator, yet avoiding numerical problems that can arise with the choice $\alpha_t = 2$.

The γ_t will be estimated as the modes of their marginal posterior densities arising from first integrating out $\boldsymbol{\theta}$ and σ^2, with respect to $\pi^R(\cdot)$, and then multiplying this marginal likelihood by the reference prior for the γ_t. There are several technical issues involved in the implementation, the details of which we delay until Section 7; for now we just assume the availability of estimates $\hat{\gamma}_t$.

With the above setup, the emulator can be defined. It is a prediction, at a new input value $\mathbf{x}^*$, of the corresponding simulator output $y(\mathbf{x}^*)$. Indeed, the predictive distribution of $y(\mathbf{x}^*)$, given $\mathbf{y}^{\mathscr{D}}$ and $\hat{\boldsymbol{\gamma}} = (\hat{\gamma}_1, \ldots, \hat{\gamma}_p)$, is a t-distribution

$$y(\mathbf{x}^*)|\mathbf{y}^{\mathscr{D}}, \hat{\boldsymbol{\gamma}} \sim t\big(\hat{y}(\mathbf{x}^*), \hat{\sigma}^2 c^{**}, n-q\big), \tag{2.3}$$

with $n-q$ degrees of freedom, where

$$\begin{aligned}
\hat{y}(\mathbf{x}^*) &= \mathbf{h}(\mathbf{x}^*)\hat{\boldsymbol{\theta}} + \mathbf{r}^T(\mathbf{x}^*)\mathbf{R}^{-1}\big(\mathbf{y}^{\mathscr{D}} - \mathbf{h}(\mathbf{x}^{\mathscr{D}})\hat{\boldsymbol{\theta}}\big),\\
\hat{\sigma}^2 &= (n-q)^{-1}\big(\mathbf{y}^{\mathscr{D}} - \mathbf{h}(\mathbf{x}^{\mathscr{D}})\hat{\boldsymbol{\theta}}\big)^T\mathbf{R}^{-1}\big(\mathbf{y}^{\mathscr{D}} - \mathbf{h}(\mathbf{x}^{\mathscr{D}})\hat{\boldsymbol{\theta}}\big),\\
c^{**} &= c(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{r}^T(\mathbf{x}^*)\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}^*) + \big(\mathbf{h}(\mathbf{x}^*) - \mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}^*)\big)^T\\
&\quad\times\big(\mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\mathbf{R}^{-1}\mathbf{h}(\mathbf{x}^{\mathscr{D}})\big)^{-1}\big(\mathbf{h}(\mathbf{x}^*) - \mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}^*)\big),
\end{aligned} \tag{2.4}$$

with $\hat{\boldsymbol{\theta}} = (\mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\mathbf{R}^{-1}\mathbf{h}(\mathbf{x}^{\mathscr{D}}))^{-1}\mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\mathbf{R}^{-1}\mathbf{y}^{\mathscr{D}}$ being the generalized least squares estimator for $\boldsymbol{\theta}$, $\mathbf{h}(\mathbf{x}^{\mathscr{D}})$ being the $n \times q$ basis design matrix with (i, j) element $h_j(\mathbf{x}_i^{\mathscr{D}})$, and $\mathbf{r}(\mathbf{x}^*) = (c(\mathbf{x}^*, \mathbf{x}_1^{\mathscr{D}}), \ldots, c(\mathbf{x}^*, \mathbf{x}_n^{\mathscr{D}}))^T$.

Note that, at the design points $\mathbf{x}_i^{\mathscr{D}}$, $1 \le i \le n$, the emulator is an interpolator of the simulator because when $\mathbf{x}^* = \mathbf{x}_i^{\mathscr{D}}$, $\mathbf{r}^T(\mathbf{x}^*)\mathbf{R}^{-1} = \mathbf{e}_i^T$, where $\mathbf{e}_i$ is an n-dimensional vector with the ith row as 1 and the others as 0. At other inputs, it provides not only a prediction of the simulator [i.e., $\hat{y}(\mathbf{x}^*)$], but also an assessment of the accuracy of the prediction; since this was developed from a partial Bayesian perspective, it also incorporates the uncertainty arising from estimating $\boldsymbol{\theta}$ and σ^2.

2.2. *The TITAN2D testbed.* The four inputs to the TITAN2D simulator are the initial flow volume V, initial angle of the flow ϕ, basal friction angle δ_{bed} and internal friction angle δ_{int}. TITAN2D produces numerous outputs, one of them being the pyroclastic flow height at every space–time grid point. The PP emulator developed herein is perfectly capable of handling the entire space–time grid, but the time component is not of particular practical interest, for the simple fact that damage from pyroclastic flow is primarily due to the largest flow that hits a given spatial location. Therefore, the simulator output of particular interest, at a given location on the island, is

$$y(V, \phi, \delta_{\text{bed}}, \delta_{\text{int}}) = \text{maximum flow height over time},$$

this being a good surrogate for the damage inflicted at the location. We will thus work with this simulator output in our illustrations and evaluations, including the ultimate goal of producing probabilistic hazard maps for the region. Actually, for reasons discussed in Bayarri et al. (2009), we fit the emulator to $\log(y + 1)$ and then transform the predictions back. Means and variances do not transform directly through this transformation, but posterior medians and quantiles do, and are what we use in actual computations; we will suppress this detail in our notation.

For the reasons discussed in Bayarri et al. (2009), the basis functions $\mathbf{h}(\cdot) = (1, V)$ will be utilized so that the mean function will simply be the regression $\theta_1 + \theta_2 V$. The design input space $\mathscr{D}$ consisted of 2048 points chosen according to a maximin Latin hypercube design over the relevant region $[10^5, 10^{9.5}] \times [0, 2\pi) \times [5.45, 18.45] \times [15, 35]$ for the four inputs.

TITAN2D was run at these 2048 inputs, and the resulting vectors of maximum flow heights over the spatial grid of the island were recorded. A complication that arises in TITAN2D is that the second input, ϕ, is periodic, ranging from 0 to 2π, so that a correlation function that respects periodicity is needed. To overcome this difficulty, we follow Spiller et al. (2014) and utilize a correlation function based on "periodic folding." Details are described in the supplementary materials [Gu and Berger (2016)]. Note that, while we focus here on prediction of hazard probabilities for the Soufrière Hill Volcano (SHV) on Montserrat Island, the methodology can be used for hazard prediction for any volcanic pyroclastic flows.

3. Parallel partial emulation.

3.1. *The PP GaSP emulator.* As discussed before, TITAN2D will generate massive data over many coordinates during each simulator run. Let k denote the total number of space–time grid points that are considered for each simulator run; with TITAN2D, k can be as big as 10^9, but, for the reasons discussed in Section 2.2, we will herein restrict consideration to only the spatial grid. Let $y_j(\mathbf{x})$ denote the simulator output at the jth coordinate so that $\mathbf{y}(\mathbf{x}) = (y_1(\mathbf{x}), \ldots, y_k(\mathbf{x}))$ is the entire simulator output arising from input $\mathbf{x}$. In this section we develop a computationally efficient and accurate emulator of that entire output.

As discussed in the introduction, we assume that an *independent* GaSP of the form (2.1) is assigned to each coordinate, with prior mean functions of the regression form $\mathbf{h}(\mathbf{x})\boldsymbol{\theta}_j$, where $\mathbf{h}(\mathbf{x})$ is a *common* q-vector of given basis functions and the $\boldsymbol{\theta}_j$ are *differing* unknown regression coefficients, *differing* unknown prior variances σ_j^2, and *common* estimated correlation parameters $\hat{\boldsymbol{\gamma}}$. Assuming common basis functions and estimated correlation parameters is the key to the computational simplification.

Let $\mathbf{y}_j^{\mathscr{D}}$ denote the column vector of simulator output values at the jth coordinate when run over the design input values, as discussed in Section 2.1. We also utilize the same standard objective prior for the mean and variance parameters

$$\pi^R(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k, \sigma_1^2, \ldots, \sigma_k^2) \propto \frac{1}{\prod_{j=1}^k \sigma_j^2}. \tag{3.1}$$

Since the GaSPs at each coordinate are independent given the range parameters $\boldsymbol{\gamma}$, the prior is of a product form in the parameters of the different coordinate GaSPs and $\hat{\boldsymbol{\gamma}}$ is common across coordinates; it is immediate that the overall GaSP, at a new input $\mathbf{x}^*$, is the product of k independent t-distributions, with that for the jth coordinate being

$$y_j(\mathbf{x}^*)|\mathbf{y}_j^{\mathscr{D}}, \hat{\boldsymbol{\gamma}} \sim t\big(\hat{y}_j(\mathbf{x}^*), \hat{\sigma}_j^2 c^{**}, n-q\big), \tag{3.2}$$

with $n-q$ degrees of freedom, where

$$\hat{y}_j(\mathbf{x}^*) = \mathbf{h}(\mathbf{x}^*)\hat{\boldsymbol{\theta}}_j + \mathbf{r}^T(\mathbf{x}^*)\mathbf{R}^{-1}\big(\mathbf{y}_j^{\mathscr{D}} - \mathbf{h}(\mathbf{x}^{\mathscr{D}})\hat{\boldsymbol{\theta}}_j\big), \tag{3.3}$$

$$\hat{\sigma}_j^2 = (n-q)^{-1}\big(\mathbf{y}_j^{\mathscr{D}} - \mathbf{h}(\mathbf{x}^{\mathscr{D}})\hat{\boldsymbol{\theta}}_j\big)^T\mathbf{R}^{-1}\big(\mathbf{y}_j^{\mathscr{D}} - \mathbf{h}(\mathbf{x}^{\mathscr{D}})\hat{\boldsymbol{\theta}}_j\big), \tag{3.4}$$

with $\hat{\boldsymbol{\theta}}_j = (\mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\mathbf{R}^{-1}\mathbf{h}(\mathbf{x}^{\mathscr{D}}))^{-1}\mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\mathbf{R}^{-1}\mathbf{y}_j^{\mathscr{D}}$ being the generalized least squares estimator for $\boldsymbol{\theta}_j$, and $\mathbf{R}$, $\mathbf{h}(\mathbf{x}^{\mathscr{D}})$, $\mathbf{r}(\mathbf{x}^*)$ and c^{**} being defined in Section 2.1. From algebraic rearrangement of (3.3), the following lemma is immediate.

LEMMA 3.1. *Letting* $\mathbf{y}^{\mathscr{D}} = (\mathbf{y}_1^{\mathscr{D}}, \mathbf{y}_2^{\mathscr{D}}, \ldots, \mathbf{y}_k^{\mathscr{D}})$ *denote the* $n \times k$ *matrix of all the simulator output at the design points, the predictive mean of the PP GaSP at the new input* $\mathbf{x}^*$, *namely,* $\hat{\mathbf{y}}(\mathbf{x}^*) = (\hat{y}_1(\mathbf{x}^*), \hat{y}_2(\mathbf{x}^*), \ldots, \hat{y}_k(\mathbf{x}^*))$, *can be expressed as*

$$\hat{\mathbf{y}}(\mathbf{x}^*) = \boldsymbol{\omega}(\mathbf{x}^*)\mathbf{y}^{\mathscr{D}}, \tag{3.5}$$

where

$$\begin{aligned}\boldsymbol{\omega}(\mathbf{x}^*) = {} & \big(\mathbf{h}(\mathbf{x}^*) - \mathbf{r}^T(\mathbf{x}^*)\mathbf{R}^{-1}\mathbf{h}(\mathbf{x}^{\mathscr{D}})\big)\big(\mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\mathbf{R}^{-1}\mathbf{h}(\mathbf{x}^{\mathscr{D}})\big)^{-1} \\ & \times \mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\mathbf{R}^{-1} + \mathbf{r}^T(\mathbf{x}^*)\mathbf{R}^{-1}.\end{aligned}$$

The weights $\boldsymbol{\omega}(\mathbf{x}^*)$ are usually called Kriging weights [Cressie (1993)]. There are two immediate important consequences of (3.5). First, the PP GaSP emulator is not only an interpolator of the simulator at the design inputs, but it is also a weighted sum of the simulator runs [each row of $\mathbf{y}^{\mathscr{D}}$ being the simulator output—at one of the n input values—over all k coordinates, and $\boldsymbol{\omega}(\mathbf{x}^*)$ being an n-vector]. This ensures that the emulator inherits the smoothness of the simulator and probably some of the dynamics. Note, in contrast, that developing a separate emulator at each coordinate would not have this property, in that this would result in different weights for the simulator output at each coordinate.

Second, the weights, $\boldsymbol{\omega}(\mathbf{x}^*)$, depend only on computation of the q-vector $\mathbf{h}(\mathbf{x}^*)$ and the n-vector $\mathbf{r}(\mathbf{x}^*)$ together with precomputable matrices and vectors. The entire computation of the emulator is thus linear in k, the key to the computational simplification. (More details of the computation are given in Section 5.1.)

Note that it is crucial that the outputs over all coordinates share the same correlation parameters $\hat{\boldsymbol{\gamma}}$. If not, then each coordinate would have a different design correlation matrix $\mathbf{R}$, requiring the inversion of an $n \times n$ matrix at each coordinate; the computational situation is actually then even worse, as shown in Section 5.1, because of the need to separately estimate the $\hat{\boldsymbol{\gamma}}_j$. As shown in Section 5.1, there is also a considerable penalty for not having the same basis elements at each coordinate, although the penalty is not nearly as severe as that for allowing differing correlation parameters.

Figure 1 shows the median (truncated at 20 meters at the volcanic center region) and interquartile range of the PP GaSP emulator of TITAN2D for a new input based on $n = 50$ simulator design runs; only 50 runs are used in this illustration because even this small number of runs seems to capture the main features of the model output. Note that the GaSP assessment of accuracy suggests small uncertainty at most of the locations. We will see in Section 5 that these internal emulator uncertainties do accurately reflect the real accuracy in emulation of TITAN2D.

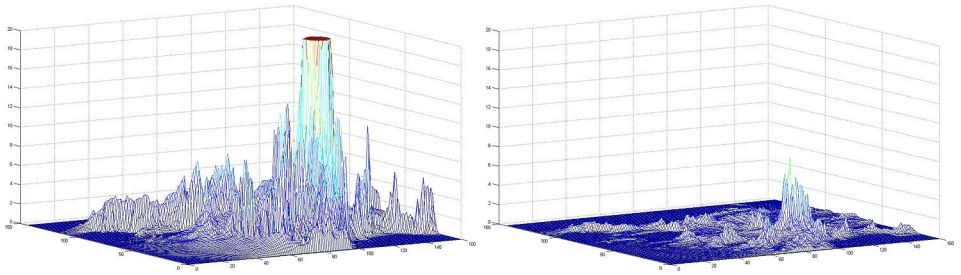

FIG. 1. *Median* (*truncated at* 20 *meters at the volcanic center region*) *and interquartile range of the GaSP emulator of "maximum flow height over time" for TITAN2D, at* 23,040 *spatial locations over Montserrat Island and for new input values* $V^* = 10^{6.9984}$, $\varphi^* = 3.3487$, $\delta^*_{\text{bed}} = 10.8790$ *and* $\delta^*_{\text{int}} = 31.0300$.

3.2. *Adding a nugget to the PP GaSP emulator.* In TITAN2D, the output flow height is almost constant (for fixed values of the other inputs), as the internal friction angle δ_{int} varies over its range $[15°, 30°]$. This was initially indicated by Bayesian model selection methodology for GaSPs [Linkletter et al. (2006), Savitsky, Vannucci and Sha (2011)] and sensitivity analysis [Iooss and Lemaître (2014)], and confirmed by the simulation study in Section 5.2.4. Using a weak input in emulation has the same drawbacks as using a weak covariate in regression—the inaccuracies introduced by incorporating the weak input or covariate into the model can lead to worse predictions than omitting them. However, if a simulator input is omitted in the emulator [Andrianakis and Challenor (2012)], the emulator can no longer be an interpolator so that the GaSP model is then inappropriate. The standard solution is to add a nugget (a noise term) to the GaSP model, such as $\tilde{y}(\cdot) = y(\cdot) + \varepsilon$, where $y(\cdot)$ is the earlier noise-free GaSP and ε is i.i.d. mean-zero Gaussian white noise. In particular, we assume that the covariance function for the new process $\tilde{y}_j(\cdot)$ at coordinate j is

$$\sigma_j^2 \tilde{c}(\mathbf{x}_l, \mathbf{x}_m) = \sigma_j^2 \{c(\mathbf{x}_l, \mathbf{x}_m) + \nu 1_{l=m}\}; \tag{3.6}$$

note that we assume the nugget parameter ν is common across all coordinates (needed for the same reasons we required common correlation parameters $\boldsymbol{\gamma}$). We parameterize the nugget in this way to allow for marginalizing out over σ_j^2 [Kazianka and Pilz (2012), Ren, Sun and He (2012)]. After adding the nugget, the covariance matrix for the design input space $\mathscr{D}$ at coordinate j is

$$\sigma_j^2 \tilde{\mathbf{R}} = \sigma_j^2 (\mathbf{R} + \nu \mathbf{I}). \tag{3.7}$$

For a new input, $\mathbf{x}^*$, the joint distribution of the new and design outputs at coordinate j is

$$\begin{pmatrix} y_j(\mathbf{x}^*) \\ \mathbf{y}_j^{\mathscr{D}} \end{pmatrix} \Big| \boldsymbol{\theta}_j, \sigma_j^2, \boldsymbol{\gamma}, \nu \sim N\left(\begin{pmatrix} \mathbf{h}(\mathbf{x}^*)\boldsymbol{\theta}_j \\ \mathbf{h}(\mathbf{x}^{\mathscr{D}})\boldsymbol{\theta}_j \end{pmatrix}, \sigma_j^2 \begin{pmatrix} \tilde{c}(\mathbf{x}^*, \mathbf{x}^*) & \mathbf{r}^T(\mathbf{x}^*) \\ \mathbf{r}(\mathbf{x}^*) & \tilde{\mathbf{R}} \end{pmatrix} \right) \tag{3.8}$$

for $1 \le j \le k$. The nugget parameter ν will be estimated along with the input correlation parameters, as discussed in Section 7.2, leading to $\hat{\boldsymbol{\gamma}}$ and $\hat{\nu}$ that will be used to develop the emulator. Indeed, the resulting PP GaSP with nugget is defined exactly as in (3.2), with the simple change of replacing $\mathbf{R}$ by $\tilde{\mathbf{R}}$ (computed using $\hat{\boldsymbol{\gamma}}$ and $\hat{\nu}$). After adding a nugget, the PP GaSP is not an interpolator of the design points, but, in the Supplementary Materials [Gu and Berger (2016)], we show that it is close to being an interpolator.

The improvement, for TITAN2D, in going from a four-input emulator to a three-input emulator with nugget, is indicated in Table 1 and Table 2 in Section 5.2. For an indication as to the overall accuracy of the PP emulator with nugget, we consider a crucial feature of the TITAN2D output, namely, the contour on the island at which the maximum flow height is 1 m; as discussed in Bayarri et al. (2009), the interior of this contour defines the region in which the pyroclastic flow is viewed

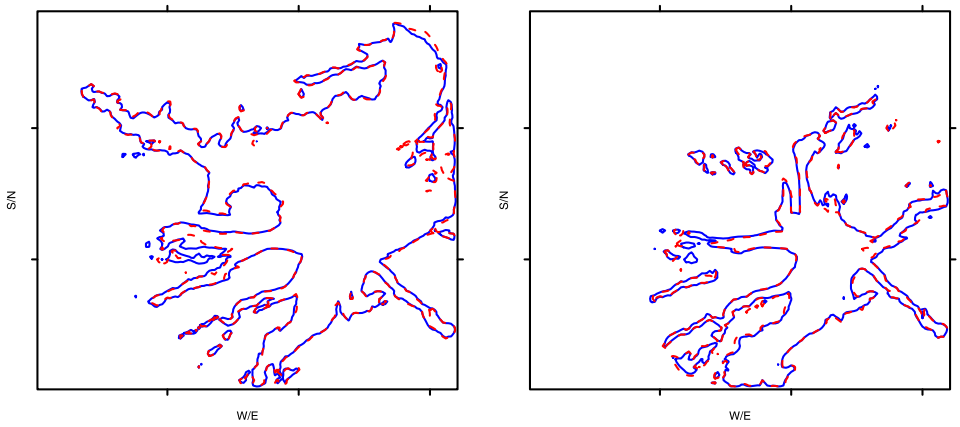

FIG. 2. *1 m spatial contours of maximum pyroclastic flow height on Montserrat Island for two held-out values of the inputs. The red dashed contour is from the actual simulator run, while the blue solid contour is the prediction from the PP GaSP emulator with 3 inputs* $(V, \delta_{\text{bed}}, \phi)$ *and an estimated nugget. The held-out testing inputs for the left figure are* $V^* = 10^{7.1368}$, $\varphi^* = 1.8484$, $\delta^*_{\text{bed}} = 12.2940$ *and* $\delta^*_{\text{int}} = 24.2140$. *Those for the right figure are* $V^* = 10^{6.8292}$, $\varphi^* = 4.5360$, $\delta^*_{\text{bed}} = 12.7880$ *and* $\delta^*_{\text{int}} = 27.3000$.

as being catastrophic. The PP emulator with nugget of TITAN2D was developed using only $n = 50$ runs of the simulator, selected to be approximately space-filling (the small number in order to hopefully see some differences between the emulator and the simulator). The 1 m contours on the island were then computed for a large number of held-out design inputs using the emulator and then the simulator runs.

Two typical results are presented in Figure 2; the red curves are actual contours from the TITAN2D simulator, while the blue curves are the contours from the emulator. The contours match surprisingly well, especially considering the challenging topography (the "holes" in the contours reflect topographical features, such as hills, known to the simulator but not directly known to the emulator) and the use of only 50 training runs.

4. Flexible hazard quantification. As discussed in Section 1, the scientific goal for this work was to enable flexible assessments of hazard from pyroclastic flow over a wide region. In particular, we focus here on developing contour plots of probabilities that maximum flow heights from SHV will exceed any threshold H of interest, over a time period T and over the entire at-risk part of Montserrat Island. Using the PP GaSP emulator, the entire distribution of flow heights over the island (as inputs vary) can be estimated, and this, in turn, can be used to answer a very wide range of hazard questions. We specifically develop the whole island hazard maps for $T = 2.5$ years and H equal 0.5, 1.0 or 2.0.

4.1. *Uncertainty in the inputs and the occurrence of pyroclastic flows.* We first need to account for the uncertainty in the inputs $\mathbf{x}^* = (V, \phi, \delta_{\text{bed}})$. The distribution

of these inputs is studied in Bayarri et al. (2009), Spiller et al. (2014), and we follow their analysis. The distribution of (V, ϕ) is assumed to be of the form

$$p(V, \phi | V_m) \propto \alpha V_m^{\alpha} V^{-\alpha-1} 1_{V>V_m} 1_{0\le\phi<2\pi},$$

that is, a uniform distribution on $[0, 2\pi)$ for ϕ and (independently) a Pareto distribution for the initial volume V. V_m was chosen to be 5×10^4, since flows smaller than this value have no impact on hazard assessments of interest. Based on data giving the volumes of observed pyroclastic flows from SHV, a full Bayesian analysis was conducted in Bayarri et al. (2009) and Spiller et al. (2014) for the Pareto shape parameter α, but the variance of the posterior was so small that we simply utilize $\alpha = 0.64$ (the posterior mean and MLE) in the ensuing analysis. The basal friction angle, δ_{bed}, is assumed to be independent of V and ϕ, and is known to be decreasing in V. Based on available data relating V to δ_{bed}, we follow Bayarri et al. (2015), Spiller et al. (2014) and fit a linear model to the following transformed V and δ_{bed}:

$$\log_{10}(\tan^{-1}(\delta_{\text{bed}})) = a + b \log_{10} V + \varepsilon, \tag{4.1}$$

where $\varepsilon \sim N(0, \sigma_{\text{bed}}^2)$. Eleven observations of (V, δ_{bed}) at Montserrat Island [Bayarri et al. (2015)] are available to fit the model and, utilizing the objective prior $\pi(a, b, \sigma_{\text{bed}}^2) = 1/\sigma_{\text{bed}}^2$, the posterior predictive distribution $\pi(\delta_{\text{bed}}|V)$ is found and utilized in the ensuing analysis. [We will call the above the posterior distribution of $(V, \phi, \delta_{\text{bed}})$, although it is only an approximate posterior in terms of V.]

Conditional on the occurrence of a pyroclastic flow (PF), denote the density of flow height at location j by $f_j(\cdot)$ and denote the cumulative distribution function as $F_j(\cdot)$. This will be estimated by the distribution of flow heights arising from the PP GaSP emulator, as the inputs $\mathbf{x}^* = (V, \phi, \delta_{\text{bed}})$ are drawn from their posterior distribution described above. Actually, we need a sample from $f_j(\cdot)$ for each coordinate j in the following, which we will (approximately) obtain by drawing a sample of inputs $(\mathbf{x}_1^*, \ldots, \mathbf{x}_{N^*}^*)$ from the input posterior distribution, and then computing the PP GaSP with nugget emulator mean (3.5) at each input, simultaneously obtaining a sample at all locations. Strictly speaking, we should sample from the PP GaSP posterior t-distributions, but, in our application, these distributions are extremely concentrated around their modes since we will be using all 2048 simulator runs to build the emulator; the variation caused by the uncertainty in $\mathbf{x}^* = (V, \phi, \delta_{\text{bed}})$ is several orders of magnitude larger than the uncertainty in the PP Gasp.

Hazard prediction, for a period of time T at location j, is based on the distribution of the maximum pyroclastic flow height, $Y_j^{\{T\}}$, that occurs over that period at the location; the following lemma gives the density and α-quantile of this distribution, under the assumption that pyroclastic flows arise from a stationary Poisson process with yearly intensity λ. [At SHV, $\lambda \approx 22$/year, as found in Bayarri et al. (2009)]. We acknowledge that stationarity can be a critical assumption, but it is the most frequently used assumption to provide tractable results [Bayarri et al. (2015), Spiller et al. (2014)].

LEMMA 4.1. *Under the assumption that pyroclastic flows arise from a stationary Poisson process with yearly intensity* λ, *the density of* $Y_j^{\{T\}}$, *the maximum flow height over that period at the location* j, *is*

$$p_j^{\{T\}}(y) = 1_{\{y=0\}} \exp(-\lambda T) + f_j(y)\lambda T \exp\{\lambda T\big(F_j(y) - 1\big)\}.$$

The α*-quantile of this distribution is*

$$y_j^\alpha = \begin{cases} 0, & \alpha \le \exp(-\lambda T), \\ F_j^{-1}\left(\dfrac{\log(\alpha)}{\lambda T} + 1\right), & \alpha > \exp(-\lambda T). \end{cases} \tag{4.2}$$

PROOF. The random number of occurrences, M, of PF's over time period T follows a Poisson distribution with mean $\tilde{\lambda} = \lambda T$. If $M = m$ were to happen over the next T years, then the maximum flow height at coordinate j is then the largest order statistic, having density $mF_j(y)^{m-1} f_j(y)$. If $M = 0$, which happens with probability $\exp(-\tilde{\lambda})$, the maximum flow height is obviously 0. Marginalizing out over M gives

$$\begin{aligned} p_j^{\{T\}}(y) &= 1_{\{y=0\}} \exp(-\tilde{\lambda}) + \sum_{m=1}^{\infty} m f_j(y) F_j(y)^{m-1} \frac{\tilde{\lambda}^m}{m!} \exp(-\tilde{\lambda}) \\ &= 1_{\{y=0\}} \exp(-\tilde{\lambda}) + f_j(y)\tilde{\lambda} \exp\{\tilde{\lambda}\big(F_j(y) - 1\big)\} \\ &\quad \times \sum_{m=1}^{\infty} \frac{\tilde{\lambda}^{m-1}}{(m-1)!} \exp\big(-\tilde{\lambda} F_j(y)\big) \\ &= 1_{\{y=0\}} \exp(-\tilde{\lambda}) + f_j(y)\tilde{\lambda} \exp\{\tilde{\lambda}\big(F_j(y) - 1\big)\}. \end{aligned}$$

Expression (4.2) is an immediate consequence. □

That we have a closed-form expression for the quantiles of $p_j^{\{T\}}$ is key to being able to efficiently employ the PP GaSP emulator to simultaneously compute hazard probabilities over all relevant locations at SHV. Simulation of M would not allow for such efficient use of the emulator.

Quantiles of $p_j^{\{T\}}$ typically transform into quantiles of F_j in the far right tails. For instance, suppose we are interested in quantiles of $p_j^{\{T\}}$ at levels $\alpha = (0.01, 0.1, 0.6, 0.95)$ when $T = 2.5$ years and $\lambda \approx 22$ times/year. Then (4.2) implies that we need the corresponding $(0.9163, 0.9581, 0.9907, 0.9990)$ quantiles of F_j. These latter quantiles will be found at each location j, as the corresponding empirical quantiles from the sample of N^* draws from F_j that were discussed above. The point here is that typically it will suffice to only retain the largest 10% of these draws in order to find the desired quantiles of $p_j^{\{T\}}$; this is a significant saving, since one must store these draws over all locations j.

Algorithm 1 Flexible full hazard map

(1) Run TITAN2D N times for each design $\mathbf{x}_i^{\mathscr{D}}$, $i = 1, \ldots, N$, and record the output pyroclastic flow $\mathbf{y}^{\mathscr{D}}$.
(2) Build the PP GaSP emulator discussed in Section 3, based on all N runs on the design points.
(3) Sample $\mathbf{x}_i^*$, for $i = 1, \ldots, N^*$, from the posterior distribution of inputs discussed in Section 4.1.
(4) Compute the PP GaSP posterior predictive mean (3.5) for each sample $\mathbf{x}_i^*$, and collect the samples at each coordinate j to provide the sample from the flow height distribution at location j.
(5) For any threshold H that is of interest, use the proportion of the predictions from the samples $\mathbf{x}_i^*$ in step (4) that are smaller than H as the estimate of $F_j(H)$.
(6) Use the estimate of $F_j(H)$ to obtain the probability of maximum flow heights over the next T years larger than H at location j, by use of (4.2).

4.2. *Quantification of the hazard at SHV.* We first fit the PP GaSP emulator [i.e., obtain estimates of $(\hat{\boldsymbol{\gamma}}, \hat{\nu})$], using all simulator runs $\mathbf{y}^{\mathscr{D}}$, utilizing the composite likelihood method discussed in Section 7.3. Then we sample $N^* = 10^5$ inputs $(\mathbf{x}_1^*, \ldots, \mathbf{x}_{N^*}^*)$ from their posterior distribution discussed in Section 4.1. At each input we compute the PP GaSP posterior predictive mean and collect the samples at each location j (i.e., the jth coordinates of the predictive means) to provide an (approximate) sample from $F_j(\cdot)$ at each location j (possibly saving only the largest 10% of samples at each location). For any threshold H, we compute the estimate of $F_j(H)$ as the proportion of samples from this distribution smaller than H, and we marginalize out the occurrence of PF to get the estimated probability that the maximum flow heights exceed H at each location using Lemma 4.1. This is summarized in Algorithm 1.

Figure 3 gives contours of the probabilities that the maximum flow heights exceed 0.5, 1 and 2 meters over the next $T = 2.5$ years over Montserrat Island. The upper row in Figure 3 shows the hazard probabilities produced by the PP GaSP with only $N = 50$ runs from TITAN2D, uniformly sampled from the available 2048 runs; and the lower rows show the results using all 2048 runs. While the results are similar, there are clear differences, especially in the areas of small hazard probability. This is because TITAN2D outputs have many zeros at these locations so that it can easily happen that all $N = 50$ runs simply report zero at a location where there is hazard. An interesting problem (outside the scope of this paper) is that of determining the minimum number of runs of TITAN2D for an accurate hazard assessment.

Belham Valley is a small region in the northwest part of Montserrat, plotted as the shaded area in Figure 3. The coastal area to the north of Belham Valley is still inhabited and so is of primary interest for risk assessment. The upper (uninhabited) parts of the valley have a large probability (≈ 0.9) to have more than 1 meter

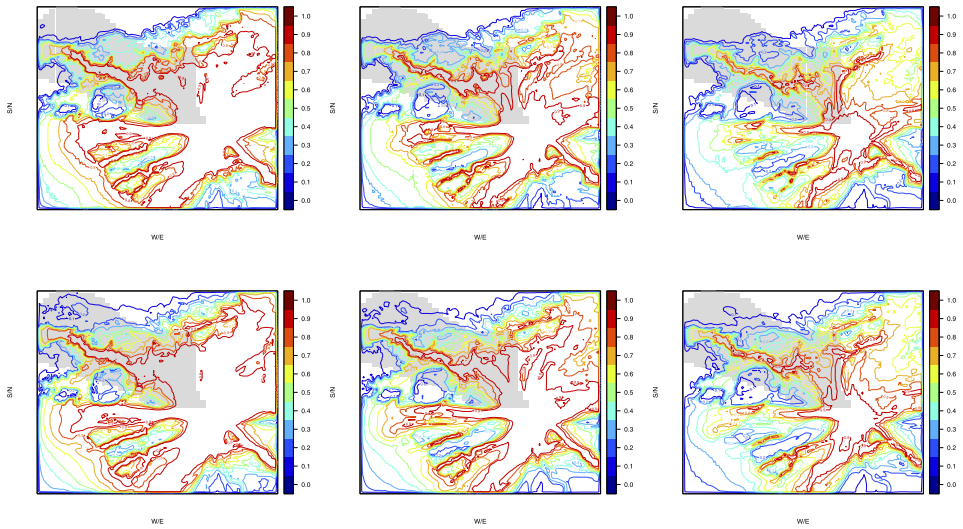

FIG. 3. *For SHV, contours of the probabilities that the maximum flow heights exceed* 0.5 (*left*), 1 (*center*) *and* 2 (*right*) *meters over the next* $T = 2.5$ *years at each location on SHV. The shaded area is Belham Valley, which is still inhabited. The results in the upper row utilized only* $N = 50$ *runs of TITAN2D to construct the PP GaSP, while the lower row results were based on utilizing all* $N = 2048$ *runs.*

flows within the next $T = 2.5$ years, while the lower parts of the valley have comparatively small hazard probability. The borders of some inhabited regions have probability larger than 0.1 of being hit by pyroclastic flows higher than 0.5 meter, which is a significant concern.

5. Validation and numerical comparisons. In this section, we study the performance of the PP emulator in the context of TITAN2D output, and compare it with the MS GaSP emulator [Conti and O'Hagan (2010)] and other emulators defined in Section 5.2.1. Initially, we thought that the MS GaSP emulator would be the gold standard since it allows for adaptation of the correlation parameters to the particular coordinate; in contrast, the PP emulator insists on the same correlation parameters across all coordinates. Quite surprisingly, we did not find this to be so. The comparisons between emulators will be in terms of computational cost and out-of-sample prediction.

5.1. *Computational cost.* It is useful to divide the computational cost of the PP emulator into three phases:

- The first [see (3.5)] is the one-time costs of computing $\tilde{\mathbf{R}}^{-1}$, $\mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\tilde{\mathbf{R}}^{-1}$ and $(\mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\tilde{\mathbf{R}}^{-1}\mathbf{h}(\mathbf{x}^{\mathscr{D}}))^{-1}$ and, estimating $(\hat{\boldsymbol{\gamma}}, \hat{\nu})$. The first three have maximum cost $O(n^3)$; we consider the cost of estimating $(\hat{\boldsymbol{\gamma}}, \hat{\nu})$ later.

- The second phase is computation of $\boldsymbol{\omega}(\mathbf{x}^*)$ in (3.5) at a new input $\mathbf{x}^*$, a computation of order $O(n^2)$ utilizing the phase one precomputations. This cost may seem minor compared to the phase one cost of $O(n^3)$, but, for many uses of the emulator, such as performing an MCMC analysis, this may have to be repeated many thousands of times, whereas the phase one computation is not repeated.
- Finally, the computation of the emulator mean in (3.5) is then $O(nk)$, which can be much larger than the phase one and two costs in our situation, since k can be much larger than n. (In the TITAN2D testbed, n is a maximum of 2048, while k can be as large as 10^9.) Similarly, it can be shown that the computational cost for computing all the variances of the PP GaSP is $O(n^2k)$; this is substantially more expensive than computing the PP emulator mean, but often it will only be necessary to compute the variances at some of the coordinates to obtain a feel for the accuracy of the emulator.

The MS emulator has different $(\hat{\boldsymbol{\gamma}}, \hat{\nu})$ and, hence, different $\tilde{\mathbf{R}}$ at each coordinate. The inversions of $\tilde{\mathbf{R}}$ thus have to be done k times in the precomputation stage, leading to a precomputational cost of order $O(n^3k)$. Even the MS emulator mean, after this precomputation, is an expensive $O(n^2k)$, essentially because a new $\boldsymbol{\omega}(\mathbf{x}^*)$ must be computed at each coordinate. Basically, when k is huge and n is large, use of the MS emulator is not computationally feasible.

Actually, the precomputation of $(\hat{\boldsymbol{\gamma}}, \hat{\nu})$ in the PP GaSP is the severest computational challenge if one attempts to use the full likelihood to estimate $(\hat{\boldsymbol{\gamma}}, \hat{\nu})$. The reason is that, for each candidate $(\hat{\boldsymbol{\gamma}}, \hat{\nu})$ used in trying to fit the full likelihood, a new inversion of $\tilde{\mathbf{R}}^{-1}$ is needed, and the subsequent computation of the likelihood (see Section 7.2) is of order $O(n^2k)$ for the PP GaSP emulator. Hence, the full cost of estimating $\hat{\boldsymbol{\gamma}}$ is $O(tn^2k) + O(tn^3)$, where t is the number of iterations needed in the estimation process. (In the testbed examples considered here, t is roughly 200.) For the MS emulator, the total computational cost involved in estimating the differing $\hat{\boldsymbol{\gamma}}$ at the coordinates is of order $O(tn^3k)$, which is prohibitive in settings such as here. The computational time in seconds for the PP GaSP and MS GaSP emulators are given in Table 1 and Table 2 for two computational scenarios; these actual times reflect the extreme theoretical disparity discussed above.

The expense of estimating $(\boldsymbol{\gamma}, \nu)$ suggests that various approximation strategies be utilized. In Section 7.3, we will consider two such strategies, basing the estimation on only subsets of the designed inputs $\mathbf{x}^{\mathscr{D}}$ and use of composite likelihoods.

5.2. *Out-of-sample prediction.* Here we compare the performance of the PP GaSP and MS GaSP emulators in out-of-sample prediction. We also include emulators from the next section in the comparison; these have been considered for situations similar to ours in the recent literature.

5.2.1. *Coregionalization emulators.* Another approach to emulation of multiple outputs is the Linear Model of Coregionalization (LMC) emulator [Fricker,

Oakley and Urban (2013)]. In this approach, the output $\mathbf{Y}(\mathbf{x})_{[k\times 1]}$ is modeled as

$$\mathbf{Y}(\mathbf{x}) = \boldsymbol{\mu}(\mathbf{x}) + \mathbf{A}\mathbf{v}(\mathbf{x}) + \boldsymbol{\varepsilon}, \tag{5.1}$$

where $\mathbf{A}$ is a $k \times k_0$ matrix, $k_0 < k$, and $\mathbf{v}(\mathbf{x}) = (\mathbf{v}_1(\mathbf{x}), \ldots, \mathbf{v}_{k0}(\mathbf{x}))^T$, with the $\mathbf{v}_i(\mathbf{x})$ being zero mean independent GaSP emulators and $\boldsymbol{\varepsilon}$ is independent noise. Denote the observed output matrix as $\mathbf{Y}_{[k\times n]} = (\mathbf{Y}(\mathbf{x}_1), \ldots, \mathbf{Y}(\mathbf{x}_n))$. In Higdon et al. (2008), the output is normalized and then represented by a singular value decomposition (SVD), $\mathbf{Y} - \bar{\mathbf{Y}}_{\text{row}} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, where $\bar{\mathbf{Y}}_{\text{row}}$ is the row mean of $\mathbf{Y}$. $\mathbf{A}$ is then estimated as the first k_0 columns of $\mathbf{U}\mathbf{D}/\sqrt{n}$. In Paulo, García-Donato and Palomo (2012), $\mathbf{A}$ is estimated as the column in the eigen-decomposition of the observed variance matrix of $\mathbf{Y}$, and $\boldsymbol{\varepsilon}$ is omitted because no dimension reduction is used. In Rougier (2008), dimension reduction with varying $\mathbf{h}(\mathbf{x})$ in each coordinate is also discussed.

Note that, when $k > n$, nonzero singular values of $\mathbf{Y}$ by SVD and eigenvalues by eigen-decomposition are equal to or smaller than n. In our situation, $k \gg n$, while the rank of the estimated $\hat{\mathbf{A}}$ is at most n, using either of these two approaches. In the numerical comparisons we will include the LMC emulator, with $\mathbf{A}$ being estimated by the eigenvectors of the observed covariance matrix of the output [Paulo, García-Donato and Palomo (2012)]. We will also compare the method of estimating the correlation parameters and nugget that is developed in Section 7, which we call *robust estimation*, with the standard method in the DiceKriging package [Roustant, Ginsbourger and Deville (2012)].

5.2.2. *Design of the numerical study.* To evaluate the accuracy of various variants of the PP emulator and alternative emulators, we divide the simulator runs into two parts, those used for development of the emulator and those used for out-of-sample assessment of accuracy. We utilized only $n = 50$ runs to design the emulator because of the extreme computational difficulty of working with the MS emulator (the primary emulator for comparison) for larger n, as discussed in Section 5.1. Also, we surprisingly found that the PP-emulator based on only 50 runs is quite accurate, and using a large number of runs to build the emulators would likely have made it more difficult to see differences or problems.

We consider two evaluation scenarios. The first encompasses the entire island except the crater, but is limited to flow volumes $6 < \log_{10} V < 7.5$; 683 runs are available in this region. The second scenario focuses on regions of the island with moderate to small expected flows, since these regions are the subject of current major risk assessment. We omit the crater region from the analysis because there is no interest in hazard prediction there, and the flows are so large that they could adversely affect the estimation of the GaSP correlation parameters. Locations where all 50 simulator runs had maximum flow heights of zero were also eliminated from the analysis. The total number of remaining spatial coordinates was 23,040; $k = 17{,}311$ coordinates for the first case and $k = 14{,}911$ for the second. Of course,

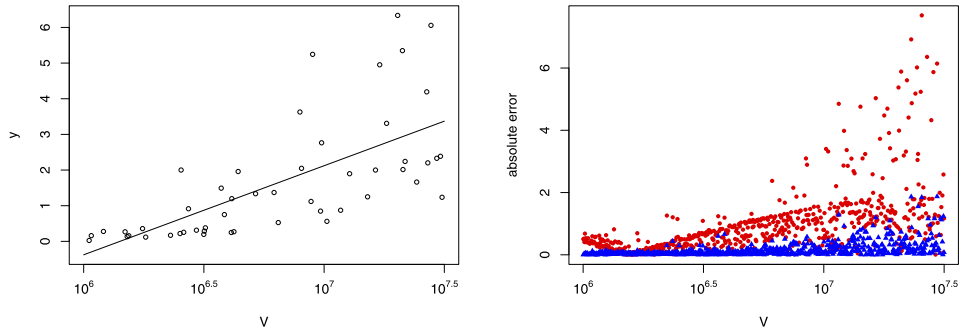

FIG. 4. *The left figure is the least squares fit of simulator output to volume for* 50 *simulator runs at a specific location. The right figure compares use of this least squares fit to estimate the outputs of* 633 *other simulator runs* (*the red dots*), *corresponding to other input values at the same location, with use of the PP GaSP* (*developed from the same* 50 *simulator runs*) *to estimate the* 633 *outputs* (*the blue triangles*). *Accuracy is measured by the absolute error of the prediction* $|y(x_i^*) - \hat{y}(x_i^*)|$.

utilizing a single emulator over such a large domain might well not work, and a natural strategy to consider is to divide the domain into more homogeneous regions and develop separate emulators over each region; luckily, this did not seem to be necessary for our scientific application.

Before proceeding with the complex emulators, it is useful to check that simple methods, such as linear regression, are not adequate for the problem. Thus, Figure 4 compares the use of simple linear regression of the output versus V at a specific location, based on the 50 training runs of the simulator, with the PP GaSP built on the same 50 runs, for predicting the 633 other simulator runs. Clearly, the linear regression estimates are far less accurate.

5.2.3. *Prediction criteria.* Diagnostics for GaSP emulation have been discussed in Bastos and O'Hagan (2009). The criteria that we focus on are out-of-sample prediction and accuracy in uncertainty quantification. In the following, we denote x_i^*, $1 \le i \le n^*$, as the held-out runs to verify the performance of the emulators; we have $n^* = 633$. The specific criteria employed are the following:

$$\mathrm{MSE} = \frac{\sum_{j=1}^{k}\sum_{i=1}^{n^*}(y_j(\mathbf{x}_i^*) - \hat{y}_j(\mathbf{x}_i^*))^2}{kn^*},$$

$$P_{\mathrm{CI}}(95\%) = \frac{1}{kn^*}\sum_{j=1}^{k}\sum_{i=1}^{n^*} 1\{y_j(\mathbf{x}_i^*) \in \mathrm{CI}_{ij}(95\%)\},$$

$$L_{\mathrm{CI}}(95\%) = \frac{1}{kn^*}\sum_{j=1}^{k}\sum_{i=1}^{n^*} \mathrm{length}\{\mathrm{CI}_{ij}(95\%)\},$$

where $\hat{y}_j(\mathbf{x}_i^*)$ is the prediction of the output of the ith held-out run, $\mathbf{x}_i^*$, at the jth spatial coordinate; $\mathrm{CI}_{ij}(95\%)$ is the 95% posterior credible interval based on (3.2);

TABLE 1

Performance of various emulators of max flow height over spatial grids in all locations except the crater area and nonflow areas. The first emulator uses all 4 *inputs, while the remaining four emulators use* 3 *inputs* $(V, \delta_{\text{bed}}, \phi)$ *and nugget(s), all with the same regressor* $\mathbf{h}(\mathbf{x}) = (1, V)$. *The emulators are evaluated based on* $n^* = 633$ *held-out inputs over* $k = 17{,}311$ *locations. The last row shows the computational time needed to estimate the correlation parameters and nuggets in the emulators (the dominant part of the computational cost) using R and* [C++]

	4 inputs	**3 inputs and estimated nugget(s)**			
	PP GaSP robust est.	**PP GaSP robust est.**	**MS GaSP robust est.**	**MS GaSP DiceKriging**	**LMC GaSP robust est.**
MSE	0.109	0.097	0.103	0.114	0.123
$P_{\text{CI}}(95\%)$	0.926	0.950	0.924	0.900	0.903
$L_{\text{CI}}(95\%)$	0.521	0.536	0.491	0.462	0.449
Time for $\boldsymbol{\gamma}$ and v (s)	50.0	28.1 [2.0]	31,337.7	4493.2	83.6

and length$\{\text{CI}_{ij}(95\%)\}$ is the length of the 95% posterior credible interval. An ideal emulator would have relatively low Mean Square Error (MSE), $P_{\text{CI}}(95\%)$ close to the 95% nominal level and short average credible interval lengths.

5.2.4. *Emulation over the noncrater region with constrained flow volumes*. Table 1 presents the results for the $k = 17{,}311$ noncrater locations with constrained flow volumes.

First, note that the computation times for the emulators reflect what was discussed in Section 5.1; the PP GaSP emulator is roughly *three orders of magnitude* faster than the MS emulator. The MS emulator with parameters estimated by DiceKriging was faster because it incorporated certain optimization techniques and the underlying codes were written in C, but it was still two orders of magnitude slower than PP GaSP using R codes. The speed of the LMC emulator was similar to PP GaSP because it projects the k-dimensional space onto a n-dimensional subspace (as discussed in Section 5.2.1).

The PP GaSP emulator based on three inputs and the nugget outperformed the PP GaSP emulator based on four inputs. It had better MSE and more accurate coverage, with only slightly longer credible intervals. This was also true for the second test situation, as evidenced in Table 2.

The PP GaSP emulator had the lowest out-of-sample MSE result among the four emulators based on three inputs and a nugget and, as importantly, produced 95% credible intervals that actually covered approximately 95% of the held-out outputs. In contrast, the other emulators were overconfident in their accuracy assessments. This is not surprising for the LMC GaSP emulator, since its projection onto an n-dimensional subspace is too restrictive, but it is surprising for the MS emulator, which we had entertained as being the gold standard because of its increased flexibility. The average length of the credible intervals for the PP emulator

TABLE 2

Performance of various emulators of max flow height over the $k = 14{,}911$ locations in the moderate to small flow area. The first emulator uses 4 inputs, while the remaining four emulators use 3 inputs $(V, \delta_{\text{bed}}, \phi)$ and nugget(s), all with the same regressor $\mathbf{h}(\mathbf{x}) = (1, V)$. The emulators are evaluated based on $n^* = 633$ held-out inputs. The last row shows the computational time needed to estimate the correlation parameters and nuggets in the emulators (the dominant part of the computational cost) using R and [C++]

	4 inputs	**3 inputs and estimated nugget(s)**			
	PP GaSP robust est.	**PP GaSP robust est.**	**MS GaSP robust est.**	**MS GaSP DiceKriging**	**LMC GaSP robust est.**
MSE	0.057	0.050	0.055	0.061	0.062
$P_{\text{CI}}(95\%)$	0.930	0.950	0.924	0.900	0.900
$L_{\text{CI}}(95\%)$	0.350	0.358	0.319	0.299	0.298
Time for $\boldsymbol{\gamma}$ and ν (s)	42.0	38.8 [2.1]	27,150.9	3835.4	81.2

with 3 inputs and a nugget was slightly longer than for the other emulators, but, again, that is very likely due to the other emulators being overconfident.

The MSE's of all of the emulators are rather impressive, especially when realizing that the output values they are predicting ranged from 0 to 40 in the noncrater area. Likewise, the small average size of the credible intervals is impressive for predicting outputs over that range.

The reason for the comparatively poor performance of the MS emulator is that fairly often (i.e., at some significant fraction of the coordinates) the estimates of the correlation parameters are bad because (i) only a limited number of computer runs are used ($n = 50$); (ii) each location will have many simulator runs with zero flow heights, which can cause problems for Gaussian processes. The first issue could be dealt with by using more simulator runs to develop the emulators, but this drastically increases the computational cost. The second issue, however, is generic in emulating the TITAN2D computer model; each pyroclastic flow will only hit some of the locations on the island, with the others receiving zero flow. In contrast, while the PP emulator may not have the optimal correlation parameters at any coordinate, the stability of their estimation ensures good average prediction.

The MS emulator implemented via the DiceKriging package estimates both the range parameters $\boldsymbol{\gamma}$ and smoothness parameters α of the power exponential correlation function, and typically results in smaller out-of-sample MSE when a large or moderate number of runs are used. However, it is not performing as well as the MS emulators with robust estimation of the range and nugget parameters, possibly because of the periodic folding adjustment [Spiller et al. (2014)] for the initial flow angle ϕ but probably because of the likely superiority of the robust estimation of the range and nugget parameters that is given in Section 7. For further discussion of this, see Gu (2016).

The LMC GaSP emulator using eigen-decomposition to estimate the orthogonal basis matrix $\mathbf{A}$ performs the worst among 5 emulators. Using an orthogonal basis with 50 dimensions does not seem to be flexible enough to capture the variations among the $k = 17{,}911$ locations.

5.2.5. *Emulation over the region having only moderate to small flows.* Table 2 presents the MSE results for the 14,911 locations in the small to moderate flow region. The PP GaSP outperforms the MS GaSP by more than 10% in terms of MSE and again has considerably better confidence properties. The degraded performance of the MS GaSP here is probably due to the fact that the small flow regions have numerous 0 max flow heights, which can cause problems in the estimation of the range and nugget parameters. And, of course, the computational advantage of the PP emulator was enormous.

6. The near irrelevance of spatial correlation in emulator construction. A seemingly natural extension of the PP emulator is to introduce spatial correlation into the model, as in Conti and O'Hagan (2010), in recognition of the fact that there is typically strong spatial dependence between simulator outputs at nearby inputs. (Recall that the PP emulator assumes each output is independent.) To keep the computation manageable, the spatial correlations and model input correlations are typically presumed to be separable, that is, the covariance function for $\mathbf{y}^{\mathscr{D}}$, conditional on $\boldsymbol{\Theta}$, is assumed to be a Kronecker product of a $k \times k$ spatial correlation matrix $\boldsymbol{\Sigma}$ and the $n \times n$ input correlation matrix $\mathbf{R}$, leading to the following matrix-normal density for the Gaussian process:

$$
\begin{aligned}
&p\big(\mathbf{y}^{\mathscr{D}} | \boldsymbol{\Sigma}, \boldsymbol{\Theta}, \boldsymbol{\gamma}\big) \\
&\quad = \frac{\exp(-\frac{1}{2}\operatorname{tr}[\boldsymbol{\Sigma}^{-1}(\mathbf{y}^{\mathscr{D}} - \mathbf{h}(\mathbf{x}^{\mathscr{D}})\boldsymbol{\Theta})^T \mathbf{R}^{-1}(\mathbf{y}^{\mathscr{D}} - \mathbf{h}(\mathbf{x}^{\mathscr{D}})\boldsymbol{\Theta})])}{(2\pi)^{nk/2}|\boldsymbol{\Sigma}|^{n/2}|\mathbf{R}|^{k/2}},
\end{aligned} \tag{6.1}
$$

where $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_k)$ is the $q \times k$ matrix of parameters of the mean function for the k spatial coordinates.

In Conti and O'Hagan (2010), a Jeffreys-type noninformative prior,

$$
\pi(\boldsymbol{\Theta}, \boldsymbol{\Sigma}) \propto |\boldsymbol{\Sigma}|^{-(k+1)/2}, \tag{6.2}
$$

was considered, since one can then exactly marginalize out $\boldsymbol{\Theta}$ and $\boldsymbol{\Sigma}$ when k is small. This does not work, however, if $k > n - q$, the situation we are considering, since there is then a nonintegrable singularity in the posterior at $\boldsymbol{\Sigma} = \mathbf{0}$.

A wide variety of other prior distributions on $\boldsymbol{\Sigma}$ can be considered, including priors that effectively give $\boldsymbol{\Sigma}$ a lower dimensional structure. Indeed, we propose one such prior in the Supplementary Materials [Gu and Berger (2016)] which is effective in smoothing random draws from the PP emulator; recall that, because of the independence assumption at each coordinate, draws directly from the PP GaSP

emulator will be quite rough, although the median, mean and quantiles of the PP GaSP are smooth.

Surprisingly, however, for *any* prior on $\boldsymbol{\Sigma}$, the resulting emulator mean will simply be the PP emulator mean (assuming the usual constant prior is used for the parameters of the mean function), and the resulting emulator variance function will almost equal the PP emulator variance function. Thus, there is no need to introduce spatial correlation structure into the emulator with regard to the response space if only the mean and pointwise variance functions are concerned. This delightful simplification is established in the next theorem.

THEOREM 6.1. *For the GaSP with separable covariance structure in* (6.1), *given correlation parameters* $\boldsymbol{\gamma}$ *and the objective prior*

$$\pi(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k | \boldsymbol{\Sigma}, \boldsymbol{\gamma}) \propto 1 \tag{6.3}$$

for the parameters of the mean function, the following hold:

1. *The posterior mean of the GaSP, for an unobserved* $\mathbf{x}^*$ *and at coordinate* j, *is identical to the PP emulator posterior mean in* (3.3).

2. *The posterior variance of the GaSP, for an unobserved* $\mathbf{x}^*$ *and at coordinate* j, *depends on* $\boldsymbol{\Sigma}$ *only through the posterior mean of the* j*th diagonal term,* $\mathrm{E}[\boldsymbol{\Sigma}_{jj} | \mathbf{y}^{\mathscr{D}}, \boldsymbol{\gamma}]$; *it is identical to the PP emulator posterior variance if* $\mathrm{E}[\boldsymbol{\Sigma}_{jj} | \mathbf{y}^{\mathscr{D}}, \boldsymbol{\gamma}] = \frac{(n-q)\hat{\sigma}_j^2}{n-q-2}$, *with* $\hat{\sigma}_j^2$ *defined in* (3.4), *under the new prior for* $\boldsymbol{\Sigma}$.

PROOF. See Appendix A. □

Note that, when $n - q$ is moderately large, as is usually the case, (3.4) will approximately equal the new posterior expectation of σ_j^2, since almost all the information about σ_j^2 is contained in the likelihood, not the prior. Thus, in practice, one can just use the PP emulator mean and variance, and ignore the spatial structure, unless draws from the emulator are required.

7. Estimating the correlation parameters. In this section, we discuss estimation of the correlation parameter $\boldsymbol{\gamma}$. First, the reference prior for $\boldsymbol{\gamma}$ in the situation of vector-valued outputs (as considered in this paper) is derived. Next, an estimation strategy is proposed, utilizing the posterior mode. Finally, to overcome the computational challenge, a composite likelihood approach is considered.

7.1. *The reference priors for vector output.* When dealing with a Gaussian process with a single real output, the reference prior under an isotropic kernel was derived in Berger, De Oliveira and Sansó (2001) and under a product correlation matrix in Bayarri et al. (2009). As recommended in Berger, De Oliveira and Sansó (2001) and Paulo (2005), we follow the strategy of first marginalizing out the parameters of the mean function (with respect to a constant prior) and then deriving

the reference prior for $\boldsymbol{\gamma}$ from the marginal likelihood; the result is given in the following theorem.

THEOREM 7.1 (Reference prior for PP GaSP without a nugget). *The reference prior of the PP GaSP for vector output has the form*

$$\pi^R(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k, \sigma_1^2, \ldots, \sigma_k^2, \boldsymbol{\gamma}) \propto \frac{\pi^R(\boldsymbol{\gamma})}{\prod_{i=1}^k \sigma_i^2},$$

with $\pi^R(\boldsymbol{\gamma}) \propto |\mathbf{I}^*(\boldsymbol{\gamma})|^{1/2}$, *where* $\mathbf{I}^*(\boldsymbol{\gamma})$ *is the expected Fisher information matrix*

$$\mathbf{I}^*(\boldsymbol{\gamma}) = \begin{pmatrix} n-q & \operatorname{tr}(\mathbf{W}_1) & \operatorname{tr}(\mathbf{W}_2) & \cdots & \operatorname{tr}(\mathbf{W}_p) \\ & \operatorname{tr}(\mathbf{W}_1^2) & \operatorname{tr}(\mathbf{W}_1\mathbf{W}_2) & \cdots & \operatorname{tr}(\mathbf{W}_1\mathbf{W}_p) \\ & & \operatorname{tr}(\mathbf{W}_2^2) & \cdots & \operatorname{tr}(\mathbf{W}_2\mathbf{W}_p) \\ & & & \ddots & \vdots \\ & & & & \operatorname{tr}(\mathbf{W}_p^2) \end{pmatrix}_{(p+1)\times(p+1)}, \tag{7.1}$$

with $\mathbf{W}_t = \dot{\mathbf{R}}_t\mathbf{Q}$, *for* $1 \le t \le p$, *where* p *is the number of range parameters in the correlation matrix* $\mathbf{R}$, $\dot{\mathbf{R}}_t$ *is the derivative of* $\mathbf{R}$ *with respect to the* t*th range parameter, and* $\mathbf{Q} = \mathbf{R}^{-1}\mathbf{P}$ *with* $\mathbf{P} = \mathbf{I} - \mathbf{h}(\mathbf{x}^{\mathscr{D}})\{\mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\mathbf{R}^{-1}\mathbf{h}(\mathbf{x}^{\mathscr{D}})\}^{-1}\mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\mathbf{R}^{-1}$.

PROOF. See Appendix B. □

THEOREM 7.2 (Reference prior for PP GaSP with a nugget). *The reference prior of the PP GaSP with a nugget, for vector output, has the form*

$$\pi^{\tilde{R}}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k, \sigma_1^2, \ldots, \sigma_k^2, \boldsymbol{\gamma}, \nu) \propto \frac{\pi^{\tilde{R}}(\boldsymbol{\gamma}, \nu)}{\prod_{i=1}^k \sigma_i^2},$$

with $\pi^{\tilde{R}}(\boldsymbol{\gamma}, \nu) \propto |\tilde{\mathbf{I}}^*(\boldsymbol{\gamma}, \nu)|^{1/2}$, *where* $\tilde{\mathbf{I}}^*(\boldsymbol{\gamma}, \nu)$ *is the expected Fisher information matrix*

$$\tilde{\mathbf{I}}^*(\boldsymbol{\gamma}, \nu) = \begin{pmatrix} n-q & \operatorname{tr}(\tilde{\mathbf{W}}_1) & \operatorname{tr}(\tilde{\mathbf{W}}_2) & \cdots & \operatorname{tr}(\tilde{\mathbf{W}}_{p+1}) \\ & \operatorname{tr}(\tilde{\mathbf{W}}_1^2) & \operatorname{tr}(\tilde{\mathbf{W}}_1\tilde{\mathbf{W}}_2) & \cdots & \operatorname{tr}(\tilde{\mathbf{W}}_1\tilde{\mathbf{W}}_{p+1}) \\ & & \operatorname{tr}(\tilde{\mathbf{W}}_2^2) & \cdots & \operatorname{tr}(\tilde{\mathbf{W}}_2\tilde{\mathbf{W}}_{p+1}) \\ & & & \ddots & \vdots \\ & & & & \operatorname{tr}(\tilde{\mathbf{W}}_{p+1}^2) \end{pmatrix}_{(p+2)\times(p+2)}, \tag{7.2}$$

with $\tilde{\mathbf{W}}_t = \dot{\tilde{\mathbf{R}}}_t\tilde{\mathbf{Q}}$, *for* $1 \le t \le p$, *where* p *is the number of range parameters in* $\tilde{\mathbf{R}}$, $\dot{\tilde{\mathbf{R}}}_t$ *is the derivative of* $\tilde{\mathbf{R}}$ *with respect to the* t*th range parameter, and* $\tilde{\mathbf{Q}} = \tilde{\mathbf{R}}^{-1}\tilde{\mathbf{P}}$ *with* $\tilde{\mathbf{P}} = \mathbf{I} - \mathbf{h}(\mathbf{x}^{\mathscr{D}})\{\mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\tilde{\mathbf{R}}^{-1}\mathbf{h}(\mathbf{x}^{\mathscr{D}})\}^{-1}\mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\tilde{\mathbf{R}}^{-1}$.

PROOF. The proof is a direct generalization of Ren, Sun and He (2012), essentially following the same steps as the proof of Theorem 7.1. □

7.2. *Marginal posterior.* We will utilize the marginal posterior density of $\boldsymbol{\gamma}$ and ν to perform the estimation for these parameters. Starting with the full likelihood, multiplying by the reference prior and integrating out the parameters of the mean function, $(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k)$, and variance parameters, $(\sigma_1^2, \ldots, \sigma_k^2)$, results in

$$p(\boldsymbol{\gamma}, \nu|\mathbf{y}^{\mathscr{D}}) \propto L(\mathbf{y}^{\mathscr{D}}|\boldsymbol{\gamma}, \nu)\big|\tilde{\mathbf{I}}^*(\boldsymbol{\gamma}, \nu)\big|^{1/2}, \tag{7.3}$$

with

$$L(\mathbf{y}^{\mathscr{D}}|\boldsymbol{\gamma}, \nu) \propto |\tilde{\mathbf{R}}|^{-k/2}\big|\mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\tilde{\mathbf{R}}^{-1}\mathbf{h}(\mathbf{x}^{\mathscr{D}})\big|^{-k/2}\prod_{i=1}^{k}\big[(\mathbf{y}_i^{\mathscr{D}})^T\tilde{\mathbf{Q}}\mathbf{y}_i^{\mathscr{D}}\big]^{-(n-q)/2}. \tag{7.4}$$

We reparameterize the range parameters and the nugget by $(\xi_1, \ldots, \xi_p, \tau) = (\log(1/\gamma_1^{\alpha_1}), \ldots, \log(1/\gamma_p^{\alpha_p}), \log(\nu))$. We then estimate the parameters $(\boldsymbol{\xi}, \tau)$ as the mode of this marginal posterior, namely,

$$(\hat{\xi}_1, \ldots, \hat{\xi}_p, \hat{\tau}) = \operatorname*{argmax}_{\xi_1,\ldots,\xi_p,\tau} L(\mathbf{y}^{\mathscr{D}}|\xi_1, \ldots, \xi_p, \tau)\pi^{\tilde{R}}(\xi_1, \ldots, \xi_p, \tau). \tag{7.5}$$

The difference between the posterior mode of $(\boldsymbol{\gamma}, \nu)$ and $(\boldsymbol{\xi}, \tau)$ arises because of the Jacobian of the transformation. As discussed in Spiller et al. (2014) and Gu (2016), the marginal likelihood alone can have bad behavior, such as being maximized as parameters go to infinity. Using the marginal posterior, with respect to the reference prior, seems to substantially eliminate such bad behavior and achieve comparatively better results.

Note that there have been a variety of parameterizations for GaSP's that have been used in the past literature other than the $\boldsymbol{\gamma}$ parameterization in (2.2). The parameterization $\beta_j = 1/\gamma_j^{\alpha_j}$ was discussed in Paulo (2005) and the parameterization $\xi_j = \log(1/\gamma_j^{\alpha_j})$ was introduced in Spiller et al. (2014). The effectiveness of the different parameterizations, when combined with use of the posterior mode, is extensively discussed in Gu (2016), where a robustness argument in favor of the above parameterization is given. For the MS GaSP with a nugget, the same strategy is used: form the marginal posterior distribution of $(\boldsymbol{\xi}, \tau)$ and utilize the posterior mode of these parameters in the MS GaSP.

One concern with using (7.4) is that the assumption of independence of coordinates is clearly wrong, so that this likelihood is almost certainly much too concentrated. This, by itself, would not be a problem, since we are simply using it to obtain estimates of the correlation parameters, but it is possible that the likelihood would also be biased in some way. To investigate this, we define the following "oracle estimator," which views the posterior predictive mean in Lemma 3.1 as a

TABLE 3
MSE comparison between PP GaSP and Oracle PP GaSP with $n = 50$ *and* $n^* = 633$

	PP GaSP	**Oracle PP GaSP**
MSE at noncrater area	0.09726675	0.09464283
MSE at small flow area	0.04964399	0.04837848

function of the range parameters and nugget $(\boldsymbol{\xi}, \tau)$ and optimizes over the choice of these parameters:

$$(\xi_1^{\text{oracle}}, \ldots, \xi_p^{\text{oracle}}, \tau^{\text{oracle}}) = \underset{\xi_1,\ldots,\xi_p,\tau}{\operatorname{argmin}} \frac{\sum_{j=1}^{k}\sum_{i=1}^{n^*}\left(y_j(\mathbf{x}_i^*) - \hat{y}_j^{\text{oracle}}(\mathbf{x}_i^*)\right)^2}{kn^*}, \tag{7.6}$$

where

$$\hat{y}_j^{\text{oracle}}(\mathbf{x}_i^*) = \boldsymbol{\omega}(\xi_1, \ldots, \xi_p, \tau)\mathbf{y}_j^{\mathscr{D}},$$

$$\boldsymbol{\omega}(\xi_1, \ldots, \xi_p, \tau) = \left(\mathbf{h}(\mathbf{x}^*) - \mathbf{r}^T(\mathbf{x}^*)\tilde{\mathbf{R}}^{-1}\mathbf{h}(\mathbf{x}^{\mathscr{D}})\right)\left(\mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\tilde{\mathbf{R}}^{-1}\mathbf{h}(\mathbf{x}^{\mathscr{D}})\right)^{-1} \times \mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\tilde{\mathbf{R}}^{-1} + \mathbf{r}^T(\mathbf{x}^*)\tilde{\mathbf{R}}^{-1}.$$

Table 3 compares the MSE of the oracle PP GaSP and the PP GaSP. For both of the regions under consideration, the PP GaSP has almost the same MSE as the oracle so that the use of (7.4) in estimating the correlation parameters and nugget seems justified.

7.3. *Using composite likelihood.* As discussed in Section 5.1, the major computational challenge in developing the PP emulator is estimating the parameters $(\boldsymbol{\gamma}, \nu)$, the computation being of order $O(tn^2k) + O(tn^3)$, with t being the number of iterations (typically about 200) needed to find a good approximation to the marginal posterior mode discussed in the previous section. A variety of strategies have been proposed to reduce this computational burden. Use of covariance tapering and compactly supported correlation functions were studied and successfully applied to large spatial datasets in Kaufman, Schervish and Nychka (2008) and Kaufman et al. (2011). Other possibilities include estimating the parameters using only some subsets of the input design points (i.e., significantly reduce n) or using only some coordinates of the simulator output (i.e, significantly reduce k).

Another approach, and that which we will adopt here, is to use the marginal composite likelihood for the input parameter estimation, in that this can be done in a way that guarantees accurate parameter estimation (at least asymptotically). The idea of composite likelihood can be traced back to the pseudo-likelihood [Besag

(1974)] and partial likelihood [Cox (1975)]. It has been studied intensively in recent years; see, for example, Lindsay, Yi and Sun (2011) and Varin, Reid and Firth (2011) for recent developments.

We will utilize the Independent Marginal Composite Likelihood (ICML) approach, replacing $L(\mathbf{y}^{\mathscr{D}}|\boldsymbol{\gamma}, \nu)$ in (7.3) by the following product of sublikelihoods:

$$L_C(\mathbf{y}^{\mathscr{D}}|\boldsymbol{\gamma}, \nu) = \prod_{t=1}^{T} \{L_t(\mathbf{y}_t^{\mathscr{D}}|\boldsymbol{\gamma}, \nu)\}, \tag{7.7}$$

where the $L_t(\mathbf{y}_t^{\mathscr{D}}|\boldsymbol{\gamma}, \nu)$ are "parts" of the full likelihood, formed from batches of subrows of the $n \times k$ matrix $\mathbf{y}^{\mathscr{D}}$, and it is assumed that there is no correlation between the different batches. Specifically, we form $T = n/n_0$ batches of the design inputs, each batch being of size n_0, by simple random sampling of the inputs. Imposing independence of the batches results in the correlation matrix over the input space

$$\tilde{\mathbf{R}}_C = \begin{pmatrix} \tilde{\mathbf{R}}_1 & & & \\ & \tilde{\mathbf{R}}_2 & & \\ & & \ddots & \\ & & & \tilde{\mathbf{R}}_T \end{pmatrix},$$

where each $\tilde{\mathbf{R}}_t$, $1 \le t \le T$ is a batch with m inputs and the other elements of the correlation matrix are 0. The composite marginal posterior for the parameters $(\boldsymbol{\gamma}, \nu)$ is then

$$p_C(\boldsymbol{\gamma}, \nu|\mathbf{y}^{\mathscr{D}}) \propto L_C(\mathbf{y}^{\mathscr{D}}|\boldsymbol{\gamma}, \nu)\pi^{\tilde{R}}(\boldsymbol{\gamma}, \nu).$$

Defining $(\hat{\boldsymbol{\gamma}}_C, \hat{\nu}_C)$ as the composite maximum likelihood estimator, under the regular conditions [Lindsay (1988), Severini (2000)], we have that, as $n \to \infty$,

$$\sqrt{n}[(\hat{\boldsymbol{\gamma}}_C, \hat{\nu}_C) - (\boldsymbol{\gamma}, \nu)] \xrightarrow{d} N(0, \mathbf{G}^{-1}),$$

where

$$\mathbf{G} = \mathbf{G}(\boldsymbol{\gamma}, \nu) = \mathbf{H}(\boldsymbol{\gamma}, \nu)\mathbf{J}^{-1}(\boldsymbol{\gamma}, \nu)\mathbf{H}(\boldsymbol{\gamma}, \nu),$$

$$\mathbf{H}(\boldsymbol{\gamma}, \nu) = -\mathrm{E}\left(\frac{\partial^2 L_C(\mathbf{y}^{\mathscr{D}}|\boldsymbol{\gamma}, \nu)}{\partial(\boldsymbol{\gamma}, \nu)^2}\right), \qquad \mathbf{J}(\boldsymbol{\gamma}, \nu) = \mathrm{Var}\left(\frac{\partial L_C(\mathbf{y}^{\mathscr{D}}|\boldsymbol{\gamma}, \nu)}{\partial(\boldsymbol{\gamma}, \nu)}\right).$$

Because of these asymptotic results, the use of the composite likelihood to estimate $(\boldsymbol{\gamma}, \nu)$ is reasonable when n is large.

In choosing n_0, we make use of the "folklore" notion that the number of design points necessary to effectively estimate p correlation parameters is $10p$. For TITAN2D, there are either 4 correlation parameters or 3 with a nugget, so n_0 should be at least 40. We then utilize $n_0 = 50$ to form each batch to see the performance.

TABLE 4

The MSE and computational time in seconds using R at the noncrater area and the small flow area based on $n = 200$ inputs. The first column uses ICML with block size $n_0 = 50$ to do estimation of the range and nugget parameters, and also uses the composite likelihood to do prediction. The second column uses the composite likelihood to do the parameter estimation, but uses the full likelihood for prediction. The third column shows the results for the full PP GaSP. The number of held-out runs for the evaluation is $n^* = 483$

MSE (and time in seconds)	**PP GaSP ICML block est, block pred**	**PP GaSP ICML block est, full pred**	**PP GaSP full lik full pred**
Noncrater area	0.088 (103.6 s)	0.063 (111.9 s)	0.062 (534.4 s)
Small flow area	0.050 (113.4 s)	0.034 (133.7 s)	0.033 (573.5 s)

Table 4 presents the MSE in prediction for three different ways of estimating the range and nugget parameters. The first two use the ICML approach with 4 blocks each and with $n = 50$; the first one utilizes averages of 4 blocks for prediction, while the second utilizes the full $n \times n$ matrix $\tilde{\mathbf{R}}$ for prediction, as it only requires one inversion. The third method is the full PP GaSP, using the full correlation matrix to do both estimation and prediction. Clearly, using the full correlation matrix for prediction is much better than merely using blocks for the prediction. Using the full likelihood for estimation of the range and nugget parameters is slightly better than using the ICML approach, but the difference is modest. And the second method needs only $O(tmnk + tm^2n)$ flops, as compared to $O(tn^2k + tn^3)$ flops for the full PP GaSP method, so the ICML approach can be very attractive.

APPENDIX A: PROOF OF THEOREM 6.1

The joint distribution of $\mathbf{y}(\mathbf{x}^*) = (y_1(\mathbf{x}^*), y_2(\mathbf{x}^*), \ldots, y_k(\mathbf{x}^*))$ and $\mathbf{y}^{\mathscr{D}}$ is a matrix normal distribution,

$$\begin{pmatrix} \mathbf{y}(\mathbf{x}^*) \\ \mathbf{y}^{\mathscr{D}} \end{pmatrix} \Big| \boldsymbol{\Theta}, \boldsymbol{\gamma}, \boldsymbol{\Sigma} \sim N_{(n+1),k}\left(\begin{pmatrix} \mathbf{h}(\mathbf{x}^*)\boldsymbol{\Theta} \\ \mathbf{h}(\mathbf{x})\boldsymbol{\Theta} \end{pmatrix}, \begin{pmatrix} c(\mathbf{x}^*, \mathbf{x}^*) & \mathbf{r}^T(\mathbf{x}^*) \\ \mathbf{r}(\mathbf{x}^*) & \mathbf{R} \end{pmatrix}, \Sigma \right),$$

where $N_{(n+1),k}(\cdot, \cdot, \cdot)$ is a $(n+1) \times k$ matrix normal distribution. From Gupta and Nagar (1999), it follows that

$$\mathrm{E}[\mathbf{y}(\mathbf{x}^*)|\mathbf{y}^{\mathscr{D}}, \Theta, \boldsymbol{\gamma}, \boldsymbol{\Sigma}] = \mathbf{h}(\mathbf{x}^*)\boldsymbol{\Theta} + \mathbf{r}^T(\mathbf{x}^*)\mathbf{R}^{-1}(\mathbf{y}^{\mathscr{D}} - \mathbf{h}(\mathbf{x}^{\mathscr{D}})\boldsymbol{\Theta}),$$

and, for the jth coordinate,

$$\mathrm{E}[\mathbf{y}_j(\mathbf{x}^*)|\mathbf{y}^{\mathscr{D}}, \boldsymbol{\theta}_j, \boldsymbol{\gamma}, \boldsymbol{\Sigma}] = \mathbf{h}(\mathbf{x}^*)\boldsymbol{\theta}_j + \mathbf{r}^T(\mathbf{x}^*)\mathbf{R}^{-1}(\mathbf{y}_j^{\mathscr{D}} - \mathbf{h}(\mathbf{x}^{\mathscr{D}})\boldsymbol{\theta}_j). \tag{A.1}$$

Using the objective prior in equation (6.3) results in

$$\boldsymbol{\theta}_j|\mathbf{y}^{\mathscr{D}}, \boldsymbol{\gamma}, \boldsymbol{\Sigma} \sim N(\hat{\boldsymbol{\theta}}_j, \Sigma_{jj}[(\mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\mathbf{R}^{-1}\mathbf{h}(\mathbf{x}^{\mathscr{D}}))]^{-1}),$$

where $\hat{\boldsymbol{\theta}}_j = \{\mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\mathbf{R}^{-1}\mathbf{h}(\mathbf{x}^{\mathscr{D}})\}^{-1}\mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\mathbf{R}^{-1}\mathbf{y}_j^{\mathscr{D}}$. Taking the expectation over $\boldsymbol{\theta}_j$ in (A.1) results in the expression for the posterior mean in (3.3), as was to be established.

For the posterior variance, after marginalizing out the parameters of the mean function with the objective prior in equation (6.3), it is shown in Conti and O'Hagan (2010) that

$$\begin{aligned}\operatorname{Var}[y_j(\mathbf{x}^*)|\mathbf{y}^{\mathscr{D}}, \boldsymbol{\gamma}, \boldsymbol{\Sigma}] = \sigma_j^2\{&(c(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{r}^T(\mathbf{x}^*)\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}^*)) \\ &+ [\mathbf{h}(\mathbf{x}^*) - \mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}^*)]^T[\mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\mathbf{R}^{-1}\mathbf{h}(\mathbf{x}^{\mathscr{D}})]^{-1} \\ &\times [\mathbf{h}(\mathbf{x}^*) - \mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}^*)]\},\end{aligned}$$

where $\sigma_j^2 = \Sigma_{jj}$. Now

$$\begin{aligned}\operatorname{Var}[y_j(\mathbf{x}^*)|\mathbf{y}^{\mathscr{D}}, \boldsymbol{\gamma}] = &\operatorname{Var}_{\boldsymbol{\Sigma}|\mathbf{y}^{\mathscr{D}},\boldsymbol{\gamma}}[\mathrm{E}[y_j(\mathbf{x}^*)|\mathbf{y}^{\mathscr{D}}, \boldsymbol{\gamma}, \boldsymbol{\Sigma}]] \\ &+ \mathrm{E}_{\boldsymbol{\Sigma}|\mathbf{y}^{\mathscr{D}},\boldsymbol{\gamma}}[\operatorname{Var}[y_j(\mathbf{x}^*)|\mathbf{y}^{\mathscr{D}}, \boldsymbol{\gamma}, \boldsymbol{\Sigma}]],\end{aligned}$$

but the first term is zero, since the posterior mean does not depend on $\boldsymbol{\Sigma}$. Noting that $\operatorname{Var}[y_j(\mathbf{x}^*)|\mathbf{y}^{\mathscr{D}}, \boldsymbol{\gamma}, \boldsymbol{\Sigma}] = \sigma_j^2 \times c^{**}$, where c^{**} is as in (2.4), it is immediate that

$$\operatorname{Var}[y_j(\mathbf{x}^*)|\mathbf{y}^{\mathscr{D}}, \boldsymbol{\gamma}] = \mathrm{E}[\sigma_j^2|\mathbf{y}^{\mathscr{D}}, \boldsymbol{\gamma}]c^{**},$$

as was to be established.

APPENDIX B: PROOF OF THEOREM 7.1

As in Berger, De Oliveira and Sansó (2001), we derive the reference prior based on the marginal likelihood after integrating out the parameters of the mean function $\boldsymbol{\theta}$ with a constant prior. The log marginal likelihood, conditional on $(\boldsymbol{\gamma}, \boldsymbol{\sigma}^2)$, is

$$\begin{aligned}\log(L(\mathbf{y}|\boldsymbol{\gamma}, \boldsymbol{\sigma}^2)) \propto &-\frac{n-q}{2}\sum_{i=1}^{k}\log(\sigma_i^2) - \frac{k}{2}\log(|\mathbf{R}|) \\ &- \frac{k}{2}\log(|\mathbf{h}^T(\mathbf{x}^{\mathscr{D}})\mathbf{R}^{-1}\mathbf{h}(\mathbf{x}^{\mathscr{D}})|) - \frac{(n-q)}{2}\sum_{i=1}^{k}\log(S_i^2),\end{aligned} \tag{B.1}$$

with

$$S_i^2 = (\mathbf{y}_i^{\mathscr{D}})^T\mathbf{Q}\mathbf{y}_i^{\mathscr{D}}. \tag{B.2}$$

As in the proof of Theorem 2 in Berger, De Oliveira and Sansó (2001), direct computation yields

$$\mathrm{E}\left(\frac{\partial \log(L(\mathbf{y}|\boldsymbol{\gamma}, \boldsymbol{\sigma}^2))}{\partial \sigma_i^2}\right)^2 = \frac{n-q}{2\sigma_i^4},$$

$$\mathrm{E}\left(\frac{\partial \log(L(\mathbf{y}|\boldsymbol{\gamma},\boldsymbol{\sigma}^2))}{\partial \sigma_i^2}\frac{\partial \log(L(\mathbf{y}|\boldsymbol{\gamma},\boldsymbol{\sigma}^2))}{\partial \sigma_j^2}\right)=0,$$

$$\mathrm{E}\left(\frac{\partial \log(L(\mathbf{y}|\boldsymbol{\gamma},\boldsymbol{\sigma}^2))}{\partial \gamma_l}\right)^2=\frac{k}{2}\operatorname{tr}(\mathbf{W}_l^2),$$

$$\mathrm{E}\left(\frac{\partial \log(L(\mathbf{y}|\boldsymbol{\gamma},\boldsymbol{\sigma}^2))}{\partial \gamma_l}\frac{\partial \log(L(\mathbf{y}|\boldsymbol{\gamma},\boldsymbol{\sigma}^2))}{\partial \gamma_m}\right)=\frac{k}{2}\operatorname{tr}(\mathbf{W}_l\mathbf{W}_m),$$

$$\mathrm{E}\left(\frac{\partial \log(L(\mathbf{y}|\boldsymbol{\gamma},\boldsymbol{\sigma}^2))}{\partial \sigma_i^2}\frac{\partial \log(L(\mathbf{y}|\boldsymbol{\gamma},\boldsymbol{\sigma}^2))}{\partial \gamma_l}\right)=\frac{1}{2\sigma_i^2}\operatorname{tr}(\mathbf{W}_l),$$

where $1 \le i \ne j \le k$ and $1 \le l \ne m \le p$. The Fisher information matrix is

$$|\mathbf{I}^*(\boldsymbol{\gamma},\boldsymbol{\sigma}^2)| \propto \begin{vmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B}^T & \mathbf{C}\end{vmatrix}_{(k+p)\times(k+p)}$$

with

$$\mathbf{A}=\begin{pmatrix}\frac{n-q}{2\sigma_1^4} & 0 & 0 & 0\\ 0 & \frac{n-q}{2\sigma_2^4} & 0 & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & \frac{n-q}{2\sigma_k^4}\end{pmatrix}_{k\times k},$$

$$\mathbf{B}=\begin{pmatrix}\frac{\operatorname{tr}(\mathbf{W}_1)}{2\sigma_1^2} & \frac{\operatorname{tr}(\mathbf{W}_2)}{2\sigma_1^2} & \cdots & \frac{\operatorname{tr}(\mathbf{W}_p)}{2\sigma_1^2}\\ \frac{\operatorname{tr}(\mathbf{W}_1)}{2\sigma_2^2} & \frac{\operatorname{tr}(\mathbf{W}_2)}{2\sigma_2^2} & \cdots & \frac{\operatorname{tr}(\mathbf{W}_p)}{2\sigma_2^2}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\operatorname{tr}(\mathbf{W}_1)}{2\sigma_k^2} & \frac{\operatorname{tr}(\mathbf{W}_2)}{2\sigma_k^2} & & \frac{\operatorname{tr}(\mathbf{W}_p)}{2\sigma_k^2}\end{pmatrix}_{k\times p},$$

$$\mathbf{C}=\begin{pmatrix}\frac{k\operatorname{tr}(\mathbf{W}_1^2)}{2} & \frac{k\operatorname{tr}(\mathbf{W}_1\mathbf{W}_2)}{2} & \cdots & \frac{k\operatorname{tr}(\mathbf{W}_1\mathbf{W}_p)}{2}\\ \frac{k\operatorname{tr}(\mathbf{W}_1\mathbf{W}_2)}{2} & \frac{k\operatorname{tr}(\mathbf{W}_2^2)}{2} & \cdots & \frac{k\operatorname{tr}(\mathbf{W}_2\mathbf{W}_p)}{2}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{k\operatorname{tr}(\mathbf{W}_1\mathbf{W}_p)}{2} & \frac{k\operatorname{tr}(\mathbf{W}_2\mathbf{W}_p)}{2} & \cdots & \frac{k\operatorname{tr}(\mathbf{W}_p^2)}{2}\end{pmatrix}_{p\times p}.$$

The result soon follows from $\pi^R(\boldsymbol{\gamma},\boldsymbol{\sigma}^2) \propto |\mathbf{I}^*(\boldsymbol{\gamma},\boldsymbol{\sigma}^2)|^{1/2}$.

Acknowledgments. The research of Mengyang Gu was part of his Ph.D. thesis at Duke University. The authors thank Abani Patra and Ramona Stefanescu from the department of Mechanical and Aerospace Engineering at University of Buffalo for providing simulation data of the TITAN2D computer model. They thank the Editor, the Associate Editor and two referees for their comments that substantially improved the article.

SUPPLEMENTARY MATERIAL

Supplement to "Parallel partial Gaussian process emulation for computer models with massive output" (DOI: 10.1214/16-AOAS934SUPP; .pdf). This supplement consists of three parts. The first part describes the "periodic folding" method for modeling the correlation between periodic inputs. The second part provides some numerical results that the PP GaSP emulator with a nugget is close to being an interpolator for the TITAN2D computer model. Part 3 discusses a prior for smoothing the draws of the PP GaSP emulator through block sampling.

REFERENCES

ANDRIANAKIS, I. and CHALLENOR, P. G. (2012). The effect of the nugget on Gaussian process emulators of computer models. *Comput. Statist. Data Anal.* **56** 4215–4228. MR2957866

BASTOS, L. S. and O'HAGAN, A. (2009). Diagnostics for Gaussian process emulators. *Technometrics* **51** 425–438. MR2756478

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

BAYARRI, M. J., BERGER, J. O., CAFEO, J., GARCIA-DONATO, G., LIU, F., PALOMO, J., PARTHASARATHY, R. J., PAULO, R., SACKS, J. and WALSH, D. (2007b). Computer model validation with functional output. *Ann. Statist.* **35** 1874–1906. MR2363956

BAYARRI, M. J., BERGER, J. O., CALDER, E. S., DALBEY, K., LUNAGOMEZ, S., PATRA, A. K., PITMAN, E. B., SPILLER, E. T. and WOLPERT, R. L. (2009). Using statistical and computer models to quantify volcanic hazards. *Technometrics* **51** 402–413. MR2756476

BAYARRI, M. J., BERGER, J. O., CALDER, E. S., PATRA, A. K., PITMAN, E. B., SPILLER, E. T. and WOLPERT, R. L. (2015). Probabilistic quantification of hazards: A methodology using small ensembles of physics-based simulations and statistical surrogates. *Int. J. Uncertain. Quantif.* **5** 297–325. MR3413743

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

BESAG, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *J. Roy. Statist. Soc. Ser. B* **36** 192–236. MR0373208

CONTI, S. and O'HAGAN, A. (2010). Bayesian emulation of complex multi-output and dynamic computer models. *J. Statist. Plann. Inference* **140** 640–651. MR2558393

COX, D. R. (1975). Partial likelihood. *Biometrika* **62** 269–276. MR0400509

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

FORRESTER, A., SOBESTER, A. and KEANE, A. (2008). *Engineering Design Via Surrogate Modelling*: *A Practical Guide*. Wiley, New York.

FRICKER, T. E., OAKLEY, J. E. and URBAN, N. M. (2013). Multivariate Gaussian process emulators with nonseparable covariance structures. *Technometrics* **55** 47–56. MR3038484

GELFAND, A. E., DIGGLE, P. J., FUENTES, M. and GUTTORP, P., eds. (2010). *Handbook of Spatial Statistics*. CRC Press, Boca Raton, FL. MR2761512

GU, M. (2016). Robust uncertainty quantification and scalable computation for computer models with massive output. Ph.D. thesis, Duke Univ.

GU, M. and BERGER, J. O. (2016). Supplement to "Parallel partial Gaussian process emulation for computer models with massive output." DOI:10.1214/16-AOAS934SUPP.

GUPTA, A. K. and NAGAR, D. K. (1999). *Matrix Variate Distributions*. CRC Press, Boca Raton.

HIGDON, D., GATTIKER, J., WILLIAMS, B. and RIGHTLEY, M. (2008). Computer model calibration using high-dimensional output. *J. Amer. Statist. Assoc.* **103** 570–583. MR2523994

IOOSS, B. and LEMAÎTRE, P. (2014). A review on global sensitivity analysis methods. Preprint. Available at arXiv:1404.2405.

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

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. *Ann. Appl. Stat.* **5** 2470–2492. MR2907123

KAZIANKA, H. and PILZ, J. (2012). Objective Bayesian analysis of spatial data with uncertain nugget and range parameters. *Canad. J. Statist.* **40** 304–327. MR2927748

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

KENNEDY, M., ANDERSON, C., O'HAGAN, A., LOMAS, M., WOODWARD, I., GOSLING, J. P. and HEINEMEYER, A. (2008). Quantifying uncertainty in the biospheric carbon flux for England and Wales. *J. Roy. Statist. Soc. Ser. A* **171** 109–135. MR2412649

LEE, L. A., CARSLAW, K. S., PRINGLE, K. J., MANN, G. W. and SPRACKLEN, D. V. (2011). Emulation of a complex global aerosol model to quantify sensitivity to uncertain parameters. *Atmos. Chem. Phys.* **11** 12253–12273.

LEE, L. A., CARSLAW, K. S., PRINGLE, K. J. and MANN, G. W. (2012). Mapping the uncertainty in global CCN using emulation. *Atmospheric Chemistry and Physics* **12** 9739–9751.

LI, R. and SUDJIANTO, A. (2005). Analysis of computer experiments using penalized likelihood in Gaussian kriging models. *Technometrics* **47** 111–120. MR2188073

LINDSAY, B. G. (1988). Composite likelihood methods. In *Statistical Inference from Stochastic Processes* (*Ithaca*, *NY*, 1987). *Contemp. Math.* **80** 221–239. Amer. Math. Soc., Providence, RI. MR0999014

LINDSAY, B. G., YI, G. Y. and SUN, J. (2011). Issues and strategies in the selection of composite likelihoods. *Statist. Sinica* **21** 71–105. MR2796854

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

LOPES, D. (2011). Development and implementation of Bayesian computer model emulators. Ph.D. thesis, Duke Univ.

MARREL, A., IOOSS, B., JULLIEN, M., LAURENT, B. and VOLKOVA, E. (2011). Global sensitivity analysis for models with spatially dependent outputs. *Environmetrics* **22** 383–397. MR2843392

PATRA, A. K., BAUER, A. C., NICHITA, C. C., PITMAN, E. B., SHERIDAN, M. F., BURSIK, M., RUPP, B., WEBBER, A., STINTON, A. J., NAMIKAWA, L. M. et al. (2005). Parallel adaptive numerical simulation of dry avalanches over natural terrain. *J. Volcanol. Geotherm. Res.* **139** 1–21.

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

PAULO, R., GARCÍA-DONATO, G. and PALOMO, J. (2012). Calibration of computer models with multivariate output. *Comput. Statist. Data Anal.* **56** 3959–3974. MR2957846

PITMAN, E. B., NICHITA, C. C., PATRA, A., BAUER, A., SHERIDAN, M. and BURSIK, M. (2003). Computing granular avalanches and landslides. *Phys. Fluids* **15** 3638–3646. MR2028451

RASMUSSEN, C. E. and WILLIAMS, C. K. I. (2006). *Gaussian Processes for Machine Learning*. *Adaptive Computation and Machine Learning*. MIT Press, Cambridge, MA. MR2514435

REN, C., SUN, D. and HE, C. (2012). Objective Bayesian analysis for a spatial model with nugget effects. *J. Statist. Plann. Inference* **142** 1933–1946. MR2903403

ROUGIER, J. (2008). Efficient emulators for multivariate deterministic functions. *J. Comput. Graph. Statist.* **17** 827–843. MR2649069

ROUGIER, J., GUILLAS, S., MAUTE, A. and RICHMOND, A. D. (2009). Expert knowledge and multivariate emulation: The thermosphere-ionosphere electrodynamics general circulation model (TIE-GCM). *Technometrics* **51** 414–424. MR2756477

ROUSTANT, O., GINSBOURGER, D. and DEVILLE, Y. (2012). DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. *J. Stat. Softw.* **51** 1–55.

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

SAVITSKY, T., VANNUCCI, M. and SHA, N. (2011). Variable selection for nonparametric Gaussian process priors: Models and computational strategies. *Statist. Sci.* **26** 130–149. MR2849913

SEVERINI, T. A. (2000). *Likelihood Methods in Statistics*. *Oxford Statistical Science Series* **22**. Oxford Univ. Press, Oxford. MR1854870

SPILLER, E. T., BAYARRI, M. J., BERGER, J. O., CALDER, E. S., PATRA, A. K., PITMAN, E. B. and WOLPERT, R. L. (2014). Automating emulator construction for geophysical hazard maps. *SIAM/ASA J. Uncertain. Quantificat.* **2** 126–152. MR3283903

VARIN, C., REID, N. and FIRTH, D. (2011). An overview of composite likelihood methods. *Statist. Sinica* **21** 5–42. MR2796852

XIAO, M., BREITKOPF, P., FILOMENO COELHO, R., KNOPF-LENOIR, C., SIDORKIEWICZ, M. and VILLON, P. (2010). Model reduction by CPOD and Kriging: Application to the shape optimization of an intake port. *Struct. Multidiscip. Optim.* **41** 555–574. MR2601473

ZHANG, H. (2004). Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. *J. Amer. Statist. Assoc.* **99** 250–261. MR2054303

M. GU
J. O. BERGER
DEPARTMENT OF STATISTICAL SCIENCE
DUKE UNIVERSITY
P.O. BOX 90251
DURHAM, NORTH CAROLINA 27708-0251
USA
E-MAIL: mg211@stat.duke.edu
berger@stat.duke.edu