International Journal for Uncertainty Quantification, 4 (1): 63–93 (2014)

DIMENSIONALITY REDUCTION FOR COMPLEX MODELS VIA BAYESIAN COMPRESSIVE SENSING

Khachik Sargsyan,[1,*] Cosmin Safta,[1] Habib N. Najm,[1] Bert J. Debusschere,[1] Daniel Ricciuto,[2] & Peter Thornton[2]

[1] *Sandia National Laboratories, 7011 East Avenue, MS 9051, Livermore, California 94550, USA*

[2] *Oak Ridge National Laboratory, 1 Bethel Valley Road, MS 6301, Oak Ridge, Tennessee 37831, USA*

Original Manuscript Submitted: 12/18/2012; Final Draft Received: 06/24/2013

Uncertainty quantification in complex physical models is often challenged by the computational expense of these models. One often needs to operate under the assumption of sparsely available model simulations. This issue is even more critical when models include a large number of input parameters. This "curse of dimensionality," in particular, leads to a prohibitively large number of basis terms in spectral methods for uncertainty quantification, such as polynomial chaos (PC) methods. In this work, we implement a PC-based surrogate model construction that "learns" and retains only the most relevant basis terms of the PC expansion, using sparse Bayesian learning. This dramatically reduces the dimensionality of the problem, making it more amenable to further analysis such as sensitivity or calibration studies. The model of interest is the community land model with about 80 *input parameters, which also exhibits nonsmooth input-output behavior. We enhanced the methodology by a clustering and classifying procedure that leads to a piecewise-PC surrogate thereby dealing with nonlinearity. We then obtain global sensitivity information for five outputs with respect to all input parameters using less than 10,000 model simulations—a very small number for an* 80*-dimensional input parameter space.*

KEY WORDS: *uncertainty quantification, surrogate model, Bayesian inference, compressive sensing, classification, polynomial chaos, community land model*

1. INTRODUCTION

Land ecosystems influence the Earth's climate system through multiple biophysical and biogeochemical feedback mechanisms. These mechanisms operate mainly through net land surface fluxes of greenhouse gases (GHG), variation in land surface albedo, and land-atmosphere exchanges of sensible and latent heat. A major objective in the development of the community land model (CLM) is to improve prediction of future states of the climate and ecosystems by including new components and processes that contribute to these feedbacks [1]. For example, estimating the sign and magnitude of the land component of the global carbon-climate feedback depends on a mechanistic representation of carbon uptake by vegetation, turnover of live plant parts to litter, and decomposition of litter by soil heterotrophs, with return of carbon to the atmosphere by respiration and long-term storage of recalcitrant forms of organic matter in soils. These dynamics are sensitive to spatial and temporal patterns of changing temperature and precipitation, which in turn depend on changes in GHG concentration, resulting in a complex array of feedbacks.

In general, predictions from computational models, including the CLM, are affected by a number of sources of uncertainty. These include assumptions made in the modeling of physical processes, input parameter values, numerical discretization, and initial and boundary conditions. To this end, the quantification of uncertainties in a physical system involves two intimately related processes. The first relates to the forward propagation of uncertainty from input parameters to output quantities of interest (QoIs). The reverse process focuses on the estimation of input parameters

*Correspond to Khachik Sargsyan, E-mail: ksargsy@sandia.gov

based on comparing model simulations with available observational or other computational data. In this study we will tackle the first aspect, related to forward uncertainty quantification (UQ), with application to the carbon-cycle feedback dynamics processes in the CLM. In particular we will perform a global sensitivity study and focus on the challenges posed by the high dimensionality of the input parameters space and the fact that computational simulations are expensive even on today's leadership computing platforms.

Global sensitivity analysis (GSA) formally connects uncertainties in model output to the underlying uncertainties present in the model inputs. This provides a formal approach to assess model reliability in particular when used in a decision-making framework [2]. In this study we employ variance-decomposition methods which are based on the decomposition of the model output variance into fractions associated with input factors and their interactions. The effects of input parameters and their interactions are quantified though Sobol indices [3, 4]. While these indices can generally be computed through Monte Carlo (MC) sampling, in practical high performance computing settings, the slow ($1/\sqrt{N}$) convergence of MC renders these methods prohibitively expensive for reasonable accuracy. In this work we are focusing on developing response surface models to be used as efficient-to-evaluate surrogate models in place of the expensive CLM simulations.

In particular we are fitting polynomial chaos (PC) expansions to the QoIs. As the number of basis terms increases factorially fast with the number of dimensions and the total polynomial order, e.g., a second-order polynomial involves about 3000 terms while a third-order polynomial leads to about 10^5 terms in 80-dimensional input parameter space. This leads to a prohibitive number of model evaluations, up to $\sim 10^6$, necessary to compute the PC basis terms. One alternative given a limited number of model runs is to evaluate the expansion coefficients by regression. Several regularization approaches were developed to constrain the problem when the system is underdetermined, i.e., the number of coefficients is larger than the number of model evaluations.

A popular approach is the Tikhonov regularization [5], also known as ridge regression. This approach adds an l_2 norm-based constraint on the vector of unknowns, in our case the PC response surface coefficients. The Tikhonov regularization effectively reduces the magnitude of coefficients for unimportant basis terms, but they are still retained in the response surface model. Alternative regularization approaches are the least absolute shrinkage and selection operator (LASSO) [6] or least angle regression (LARS) [7, 8]. These techniques use the l_1 norm of the parameter vector and result in typically many coefficients driven to zero, leading to a sparse response surface solution if the underlying physical system exhibits sparsity. Note that the sparsity is very commonly present in physical systems.

While the l_1 regularization approach, or compressive sensing, has been recently applied to polynomial bases [9], it is useful to cast the problem in a Bayesian framework in presence of limited information. In the Bayesian formulation, the regularization terms are embedded in the prior formulation while the regression term is incorporated in the likelihood [10]. In this context, the l_1 penalty corresponds to independent Laplace priors on the expansion coefficients. Independent studies by Park and Casella [11] and Ji and Carin [12] proposed similar Bayesian approaches to the l_1 regularization approach, henceforth referred to as Bayesian compressive sensing (BCS), almost identical to the relevance vector machine (RVM) sparse learning technique described in [13], and applied recently in the multi-output context for uncertainty quantification in [14]. Furthermore, Babacan et al. [15] introduced a numerical approach for BCS based on a greedy algorithm, allowing its efficient application in practical image processing problems. This paper builds on this methodology and further introduces an iterative approach for selecting higher-order PC terms, making the combined approach feasible for large dimensional configurations.

A striking feature of the CLM is the fact that simulations can be broadly categorized into two groups. For some input parameters, the modeled vegetation dies off in a long term, forcing some outputs of interest to vanish. This leads to two quantitatively different regimes for the forward function that describes the relationship between input parameters and output quantities of interest. It is well-recognized [16] that polynomial-based orthogonal expansions have poor accuracy for representing such nonsmooth behavior. A number of studies addressed this issue in PC-based UQ by employing domain decomposition [17–20] and/or basis enrichment methods [21–24]. Brute-force domain decomposition methods suffer from the curse of dimensionality as they split the domains parallel to axes, dimension-by-dimension. While smarter domain decomposition methods try to "learn" the high-gradient regions and split the input space accordingly, these methods, as well as basis enrichment methods require judicious, *ad hoc* analysis that is computationally intractable in higher dimensions. In such settings, we will employ data-driven decomposition, relying on techniques borrowed from the machine learning community. Classification algorithms, e.g., the random

decision forest (RDF) approach [25], are trained on existing sets of samples, partitioned such that in each set, the QoIs exhibit smooth behavior. The iterative BCS algorithm is used to find PC-based response surfaces corresponding to each partition. The QoIs at new sample points are obtained by evaluating the most appropriate response surface model according to the classification algorithm.

This paper is organized as follows. In Section 2 we introduce PC surrogates in general. Section 3 focuses on the BSC methodology for building sparse PC surrogates, and presents several numerical tests to illustrate the performance of the algorithm. The iterative procedure for selecting basis terms in the setting of high-dimensional problems is proposed and discussed in Section 4. Next, in Section 5, we introduce and discuss piecewise-PC expansions as a tool to handle strongly nonlinear models. The application of the techniques to the CLM is demonstrated in Section 6. Finally, we conclude and discuss the results in Section 7.

2. POLYNOMIAL CHAOS SURROGATE

Consider a *forward* model $z = f(\boldsymbol{\lambda})$, where $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_{\tilde{d}})$ is a $\tilde{d}$-dimensional input parameter vector and z is a scalar output. PC spectral expansions [26, 27] view both input parameters and the output of interest as random variables and expand them as a series of orthogonal polynomials $\Psi_k(\boldsymbol{\xi})$ of standard, i.i.d. random variables $(\xi_1, \xi_2, \ldots, \xi_d) = \boldsymbol{\xi}$. This is essentially a parameterization of all inputs

$$\lambda_i \simeq \sum_{k=0}^{K_{\text{in}}-1} \lambda_{i,k} \Psi_k(\boldsymbol{\xi}) \tag{1}$$

and the output

$$z \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\xi}) \tag{2}$$

by sets of *deterministic* numbers $\lambda_{i,k}$ and c_k for $i = 1, 2, \ldots, \tilde{d}$ and $k = 0, 1, \ldots, K_{(\text{in})} - 1$. Note that, in general, the number of input parameters $\tilde{d}$ does not need to coincide with the *stochastic dimension* d. The number of basis terms in the input and output PC expansions are K_{in} and K, respectively. For forward uncertainty propagation and response surface construction, one typically fixes K_{in} initially, while K takes values depending on the output representation accuracy requirements and truncation rules.

In Eqs. (1) and (2) the polynomials $\Psi_k(\cdot)$ are taken by construction to be *orthonormal* with respect to the probability density function (PDF) $\pi(\boldsymbol{\xi})$ of the underlying variable $\boldsymbol{\xi}$:

$$\langle \Psi_i(\boldsymbol{\xi}) \Psi_j(\boldsymbol{\xi}) \rangle = \int \Psi_i(\boldsymbol{\xi}) \Psi_j(\boldsymbol{\xi}) \pi(\boldsymbol{\xi}) d\boldsymbol{\xi} = \delta_{i,j}. \tag{3}$$

Each index k of the multivariate polynomial $\Psi_k(\boldsymbol{\xi})$ corresponds to a vector *multi-index* $\boldsymbol{p}(k) = (p_1^{(k)}, p_2^{(k)}, \ldots, p_d^{(k)})$. Further in the text, for clarity of presentation, we will drop the explicit dependence notation in $\boldsymbol{p}(k)$ or $p_i^{(k)}$, and simply use multi-index $\boldsymbol{p}$ or scalar index k, depending on the context. The multivariate polynomial $\Psi_{\boldsymbol{p}}(\cdot)$ is a function of d variables $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_d)$ defined by

$$\Psi_{\boldsymbol{p}}(\boldsymbol{\xi}) = \psi_{p_1}(\xi_1) \psi_{p_2}(\xi_2) \cdots \psi_{p_d}(\xi_d), \tag{4}$$

where $\psi_{p_i}(\xi)$ is the standard one-dimensional orthonormal polynomial of degree p_i, for $i = 1, 2, \ldots, d$. By convention, the sum of all degrees $p_1 + p_2 + \cdots + p_d$ is called the order of the multivariate polynomial $\Psi_{\boldsymbol{p}}(\boldsymbol{\xi})$ or $\Psi_k(\boldsymbol{\xi})$ if one enumerates the polynomials by a single scalar index.

Two of the most commonly used PC expansions for continuous random variables are Gauss-Hermite (Hermite polynomials as functions of standard normal random variables) and Legendre-uniform (Legendre polynomials as functions of uniform random variables) PC expansions [28]. While the major conclusions of this work are independent of the basis choice, we will focus on Legendre-uniform (LU) PC as the more appropriate choice when one deals

with input parameters of bounded support. Note that while the conventional PC expansions use Hermite or Legendre polynomials *without* normalization, we will use the normalized polynomial bases for convenience and to avoid biasing the basis selection in Section 3.

The first five *univariate* Legendre polynomials are

$$\begin{aligned} L_0(\xi) &= 1 \\ L_1(\xi) &= \xi \\ L_2(\xi) &= \frac{1}{2}\left(3\xi^2 - 1\right) \\ L_3(\xi) &= \frac{1}{2}\left(5\xi^3 - 3\xi\right) \\ L_4(\xi) &= \frac{1}{8}\left(35\xi^4 - 30\xi^2 + 3\right), \end{aligned} \tag{5}$$

satisfying the recurrence relation [29] for all $m = 1, 2, \ldots$

$$L_{m+1}(\xi) = \frac{2m+1}{m+1}\xi L_m(\xi) - \frac{m}{m+1}L_{m-1}(\xi). \tag{6}$$

Since the L_2 norm of a Legendre polynomial $L_m(\xi)$ with respect to the density of ξ $\pi(\xi) = 1/2$ is $||L_m(\xi)|| = \sqrt{1/(2m+1)}$, the normalized, univariate Legendre basis functions are

$$\psi_m(\xi) = \sqrt{2m+1}L_m(\xi). \tag{7}$$

As long as the input PC expansion (PCE) is known, the output PCE can serve as a surrogate model for the function $z = f(\boldsymbol{\lambda})$ that is computationally cheap and allows simple extraction of the statistical properties of z. Since the goal is to obtain a surrogate model that is *uniformly* accurate across a given input range, we employ linear PCEs for the input parameters, effectively assuming that the input λ_i is a linear transform away from the underlying uniform random variable ξ_i, i.e. $\lambda_i = \lambda_{i,0} + \lambda_{i,1}\xi_i$. The goal is to build an output PC expansion (2) given a set of input-output pairs $\{(\boldsymbol{\lambda}_i, z(\boldsymbol{\lambda}_i))\}_{i=1}^N$ as results of N model simulations, called *training* runs. Note that, since the relationship between physical parameter vector $\boldsymbol{\lambda}$ and the PC input $\boldsymbol{\xi}$ is known, our task is essentially to construct a polynomial regression to obtain PC coefficients c_k in (2) given an input-output data set $\mathcal{D} = \{(\boldsymbol{\xi}_i, z_i)\}_{i=1}^N$. Such a model can then serve as a surrogate for the model $f(\cdot)$ and be queried instead of it in studies that require prohibitively many forward model simulations, such as global sensitivity analysis, optimization, or calibration studies.

In PCEs (1) and (2) a finite number of basis terms are used, according to some predefined truncation or basis selection rule. Typical *isotropic*—i.e., all dimensions are treated equally—truncation rules are listed below.

- *Total degree (TD)*: the classical and most frequently used rule truncates the PCE according to the total degree of basis polynomials,

$$\sum_{i=1}^{d} p_i \leq p, \tag{8}$$

  i.e., all polynomials of total degree $\leq p$ are retained.

- *Tensor product (TP)*: this rule bounds the polynomial degree per dimension, i.e., all multivariate polynomials with univariate components of degree $\leq p$ are retained,

$$\max_{i=1,\ldots,d} p_i \leq p. \tag{9}$$

- *Hyperbolic cross (HC)*: the product structure of the rule encourages lower-rank, higher-degree structure of the corresponding multi-index,

$$\prod_{i=1}^{d}(p_i + 1) \leq (p+1). \tag{10}$$

- l_q *truncation (LQ)*: this rule is based on the l_q norm of the degree vector,

$$\sum_{i=1}^{d} p_i^q \leq p^q. \tag{11}$$

  Note that the TD truncation rule is a special case when $q = 1$. The typical setting is $0 < q < 1$, which promotes low-rank interactions similar to the HC rule.

In all the truncation rules above, we will refer to p as generalized degree. Note that one can build the truncation rules to employ *anisotropy* with respect to dimensions. For example, one can generalize the TD rule to $\sum_{i=1}^{d} w_i p_i \leq p$ for $0 \leq w_i \leq 1$, where smaller weights correspond to more important dimensions by allowing for larger degrees in these dimensions.

Typically, physical models are parameterized in such a way that only low-rank interactions matter, i.e., the inputs tend to affect the outputs mainly individually (rank 1) or pairwise (rank 2). This is the basis of the high dimensional model representation (HDMR), also known as analysis of variance (ANOVA) decomposition [30]. The HDMR truncation of PC promotes low-rank interaction terms explicitly and allows more flexibility. One first retains univariate terms up to a certain generalized degree $p^{(1)}$, followed by bivariate terms up to a generalized degree $p^{(2)}$ and so on. Figure 1 illustrates various truncation options for a two-dimensional basis.

The truncation options above are chosen as a preprocessing step. However, often one needs a more customized basis selection that best explains the dataset at hand. Moreover, in a high-dimensional setting, the number of basis terms retained by classical truncation rules could be prohibitively large, necessitating adaptive, iterative strategies. In

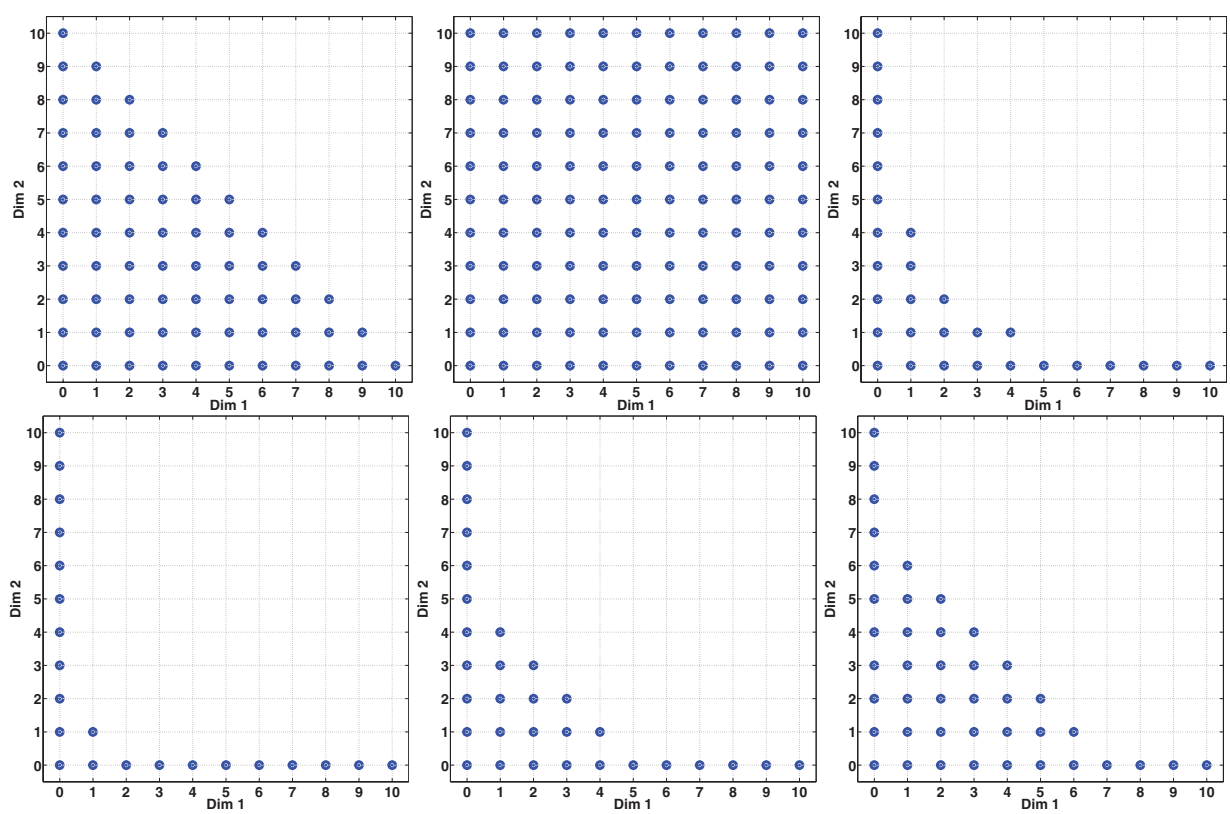


FIG. 1: Illustration of various isotropic truncation options with generalized degree $p = 10$. A dot (i, j) corresponds to a basis term $\psi_i(\xi_1)\psi_j(\xi_2)$. Top row, left to right: TD, TP, HC. Bottom row, left to right: LQ with $q = 1/3$, LQ with $q = 0.5$, and HDMR with TD truncation of each lower-rank subset with degrees $p^{(1)} = 10$ and $p^{(2)} = 7$.

the following, we essentially recast the truncation problem as a basis selection problem and apply techniques inspired by sparse signal processing, as well as statistical learning, to polynomial bases.

3. BAYESIAN COMPRESSIVE SENSING FOR POLYNOMIAL REGRESSION

Bayesian methods are well suited to deal with incomplete, sparse information [31]. Typically, the outcome of a Bayesian approach consists of a posterior probability distribution, describing our knowledge of the quantities under study. Bayes' formula, in the context of inferring a PC expansion for the quantities of interest, based on available data $\mathcal{D}$, can be written as[1]

$$q(\boldsymbol{c}) \propto L_{\mathcal{D}}(\boldsymbol{c})p(\boldsymbol{c}). \tag{12}$$

Here the likelihood $L_{\mathcal{D}}(\boldsymbol{c})$ is a measure of a goodness-of-fit of the polynomial representation

$$z \simeq z_{\boldsymbol{c}}(\boldsymbol{\xi}) = \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\xi}) \tag{13}$$

to the data. We will assume a Gaussian noise model with standard deviation σ to write

$$L_{\mathcal{D}}(\boldsymbol{c}) = (2\pi\sigma^2)^{-N/2} \exp\left(-\sum_{i=1}^{N} \frac{(z_i - z_{\boldsymbol{c}}(\boldsymbol{\xi}_i))^2}{2\sigma^2}\right). \tag{14}$$

This Gaussian assumption is merely a modeling choice. It corresponds to the statistical discrepancy between the complex model and the surrogate model. One should expect this discrepancy to have a vanishing mean, while the size σ of the standard deviation of this discrepancy represents a tolerance to the fitting error. With these restrictions, i.e., given a mean and a standard deviation, the best assumption is Gaussian in the sense of the maximum entropy principle [32]. To some extent, this assumption is similar to classical Gaussian error assumptions in least-square regression. As for the correlations, one expects the discrepancy errors between the full and surrogate models to be uncorrelated in the presence of only sparse data—the average distance between data points is expected to be larger than the correlation length in the discrepancy.

The prior distribution $p(\boldsymbol{c})$ incorporates prior information on the object of inference, i.e., the PC mode vector $\boldsymbol{c}$. The posterior distribution $q(\boldsymbol{c})$ is the main outcome of the inference process, and it corresponds to the current knowledge about the inferred values of $\boldsymbol{c}$ *given* the data set $\mathcal{D}$.

While in principle the Bayesian procedure outlined above could be used to determine the full vector of coefficients $\boldsymbol{c}$ of all basis functions, this is in practice not always feasible. If the output z depends on many parameters, then $\boldsymbol{\xi}$ and the PCE (13) will be high dimensional, and if the forward model is computationally expensive to evaluate, then the number of samples required to determine all the terms in this expansion would be prohibitively large. Moreover, whether in the case of sparse information or even in abundance of data, some input parameters might be *unidentifiable*, i.e., the available information does not inform on these parameters. Our task of finding the best possible PC representation given the available data will, in particular, automatically detect and exclude the unidentifiable inputs from the surrogate representation. To this goal, we rely on BCS [12, 15] to determine a sparse set of basis functions that is best supported by the data, as outlined below.

The key in inferring a sparse set of PC modes is the usage of *sparsity* priors that "encourage" the modes to have nearly vanishing values, unless there is strong support in the data for those PC modes. This leads to a sparse set of basis functions. A common sparsity prior is the Laplace prior of a form

$$p(\boldsymbol{c}) = (\alpha/2)^{K+1} \exp\left(-\alpha \sum_{k=0}^{K} |c_k|\right). \tag{15}$$

[1]Note that the data set $\mathcal{D}$ actually is a set of training simulation results of the model $f(\cdot)$

In this case the *maximum a posteriori* (MAP) estimate of the object of inference $\boldsymbol{c}$, i.e., the vector $\boldsymbol{c}$ that maximizes the posterior $q(\boldsymbol{c})$ from (12), coincides with the solution of the optimization problem

$$\arg\max_{\boldsymbol{c}} \left(\log L_{\mathcal{D}}(\boldsymbol{c}) - \alpha||\boldsymbol{c}||_1\right). \tag{16}$$

Clearly, the prior distribution corresponds to the l_1 regularization term $\alpha||\boldsymbol{c}||_1$. The optimization problem (16) corresponds to the classical compressive sensing algorithm that is extensively used in the signal processing community [33]. The positive parameter α controls the relative importance of the penalty with respect to the goodness-of-fit. It is typically fixed at a user-defined value. In a hierarchical Bayesian setting, however, it can be endowed with a prior distribution and marginalized over in the posterior distribution.

Following [15], we use a hierarchical construction with Gaussian prior distribution on the PC coefficients $\boldsymbol{c}$

$$p(c_k|s_k^2) = \frac{1}{\sqrt{2\pi s_k^2}} \exp\left(-\frac{c_k^2}{2s_k^2}\right) \tag{17}$$

and gamma prior distribution on the prior widths s_k^2

$$p(s_k^2|\alpha^2) = \frac{\alpha^2}{2} \exp\left(-\frac{\alpha^2 s_k^2}{2}\right). \tag{18}$$

The regression technique with the hierarchical prior construction (17) is commonly referred to as relevance vector machine (RVM) [13], although the typical RVM usage is associated with localized, kernel basis functions instead of polynomial bases. For convenience, we will use a formal notation $\boldsymbol{s}^2$ for the vector of prior variances. Marginalizing over the hyperparameters $\boldsymbol{s}^2$, one obtains the Laplace prior (15). Indeed,

$$p(\boldsymbol{c}|\alpha^2) = \int_0^\infty \prod_{k=0}^{K-1} p(c_k|s_k^2)p(s_k^2|\alpha^2)ds_k^2 = \prod_{k=0}^{K-1} \frac{\alpha}{2} e^{-\alpha|c_k|}. \tag{19}$$

This combination of a Gaussian and gamma distributions leading to a Laplace distribution has been implemented in the Bayesian LASSO methodology, which eventually can be cast into the optimization form (16) as well [11, 34]. The parameter α can further be modeled hierarchically. The major simplifying assumption at this stage is in the fact that hyperparameter values are not being marginalized over. Instead they are fixed—in practice, updated—according to an evidence maximization procedure, effectively assuming very narrow priors for them. For details of the procedure, see [13, 15, 35]. The hyperparameters σ^2, $\boldsymbol{s}^2$, and α are fixed at the values that maximize the *evidence* or the integrated likelihood

$$\begin{aligned} E(\sigma^2, \boldsymbol{s}^2, \alpha) &= \int_{\mathbb{R}^K} L_{\mathcal{D}}(\boldsymbol{c}; \sigma^2) p(\boldsymbol{c}|\boldsymbol{s}^2) p(\boldsymbol{s}^2|\alpha) p(\alpha) p(\sigma^2) d\boldsymbol{c} \\ &\propto \quad p(\sigma^2)p(\alpha)p(\boldsymbol{s}^2|\alpha)\sigma^{-1}|\boldsymbol{C}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^2}\boldsymbol{y}^T\boldsymbol{C}^{-1}\boldsymbol{y}\right), \end{aligned} \tag{20}$$

where $\boldsymbol{C} = \boldsymbol{I} + \boldsymbol{\Psi}\boldsymbol{S}^{-1}\boldsymbol{\Psi}^T$. Here $\boldsymbol{\Psi}$ is an $N \times K$ *projection* matrix with entries $\boldsymbol{\Psi}_{ik} = \Psi_k(\boldsymbol{\xi}_i)$ and $\boldsymbol{S} = \mathrm{diag}(\sigma^2/s_0^2, \ldots, \sigma^2/s_{K-1}^2)$.

The evidence maximization is in practice with respect to the inverse variances $1/s_k^2$ instead of s_k^2. It turns out that, for many basis terms, the inverse variance that maximizes the evidence grows indefinitely, i.e., $s_k^2 \to 0$. These are the terms that will be purged from the basis set. In practice, however, a much faster heuristic optimization algorithm is used that has weaker memory requirements and deletes and adds basis functions iteratively using analytic manipulations rather than numerical ones; see [15, 35]. The algorithm involves a stopping criterion parameter ϵ, which is compared to the total relative change in the maximal value of the evidence, $(M^{(n)} - M^{(n-1)})/(M^{(n)} - M^{(1)}) < \epsilon$, where $M^{(n)}$ is the approximate value for the maximum of the evidence at the nth iteration. Smaller values of ϵ make the iterative

procedure progress further, leading to fewer retained basis terms. In all the subsequent tests, unless noted otherwise, a value of $\epsilon = 10^{-9}$ has been used.

Let us denote by K' the number of retained basis functions, and reindex them using the same subscript k, for $k = 0, 1, \ldots, K' - 1$. Together with the likelihood (14) and the prior (17), for fixed values of the hyperparameters s_k^2, σ^2, one obtains a Gaussian posterior distribution for $\boldsymbol{c}$ with mean and variance, respectively [12, 15],

$$\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Psi}^T\boldsymbol{y} \qquad \text{and} \qquad \boldsymbol{\Sigma} = \sigma^2(\boldsymbol{\Psi}^T\boldsymbol{\Psi} + \boldsymbol{S})^{-1}, \tag{21}$$

where $\boldsymbol{\Psi}$ is an $N \times K'$ projection matrix with entries $\boldsymbol{\Psi}_{ik} = \Psi_k(\boldsymbol{\xi}_i)$ and $\boldsymbol{S} = \text{diag}(\sigma^2/s_0^2, \ldots, \sigma^2/s_{K'-1}^2)$. Indeed, note that, in matrix forms, and focusing on the dependency on $\boldsymbol{c}$ only,

$$\begin{aligned} q(\boldsymbol{c}) \propto L_{\mathrm{D}}(\boldsymbol{c})p(\boldsymbol{c}) \quad &\propto \quad \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{\Psi}\boldsymbol{c})^T(\boldsymbol{y} - \boldsymbol{\Psi}\boldsymbol{c})\right)\exp\left(-\frac{1}{2\sigma^2}\boldsymbol{S}\boldsymbol{c}\right) \propto \\ &\propto \quad \exp\left(-\frac{1}{2\sigma^2}\boldsymbol{c}^T(\boldsymbol{\Psi}^T\boldsymbol{\Psi} + \boldsymbol{S})\boldsymbol{c} + \frac{1}{\sigma^2}\boldsymbol{c}^T\boldsymbol{\Psi}^T\boldsymbol{y}\right) \propto \\ &\propto \quad \exp\left(-\frac{1}{2}(\boldsymbol{c} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{c} - \boldsymbol{\mu})\right). \end{aligned} \tag{22}$$

The final surrogate model is therefore a Gaussian process with mean

$$m(\boldsymbol{\xi}) = \sum_{k=0}^{K'-1} \mu_k \Psi_k(\boldsymbol{\xi}) \tag{23}$$

and covariance function

$$C(\boldsymbol{\xi}, \boldsymbol{\xi}') = \sum_{i,j=0}^{K'-1} \Psi_i(\boldsymbol{\xi})\Sigma_{ij}\Psi_j(\boldsymbol{\xi}'). \tag{24}$$

3.1 Numerical Tests

In this section, we illustrate the application of the BCS algorithm on analytically tractable forward functions with known sparsity structures. In all tests of this section, the input parameter vector $\boldsymbol{\lambda}$ is identified with $\boldsymbol{\xi}$, to avoid an extra layer arising from the input PC (1). In principle, one could associate this setting with simple linear input PCEs $\lambda_i = \xi_i$. This simplification for the test problems does not diminish the generality of the methods presented.

For these synthetic test cases, we will consider two error measures, the goodness-of-fit of the resulting representation at the $N = N_t$ training points and at N_v *validation* points. Both the training and validation points are selected randomly, according to a uniform, Latin hypercube sampling (LHS) scheme. We rely on a relative L_2 validation error

$$E_v = \sqrt{\frac{\sum_{i=1}^{N_v}\left(z(\boldsymbol{\xi}_i) - z_{\boldsymbol{\mu}}(\boldsymbol{\xi}_i)\right)^2}{\sum_{i=1}^{N_v} z(\boldsymbol{\xi}_i)^2}}, \tag{25}$$

where the posterior mean PC mode vector $\boldsymbol{c} = \boldsymbol{\mu}$ and $\boldsymbol{\xi}_i$'s are the validation points for $i = 1, 2, \ldots, N_v$. The error at the training points $E_t(\boldsymbol{c})$ is defined similarly and will be used as a second error measure. Note that while our construction leads to an *uncertain* response surface, since the polynomial representation coefficients $\boldsymbol{c}$ are associated with a posterior probability distribution, we report the error based on the mean estimate. However, although the BCS algorithm uses evidence maximization, it is still prone to overfitting, if the stopping criterion ϵ in the heuristic optimization approach is too loose. In the tests below, overfitting will be detected by comparing the validation error E_v from (25) to the training error E_t. If these two errors are close to each other, we conclude that the corresponding PC surrogate is robust; otherwise it is overfitting, i.e., it is too complex and has too many degrees of freedom for the particular data at hand.

First, consider the following, d-dimensional, kth-order polynomial function

$$f(\boldsymbol{\xi}) = \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\xi}), \tag{26}$$

where only few coefficients are nonzero. Namely, we randomly select s terms from the total number $K = (d + k)!/(d!k!)$ and set the corresponding coefficients to be equal to 1, while the rest of the polynomial coefficients are set to 0. The number of nonzero coefficients s is essentially the *sparsity* of the model. For each model, we sample N evaluations and use those as data for a BCS run to recover the nonzero coefficients. A single BCS run is considered a *success* if only the s nonvanishing terms are picked. Successful runs lead to exact recovery of the nonzero terms with remaining relative L_2 errors purely due to the fact that one has a finite number of measurements.

Figure 2 illustrates the results of three different tests, $s = 20$ and $s = 50$ with a fifth-order, five-dimensional polynomial, as well as $s = 20$ with a third-order, 10-dimensional polynomial. We report the dependence of the average L_2 relative errors E_t and E_v and the success rate per 100 replica runs for various values of the number of measurements. Replica runs differ only by the training sample set. The success rate is defined as the fraction of runs that lead to the recovery of the nonzero bases. The relative errors are computed both for the training points and on a separate validation set of points of the same cardinality, proving that the failed runs suffer from overfitting.

Furthermore, Fig. 3 demonstrates the full dependence of the success rate on the number of measurements for a 5-dimensional problem with a 10th-order initial basis. Here, in order to save some computational effort, we used $R = 10$ replica simulations to determine the success rate. The results illustrated in Figs. 2 and 3 show that the algorithm performance depends on the sparsity of the problem and is essentially independent of the full dimensionality. Using Fig. 3, one can also draw a "rule-of-thumb" that for a somewhat safe recovery of the true solution one needs at least approximately $5s$ measurements, where s is the sparsity of the problem.

Genz functions [36, 37] provide a convenient way to dial-in dimensional importances, i.e., the sparsity of the model can be controlled up front. Consider a Genz-exponential function $f(\boldsymbol{\xi}) = \exp\left(\sum_{i=1}^{d} a_i \xi_i\right)$. All Genz functions, and the exponential function in particular, share a parameter vector $\boldsymbol{a} = (a_1, a_2, \ldots, a_d)$ that controls how important each parameter is for the output. These coefficients are set to follow a power rule $a_i = i^{-M}$. We will first focus on the $M = 1$ case and illustrate further how the BCS algorithm detects the important dimensions. The dialed-in dimensional importances a_i are randomly shuffled for the purpose of this illustration. The BCS algorithm is employed to distill the first-order terms only. Eventually all terms are picked as relevant with a high tolerance parameter. However, the order by which the basis terms are picked corresponds to hypothetical situations with stricter tolerances. The outcome of the algorithm clearly shows the order of importance of the dimensions: the earlier a specific term is selected, the more important it is. Figure 4 shows the values of dimensional importances, as well as the order by which the algorithm picks them. This proof-of-concept demonstrates that the BCS algorithm detects the dimensional importances in the expected sequence. Figure 4 also demonstrates the clear correlation between the dialed-in dimensional importance coefficients and the PC coefficients of the resulting linear expansions. Note that, for the given BCS stopping criterion, the algorithm detects the higher-importance dimensions better than the lower-importance ones.

For further tests, we will focus on the Genz-oscillatory function

$$f(\boldsymbol{\xi}) = \cos(2\pi e + \sum_{i=1}^{d} a_i \xi_i), \tag{27}$$

where the horizontal shift parameter has been chosen to avoid extra structure in the problem due to the even character of the cosine function. Note that despite the name, the function may not show oscillatory behavior on the hypercube $\boldsymbol{\xi} \in [-1, 1]^d$, depending on the values of a_i's.

The power parameter M controls the sparsity of the model. For a more insightful sparsity indicator, consider the effective dimensionality, d_{eff} defined as the number of the most influential dimensions contributing to 95% of the overall parameter sum $\sum_{i=1}^{d} a_i$, i.e.,

$$d_{\text{eff}} = \arg\min_{d'} \left(\frac{\sum_{i=1}^{d'} a_i}{\sum_{i=1}^{d} a_i} > 0.95 \right). \tag{28}$$

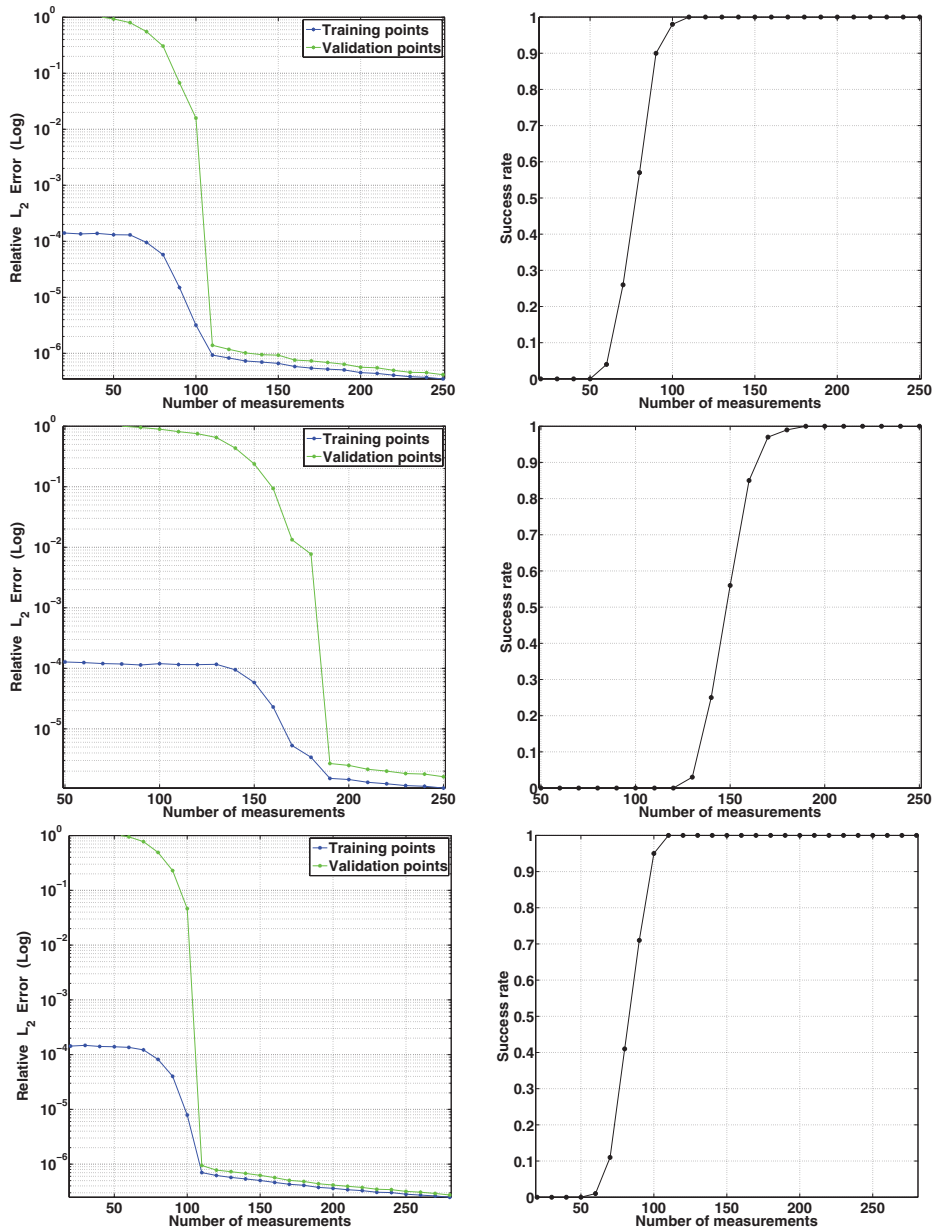


FIG. 2: Relative error (left column) and exact recovery success rate (right column) for three test functions of form (26). Top row: $p = 5$, $d = 5$, $s = 20$. Middle row: $p = 5$, $d = 5$, $s = 50$. Bottom row: $p = 3$, $d = 10$, $s = 20$. The test functions are created using TD truncation of degree p with a d-dimensional input and exactly s randomly chosen coefficients set to unity, while the other coefficients vanish.

As one would expect, the higher the effective dimensionality of the model, the less accurate the surrogate of a given maximal order is, for the same number of measurements. Indeed, Fig. 5 illustrates an application of the algorithm for a 50-dimensional Genz-oscillatory function with $M = 2$ starting with an initial overcomplete basis of third-order. The plots, from left to right, correspond to three different sets of measurements, with $N = 1000$, $N = 5000$, and $N =$ 10,000. A validation set with the same cardinality N is set aside to estimate the relative error on these points as well.

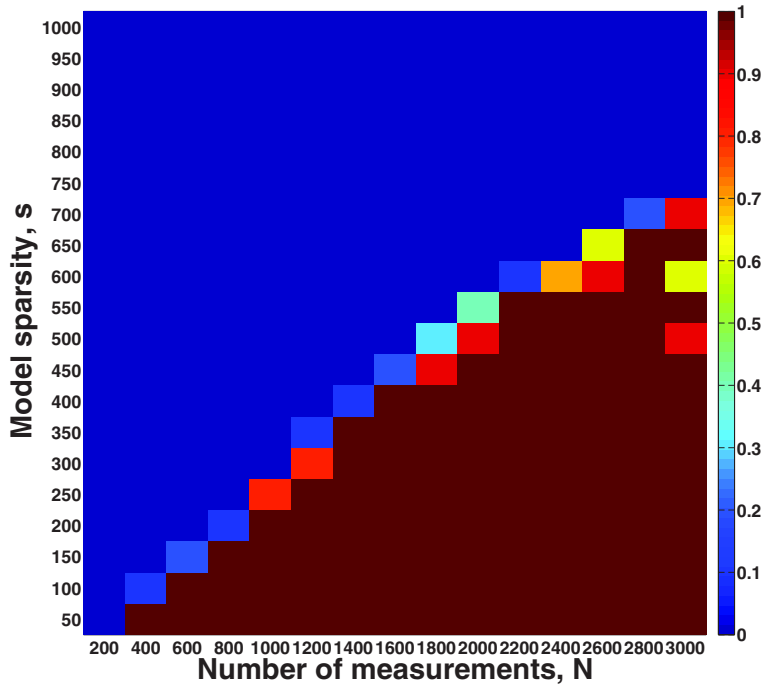


FIG. 3: Dependence of the sparse basis recovery success rate, out of 10 replicas, on the number of measurements and the model sparsity, i.e., the number of nonzero polynomial coefficients in the polynomial model (26).

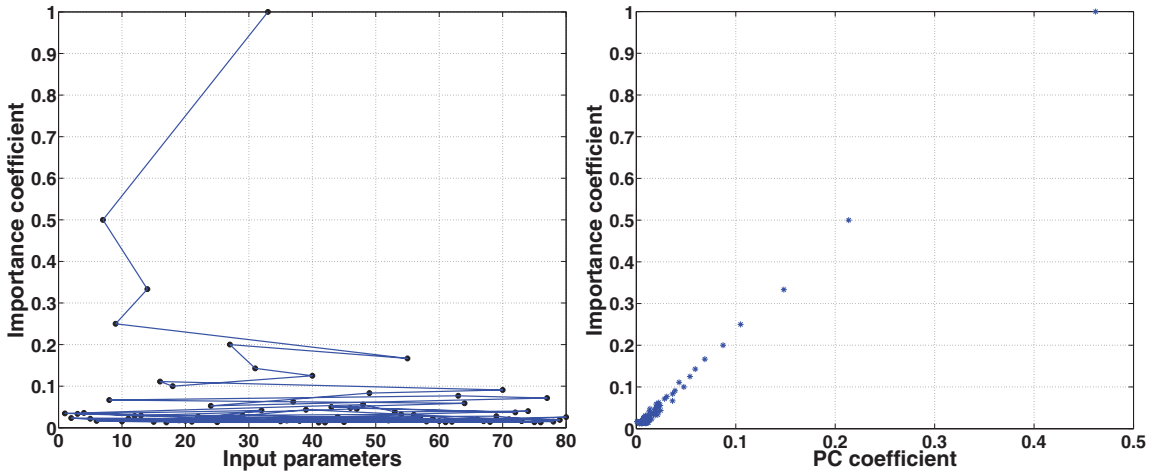


FIG. 4: Results of a first-order BCS algorithm with a 50-dimensional test function and $N_t = 1000$ training samples. The dimensional importances are shown with black dots, while the sequence in which the procedure picks the important dimensions is highlighted by the blue lines joining the dots, starting from the top. The right plot illustrates the correlation between the importance coefficients and the coefficients of the resulting linear PCEs.

In the case of overfitting, the error measure for the validation dataset is drastically above that of a training dataset, indicating that the corresponding model is not trustworthy. Clearly, the larger the training dataset, the more effective dimensions are captured accurately, while for a sparse dataset ($N = 1000$, the left plot), unless only a few dimensions matter, the resulting PC surrogate overfits the data badly.

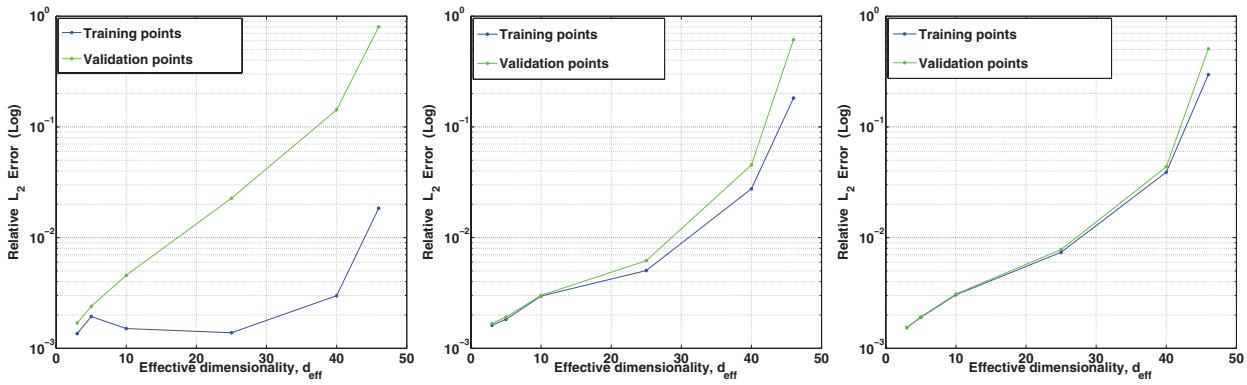


FIG. 5: Relative L_2 error measures for both training and validation datasets as the effective dimensionality increases, for three different values of the number of measurements, $N = 1000$ (left plot), $N = 5000$ (middle plot), and $N =$ 10,000 (right plot).

Next, consider again the function (27) with $d = 5$ and $a_i = i^{-2}$. Since the dimensionality is not too high, one can compute the true PC coefficients up to reasonably high order. Figure 6 shows the true, scaled coefficients up to fifth-order, computed using very accurate, level-14 Clenshaw-Curtis sparse quadrature. The BCS is applied in three different scenarios, for varying numbers of measurements. It is clear that with a lower number of measurements, some spurious, nonphysical PC modes appear due to lack of data. On the other extreme, when there is a large amount of data available, BCS does a very good job recognizing and retaining only the most important PC modes. Note that these results are in no contradiction with the "rule-of-thumb" described earlier in the paper: to recover approximately 100 terms accurately, one needs to make use of approximately $5 \times 100 = 500$ measurements.

The next illustration, Fig. 7, further demonstrates the effect of the number of measurements on the quality of the surrogate for the same function (27). The left plot corresponds to a 5-dimensional, 10th-order basis, while the right plot corresponds to a 10-dimensional, 5th-order basis. Note that, in both cases, the overcomplete basis sets have the same cardinality, $K = (10 + 5)!/(10!5!) = 3003$. The error bars and the plotted points correspond to $0.25, 0.50$ and 0.75 quantiles out of 10 identical simulations that differ only by the distribution of the training points. The validation

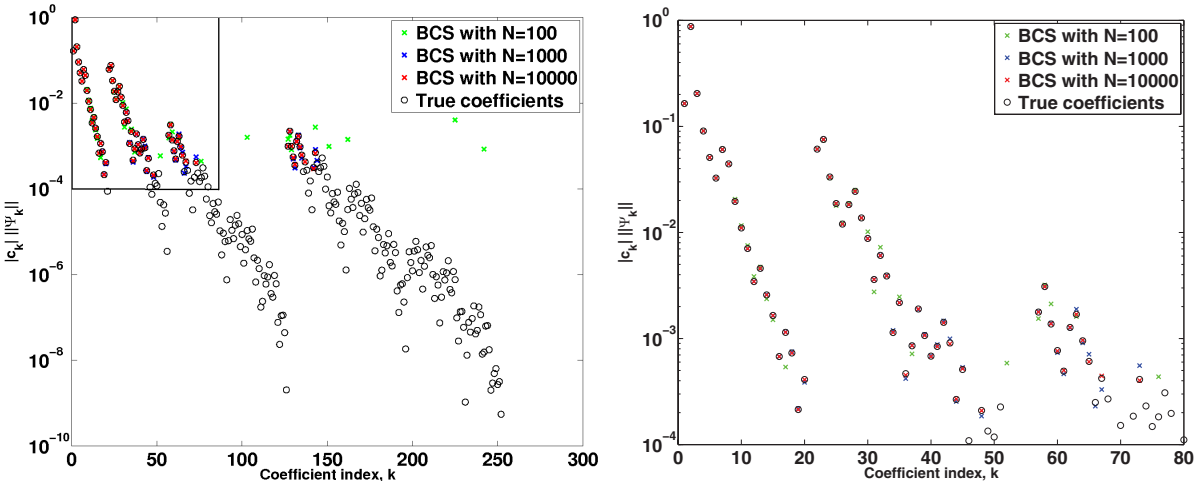


FIG. 6: Comparison between the exact PC coefficients of the function (27) with $a_i = i^{-2}$, $d = 5$, and the coefficients recovered by the BCS algorithm for three different values of the number of measurements. The plot on the right is a zoomed-in version of the upper-left corner.

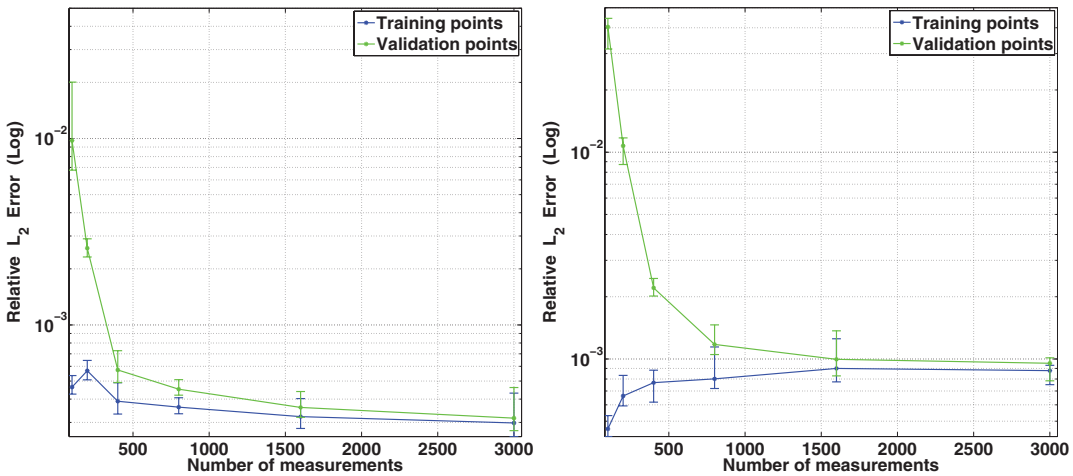


FIG. 7: The dependence of both training and validation relative errors on the number of measurement for 5- and 10-dimensional Genz oscillatory functions. The initial, overcomplete basis is taken with a TD truncation of orders 10 and 5, respectively, leading to the same number of basis terms in both cases. The error bars and the plotted points correspond to $0.25, 0.50$, and 0.75 quantiles out of 10 identical simulations that differ only by the distribution of the training points. Note that the initial increase in the error in the right plot is simply a consequence of strong overfitting.

sets, as before, have been chosen with the same cardinality as the corresponding training sets. The sparsity-controlling parameter M has been set to equal to 2, leading to effective dimensionalities $d_{\text{eff}} = 4$ and $d_{\text{eff}} = 6$ for the 5D and 10D problems, respectively. The discrepancy of the validation error E_v and training error E_t indicates overfitting. Clearly, as more training data arrive the quality of the PC surrogate improves—even if the error on training points increases, the corresponding models are more robust, since there is not a large gap between the validation error and the training error. The initial increase in the training error in the right plot in Fig. 7 reflects the fact that the data at hand do not cover the high-dimensional input parameter space well enough, leading to simpler, more smooth input-output relationships while with more and more data, the higher order relationships are exposed, rendering a polynomial fit of a fixed order less accurate even at the training points.

In order to demonstrate the dependence of the surrogate quality on the BCS stopping criterion ϵ we have tested the BCS algorithm with a gradually looser criterion on a single data set, for $N = 10{,}000$ measurements. We used the default, Genz-oscillatory function, with $d = 50$, $M = 2$ and started with an overcomplete basis of a total degree $p = 3$. Clearly, finding the right value for the stopping criterion is important, as very loose criteria might lead to too many retained terms causing an overfitting surrogate; see Fig. 8.

Figure 9 demonstrates convergence with the order of the initial overcomplete basis, for a given dataset of cardinality $N = 1000$, with a fixed BCS stopping criterion $\epsilon = 10^{-9}$, for a five-dimensional Genz-oscillatory function. Again, as before, the error bars correspond to 0.25 and 0.75 quantiles out of 10 replica runs. With higher order of the initial basis, the BCS algorithm searches among more basis terms, therefore it is able to find a sparse PC surrogate with a smaller relative error.

4. ITERATIVE PROCEDURE WITH BCS

With a total-degree truncation and in the presence of a large number of dimensions, one cannot generally afford to build an initial PC basis of high order. Here we propose an iterative procedure that allows enriching the basis with higher order polynomials while maintaining the dimensionality reduction. Given a multi-index set $\mathcal{S}$ corresponding to the current basis, we add a basis term only if it is *admissible*. A conservative definition of the admissibility is the following. We call a new basis term admissible, if by subtracting an order from any nonzero dimension one never obtains a multi-index outside the set $\mathcal{S}$. In other words,

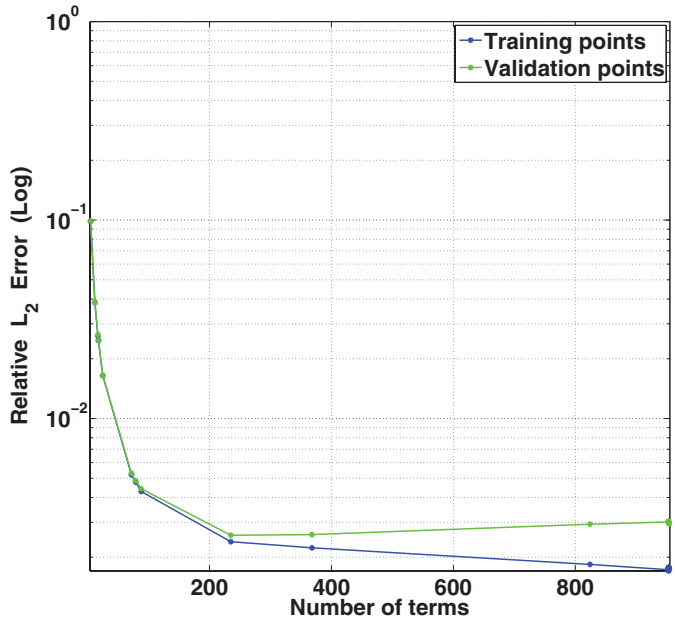


FIG. 8: The surrogate accuracy is plotted against the number of retained terms in the polynomial expansion by varying the BCS stopping criterion.

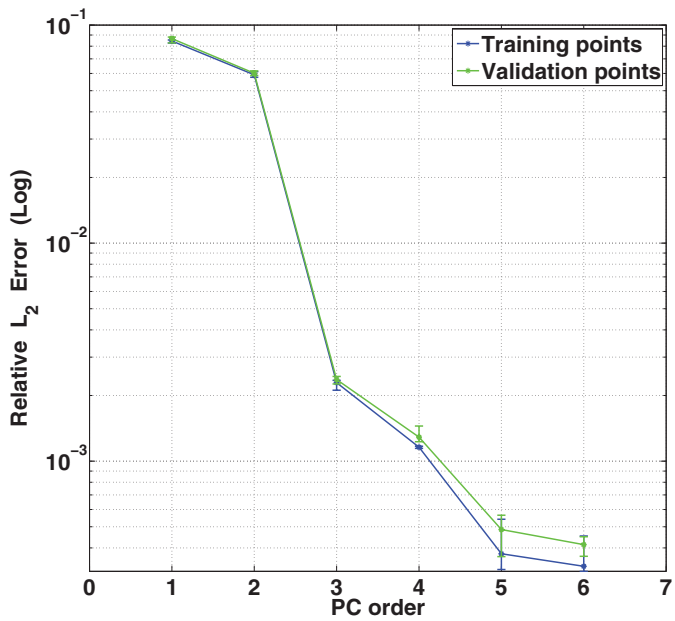


FIG. 9: Illustration of the surrogate convergence with increasing order of the initial, overcomplete basis. The error bars and the plotted points correspond to $0.25, 0.50$, and 0.75 quantiles out of 10 identical simulations that differ only by the distribution of the training points.

$$\boldsymbol{p} = (p_1, \ldots, p_d) \text{ is added to } \mathcal{S}, \text{ if } \boldsymbol{p} - \boldsymbol{e}_i \in \mathcal{S}, \text{ for all } i = 1, \ldots, d, \tag{29}$$

where $\boldsymbol{e}_i = (0, \ldots, 1, \ldots, 0)$ with 1 in the ith position. The full algorithm then reads as follows:

- Step 0. Let $\tilde{\mathcal{S}}$ be a set of multi-indices with a total degree $\leq l_0$, where l_0 is the initial PC order,
- Step 1. Run the BCS algorithm to reduce the current basis set $\tilde{\mathcal{S}} \to \mathcal{S}$.
- Step 2. Enrich the current basis by all admissible basis terms and call the new basis multi-index set $\tilde{\mathcal{S}}$. Repeat from Step 1 until the maximal order l is reached.

Note that the admissibility condition can sometimes prove to be too restrictive. In Section 6, we apply a less constrained admissibility condition, therefore, a less conservative basis enrichment scheme in order to enhance the basis search by having a larger set of an overcomplete basis at each step. The general flowchart of the iterative BCS algorithm is schematically depicted in Fig. 10.

The resulting representation reads as follows:

$$z_{\boldsymbol{c}}(\boldsymbol{\xi}) = \sum_{\boldsymbol{p} \in \mathcal{S}} c_{\boldsymbol{p}} \Psi_{\boldsymbol{p}}(\boldsymbol{\xi}), \tag{30}$$

where the PC modes $\boldsymbol{c_p}$ are described by a multivariate Gaussian posterior of a dimensionality that is equal to the cardinality of $\mathcal{S}$.

4.1 Numerical Tests

Consider a forward function $f(\xi_1, \xi_2) = \psi_5(\xi_1)$, i.e., a normalized Legendre polynomial of degree 5 in the first dimension. Clearly, a PC surrogate of this function can be exact and including only one term, corresponding to the multi-index $(5, 0)$. We start the iterative BCS (iBCS) procedure from a second total-order PC and iteratively grow (by admissibility) and shrink (by BCS) the basis set. Figure 11 illustrates the results. At the iteration that includes the true basis term $(5, 0)$, that and only that term is selected. This test also illustrates how the iterative procedure increases the basis set in the right direction, so that the final iteration includes the correct term. In all but the last iteration, the basis set does not include the correct basis. However, the algorithm tends to keep terms "closer" to $(5, 0)$ recognizing a higher relative importance of the first dimension compared to the second dimension. Note, however, that in the two-dimensional setting the application of the iterative procedure is merely for illustration, since in such low-dimensional setting one could build an initial, overcomplete basis set of a large enough order to capture the correct basis term with a single BCS application. The usage of the iterative procedure becomes necessary in very high dimensional problems, where the construction of an overcomplete basis with a standard truncation rule (e.g., total-degree, or tensor-product) is computationally prohibitive. For example, in an 80-dimensional problem, a TD truncation of second order leads to $82!/(80!2!) = 3321$ terms, while the third order involves $83!/(80!3!) = 91{,}881$ terms. The number of basis terms for a TP-truncated expansion grows even faster, in an exponential fashion.

Consider now an 80-dimensional example with the cosine test function (27) and $N = 10{,}000$ training samples. We used dimensionality importance decaying exponent $M = 2$, i.e., $a_i = i^{-2}$ in (27), which corresponds to an effective

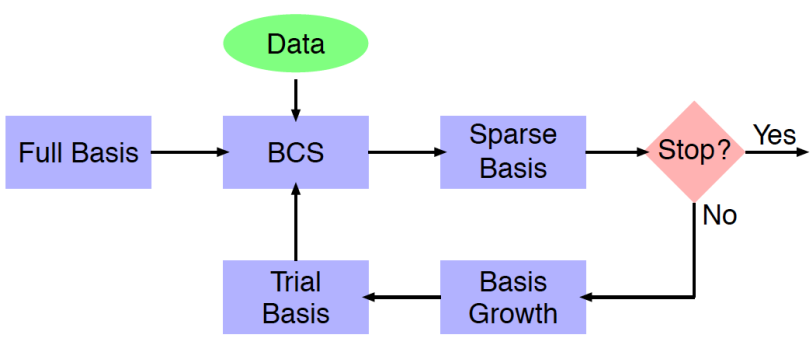


FIG. 10: Illustration of the iterative BCS algorithm. The choice of initial basis and the basis growth rule is often problem-dependent and is dictated by computational capabilities at hand.

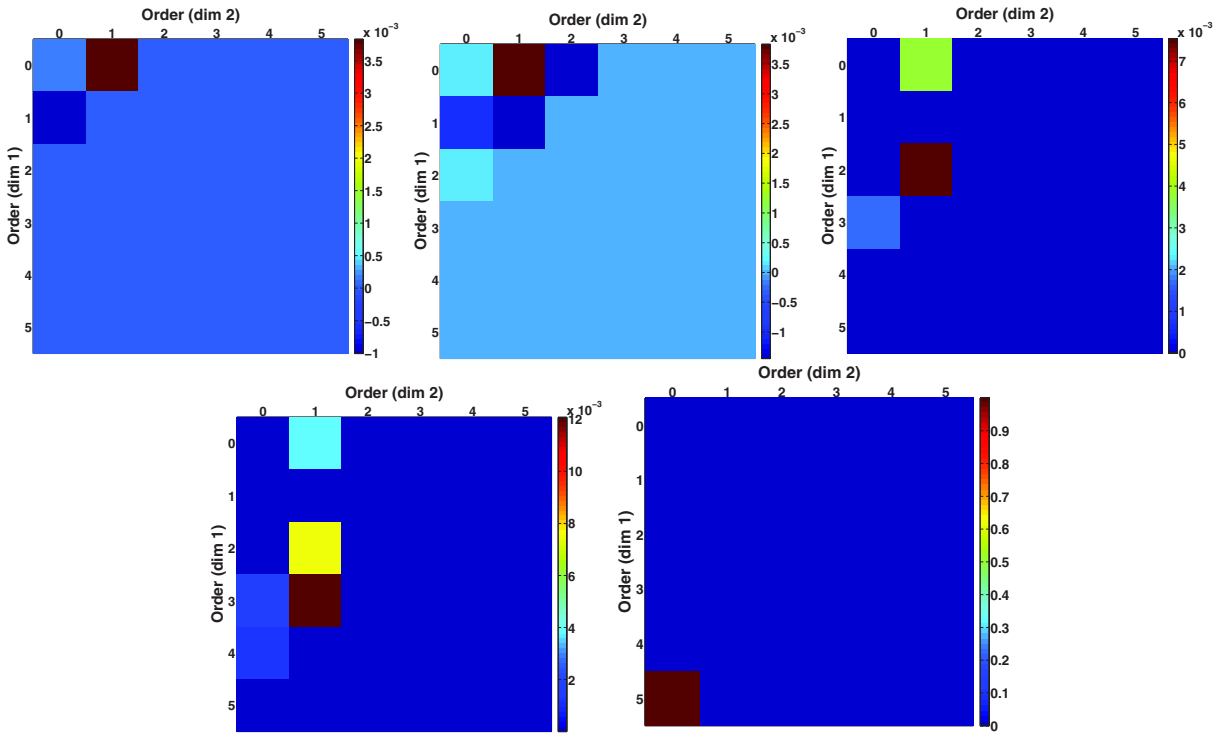


FIG. 11: Illustration of the iterative basis growth strategy. The direction of successive iterations is from left to right and from top to bottom. The true solution has a single PC term. The basis set grows in the "right" direction in order to eventually capture the correct basis term. The (i, j)-square corresponds to the PC coefficient of the term $\Psi_i(\xi_1)\Psi_j(\xi_2)$. Note that for each plot, the colormap is different for the clarity of visualization.

dimensionality $d_{\text{eff}} = 11$ according to the definition (28). At the first iteration, we take an initial basis set of total order 2 with over 3000 terms (note that, as mentioned in the previous paragraph, the third-order expansion leads to a very large number of basis terms with a computationally intractable projection matrix). With every iteration we include higher order basis terms in our search. However, as Fig. 12 illustrates, the relative errors do not decrease further after the third-order. The reason is that we have used a very strict BCS stopping criterion to avoid overfitting. However, the number of training points sets the limit: with $N = 10{,}000$ samples in an 80-dimensional space, we can robustly "learn" the model only up to certain accuracy. For a less strict criterion ϵ one can reduce the error on the training points, but the validation error E_v will increase indicating that the corresponding model is overfitting and should not be used for accurate predictions. Nevertheless, note the number of basis terms retained compared to the number of basis terms corresponding to a full total-order basis truncation of the respective order. We refer to Section 6, where for the problem at hand, a more aggressive basis growth rule is implemented that allows a more exhaustive basis search at each iteration.

5. PIECEWISE PC EXPANSIONS FOR NONSMOOTH FUNCTIONS

In many situations, the training dataset suggests either strong nonlinearity, nonsmoothness of the forward function, or more than one qualitatively different kind of behavior. In either case, decomposing the input into regions with relatively more smooth functional behavior is a key for building more accurate surrogates. This type of multi-resolution analysis leading to local PC representations could be prohibitively expensive in high-dimensional settings if the input domain is decomposed in a brute-force fashion, parallel to axes [17–19]. More advanced methods that "tile" the input domain appropriately based on discontinuity detection algorithms also suffer from the curse of dimensionality, mainly

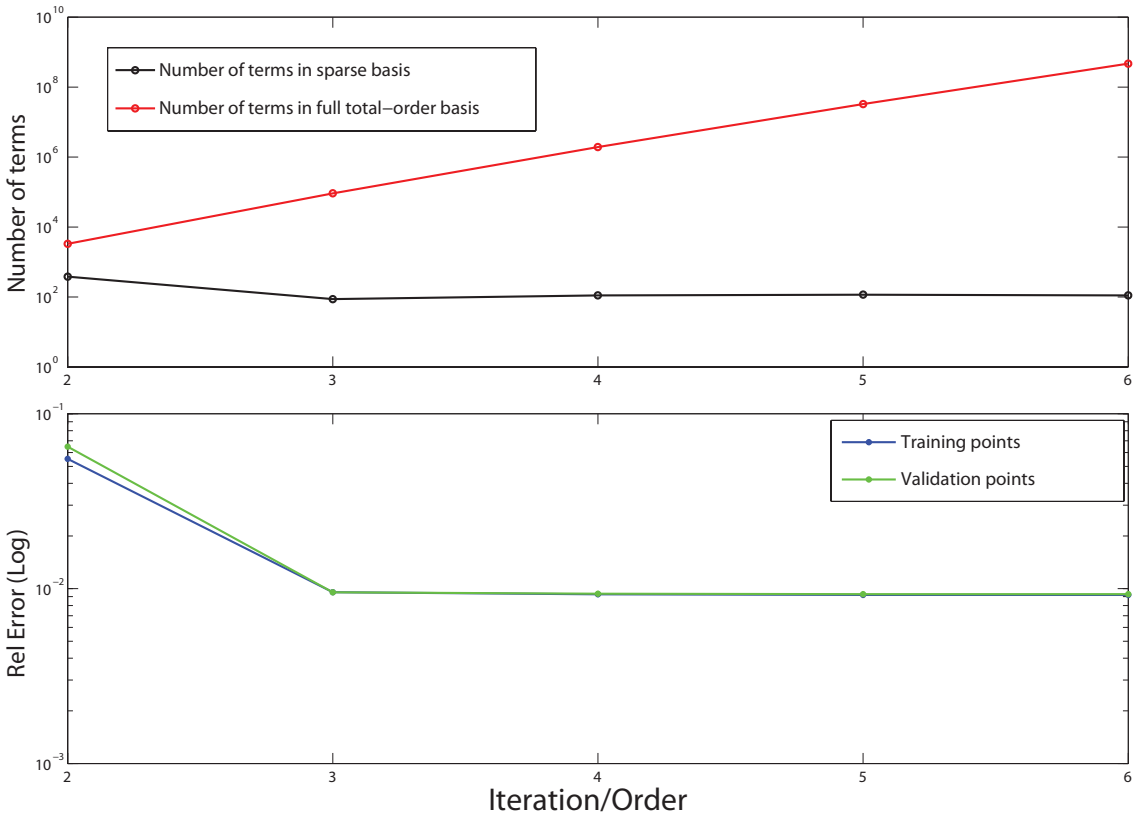


FIG. 12: Illustration of the iterative BCS algorithm for an 80-dimensional cosine test function with $N = 10000$ training points. The closeness of validation and training errors indicates that a robust model is obtained with a relative error of $\sim$1%, while using much fewer basis terms than the corresponding total-order basis.

due to difficulties associated with representations of high-dimensional discontinuity manifolds [20, 38]. Instead, in this work we employ a data-driven decomposition based on a clustering method informed by the values of the function outputs. Below we propose formal steps to build a piecewise PC surrogate enhanced with training dataset clustering. For simplicity, we describe a two-cluster case—the generalization to a multi-cluster PC expansion is straightforward.

- Cluster the training dataset into nonoverlapping subsets $\mathcal{D}_1$ and $\mathcal{D}_2$, within each of which the behavior of the function is better suited for a polynomial approximation.
- Construct global PC expansions $g_i(\boldsymbol{\xi}) = \sum_k c_{ik}\Psi_k(\boldsymbol{\xi})$ using each dataset individually ($i = 1, 2$).
- Declare a piecewise-PC surrogate

$$g_s(\boldsymbol{\xi}) = \begin{cases} g_1(\boldsymbol{\xi}) & \text{if } \boldsymbol{\xi} \in \mathcal{D}_1 \\ g_2(\boldsymbol{\xi}) & \text{if } \boldsymbol{\xi} \in \mathcal{D}_2. \end{cases} \tag{31}$$

The initial division of the samples into two or more clusters is problem-dependent but has the overarching goal of making each cluster of samples more amenable to a smooth representation. Note that the last step (31) requires a classification mechanism, since one needs to determine which cluster $\boldsymbol{\xi}$ belongs to. We apply the random decision forest (RDF) algorithm that is the state-of-the-art in machine learning and suits particularly well for high-dimensional, sparse datasets, in terms of both accuracy and computational complexity [25]. The RDF classification algorithm essentially generates an ensemble of decision trees that classify a given point to one of the clusters by recursively

splitting the input space parallel to the axes. The decision reported by RDF then is the "majority vote" of these decision trees.

It is implicitly assumed that the regions $\mathcal{D}_1$ and $\mathcal{D}_2$ are connected, i.e., do not consist of separated subregions. In the latter case, although one could build a global PC expansion over a nonconnected region, it is preferable to generalize the algorithm to the larger number of clusters.

Below we will illustrate the piecewise-PC construction and describe its advantages and drawbacks based on an example that shows behavior similar to our main application from Section 6. Namely, let us take the following bivariate function:

$$f(\xi_1, \xi_2) = e^{\sum_{i=1}^{2} c_i |\xi_i|} (\xi_2^3 - \xi_1)^+, \tag{32}$$

where $c_1 = 1$, $c_2 = 0.25$, and $x^+ = x$ if $x > 0$ and vanishes otherwise. The function is continuous, but it goes through a nonsmooth transition when crossing the curve $\xi_2^3 = \xi_1$. Let us assign all samples that lead to a nonzero output to the cluster $\mathcal{D}_1$, while the rest of the samples are assigned to the cluster $\mathcal{D}_2$. Clearly, the PC expansion for $\mathcal{D}_2$ is a trivial one, $g_2(\boldsymbol{\xi}) = 0$. The PC expansion for $\mathcal{D}_1$ is a global pth-order expansion $g_1(\boldsymbol{\xi})$. However, one only retains its values corresponding to inputs $\boldsymbol{x}$ that are classified as belonging to $\mathcal{D}_1$. Figure 13 illustrates the piecewise-PC surrogate constructed based on 150 training points. Note, however, that in high-dimensional problems, one should be aware of misclassification errors when comparing the surrogate model to the full model runs. More detailed discussion on this issue is relegated to Section 6.

6. APPLICATION TO THE CLM

The main motivational model for the algorithmic development in the previous sections is CLM-CN, the community land model with carbon/nitrogen biogeochemistry [39], which employs a nested computational grid hierarchy designed to represent the spatial heterogeneity of the land surface. Each computational cell can be composed of multiple land units (e.g., ice, vegetation, or urban area), each land unit can incorporate one or more independent soil

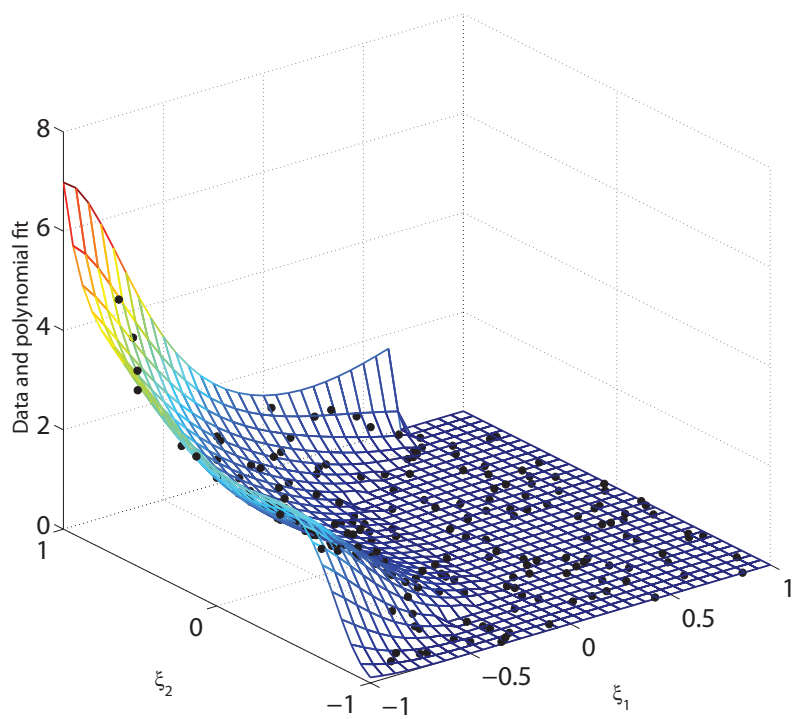


FIG. 13: Illustration of a two-dimensional response surface where a global PC expansion is enhanced by a classification in order to alleviate the effects of nonlinearities in the forward function.

columns, and each vegetated soil column may contain multiple plant functional types (PFTs). CLM-CN is a state-of-the-art land surface model and has been used both offline and as part of the fully coupled community Earth system model (CESM) to make future climate projections [40]. CLM-CN generally performs well against global benchmark datasets, both for offline simulations [41, 42] and within fully coupled simulations [43]. Here we perform offline simulations driven by site-observed meteorology at a single computational cell centered over the Niwot Ridge flux tower site in Colorado [44], modeled as a single plant functional type: temperate evergreen needleleaf forest. The results presented are steady-state vegetation produced by cycling present-day climate; past disturbances are not included. For these simulations at a single site for a 100-year horizon, the computational cost of a single simulation is quite high—approximately 10 h on a single processor. A total of 39 PFT-level physiology parameters exist in an input file, and an additional 42 constants primarily controlling soil, hydrology, and phenology processes were pulled out of the code and recast as input parameters. Thus the number of input parameters that control the model behavior is 81 in this case; see Tables 1 and 2. Clearly, such an expensive and high-dimensional model would benefit from a surrogate construction to enable sensitivity analysis or to perform optimization or calibration. However, building the surrogate involves several challenges indicated earlier in this paper: e.g., the curse of dimensionality, as well as the computational cost, and, therefore, the sparsity of model runs.

6.1 Generation of Input Parameter Ensembles

Tables 1 and 2 present the list of model input parameters explored in our study. Beside range restrictions, the input parameters need to satisfy the following constraints, by definition, or in order to remain consistent with associated physics:

$$\begin{aligned} \lambda_{18} &< \lambda_{22}, \\ \lambda_{30} + \lambda_{31} + \lambda_{32} &= 1, \\ \lambda_{33} + \lambda_{34} + \lambda_{35} &= 1. \end{aligned} \tag{33}$$

For example, Fig. 14 illustrates two sets of 1000 uniform samples on polygons that are obtained due to constraints $\lambda_{33} + \lambda_{34} + \lambda_{35} = 1$ and $\lambda_{18} < \lambda_{22}$, respectively.

In order to have proper coverage of the input parameter space that respects the constraints and uses all available information, the training set of input parameters is taken to be uniformly distributed on the constrained space. This is consistent with the maximum entropy principle, see [32].

In order to generally map a constrained input space to an unconstrained one, we introduce a transformation that maps the input parameter vector $\boldsymbol{\lambda}$ with dependent or constrained components to a vector of *i.i.d.* uniform variables $\boldsymbol{\xi}$. For this purpose, we use the Rosenblatt transformation [45], which is essentially a generalization of the standard cumulative distribution function (CDF) transformation to multiple dimensions.

To clarify the upcoming notation, let us remove one input parameter from the triple $(\lambda_i, \lambda_j, \lambda_k)$ for each constraint of a form $\lambda_i + \lambda_j + \lambda_k = 1$, since one of the parameters in the triple is completely determined by the other two. With the appropriate shifting of the indices, we will be left with $d = \tilde{d} - n_t$ input parameters, where n_t is the number of input parameter triples that sum up to one. In the present study, $n_t = 2$ and therefore $d = 79$.

Now, given a vector of random variables $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_d)$ with known joint CDF $F(\lambda_1, \ldots, \lambda_d)$, one can obtain a set of ξ_i's that are independent uniform random variables on $[-1, 1]$ for all $i = 1, 2, \ldots, d$, using the *scaled* conditional cumulative distributions

$$\begin{aligned} \xi_1 &= R_1(\lambda_1) \\ \xi_2 &= R_{2|1}(\lambda_2|\lambda_1) \\ \xi_3 &= R_{3|2,1}(\lambda_3|\lambda_2, \lambda_1) \\ &\vdots \\ \xi_d &= R_{d|d-1,\ldots,1}(\lambda_d|\lambda_{d-1}, \ldots, \lambda_1). \end{aligned} \tag{34}$$

TABLE 1: CLM input parameters: Part one

Notation	Name	Default	Min	Max	Units	Description
λ_1	displar	0.67	0.1	1	m	displacement length: canopy top
λ_2	dleaf	0.04	0.01	0.1	m	characteristic leaf dimension
λ_3	mp	6	3	16	none	slope of conductance to photosynthesis
λ_4	qe25	0.06	0.04	0.08	umol	C/umol phot Quantum efficiency
λ_5	rholvis	0.07	0.01	1	none	leaf reflectance (vis)
λ_6	rholnir	0.35	0.01	1	none	leaf reflectance (nir)
λ_7	rhosvis	0.16	0.01	1	none	stem relectance (vis)
λ_8	rhosnir	0.39	0.01	1	none	stem reflectance (nir)
λ_9	taulvis	0.05	0.01	1	none	leaf transmittance (vis)
λ_{10}	taulnir	0.1	0.01	1	none	leaf transmittance (nir)
λ_{11}	tausvis	0.001	0.0001	0.01	none	stem transmittance (vis)
λ_{12}	tausnir	0.001	0.0001	0.01	none	stem transmittance (nir)
λ_{13}	xl	0.01	0.01	1	none	leaf/stem orientation index
λ_{14}	roota_par	7	1	20	m-1	rooting distribution parameter
λ_{15}	rootb_par	2	0.5	10	m-1	rooting distribution parameter
λ_{16}	slatop	0.01	0.08	0.12	m2/gC	SLA at top of canopy
λ_{17}	dsladlai	0.0012	0.001	0.007	m2/gC/LAI	SLA/dLAI
λ_{18}	leafcn	35	23	70	gC/gN	leaf C to N ratio
λ_{19}	flnr	0.05	0.04	0.1	none	frac of leaf N in Rubisco
λ_{20}	smpso	$-66{,}000$	$-120{,}000$	$-20{,}000$	mm	soil water pot. at full opening
λ_{21}	smpsc	$-255{,}000$	$-300{,}000$	$-120{,}000$	mm	soil water pot. at closure
λ_{22}	lflitcn	70	39	143	gC/gN	leaf litter C:N
λ_{23}	frootcn	42	25	85	gC/gN	fine root C:N
λ_{24}	livewdcn	50	25	75	gC/gN	live wood C:N
λ_{25}	deadwdcn	500	200	1400	gC/gN	dead wood C:N
λ_{26}	froot_leaf	1	0.3	5	gC/gC	new fine root alloc C /leaf C
λ_{27}	stem_leaf	1.5	0.6	5.3	gC/gC	new stem alloc C per leaf C
λ_{28}	croot_stem	0.3	0.1	0.7	gC/gC	new croot alloc C per stem C
λ_{29}	flivewd	0.1	0.06	0.28	none	fraction of new wood that is live
λ_{30}	lf_flab	0.25	0.14	0.54	none	leaf litter labile fraction
λ_{31}	lf_fcel	0.5	0.37	0.49	none	leaf litter cellulose fraction
λ_{32}	lf_flig	0.25	0.1	0.38	none	leaf litter lignin fraction
λ_{33}	fr_flab	0.25	0.18	0.25	none	fine root labile fraction
λ_{34}	fr_fcel	0.5	0.38	0.5	none	fine root cellulose fraction
λ_{35}	fr_flig	0.25	0.16	0.36	none	fine root lignin fraction
λ_{36}	leaf_long	1.5	2	10	yr	leaf longevity
λ_{37}	resist	0.12	0	0.5	none	fire resistance index
λ_{38}	grperc	0.3	0.2	0.4	none	growth respiration factor 1
λ_{39}	grpnow	1	0	1	none	growth respiration factor 2
λ_{40}	bdnr	0.25	0	0.8	(1/s)	bulk denitrification rate

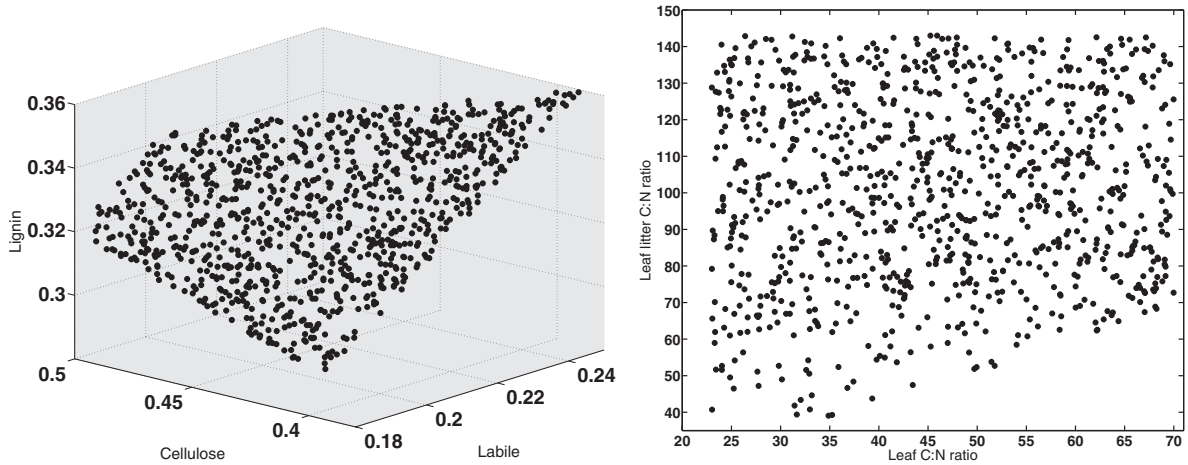
Each map $R_*(\cdot)$ is a scaled version of the corresponding CDF $F_*(\cdot)$ to ensure $\xi_i \in [-1, 1]$. That is, $R_1(\lambda_1) = 2F_1(\lambda_1) - 1$ and, similarly, for the rest of the conditional CDFs in (34).

This map, denoted by the shorthand notation $\boldsymbol{\xi} = R(\boldsymbol{\lambda})$, is called the *Rosenblatt transformation* (RT) [45]. Note that the RT is not unique: by ordering the λ_i's in different ways, one can obtain $d!$ different sets of uniform random variables.

TABLE 2: CLM input parameters: Part two

Notation	Name	Default	Min	Max	Units	Description
λ_{41}	dayscrecover	300	1	90	days	days to recover negative cpool
λ_{42}	rc_npool	100	0.5	50	none	resistance for uptake from plant npool
λ_{43}	br_mr	$2.53e-06$	$4e-07$	$1e-05$	gC/gN/s	base rate for maintenance respiration
λ_{44}	q10_mr	1.5	1	4.5	none	q10 for maintenance respiration
λ_{45}	cn_s1	12	8	20	gC/gN	carbon:nitrogen for SOM 1
λ_{46}	cn_s2	12	8	20	gC/gN	carbon:nitrogen for SOM 2
λ_{47}	cn_s3	10	6	20	gC/gN	carbon:nitrogen for SOM 3
λ_{48}	cn_s4	10	6	20	gC/gN	carbon:nitrogen for SOM 4
λ_{49}	rf_l1s1	0.39	0.35	0.45	none	resp. fraction for litter 1 → SOM 1
λ_{50}	rf_l2s2	0.55	0.385	0.715	none	resp. fraction for litter 2 → SOM 2
λ_{51}	rf_l3s3	0.29	0	0.9	none	resp. fraction for litter 3 → SOM 3
λ_{52}	rf_s1s2	0.28	0.26	0.3	none	resp. fraction for SOM 1 → SOM 2
λ_{53}	rf_s2s3	0.46	0.032	0.6	none	resp. fraction for SOM 2 → SOM 3
λ_{54}	rf_s3s4	0.55	0	1	none	resp. fraction for SOM 3 → SOM 4
λ_{55}	k_l1	1.2	0.9	1.5	1/day	decomp rate for litter 1
λ_{56}	k_l2	0.0726	0.05	0.1	1/day	decomp rate for litter 2
λ_{57}	k_l3	0.0141	0.005	0.028	1/day	decomp rate for litter 3
λ_{58}	k_s1	0.0726	0.038	0.11	1/day	decomp rate for SOM 1
λ_{59}	k_s2	0.0141	0.005	0.022	1/day	decomp rate for SOM 2
λ_{60}	k_s3	0.0014	0.0004	0.005	1/day	decomp rate for SOM 3
λ_{61}	k_s4	0.0001	0	0.0004	1/day	decomp rate for SOM 4
λ_{62}	k_frag	0.001	0.0002	0.005	1/day	fragmentation rate for CWD
λ_{63}	cwd_fcel	0.769	0.66	0.81	none	fraction of cellulose in CWD
λ_{64}	dnp	0.01	0.001	0.1	none	denitrification proportion
λ_{65}	minpsi_hr	-10	-15	-5	MPa	minimum psi for heterotrophic resp
λ_{66}	q10_hr	1.5	1	4.5	none	q10 for heterotrophic respiration
λ_{67}	r_mort	0.02	0.002	0.2	1/year	mortality rate
λ_{68}	sf_minn	0.1	0.02	0.4	none	solulble fraction of mineral N
λ_{69}	crit_dayl	39,300	35,000	45,000	seconds	critical daylength for senescence onset
λ_{70}	ndays_on	30	5	60	days	no. of days to complete leaf onset
λ_{71}	ndays_off	15	5	40	days	no. of days to complete leaf offset
λ_{72}	fstor2tran	0.5	0.1	1	none	fraction of strage to move to transfer
λ_{73}	crit_onset_fdd	15	5	30	days	no. of freezing days to set GDD counter
λ_{74}	crit_onset_swi	15	5	30	days	no. of water stress-free days for leaf onset
λ_{75}	soilpsi_on	-2	-5	-0.75	MPa	critical soil water potential for leaf onset
λ_{76}	crit_offset_fdd	15	5	30	days	no. of freezing days for leaf offset
λ_{77}	crit_offset_swi	15	5	30	days	no. of water stress days for leaf offset
λ_{78}	soilpsi_off	-2	-5	-0.75	MPa	critical soil water potential for leaf offset
λ_{79}	lwtop_ann	0.7	0.5	1	1/year	live wood turnover proportion
λ_{80}	gddfunc_p1	4.8	3	7	none	gdd threshold parameter 1
λ_{81}	gddfunc_p2	0.13	0.05	0.3	none	gdd threshold parameter 2

The inverse of the RT will be employed to map the unconstrained $\boldsymbol{\xi}$ space (i.e., $[-1,1]^d$) to the space of CLM input parameters to perform proper sampling of the constrained input space. Namely, we sampled $N' = 10{,}000$ uniform *i.i.d.* random values on the 79-dimensional $\boldsymbol{\xi}$ space, using LHS to ensure good coverage of the space. These



(a) triple $(\lambda_{33}, \lambda_{34}, \lambda_{35})$ (b) pair $(\lambda_{18}, \lambda_{22})$

FIG. 14: An illustration of 1000 input parameter samples for some of the constrained inputs with their range given in Tables 1 and 2, obtained via the inverse Rosenblatt transformation.

samples correspond to $10{,}000$ sets of input parameter vectors for CLM, each of size 81. However, 17 runs failed or crashed during forward model simulation, resulting in an effective set of $N = 9983$ samples. With the Rosenblatt transformation in place, one can build a PC-based surrogate with respect to $\boldsymbol{\xi}$, as described in Section 2.

We applied the iterative BCS algorithm for polynomial basis reduction, described in Section 4, to the CLM with five output quantities of interest (QoI), shown in Table 3. For each QoI, a 10-year average at the end of a 100-year CLM simulation is taken.

Let us study the leaf area index (output LAI) more closely. Figure 15 illustrates all the available model simulation results, on the log-axis. Notice the clear separation of the output values. Many simulations either lead to vanishing LAI, or to a small value that is believed to converge to zero *if* the simulations were run for a longer time horizon. While our ongoing work includes more robust, longer-term CLM simulations, for the present purpose we will categorize all runs with $LAI < LAI_{thr} = 0.3$ as effectively leading to a dead vegetation state. The choice of threshold value is somewhat *ad hoc*: the sorted simulation plot in Fig. 15(b), as well as preliminary tests, suggested the value of $LAI_{thr} = 0.3$. Also, as these log-axis plots suggest, the logarithm of the LAI is considered for PC surrogate building purposes. Besides the fact that the logarithm of this dataset looks more amenable to a polynomial representation, it also ensures that the resulting surrogate model for LAI stays non-negative.

Therefore, we obtain a two-regime surrogate for this QoI:

$$LAI(\boldsymbol{\lambda}) = \begin{cases} \exp(\sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\xi})) & \text{if } \boldsymbol{\xi} \in \mathcal{D} \\ 0 & \text{otherwise} \end{cases}, \tag{35}$$

where $\mathcal{D}$ is the input parameter region that leads to a nonzero LAI. Recall that the CLM parameters $\boldsymbol{\lambda}$ and the PC surrogate arguments $\boldsymbol{\xi}$ are related via the Rosenblatt transformation $\boldsymbol{\xi} = R(\boldsymbol{\lambda})$. In order to determine whether the

TABLE 3: CLM output quantities of interest

Notation	Name	Units	Description
y_1	LAI	none	Leaf area index
y_2	TOTVEGC	gC/m^2	Total vegetation carbon
y_3	GPP	gC/m^2/s	Gross primary production
y_4	HR	gC/m^2/s	Heterotrophic respiration
y_5	FPSN	mmol/(m^2 s)	Photosynthesis

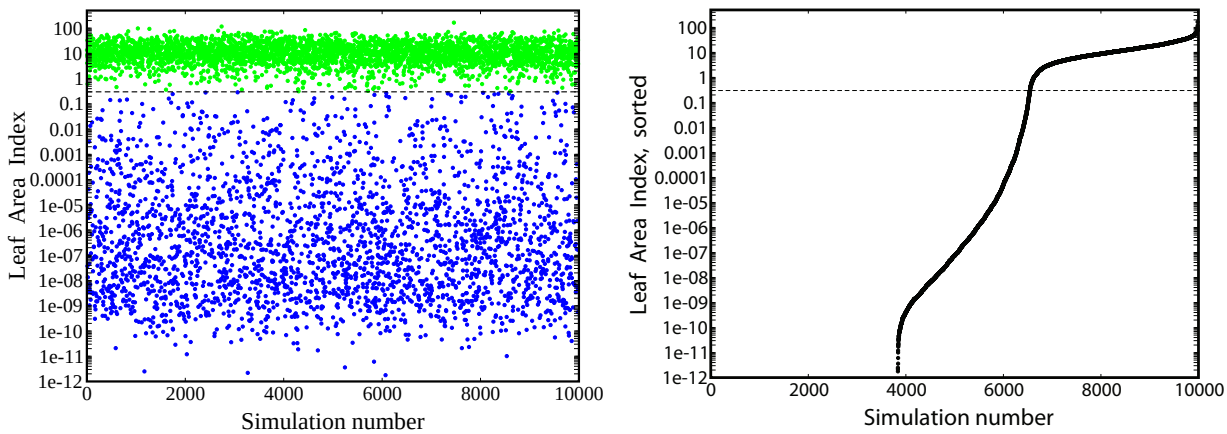


FIG. 15: The dataset for the output LAI, corresponding to 9983 CLM simulations. The dataset is clustered into two categories, below the threshold 0.3 it is believed that the simulations correspond to a dead vegetation state, in the long run. The sorted plot on the right also suggests that there is a "kink" near the threshold value below which a qualitatively different outcome is expected.

point of interest ξ, belongs to $\mathcal{D}$, the RDF algorithm [25] for classification is launched, as described in Section 5. However, having no more than $10{,}000$ samples in a 79-dimensional space leads to an approximately 15% likelihood of misclassification.

The basis selection and proper coverage of all possible bases for the expansion in (35) needs special attention for such a high-dimensional system. For example, note that a total-degree expansion of order 3 includes almost 10^5 polynomial terms. It is expected that using a more flexible, HDMR-structured basis set would be helpful, as physical models tend to be most influenced by lower-dimensional input interactions [46, 47]. Accordingly, our initial basis set choice for the iterative BCS algorithm described in Section 4, is an HDMR basis set with maximal order pair $(9, 2)$, i.e., univariate bases of orders up to 9 (a total of $79 \cdot 9 = 711$ terms) and bivariate terms of total order up to 2 (a total of $78 \times 79/2 = 3160$ terms of the form $\xi_i \xi_j$ for $i, j = 1, \ldots, 79$ and $i < j$). Together with the constant basis term, this leads to an initial basis of size $K_1 = 3793$. Further, in order to control the number of basis terms and avoid overfitting, we employ a modified approach compared to Section 4, as follows. At each iteration, we (1) take a less conservative order addition scheme where one adds the higher-dimensional *admissible* coupling terms while also adding all terms up to total order 9 on the current selected basis; and (2) retain only the first 1000 selected basis terms. The iBCS approach is then carried forward for 3 iterations with a final result of 1000 basis terms.

However, since only about a third of the CLM simulations leads to nonzero outputs, the resulting dataset is sparse given the 79-dimensional space, and a 1000-term basis set necessarily leads to overfitting and poor predictive qualities. Therefore, we further adjust our use of iBCS to further guard against overfitting by retaining only those terms that are most *robustly* supported by the data. This is done as follows. We randomly choose 10 different data subsets, each with 9000 data points out of the full set of 9983 samples, and apply the iBCS algorithm to each dataset. Then, we retain only the basis polynomials that are selected in *each* of the 10 replica runs. As Fig. 16(a) illustrates, out of more than 8000 basis terms that appeared at least once, only a small subset of 69 terms is consistently picked. Figure 16(b) shows that the coefficients associated with the retained terms do not fluctuate much, confirming the robustness of this truncated basis set. Subsequently, we apply a single, noniterative BCS step using all 9983 data points and this truncated basis set in order to find PC coefficients corresponding to these bases. This last application of BCS uses the default stopping criterion $\epsilon = 10^{-9}$ and does not remove any more basis terms.

Figure 17 demonstrates a typical outcome of the algorithm, except for the fact that we set aside 983 validation points merely for illustration and use the rest of the samples and the truncated basis set for the surrogate construction. The piecewise-PC surrogate is indicated by the red dots and the samples are ordered according to the values of the surrogate. The gray region corresponds to the $5\% - 95\%$ quantiles of the posterior predictive distribution. The posterior

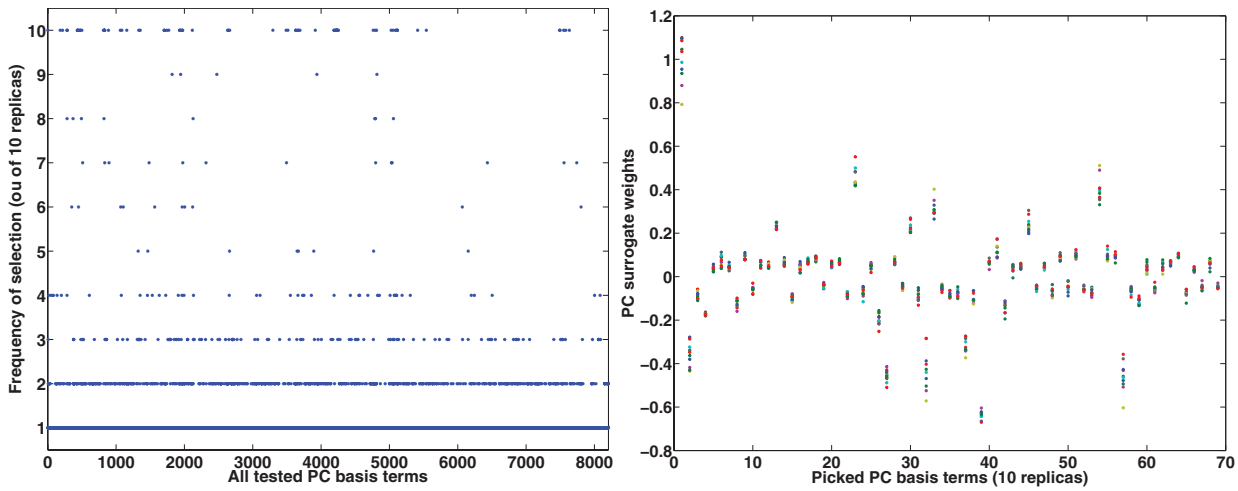


FIG. 16: The left plot shows the frequencies that each basis term appears in one of the 10 replica simulations (only the bases that appeared at least once are shown). The order of the basis terms is irrelevant. The right plot illustrates the weights across all 10 replicas for the bases that have been selected all 10 times. The small variability suggests that these bases and the associated weights are robust.

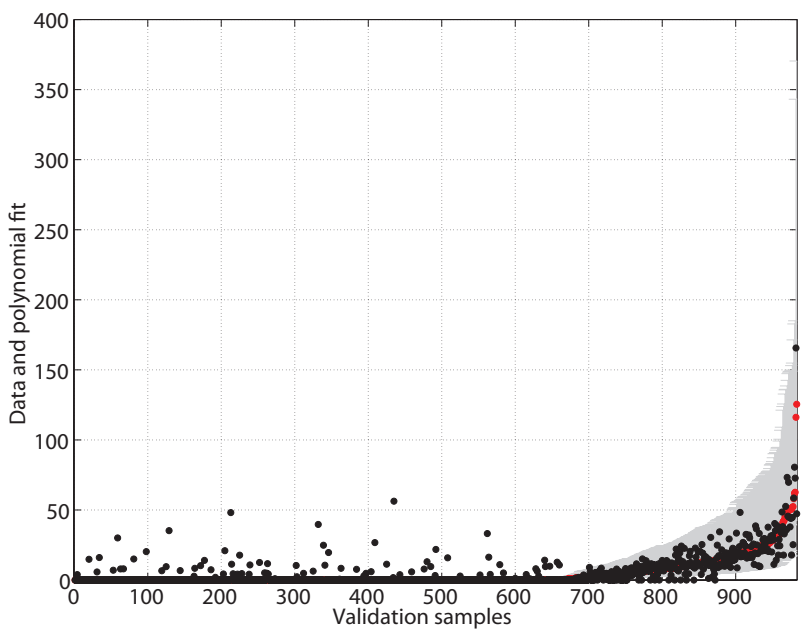


FIG. 17: Comparison between a validation dataset and the corresponding piecewise-PC model predictions. The 80-dimensional input is ordered according to the mean of the predictions (red dots). The gray region indicates $5\% - 95\%$ confidence region for the predicted values according to the posterior predictive distribution, while the black dots show the corresponding CLM outputs. Note that about 15% points are misclassified.

predictive distribution is a combination of the discrepancy σ and the "pushed-forward" posterior, i.e., the distribution that arises for the output, given the posterior distribution on the PC coefficients. Thus, since the coefficient vector $\boldsymbol{c}$ of a PC expansion $g_{\boldsymbol{c}}(\boldsymbol{\xi}) = \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\xi})$ has a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance

matrix $\boldsymbol{\Sigma}$, see (21), then the expansion itself is a Gaussian process with mean $m(\boldsymbol{\xi})$ and covariance function $C(\boldsymbol{\xi}, \boldsymbol{\xi}')$ from (23) and (24), respectively. The posterior predictive distribution at each fixed $\boldsymbol{\xi}^*$ will therefore be a Gaussian with mean $m(\boldsymbol{\xi}^*)$ and variance $\sigma^2 + C(\boldsymbol{\xi}^*, \boldsymbol{\xi}^*)$. Indeed, by definition [48], the posterior predictive distribution is the distribution of a hypothetical new data point $f^* = f(\boldsymbol{\xi}^*)$ given the current dataset of model simulations, marginalized over the PC parameter posterior distribution, i.e.,

$$\begin{aligned} p_{\text{pp}}(f^*) &= \int p(f^*|\mathcal{D}, \boldsymbol{c}) q(\boldsymbol{c}) d\boldsymbol{c} \\ &= \int \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(f^* - g_{\boldsymbol{c}}(\boldsymbol{\xi}^*))^2}{2\sigma^2}\right) \frac{1}{(2\pi)^{K/2}|\boldsymbol{\Sigma}|} \exp\left(-\frac{1}{2}(\boldsymbol{c} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{c} - \boldsymbol{\mu})\right) d\boldsymbol{c}, \end{aligned} \tag{36}$$

which, after some rearranging, allows integrating $\boldsymbol{c}$ out, leading to

$$p_{\text{pp}}(f^*) \propto \exp\left(-\frac{1}{2(\sigma^2 + C(\boldsymbol{\xi}^*, \boldsymbol{\xi}^*))}(f^* - m(\boldsymbol{\xi}^*))^2\right), \tag{37}$$

where $m(\boldsymbol{\xi})$ and $C(\boldsymbol{\xi}, \boldsymbol{\xi}')$ are defined in (23) and (24).

However, note that the surrogate was built for the logarithm of the observable, hence the posterior predictive distribution, the quantiles of which are illustrated in Fig. 17, is lognormal on the points that are classified as vegetation-growth points. The posterior predictive distribution can further be invoked in probabilistic calibration studies, as it essentially encapsulates probabilistic prediction of the CLM output at input value $\boldsymbol{\xi}$.

The misclassification of 15% of points is a direct consequence of not having enough training simulations. With less than $10{,}000$ points in an 80D space, it is difficult to expect an improvement in the value of the misclassification factor $f \approx 0.15$. A simple order-of-magnitude argument is useful to illustrate this point. Consider a hypothetical approximation $z^*(\boldsymbol{\xi})$ to a function $g(\boldsymbol{\xi})$ that is within a relative error ϵ at each point, i.e., $|z^*(\boldsymbol{\xi}) - z(\boldsymbol{\xi})|/|z(\boldsymbol{\xi})| \approx \epsilon$, for all values of $\boldsymbol{\xi}$. This leads to $E_v \approx \epsilon$ according to (25). Misclassification, however, introduces fN terms that are of $\mathcal{O}(1)$ in the numerator of (25). Taking into account the normalization of the L_2 norm, one arrives at the estimate $E_v \approx \sqrt{f}$. Therefore, we note that the relative L_2 error will at least be $\mathcal{O}(\sqrt{f}) \approx 0.4$ purely due to misclassification, hence there is not much use of invoking E_v to evaluate the predictive fidelity of the model. One can compute posterior predictive scores instead [49], to measure how predictive the resulting PC surrogate is. In principle, the uncertainty range of the posterior predictive distribution should reflect the spread in the data. In our case, it does, as can be seen from Fig. 17, except for the points that are misclassified.

The resulting surrogate allows the extraction of global sensitivity information from the piecewise PC representation. Having built the PC-based surrogate over a range of variation of input parameters, one can extract variance-based sensitivity information. The main effect sensitivity indices S_i are defined as

$$S_i = \frac{\text{Var}[\mathbb{E}(g_{\boldsymbol{c}}(\boldsymbol{\xi})|\xi_i)]}{\text{Var}[g_{\boldsymbol{c}}(\boldsymbol{\xi})]}, \tag{38}$$

for $i = 1, \ldots, d$, while the joint sensitivity indices S_{ij} are

$$S_{ij} = \frac{\text{Var}[\mathbb{E}(g_{\boldsymbol{c}}(\boldsymbol{\xi})|\xi_i, \xi_j)]}{\text{Var}[g_{\boldsymbol{c}}(\boldsymbol{\xi})]} - S_i - S_j, \tag{39}$$

for $i, j = 1, \ldots, d$. These are the so-called Sobol indices [3, 4]. While the posterior distributions for the PC modes $\boldsymbol{c}$ are available, we rely on their mean estimates for the computation of these indices, noting that one can in principle associate uncertainties with the sensitivity indices by "pushing" the posterior distribution of $\boldsymbol{c}$ forward through formulas (38) and (39). The variances in the numerators of (38) and (39) are with respect to the fixed variables ξ_i or (ξ_i, ξ_j), while the expectations are with respect to the rest of the variables. The sensitivity index S_i can be interpreted as the fraction of the variance in the output that can be attributed to the ith input only, while S_{ij} is the variance fraction that is due to the joint contribution of ith and jth inputs only. While there are exact formulas for extracting such sensitivity information from a plain PC expansion $\sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\xi})$, in our context we needed to resort to Monte Carlo estimates

of (38) and (39) because of the log-transform as well as a classification step in (35). Indeed, the Sobol indices (38) and (39) can be written in integral forms, but these integrals will not be analytically tractable for irregular domains that appear in the input parameter space due to classification. Monte Carlo estimates themselves are enhanced by techniques described in [50] allowing efficient estimation of conditional variances in the sensitivity formulae above.

The sensitivities are illustrated in Fig. 18 and are computed in the following way. One first generates two sets of M independent, uniform input samples. Besides these two sets, the piecewise-PC surrogate model is evaluated at sample sets that are obtained when one replaces one (for main effect sensitivities) or two (for joint sensitivities) dimensions of the first set by the corresponding values of the second set [50]. This leads to k more sets, each with M samples, where k is the number of indices to be computed. Therefore, a total of $(k+2)M$ surrogate model evaluations are required. For the main effect sensitivity indices $k = d$ and $M = 100{,}000$. However, even with such a large number of samples, only the sensitivity indices with larger magnitudes are computed accurately. Therefore, we down-select the input parameters based on the main effect indices, retaining only the ones above 10^{-3}, and compute the joint sensitivity indices only for pairs involving the most important parameters, i.e., $k = d'(d'-1)/2$, where $d' = 18$ in this particular case of output LAI. Finally, we used $M = 10{,}000$ independent Monte Carlo samples for the joint effect sensitivities. We note that for all outputs the main and joint effect sensitivities summed up to approximately 0.9, indicating that most of the dependences are captured within these indices.

Figures 19 and 20 illustrate results for four more outputs of interest from Table 3. These outputs are quite correlated with each other, therefore it is expected that the most important input parameters are generally shared between all the outputs. Indeed, as Fig. 21 demonstrates, the more important parameters are generally shared between the outputs. There are only 21 input parameters that have main effect sensitivity indices for at least one output over a threshold 10^{-3}. This provides a significant dimensionality reduction mechanism and can serve as a basis for further, more accurate studies on lower-dimensional input parameter space.

Quantifying the sensitivity of the model outputs to the input parameters yields important insights about CLM. We find that the three most important parameters for all QoIs are leafcn (leaf carbon to nitrogen ratio), flnr (fraction of

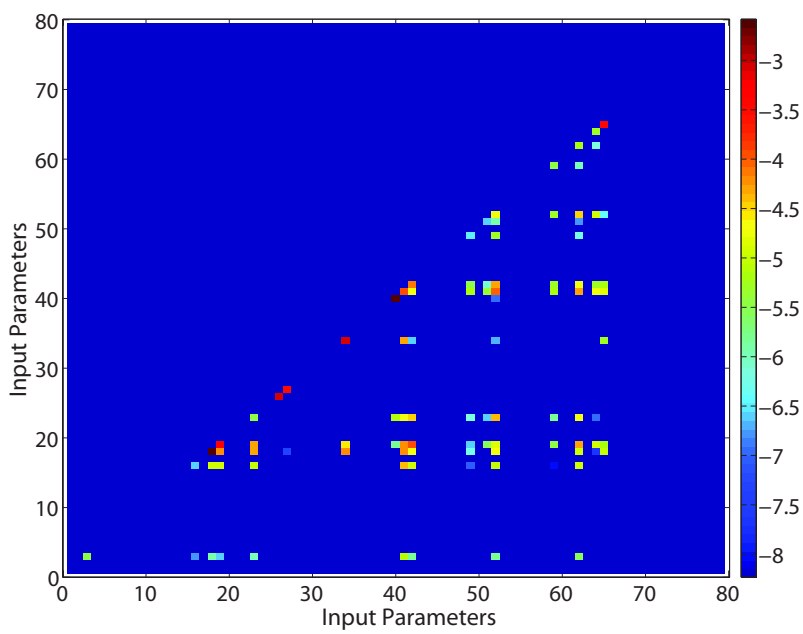
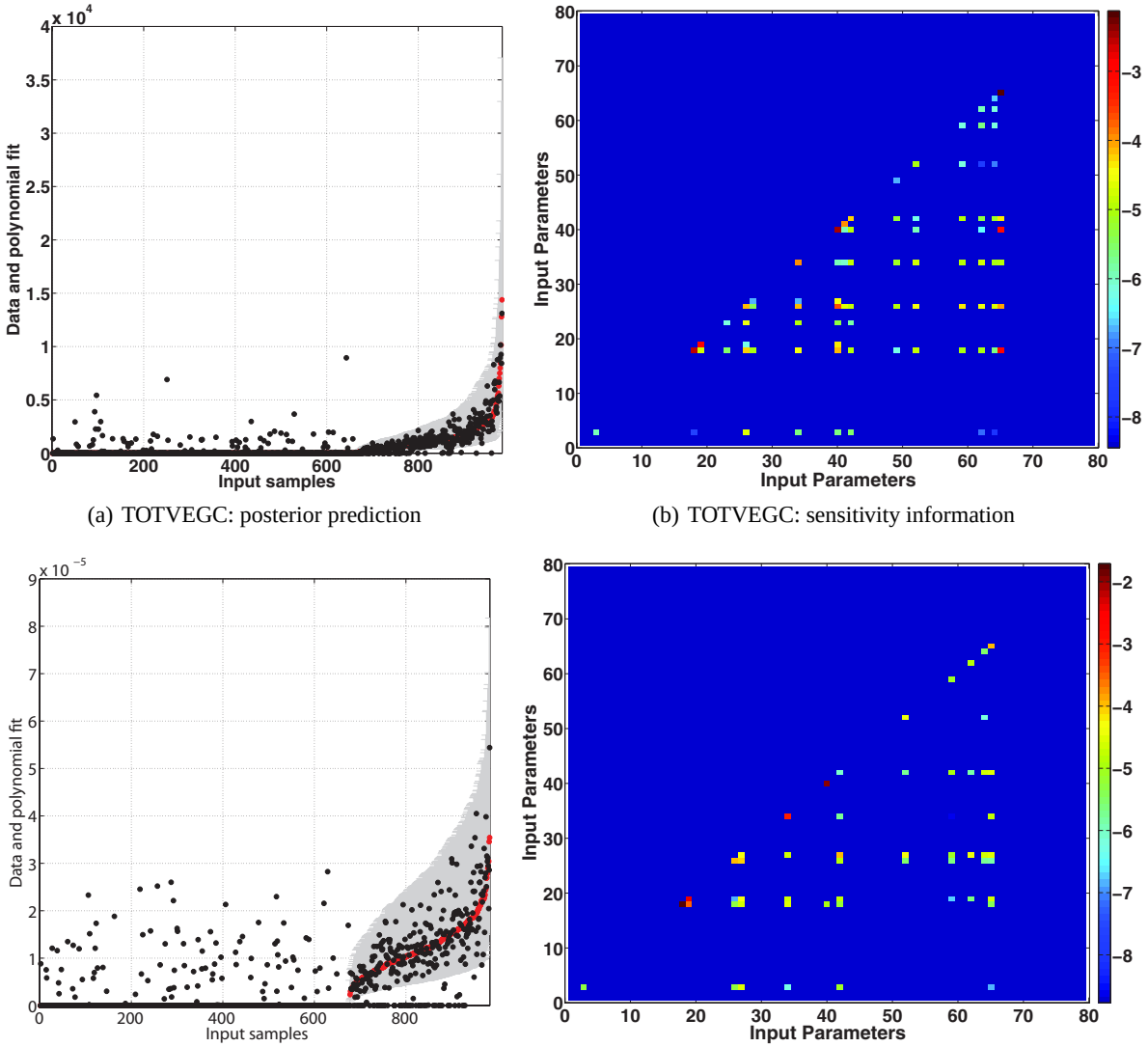


FIG. 18: Variance-based sensitivity information for LAI based on a Monte Carlo sampling of the piecewise-PC surrogate (35). The diagonal terms correspond to the main effect sensitivity indices, while the off-diagonal entries (computed only for variables with non-negligible main effect sensitivity) correspond to joint sensitivity indices. The color bar corresponds to the natural logarithmic axis.



(a) TOTVEGC: posterior prediction

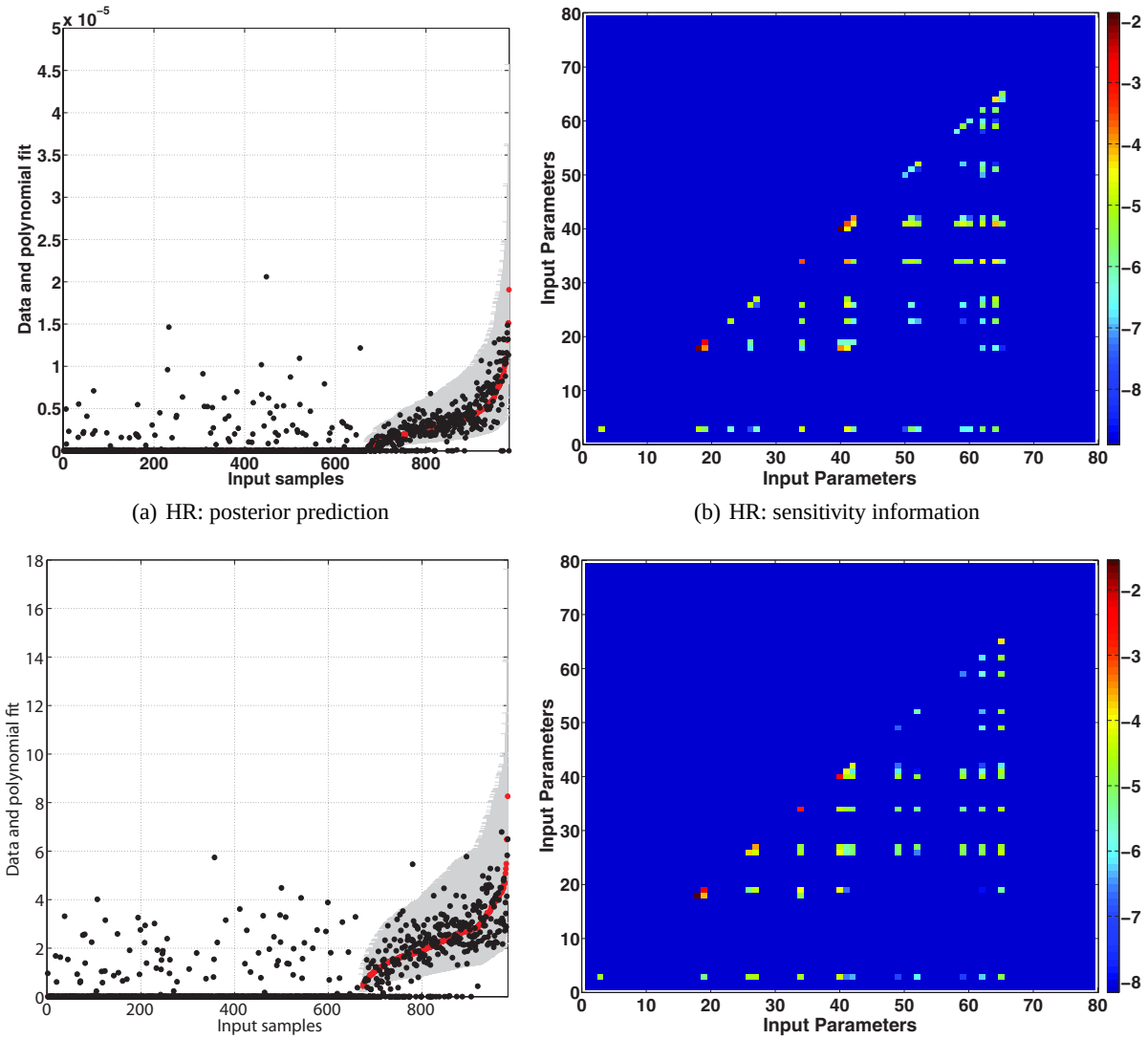
(b) TOTVEGC: sensitivity information

(c) GPP: posterior prediction

(d) GPP: sensitivity information

FIG. 19: Posterior predictive and sensitivity analysis results for outputs TOTVEGC and GPP. The color bars correspond to the natural logarithmic axis.

leaf nitrogen in RuBisCO, the primary enzyme used in photosynthesis), and rc_npool (a resistance term or buffering factor for the internal plant nitrogen pool). These parameters are primary controls on plant photosynthetic capacity for a given leaf area, with leafcn and flnr basically controlling the amount of chlorophyll and the rc_npool parameter controlling the nitrogen limitation factor. The parameters froot_leaf and stem_leaf control the allocation of photosynthate to the fine root and stem relative to leaf carbon, respectively, and are important drivers of the equilibrium LAI. Leaf longevity leaf_long also plays a key role in equilibrium LAI by controlling the balance between growth and turnover. Finally, r_mort (plant mortality rate) is a strong control on equilibrium total vegetation carbon (TOTVEGC) and LAI. The mortality rate represents plant death due to natural causes other than fire or harvest, which are represented separately, and controls the amount of biomass that is allowed to build up in the system before turning over to litter.



(a) HR: posterior prediction

(b) HR: sensitivity information

(c) FPSN: posterior prediction

(d) FPSN: sensitivity information

FIG. 20: Posterior predictive and sensitivity analysis results for outputs HR and FPSN. The color bars correspond to the natural logarithmic axis.

7. DISCUSSION AND FUTURE WORK

In this work, we have implemented a polynomial chaos (PC) surrogate construction that is specifically tuned for high-dimensional problems. Inspired by the community land model (CLM) that takes very high-dimensional input and is computationally expensive, we presented a sparse learning technique to select the most important basis terms in PC spectral expansions that represent the input-output relationship. While the idea of compressive sensing to find sparse representations is not new and is extensively used in the image processing community, it has only recently gained momentum in the context of spectral expansions. The availability of only a sparse set of model simulations motivates the use of Bayesian machinery in order to derive robust predictions with uncertainties associated with lack of knowledge. Starting from the Bayesian compressive sensing (BCS) methodology we extended its use to polynomial bases,

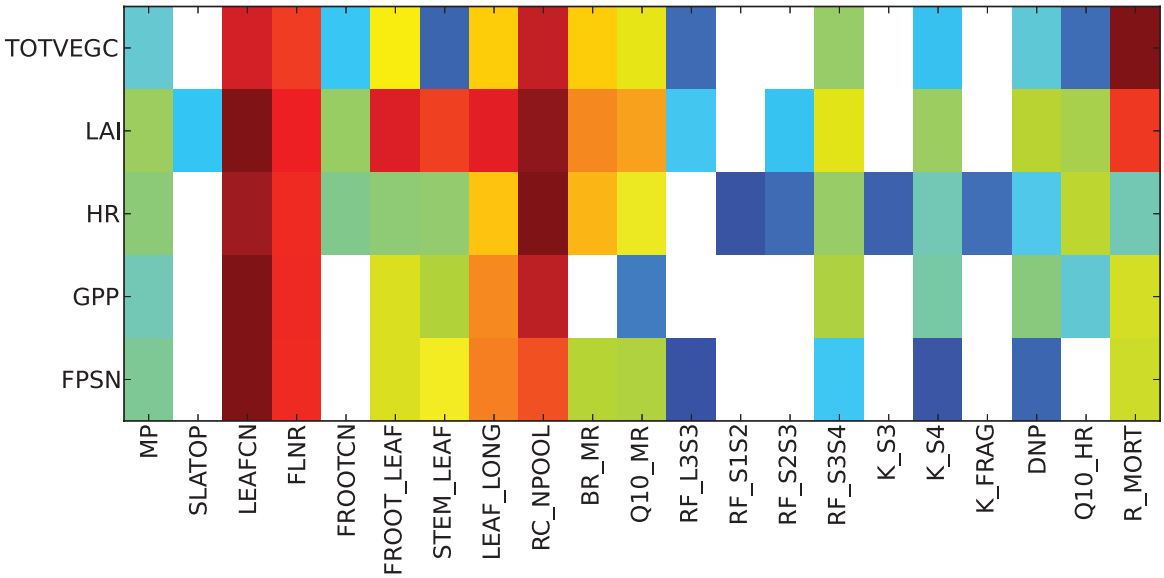


FIG. 21: Main effect sensitivity indices for five outputs of interest. For each output, a log scaling is applied and the sensitivities are shifted so that the maximal value of the log sensitivity is zero, for clarity of the color map presentation. The input parameters that have sensitivities above a threshold value 10^{-3} for at least one of the outputs are shown.

as well as enhanced it with an iterative procedure that allows dimensionality reduction and accuracy/order improvement within each iteration. Another characteristic feature of the forward function, i.e., the input-output relationship, of CLM, is the presence of two qualitatively different regimes—dead and live vegetation regions. In order to accommodate such behavior with smooth, polynomial bases, we developed a clustering- and classification-based piecewise PC construction.

We concluded the CLM analysis by extracting variance-based global sensitivity information from the piecewise-PC model. Equipped with such knowledge, our future work will be focused on building input ensembles on lower-dimensional input spaces that will carry more information about the forward model itself. The surrogates derived from these ensembles will have higher predictive fidelity and will eventually serve for calibration purposes.

Note that the BCS algorithm, although heavily relying on the evidence maximization, does not guard completely against overfitting and may not truly maximize the evidence. However, validation error measures and proper tuning of the BCS stopping criterion help to construct a sparse model that best explains the available sparse data. Whether the fixed-hyperparameter formulation is a good approximation to the full hierarchical Bayes formulation with Laplace priors is outside the scope of the paper and is a matter of future work. We are exploring the relationship between model selection with BCS and Bayes factors (BF) that rely on the exact evidence maximization and the principle of parsimony. While the basis addition and robust model selection for the CLM includes some *ad hoc*, problem-dependent analysis, the general workflow we propose here allows building sparse models semiautomatically given a set of only few training simulations in a high-dimensional input space.

ACKNOWLEDGMENTS

This work was supported by the US Department of Energy, Office of Science, under the project “Climate Science for a Sustainable Energy Future,” funded by the Biological and Environmental Research (BER) program. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the US Department of Energy’s National Nuclear Security Administration under contract DE-AC04-94-AL85000.

REFERENCES

1. Solomon, S., Qin, D., Manning, M., Chen, Z., Marquis, M., Averyt, K., Tignor, M., and Miller, H., Climate Change 2007: The Physical Science Basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change (IPCC), Cambridge, UK, 2007.
2. Saltelli, A., Tarantola, S., Campolongo, F., and Ratto, M., *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*, Wiley, New York, 2004.
3. Sobol, I. M., Sensitivity estimates for nonlinear mathematical models, *Math. Modeling and Comput. Exper.*, 1:407–414, 1993.
4. Campolongo, F., Saltelli, A., Sørensen, T., and Tarantola, S., Hitchhiker's guide to sensitivity analysis, In: Saltelli, A., Chan, K., and Scott, E. (Eds.), *Sensitivity Analysis*., Wiley, Chichester, 2000.
5. Tychonoff, A. N. and Arsenin, V. Y., *Solution of Ill-Posed Problems*, Winston & Sons, Washington, DC, 1977.
6. Tibshirani, R., Regression shrinkage and selection via the Lasso, *J. R. Stat. Soc. Ser. B (Methodological)*, 58:267–288, 1996.
7. Blatman, G. and Sudret, B., Sparse polynomial chaos expansions and adaptive stochastic finite elements using a regression approach, *Comptes Rendus Mécanique*, 336(6):518–523, 2008.
8. Blatman, G. and Sudret, B., Adaptive sparse polynomial chaos expansion based on least angle regression, *J. Comput. Phys.*, 230(6):2345–2367, 2011.
9. Doostan, A. and Owhadi, H., A non-adapted sparse approximation of PDEs with stochastic inputs, *J. Comput. Phys.*, 230(8):3015–3034, 2011.
10. Bishop, C. M. and Tipping, M. E., *Advances in Learning Theory: Methods, Models and Applications*, chapter Bayesian Regression and Classification, Amsterdam, IOS Press, 2003.
11. Park, T. and Casella, G., The Bayesian Lasso, *J. Am. Stat. Assoc.*, 103(482):681–686, 2008.
12. Ji, S., Xue, Y., and Carin, L., Bayesian compressive sensing, *IEEE Trans. Signal Proc.*, 56(6):2346–2356, 2008.
13. Tipping, M. E., Sparse Bayesian learning and the relevance vector machine, *J. Mach. Lear. Res.*, 1:211–244, 2001.
14. Bilionis, I. and Zabaras, N., Multidimensional adaptive relevance vector machines for uncertainty quantification, *SIAM J. Sci. Comput.*, 34(6):B881–B908, 2012.
15. Babacan, S., Molina, R., and Katsaggelos, A., Bayesian compressive sensing using Laplace priors, *IEEE Trans. Image Proces.*, 19(1):53–63, 2010.
16. Le Maître, O. and Knio, O., *Spectral Methods for Uncertainty Quantification*, Springer, New York, 2010.
17. Le Maître, O., Knio, O., Debusschere, B., Najm, H., and Ghanem, R., A multigrid solver for two-dimensional stochastic diffusion equations, *Comp. Meth. App. Mech. Eng.*, 192:4723–4744, 2003.
18. Le Maître, O., Najm, H., Ghanem, R., and Knio, O., Multi-resolution analysis of Wiener-type uncertainty propagation schemes, *J. Comput. Phys.*, 197:502–531, 2004.
19. Le Maître, O., Najm, H., Pébay, P., Ghanem, R., and Knio, O., Multi-resolution-analysis scheme for uncertainty quantification in chemical systems, *SIAM J. Sci. Comput.*, 29(2):864–889, 2007.
20. Sargsyan, K., Safta, C., Debusschere, B., and Najm, H., Uncertainty quantification given discontinuous model response and a limited number of model runs, *SIAM J. Sci. Comput.*, 34(1):B44–B64, 2012.
21. Daux, C., Moës, N., Dolbow, J., Sukumar, N., and Belytschko, T., Arbitrary branched and intersecting cracks with the extended finite element method, *Int. J. Numer. Meth. Eng.*, 48:1741–1760, 2000.
22. Fleming, M., Chu, Y., Moran, B., and Belytschko, T., Arbitrary branched and intersecting cracks with the extended finite element method, *Int. J. Numer. Meth. Eng.*, 40:1483–1504, 1997.
23. Nouy, A. and Clement, A., Extended Stochastic Finite Element Method for the numerical simulation of heterogenuous materials with random material interfaces, *Int. J. Numer. Meth. Eng.*, 83(10):1312–1344, 2010.
24. Ghosh, D. and Ghanem, R., Stochastic convergence acceleration through basis enrichment of polynomial chaos expansions, *Int. J. Numer. Meth. Eng.*, 73:162–174, 2008.
25. Breiman, L., Random forests, *Mach. Lear.*, 45(1):5–32, 2001.
26. Wiener, N., The homogeneous chaos, *Am. J. Math.*, 60:897–936, 1938.

27. Ghanem, R. and Spanos, P., *Stochastic Finite Elements: A Spectral Approach*, Springer Verlag, New York, 1991.
28. Xiu, D. and Karniadakis, G., The Wiener-Askey polynomial chaos for stochastic differential equations, *SIAM J. Sci. Comput.*, 24(2):619–644, 2002.
29. Abramowitz, M. and Stegun, I., *Handbook of Mathematical Functions*, Dover, New York, 1970.
30. Rabitz, H. and Alis, O. F., General foundations of high-dimensional model representations, *J. Math. Chem.*, 25:197–233, 1999.
31. Sivia, D., *Data Analysis: A Bayesian Tutorial*, Oxford Science, Oxford, 1996.
32. Jaynes, E., *Probability Theory: The Logic of Science*, G. L. Bretthorst, Ed., Cambridge University Press, Cambridge, UK, 2003.
33. Candès, E., Romberg, J., and Tao, T., Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, *IEEE Trans. Inf. Theory*, 52(2):489–509, 2006.
34. Hans, C., Bayesian Lasso regression, *Biometrica*, 96:835–845, 2009.
35. Tipping, M. E. and Faul, A. C., Fast marginal likelihood maximisation for sparse Bayesian models, in *Proc. of the Ninth Int. Workshop on Artificial Intelligence and Statistics*, Bishop, C. M. and Frey, J. (Eds.), Key West, FL, Jan. 3–6, 2003.
36. Genz, A., Fully symmetric interpolatory rules for multiple integrals, *SIAM J. Num. Anal.*, 23(6):1273–1283, 1996.
37. Genz, A. and Keister, B. D., Fully symmetric interpolatory rules for multiple integrals over infinite regions with gaussian weight, *J. Comput. Appl. Math.*, 71(2):299–309, Jul 1996.
38. Chantrasmi, T., Padé-Legendre method for uncertainty quantification with fluid dynamics applications, PhD thesis, Stanford University, 2011.
39. Thornton, P., Lamarque, J., Rosenbloom, N., and Mahowald, N., Influence of carbon-nitrogen cycle coupling on land model response to CO_2 fertilization and climate variability, *Global Biogeochem. Cycles*, 21(4), 2007.
40. Gent, P., Danabasoglu, G., Donner, L., Holland, M., Hunke, E., Jayne, S., Lawrence, D., Neale, R., Rasch, P., Vertenstein, M., Worley, P., Yang, Z.-L., and Zhang, M., The community climate system model version 4, *J. Climate*, 24(19):4973–4991, 2011.
41. Mao, J., Thornton, P., Shi, X., Zhao, M., and Post, W., Remote sensing evaluation of CLM4 GPP for the period 2000-09, *J. Climate*, 25(15):5327–5342, 2012.
42. Shi, X., Mao, J., Thornton, P., Hoffman, F., and Post, W., The impact of climate, CO_2, nitrogen deposition and land use change on simulated contemporary global river flow, *Geophys. Res. Lett.*, 38(8), 2011.
43. Randerson, J., Hoffman, F., Thornton, P., Mahowald, N., Lindsay, K., Lee, Y., Nevison, C., Doney, S., Bonan, G., Stöckli, R., Covey, C., Running, S., and Fung, I., Systematic assessment of terrestrial biogeochemistry in coupled climate-carbon models, *Global Change Biol.*, 15(10):2462–2484, 2009.
44. Monson, R., Turnipseed, A., Sparks, J., Harley, P., Scott-Denton, L., Sparks, K., and Huxman, T., Carbon sequestration in a high-elevation, subalpine forest, *Global Change Biol.*, 8(5):459–478, 2002.
45. Rosenblatt, M., Remarks on a multivariate transformation, *Ann. Math. Stat.*, 23(3):470–472, 1952.
46. Shorter, J. A., Ip, P. C., and Rabitz, H. A., An efficient chemical kinetics solver using high dimensional model representation, *J. Phys. Chem. A*, 103(36):7192–7198, 1999.
47. Li, G., Rosenthal, C., and Rabitz, H., High dimensional model representations, *J. Phys. Chem. A*, 105:7765–7777, 2001.
48. Lynch, S. and Western, B., Bayesian posterior predictive checks for complex models, *Sociol. Methods Res.*, 32(3):301–335, 2004.
49. Gneiting, T. and Raftery, A., Strictly proper scoring rules, prediction, and estimation, *J. Am. Stat. Assoc.*, 102:359–378, 2007.
50. Saltelli, A., Making best use of model evaluations to compute sensitivity indices, *Comput. Phys. Commun.*, 145:280–297, 2002.