



Gaussian process emulation of computer models: Models with massive output

Mengyang Gu and Jim Berger

Johns Hopkins University and Duke University

in collaboration with

**Susie Bayarri, Eliza Calder, Abani Patra, Bruce Pitman,
Elaine Spiller, and Robert Wolpert**

(U Valencia, U Edinburgh, U Buffalo, U Buffalo, Marquette U, Duke U)

Statistics Conference

in honor of Aad van der Vaart's 60th birthday

Leiden

June 20, 2019



(Some of) Aad's Work on Gaussian Processes

- van Zanten, J.H. and van der Vaart, A.W., (2006). Regularization by Gaussian process priors. Qualitative Assumptions and Regularization in High-dimensional Statistics, 41–43, (eds: Lutz Dumbgen, Jon A. Wellner). Mathematisches Forschungsinstitut Oberwolfach.
- van der Vaart, A.W. and van Zanten, J.H., (2007). Bayesian inference with rescaled Gaussian process priors. *Electronic Journal of Statistics* **1**, 433–448.
- van der Vaart, A.W. and van Zanten, J.H., (2008). Rates of contraction of posterior distributions based on Gaussian process priors. *Annals of Statistics* **36**, 1435–1463.
- van der Vaart, A.W. and van Zanten, J.H., (2008). Reproducing kernel Hilbert spaces of Gaussian priors. In *Pushing the limits of contemporary statistics: contributions in honor of Jayanta K. Ghosh*. Institute Mathematical Statistics Collect **3**, 200–222.



- van der Vaart, A.W. and van Zanten, J.H., (2009). Adaptive Bayesian estimation using a Gaussian random field with inverse Gamma bandwidth. *Annals of Statistics* **37**, 2655–2675.
- van der Vaart, A.W., (2010). Bayesian Regularization. In *Proceedings of the International Congress of Mathematicians*, Hyderabad, India (eds: Rajendra Bhatia, Arup Pal, G Rangarajan, V Srinivas, M Vanninathan). WorldScientific.
- Knapik, B., van der Vaart, A.W. and van Zanten, J.H., (2011). Bayesian inverse problems with Gaussian priors. *Annals of Statistics* **39**, 2626–2657.
- van der Vaart, A.W. and van Zanten, J.H., (2011). Information rates of nonparametric Gaussian process methods. *Journal Machine Learning* **12**, 2095–2119.
- Sniekers, S. and van der Vaart, A.W., (2015). Adaptive Bayesian credible sets in regression with a Gaussian process prior. *Electronic Journal of Statistics* **9**, 2475–2527.



Three Definitions:

Simulator: A computer model constructed to simulate some real process, e.g., a climate model is constructed to try to simulate future climate.

Emulator: A statistical model constructed to approximate a simulator (usually because the simulator takes too long – many hours or many days – to run).

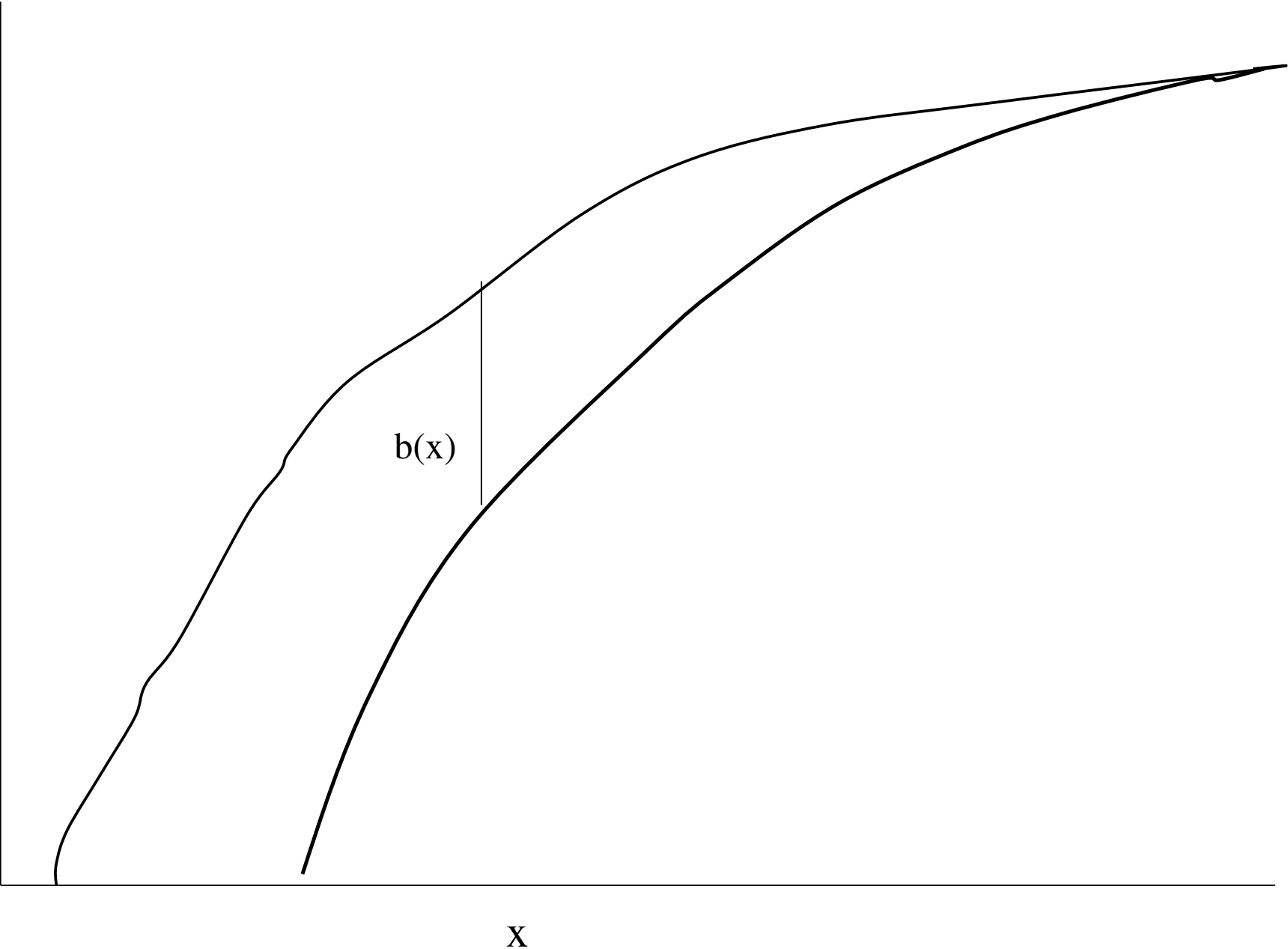
Uncertainty Quantification (UQ): The process of relating and combining simulators with data and uncertainty.



SIAM/ASA Journal on
**UNCERTAINTY
QUANTIFICATION**

siam®
Society for Industrial and
Applied Mathematics

ASA
AMERICAN STATISTICAL
ASSOCIATION





I. Functional emulators for massive output via Parallel Partial (PP) emulation (Gu et al. (2016))



TITAN2D is a computer model (based on PDE's) that simulates pyroclastic flow on Montserrat (Bayarri et al. (2009); Dalbey et al. (2012))



TITAN2D Inputs: *Initial conditions:* x_1 = flow volume (V); x_2 = flow direction (φ); *Model parameters:* x_3 = basal friction (δ_{bed}); x_4 = internal friction (δ_{int}).

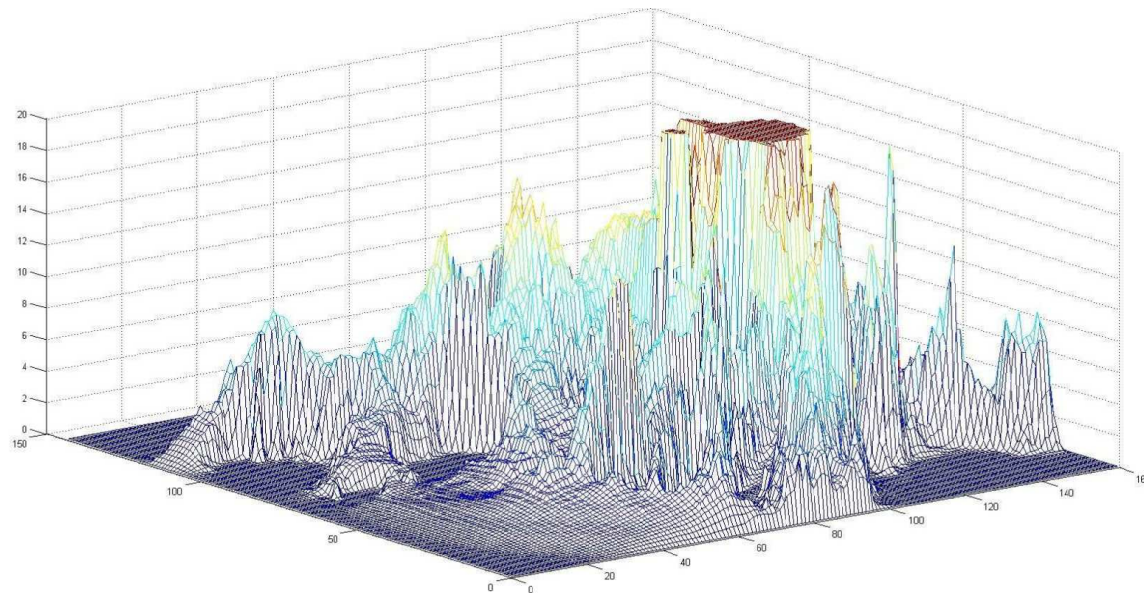


Background of the application: The simulator, TITAN2D, for given inputs V , φ , δ_{bed} , and δ_{int} , yields a description of the pyroclastic flow over a large space-time grid of points. Each run of TITAN2D takes two hours.

Of primary interest is the **maximum (over time) flow height**, $y^M(V, \varphi, \delta_{bed}, \delta_{int})$, at k spatial locations over the island.

- Flow heights $y^M(V, \varphi, \delta_{bed}, \delta_{int}) > 1\text{m}$ are deemed to be catastrophic.

The figure below gives maximum flow height at 24,000 spatial locations for inputs $V = 10^{7.462}$, $\varphi = 2.827$, $\delta_{bed} = 11.111$, and $\delta_{int} = 27.7373$





The analysis begins

- by choosing a Latin hypercube design to select $m = 2048$ (or $m = 50$) design points in the feasible input region

$$\mathbf{X} = [10^5 \text{m}^3, 10^{9.5} \text{m}^3] \times [0, 2\pi] \times [5, 18] \times [15, 35];$$

- running TITAN2D at these preliminary points, yielding the ‘data’

$$\mathbf{y}^D = \begin{pmatrix} \mathbf{y}^M(\mathbf{x}_1) \\ \mathbf{y}^M(\mathbf{x}_2) \\ \vdots \\ \mathbf{y}^M(\mathbf{x}_m) \end{pmatrix} = \begin{pmatrix} (y_1^M(\mathbf{x}_1), y_2^M(\mathbf{x}_1), \dots, y_k^M(\mathbf{x}_1)) \\ (y_1^M(\mathbf{x}_2), y_2^M(\mathbf{x}_2), \dots, y_k^M(\mathbf{x}_2)) \\ \vdots \\ (y_1^M(\mathbf{x}_m), y_2^M(\mathbf{x}_m), \dots, y_k^M(\mathbf{x}_m)) \end{pmatrix};$$

- constructing the emulator (approximation) to TITAN2D from $\mathbf{y}^D$ (in general, a matrix of size 2048×10^9).



GaSP Emulators

Model the *real-valued for now* simulator output $y^M(\mathbf{x})$ as an *unknown* function via a *Gaussian stochastic process*:

$$y^M(\cdot) \sim \text{GaSP}(\mu(\cdot), \sigma^2 c(\cdot, \cdot)),$$

with mean function $\mu(\mathbf{x})$, variance σ^2 , and correlation function $c(\cdot, \cdot)$ if, for any inputs $\{\mathbf{x}_1, \dots, \mathbf{x}_m\}$ from $\mathbf{X}$,

$$(y^M(\mathbf{x}_1), \dots, y^M(\mathbf{x}_m)) \sim \text{MVN}((\mu(\mathbf{x}_1), \dots, \mu(\mathbf{x}_m)), \sigma^2 \mathbf{C}), \quad (1)$$

where $\mathbf{C}$ is the correlation matrix with (i, j) element $c_{i,j} = c(\mathbf{x}_i, \mathbf{x}_j)$, with the separable Matérn correlation function

$$c(\mathbf{x}, \mathbf{x}^*) = \prod_{j=1}^d \left[\left(1 + \frac{\sqrt{5} |\mathbf{x}_j - \mathbf{x}_j^*|}{\gamma_j} + \frac{5 |\mathbf{x}_j - \mathbf{x}_j^*|^2}{3\gamma_j^2} \right) \exp \left(-\frac{\sqrt{5} |\mathbf{x}_j - \mathbf{x}_j^*|}{\gamma_j} \right) \right].$$



The simplest imaginable GaSP for the k -dimensional

$$\mathbf{y}^M(\mathbf{x}) = (y_1^M(\mathbf{x}), y_2^M(\mathbf{x}), \dots, y_k^M(\mathbf{x})):$$

An *independent* GaSP is assigned to each coordinate $y_j^M(\mathbf{x})$, with

- prior mean functions of the regression form $\mu_j(\mathbf{x}) = \Psi(\mathbf{x})\boldsymbol{\theta}_j$, where $\Psi(\mathbf{x})$ is a *common* l -vector of given basis functions and the $\boldsymbol{\theta}_j$ are *differing* unknown regression coefficients;
- *differing* unknown prior variances σ_j^2 ;
- *common* estimated correlation parameters $\hat{\gamma}$.

In the analysis,

- The $\boldsymbol{\theta}_j$ and σ_j^2 are dealt with in a fully Bayesian manner using $\pi(\boldsymbol{\theta}_j, \sigma_j^2) = 1/\sigma_j^2$.
- $\hat{\gamma}$ is the marginal posterior mode arising from use of the reference prior for correlation parameters, as implemented in **RobustGaSP: Robust Gaussian Stochastic Process Emulation in R** by Gu and Palomo.
 - Note that maximum likelihood can give very poor performance here.



The mean function of the posterior GaSP for $y_j^M(\mathbf{x}^*)$ at new input $\mathbf{x}^*$ is

$$\hat{\mu}_j(\mathbf{x}^*) = \Psi(\mathbf{x}^*)\hat{\boldsymbol{\theta}}_j + C(\mathbf{x}^*)\mathbf{C}^{-1}(\mathbf{y}_j^D - \Psi\hat{\boldsymbol{\theta}}_j),$$

where $\mathbf{y}_j^D$ is the j^{th} column of $\mathbf{y}^D$ and $\hat{\boldsymbol{\theta}}_j = (\Psi'\mathbf{C}^{-1}\Psi)^{-1}\Psi'\mathbf{C}^{-1}\mathbf{y}_j^D$, with Ψ being the $m \times l$ matrix whose rows are the bases evaluated at each design point, $\mathbf{C}$ being the earlier specified $m \times m$ correlation matrix, $\Psi(\mathbf{x}^*) = (\Psi_1(\mathbf{x}^*), \dots, \Psi_l(\mathbf{x}^*))$, and $C(\mathbf{x}^*) = (c(\mathbf{x}_1, \mathbf{x}^*), \dots, c(\mathbf{x}_m, \mathbf{x}^*))$.

This can be rewritten

$$\hat{\mu}_j(\mathbf{x}^*) = \sum_{i=1}^m h_i(\mathbf{x}^*) y_{ij}^D,$$

where $h_i(\mathbf{x}^*)$ is the i^{th} element of the m -vector

$$\mathbf{h}(\mathbf{x}^*) = [(\Psi(\mathbf{x}^*) - C(\mathbf{x}^*)\mathbf{C}^{-1}\Psi)(\Psi'\mathbf{C}^{-1}\Psi)^{-1}\Psi'\mathbf{C}^{-1} + C(\mathbf{x}^*)\mathbf{C}^{-1}] .$$

As Ψ and $\mathbf{C}$ (and the functions of them) can be pre-computed, computing $\mathbf{h}(\mathbf{x}^*)$ (at a new $\mathbf{x}^*$) requires roughly m^2 numerical operations.



Finally, we can write the complete parallel partial posterior (PP) mean vector (the emulator of the full simulator output at a new input $\mathbf{x}^*$) as

$$\hat{\boldsymbol{\mu}}(\mathbf{x}^*) = (\hat{\mu}_1(\mathbf{x}^*), \dots, \hat{\mu}_k(\mathbf{x}^*)) = \mathbf{h}(\mathbf{x}^*) \mathbf{y}^D.$$

- The overall computational cost is just $O(mk)$ when $k \gg m$.
 - It is crucially important to have differing $\boldsymbol{\theta}_j$ and σ_j^2 at each coordinate, but this comes with essentially no computational cost.
 - The cost increases by nine orders of magnitude if separate emulators are developed for each $y_j^M(\cdot)$ and, surprisingly, predictive performance degrades!
- Computation of all the PP emulator variances is $O(m^2k)$, but one rarely needs to compute all of them.
- The emulator is an interpolator so, when $\mathbf{x}^*$ equals one of the runs $\mathbf{x}_i$, the emulator will return the exact values from the computer run.
- As the emulator mean is just a weighted average of the actual simulator runs, it hopefully captures some of the dynamics of the process.

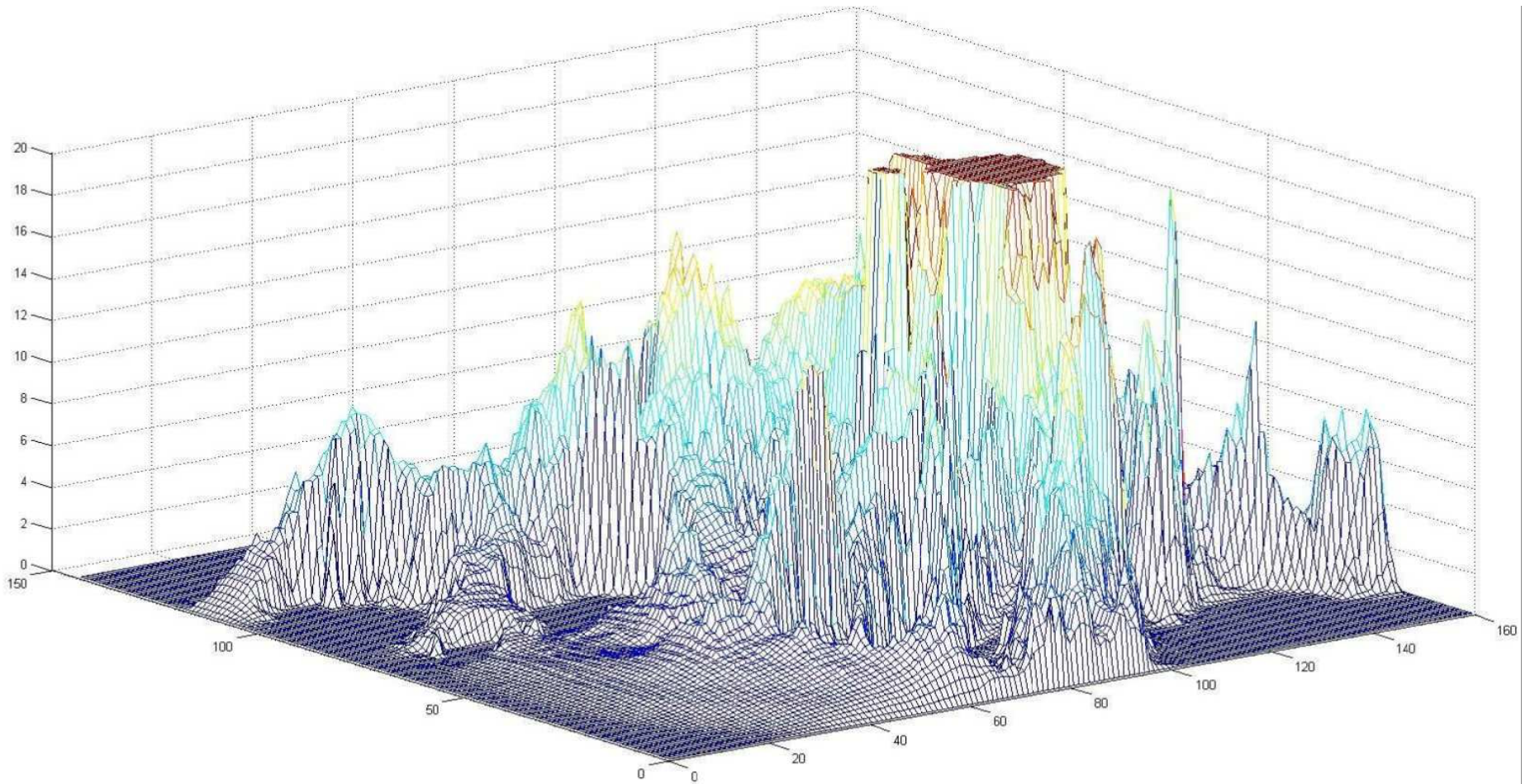


Figure 1: The mean of the emulator of ‘maximum flow height over time’ from TITAN2D, at 24,000 spatial locations over Montserrat and for new input values $V = 10^{7.462}$, $\varphi = 2.827$, $\delta_{bed} = 11.111$, and $\delta_{int} = 27.7373$.

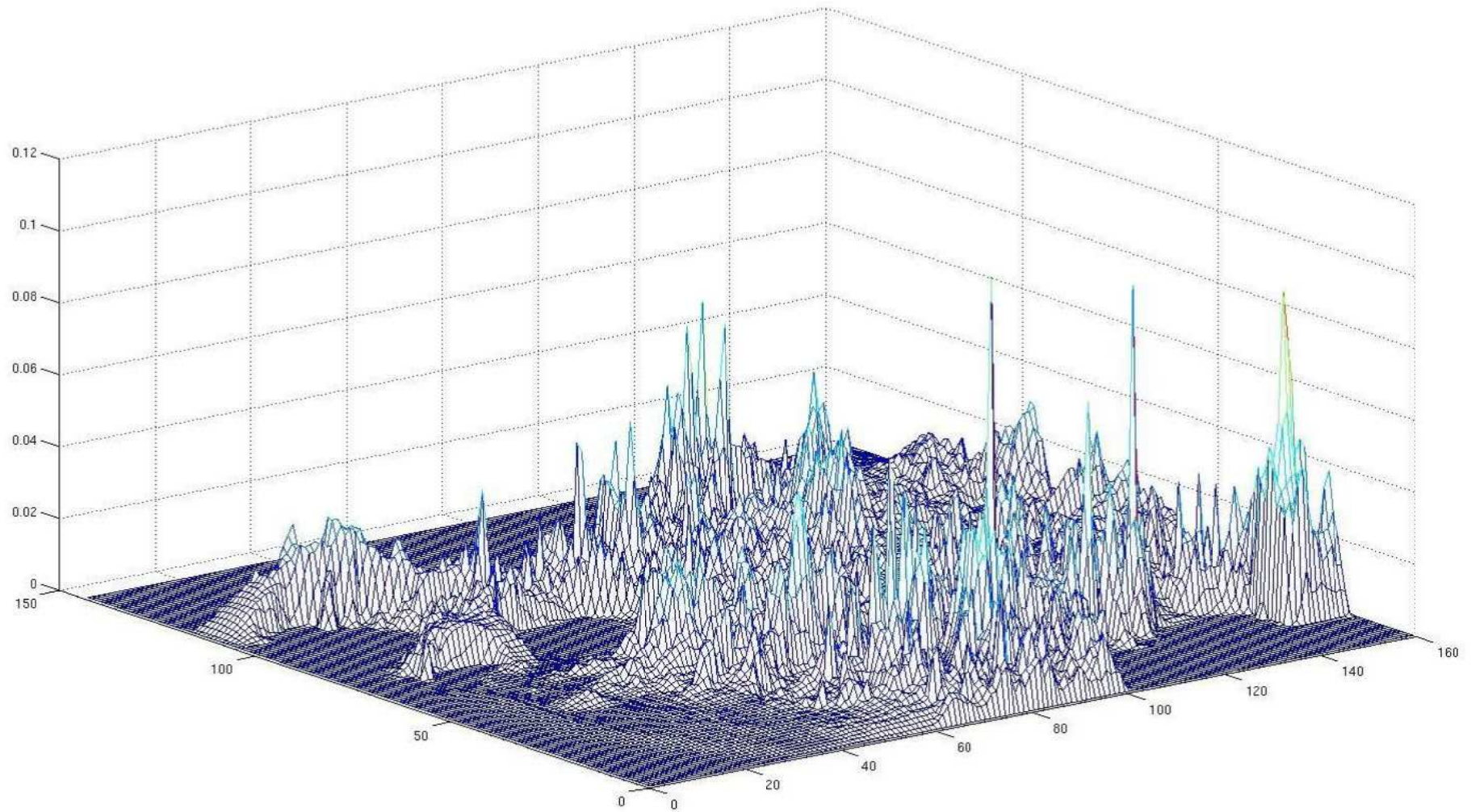


Figure 2: Variance of the emulator of ‘maximum flow height over time’ from TITAN2D, at 24,000 spatial locations over Montserrat and for new input values $V = 10^{7.462}$, $\varphi = 2.827$, $\delta_{bed} = 11.111$, and $\delta_{int} = 27.7373$.



Movie Time

Determination of 1m contours of maximum flow over time at $k = 23,040$ spatial locations, using $m = 50$ simulator runs at various inputs to develop the emulator.



The spatial ‘elephant in the room’

Is the key (and clearly invalid) assumption that simulator output values at all coordinates (e.g., space-time locations) are independent.

The usual attempted solution: Introduce a second spatial process over the output coordinates of $\mathbf{y}^M(\mathbf{x}) = (y_1^M(\mathbf{x}), y_2^M(\mathbf{x}), \dots, y_k^M(\mathbf{x}))$ to reflect the clear dependence. Usual assumptions on this process:

- It is also a Gaussian process, with correlation function $\lambda(i, j)$, leading to the $k \times k$ correlation matrix $\mathbf{\Lambda}$.
- Because k is huge, the process must be chosen so that $\mathbf{\Lambda}$ is sparse (e.g., only allow correlation with nearby points), to allow for the needed inversions of $\mathbf{\Lambda}$.
- Separability with the GaSP over the process input space is assumed, so that the covariance matrix of the joint Gaussian process is (letting $\boldsymbol{\sigma}$ denote the diagonal matrix of coordinate standard deviations)

$$\Sigma = \boldsymbol{\sigma} \mathbf{\Lambda} \boldsymbol{\sigma} \otimes \mathbf{C}, \quad \text{and thus} \quad \Sigma^{-1} = \boldsymbol{\sigma}^{-1} \mathbf{\Lambda}^{-1} \boldsymbol{\sigma}^{-1} \otimes \mathbf{C}^{-1}.$$

The problem: It is difficult to add plausible spatial structure while keeping the computation manageable when k is huge.



The Surprise: The spatial elephant can (mostly) be ignored, as the PP emulator will give essentially the same answers. Indeed, for *any* spatial structure Λ , the following can be shown:

- The emulator mean of $\mathbf{y}^M(\mathbf{x}^*) = (y_1^M(\mathbf{x}^*), \dots, y_k^M(\mathbf{x}^*))$, at a new input $\mathbf{x}^*$, is exactly the same as the PP emulator mean. (Intuition: it does not matter that the $y_i^M(\mathbf{x}^*)$ are spatially related, as they are all unknown.)
- The emulator variance at coordinate j is still $\hat{\sigma}_j^2 V_j(\mathbf{x}^*)$, with only $\hat{\sigma}_j^2$ depending on the spatial structure, and only in a minor way; thus one can just use the (slightly conservative) PP emulator variance.

The remaining little elephant: If one actually needs random draws from the emulator, the PP emulator's draws will be too rough (because each coordinate is independent), which might be harmful in some applications.

- A relatively simple fix to obtain smoother draws is to divide the grid into squares of moderate size s (e.g., $s = 4$), have the squares be independent, but allow a dependent spatial process in each square.
- If Λ in each square is assigned the objective prior $\pi(\Lambda) = |\Lambda|^{-s}$, the mean and variance of the emulator will then be the same as the PP emulator.



Introduction of a nugget

Often certain inputs have very little effect on the simulator output, and emulators that ignore that input can do better at prediction. But, for deterministic simulators, one must then introduce a ‘nugget’ (i.i.d. Gaussian errors) in the GaSP model.

We simply let the correlation matrix be $\mathbf{C} + \xi \mathbf{I}$, renormalized to be a correlation matrix, with ξ unknown. The computations are then only slightly more complicated.

Example: In TITAN2D, δ_{int} has only a minor effect, so we will investigate

- the full 4 input emulator,
- the 3 input emulator with δ_{int} removed and a nugget inserted.



Emulator:	PP GaSP	PP GaSP	MS GaSP	LMC GaSP
Parameter estimation:	robust est.	robust est.	robust est.	DiceKriging
	4 inputs	3 inputs and an estimated nugget		
Mean Square Error	0.109	0.097	0.103	0.137
95% CI Coverage	0.926	0.950	0.924	0.909
95% CI Length	0.521	0.536	0.491	0.478
time (s) using R	50.0	28.1	31337.7	3407.6

Table 1: Performance of various emulators, developed from 50 simulator runs, of max flow height over all spatial locations except the crater and non-flow areas.

- The first emulator uses all 4 inputs while the remaining three emulators use 3 inputs (V, δ_{bed}, ϕ) and a nugget, all with the same regressor $\mathbf{h}(\mathbf{x}) = (1, V)$.
- The LMC emulator uses coregionalization with SVD output decomposition.
- Evaluations based on $n^* = 633$ held-out inputs over $k = 17,311$ locations.
- The last row shows the computational times of the emulators, using R.



Thanks All



Thanks and Happy Birthday Aad





References

- Bayarri, M. J., J. O. Berger, E. S. Calder, K. Dalbey, S. Lunagomez, A. K. Patra, E. B. Pitman, E. T. Spiller, and R. L. Wolpert (2009). Using statistical and computer models to quantify volcanic hazards. *Technometrics* 51, 402–413.
- Dalbey, K., M. Jones, E. B. Pitman, E. S. Calder, M. Bursik, and A. K. Patra (2012). Hazard risk analysis using computer models of physical phenomena and surrogate statistical models. *Int. J. for Uncertainty Quantification*. To appear.
- Gu, M., J. O. Berger, et al. (2016). Parallel partial gaussian process emulation for computer models with massive output. *The Annals of Applied Statistics* 10(3), 1317–1347.