SIAM J. SCI. COMPUT.
Vol. 34, No. 6, pp. B881–B908

MULTIDIMENSIONAL ADAPTIVE RELEVANCE VECTOR MACHINES FOR UNCERTAINTY QUANTIFICATION*

ILIAS BILIONIS† AND NICHOLAS ZABARAS‡

Abstract. We develop a Bayesian uncertainty quantification framework using a local binary tree surrogate model that is able to make use of arbitrary Bayesian regression methods. The tree is adaptively constructed using information about the sensitivity of the response and is biased by the underlying input probability distribution. The local Bayesian regressions are based on a reformulation of the relevance vector machine model that accounts for the multiple output dimensions. A fast algorithm for training the local models is provided. The methodology is demonstrated with examples in the solution of stochastic differential equations.

Key words. relevance vector machine, sparse Bayesian learning, multioutput, adaptivity, uncertainty quantification

AMS subject classifications. 62F15, 65C60, 60H15, 62K20

DOI. 10.1137/120861345

1. Introduction.

Uncertainty quantification (UQ) is a field of great importance in practically all engineering tasks. Physical models require as input certain parameters, such as physical constants, equations of state, geometric specification of objects, boundary conditions, initial conditions, and so on. In general, exact knowledge of these quantities is impossible either due to measurement errors or because they are truly random. As a consequence, both the input parameters and the physical responses have to be modeled as random variables. The goal of UQ is to study the propagation of uncertainty from the input parameter space to the response space. The most celebrated method for the solution of UQ problems is the Monte Carlo (MC) method. MC is widely accepted because it can uncover the complete statistics of the solution, while having a convergence rate that is (remarkably) independent of the input dimension. Nevertheless, it quickly becomes inefficient in high-dimensional and computationally intensive problems, where only a few samples can be observed. Such difficulties have been (partially) alleviated by improved sampling techniques such as Latin hypercube sampling [25] and multilevel MC [16, 17].

Another approach to UQ is the so-called stochastic finite element method [15]. It involves the projection of the response on a space spanned by orthogonal polynomials

*Submitted to the journal's Computational Methods in Science and Engineering section January 4, 2012; accepted for publication (in revised form) September 24, 2012; published electronically December 13, 2012. This research was supported by an OSD/AFOSR MURI09 award on uncertainty quantification, the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research, and the Computational Mathematics program of the National Science Foundation (NSF) (awards DMS-0809062 and DMS-1214282). This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. Additional computing resources were provided by the NSF through TeraGrid resources provided by NCSA under grant number TG-DMS090007.

http://www.siam.org/journals/sisc/34-6/86134.html

†Center for Applied Mathematics, Cornell University, Ithaca, NY 14853 and Materials Process Design and Control Laboratory, Sibley School of Mechanical and Aerospace Engineering, 101 Frank H. T. Rhodes Hall, Cornell University, Ithaca, NY 14853-3801 (ib227@cornell.edu).

‡Corresponding Author. Materials Process Design and Control Laboratory, Sibley School of Mechanical and Aerospace Engineering, 101 Frank H. T. Rhodes Hall, Cornell University, Ithaca, NY 14853-3801 and Center for Applied Mathematics, Cornell University, Ithaca, NY 14853 (zabaras@cornell.edu).

of the random variables and the solution of a system of coupled deterministic equations involving the coefficients of the expansion in these polynomials. The scheme was originally developed for Gaussian random variables that correspond to Hermite polynomials (polynomial chaos (PC)). It was later generalized to include other types of random variables (generalized PC (gPC)) [42]. Due to the global support of the polynomials used, gPC suffers from the well-known Gibbs phenomenon in the presence of discontinuities in the random space. The multielement generalized polynomial chaos (ME-gPC) method [39, 40] was introduced in order to address exactly this issue. The idea of the multielement (ME) approach is to decompose the stochastic space in disjoint elements and then employ gPC on each element. However, the coupled nature of the equations that determine the coefficients of the polynomials make the application of the method to high input dimensions extremely difficult (*curse of dimensionality*).

Throughout the paper, we assume that we have at hand a well-established computer code that emulates the physical system. In fact, we will investigate the propagation of uncertainty from the input of the computer code to the output by learning the response surface using well-selected observations. Any modeling or discretization error will be ignored in this study. The so-called stochastic collocation methods have been designed to deal with this situation. The response is represented as an interpolative polynomial of the random input constructed by calls to the computer code at specific input points. However, the construction of the set of interpolation points is nontrivial, especially in high-dimensional settings. In [1], a Galerkin-based approximation was introduced alongside a collocation scheme based on a tensor product rule using one-dimensional Gauss quadrature points. Despite its appeal, the method scales badly with the number of random input dimensions. Alternatively, sparse grids (SGs) [21, 13] based on the Smolyak algorithm [36] have a weaker dependence on the input dimensionality. In [41, 44, 32], the Smolyak algorithm is employed to build SG interpolants in high-dimensional input spaces based on Lagrange interpolation polynomials. Similarly to gPC, such methods also fail to capture local features of the response. From the above discussion, it is apparent that discontinuities in the stochastic space must be dealt with using a basis with local support. In [30], the authors developed an adaptive version of SG collocation (SGC) based on localized hat functions called adaptive SGC (ASGC). ASGC is able to refine the SG only in important regions of the stochastic space, e.g., near a discontinuity. Nevertheless, the piecewise-linear nature of the scheme performs poorly when only a few samples are used, while adverse functions can trick the adaptive algorithm into stopping without converging.

Highly sophisticated computer codes modeling real-life phenomena (such as weather, ocean waves, earthquakes, etc.) might take hours or even days to complete a single run in massively parallel systems. Therefore, we are necessarily limited to observing only a few realizations. Motivated by this situation, we would like to consider the problems of (1) selecting the most informative observations and (2) quantifying the uncertainty in the prediction of the statistics. From the above-mentioned methods, ASGC addresses only problem (1), albeit in an ad hoc manner. In order to deal with (1) and (2) in a principled, information-theoretic way, a fully Bayesian framework is necessary. In such a framework, the Bayesian nature of the predictive distribution of the response surfaces induces a predictive distribution of the desired statistics, capturing in this way the epistemic uncertainty introduced by using the surrogate instead of the deterministic solver for their calculation. Similar ideas were developed in [14], where the authors approach this problem by inferring the coefficients of a PC expansion using Bayesian techniques. Furthermore, localized/nonstationary

features are the rule rather than the exception in most UQ problems. For example, discontinuities in the response might occur as a result of a bifurcation when the input crosses a given hypersurface [40, 30]. As mentioned earlier, global methods have been shown to perform extremely poorly in these situations. On the other hand, it has been repeatedly demonstrated that a local approach is necessary in order to effectively capture discontinuities and localized features (ME-gPC [39, 40], ASGC [30]). It is therefore evident that a local approach to uncertainty propagation is required if capturing discontinuities is desired. Similarly to [19], in [3] we built a tree surrogate of Gaussian processes (GPs). The aforementioned work used attributes specific to GPs for the adaptation criteria. In this work, we extend our tree construction methodology to arbitrary input distributions and arbitrary local Bayesian regressions. As a replacement for GPs, we develop a multioutput version of the relevance vector machine (RVM) [38], which we call multioutput RVM (MRVM). RVM is a Bayesian sparse kernel technique for regression and classification that shares many characteristics with the well-established support vector machines (SVMs) [6]. Section 7.2 of [4] discusses the similarities and differences between SVM and RVM. Let us briefly mention that RVM diminishes several of the difficulties present in SVM; for example, (1) one does not have to use cross-validation techniques in order to choose the complexity parameters and (2) the basis functions employed in regression can be arbitrary instead of strictly positive definite kernels.

The beginning of section 2 provides some basic definitions and assumptions. In section 2.1, we introduce the MRVM model for arbitrary local basis functions, and in section 2.2, we provide a fast training algorithm based on the evidence approximation. Various derivations and specific details of the constituents of the algorithm can be found in Appendices A to D. In section 2.3, we make a specific choice for the basis functions, namely local square exponential (SE) kernels centered on top of each observed input point and a locally defined set of orthogonal polynomials. Sections 2.4 and 2.5 discuss how the local and global statistics, respectively, can be evaluated, while in section 2.6 we devise a scheme that allows us to sample from the predictive distribution of the statistics. In section 2.7, we introduce our tree construction methodology, which is largely independent of the specifics of the local Bayesian regression model. The method is demonstrated numerically for various UQ problems in section 3. Finally, we conclude in section 4.

2. Methodology. Let $\mathbf{X}$ represent the parameter space of a physical model and $p(\mathbf{x})$ a probability density function (PDF) defined on $\mathbf{X}$. $\mathbf{X}$ will be referred to as the *stochastic input space*. We assume that the stochastic problem has been formulated in such a way that $\mathbf{X}$ is a rectangle of $\mathbb{R}^K$ for some $K \geq 1$, that is, $\mathbf{X} = \times_{k=1}^{K}[a_k, b_k]$, with $-\infty \leq a_k < b_k \leq +\infty$ the upper and lower bounds of each dimension. Furthermore, we suppose that all dimensions of $\mathbf{X}$ are independent. This is just a convenient assumption present in many other UQ methodologies. If this assumption did not hold, then one would have to transform the input so that it did. Such a transformation always exists [34]. Finding it, however, is beyond the scope of the present work. Thus, we may write $p(\mathbf{x}) = \prod_{k=1}^{K} p_k(x_k)$, where p_k is the PDF pertaining to the kth input dimension.

Let us now consider the multioutput function $\mathbf{f} : \mathbf{X} \to \mathbb{R}^M$ representing the result of a computer code (deterministic solver) modeling a physical system; i.e., at a given input point $\mathbf{x} \in \mathbf{X}$, the predicted response of the system is $\mathbf{f}(\mathbf{x})$. We will write

$$\mathbf{f} = (f_1, \ldots, f_M)$$

and refer to f_r as the rth output of the response function, $r = 1, \ldots, M$. In this work, we will identify $\mathbf{f}(\mathbf{x})$ as the true response of an underlying physical system and we will ignore any modeling errors. The input probability distribution induces a probability distribution on the output. The UQ problem involves the calculation of the statistics of the output $\mathbf{y} = \mathbf{f}(\mathbf{x})$. Quantities of particular interest are the *q-moments* $\mathbf{m}^q = (m_1^q, \ldots, m_M^q)$, defined for $q \geq 1$ and $r = 1, \ldots, M$ by

$$m_r^q := \int_{\mathbf{X}} f_r^q(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}, \tag{2.1}$$

as well as functions of them. In particular, the *mean* $\mathbf{m} = (m_1, \ldots, m_M)$ is given by

$$m_r := m_r^1 = \int_{\mathbf{X}} f_r(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}, \tag{2.2}$$

and the *variance* $\mathbf{v} = (v_1, \ldots, v_M)$ is given by

$$v_r := \int_{\mathbf{X}} \left(f_r(\mathbf{x}) - m_r\right)^2 p(\mathbf{x}) d\mathbf{x} = m_r^2 - (m_r^1)^2. \tag{2.3}$$

The statistics will be calculated by interrogating a surrogate of $\mathbf{f} : \mathbf{X} \to \mathbb{R}^M$. This surrogate will be put together from local surrogates defined over $I \geq 1$ *elements* of the stochastic space $\mathbf{X}^i \subset \mathbf{X}$ such that

$$\mathbf{X} = \cup_{i=1}^I \mathbf{X}^i \text{ and } \operatorname{int}(\mathbf{X}^i) \cap \operatorname{int}(\mathbf{X}^j) = \emptyset \quad \forall i, j \in I, i \neq j, \tag{2.4}$$

where $\operatorname{int}(\mathbf{X}^i)$ denotes the interior of the set $\mathbf{X}^i$ under the usual Euclidean metric of $\mathbb{R}^K$. The response surface is correspondingly decomposed as

$$\mathbf{f}(\mathbf{x}) := \sum_{i=1}^I \mathbf{f}^i(\mathbf{x}) 1_{\mathbf{X}^i}(\mathbf{x}), \tag{2.5}$$

where $1_{\mathbf{X}^i}(\mathbf{x})$ is the indicator function of $\mathbf{X}^i$ and $\mathbf{f}^i(\cdot)$ is the restriction of $\mathbf{f}(\cdot)$ on $\mathbf{X}^i$. The local surrogates will be identified as MRVMs defined over the stochastic element $\mathbf{X}^i$. These MRVMs will be trained by observing $\mathbf{f}^i(\cdot)$. The MRVM model is outlined in section 2.1, and a training algorithm is provided in section 2.2. The predictive mean of the MRVMs will be used to derive semianalytic estimates of all moments $\mathbf{m}^q$ (sections 2.4 and 2.5). An addendum of the Bayesian treatment is the ability to provide error bars for the point estimates of the moments (section 2.6). This feature is absent from most current UQ methods. Our aim is to create a surrogate by making as few calls to the computer program as possible. This is achieved by adaptively decomposing the domain (tree construction) based on the predicted variability of the response function as well as biasing by the underlying input probability density $p(\mathbf{x})$ (section 2.7).

2.1. MRVM. We turn our focus to a single element of the stochastic space $\mathbf{X}^i \subset \mathbf{X}$ and discuss the construction of a local surrogate model based on some already observed data. The choice of the elements is the subject of section 2.7. All quantities introduced herein are local to the element $\mathbf{X}^i$. However, in order to avoid unnecessarily cumbersome notation, we do not explicitly show this dependence. We assume that we have observed a fixed number $N \geq 1$ of data points

$$\mathcal{D} := \left\{ \left(\mathbf{x}^{(n)}, \mathbf{y}^{(n)} = \mathbf{f}\left(\mathbf{x}^{(n)}\right) \right) \right\}_{n=1}^N, \ \mathbf{x}^{(n)} \sim p^i(\mathbf{x}), \tag{2.6}$$

where $p^i(\mathbf{x})$ is the conditional PDF on $\mathbf{X}^i$ defined by

$$p^i(\mathbf{x}) = \frac{p(\mathbf{x})}{P\left(\mathbf{X}^i\right)} 1_{\mathbf{X}^i}(\mathbf{x}) \tag{2.7}$$

and $P\left(\mathbf{X}^i\right)$ is the probability that a random input point falls in $\mathbf{X}^i$:

$$P\left(\mathbf{X}^i\right) := \int_{\mathbf{X}^i} p(\mathbf{x}) d\mathbf{x}. \tag{2.8}$$

Because we wish to model all outputs simultaneously, it is necessary to scale them so that they exhibit the same signal strength. Toward this end, let us introduce the *observed means*

$$\mu_{\text{obs},r} = \frac{1}{N} \sum_{n=1}^{N} y_r^{(n)} \tag{2.9}$$

and the *observed variances*

$$\sigma^2_{\text{obs},r} = \frac{1}{N-1} \sum_{n=1}^{N} (y_r - \mu_{\text{obs},r})^2 \tag{2.10}$$

for $r = 1, \ldots, M$ of the data $\mathcal{D}$. We will be modeling the *scaled response functions* $g_r : \mathbf{X}^i \to \mathbb{R}$ defined by

$$g_r(\mathbf{x}) = \frac{f_r(\mathbf{x}) - \mu_{\text{obs},r}}{\sigma_{\text{obs},r}}, \ r = 1, \ldots, M. \tag{2.11}$$

Obviously, this definition depends on the actual observations. However, we expect that if N is big or if the stochastic element under investigation is small, then it is a good approximation to the ideal scaling, i.e., zero mean and unit variance for all outputs.

We model the mean of each g_r as

$$\mu_{g_r}(\mathbf{x}) = \mathbf{w}_r^T \boldsymbol{\phi}(\mathbf{x}), \tag{2.12}$$

where $S \geq 1$, $\mathbf{w}_r = (w_{r1}, \ldots, w_{rS})^T$ is the vector of the weights for the rth output, and $\boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_S(\mathbf{x}))^T$ are some basis functions defined over the element $\mathbf{X}^i$. At this point, let us introduce the scaled version of the observations $\mathcal{D}_{\text{sc}} = \{(\mathbf{x}^{(n)}, \mathbf{z}^{(n)})\}_{n=1}^N$, where $\mathbf{z}^{(n)} = (z_1^{(n)}, \ldots, z_M^{(n)})^T$ with $\mathbf{z}_r^{(n)} = \frac{y_r^{(n)} - \mu_{\text{obs},r}}{\sigma_{\text{obs},r}}$, and let $\mathcal{D}_{\text{sc},r} = \{(\mathbf{x}^{(n)}, z_r^{(n)})\}_{n=1}^N$ be the observations in $\mathcal{D}_{\text{sc}}$ that pertain to the rth output dimension. We assume that, given the weights, the scaled observations $z_r = g_r(\mathbf{x})$ are normally distributed about $\mu_{g_r}(\mathbf{x})$ with inverse variance $\beta > 0$, i.e.,

$$p(z_r | \mathbf{x}, \mathbf{w}_r, \beta) = \mathcal{N}\left(z_r | \mu_{g_r}(\mathbf{x}), \beta^{-1}\right), r = 1, \ldots, M. \tag{2.13}$$

Under the assumption that the added noise is independent for each sample point, the *likelihood* of the scaled data $\mathcal{D}_{\text{sc},r}$ related to the rth output can be written as

$$p(\mathcal{D}_{\text{sc},r} | \mathbf{w}_r, \beta) = \prod_{n=1}^{N} p(z_r^{(n)} | \mathbf{x}^{(n)}, \mathbf{w}_r, \beta), \tag{2.14}$$

and assuming the output dimensions are conditionally independent given the weights, the likelihood of all the scaled observed data $\mathcal{D}_{\text{sc}}$ is

$$p(\mathcal{D}_{\text{sc}}|\mathbf{W},\beta) = \prod_{r=1}^{M} p(\mathcal{D}_{\text{sc},r}|\mathbf{w}_r,\beta), \tag{2.15}$$

where $\mathbf{W}$ is the $M \times S$ matrix whose rows are given by $\mathbf{w}_r$. We impose a *prior* probability density on each weight w_{rs} of the form

$$p(w_{rs}|\alpha_s) = (2\pi)^{-1/2}\alpha_s^{1/2}\exp\left\{-\frac{\alpha_s w_{rs}^2}{2}\right\},\ s = 1,\dots,S, \tag{2.16}$$

where $\alpha_s, s = 1,\dots,S$, are unknown *positive* hyperparameters (one for each basis function). We will collectively denote these by $\boldsymbol{\alpha} = (\alpha_1,\dots,\alpha_S)^T$. Notice that all output dimensions $r = 1,\dots,M$ share the same $\boldsymbol{\alpha}$. Furthermore, we assume that the weights of the rth output are conditionally independent given $\boldsymbol{\alpha}$:

$$p(\mathbf{w}_r|\boldsymbol{\alpha}) = \prod_{s=1}^{S} p(w_{rs}|\alpha_s),$$

as well as that all the $\mathbf{w}_r$'s are conditionally independent given $\boldsymbol{\alpha}$:

$$p(\mathbf{W}|\boldsymbol{\alpha}) = \prod_{r=1}^{M} p(\mathbf{w}_r|\boldsymbol{\alpha}).$$

Following [38], it is easy to show using matrix identities that the *posterior distribution* of the weights $\mathbf{w}_r$ is given by

$$p(\mathbf{w}_r|\mathcal{D}_{\text{sc},r},\boldsymbol{\alpha},\beta) = \mathcal{N}(\mathbf{w}_r|\boldsymbol{\mu}_r,\boldsymbol{\Sigma}), \tag{2.17}$$

where

$$\boldsymbol{\Sigma} = \left(\text{diag}(\boldsymbol{\alpha}) + \beta\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1} \text{ and } \boldsymbol{\mu}_r = \beta\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\mathbf{z}_r, \tag{2.18}$$

where $\text{diag}(\boldsymbol{\alpha})$ is a diagonal matrix with $\boldsymbol{\alpha}$ at the diagonal, $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\mathbf{x}^{(1)})\dots\boldsymbol{\phi}(\mathbf{x}^{(N)})]^T$ is the $N \times S$ *design matrix*, and $\mathbf{z}_r = (z_r^{(1)},\dots,z_r^{(N)})^T$ is the scaled version of the observed rth outputs. Under the *evidence approximation* (see Chap. 3 of [4]), point estimates of $\boldsymbol{\alpha}$ and β can be found by maximizing the *marginal likelihood* defined by

$$p(\mathcal{D}_{\text{sc}}|\boldsymbol{\alpha},\beta) = \int p(\mathcal{D}_{\text{sc}}|\mathbf{W},\beta)p(\mathbf{W}|\boldsymbol{\alpha})d\mathbf{W}. \tag{2.19}$$

We will calculate this integral analytically and provide an efficient algorithm for its maximization in section 2.2.

To conclude this section, let us give the predictive distribution of our model. Let $\boldsymbol{\alpha}$ and β be the hyperparameters that maximize (2.19). From [38] and scaling back to the original function $f_r(\mathbf{x})$, it is easy to show that the predictive distribution is

$$p(y_r|\mathbf{x},\mathcal{D}) = \mathcal{N}(y_r|\mu_{f_r}(\mathbf{x}),\sigma_{f_r}^2(\mathbf{x})), \tag{2.20}$$

where the *predictive mean* is given by

$$\mu_{f_r}(\mathbf{x}) = \sigma_{\text{obs},r}\boldsymbol{\mu}_r^T\boldsymbol{\phi}(\mathbf{x}) + \mu_{\text{obs},r} \tag{2.21}$$

and the *predictive variance* by

$$\sigma_{f_r}^2(\mathbf{x}) = \sigma_{\text{obs},r}^2\left(\beta^{-1} + \boldsymbol{\phi}(\mathbf{x})^T\boldsymbol{\Sigma}\boldsymbol{\phi}(\mathbf{x})\right). \tag{2.22}$$

Remark 1. *Treating the outputs jointly.* One can argue that treating the outputs completely independently (i.e., each output having its own set of hyperparameters) would raise the need for scaling them and would also increase the flexibility of the model. The main drawback of such an approach is that inference of the hyperparameters has to be carried out as many times as the number of outputs times the number of elements. In high-dimensional output scenarios, this approach becomes computationally intense. On the other hand, the joint treatment presented in this section requires the inference of the hyperparameters only once per element. We have chosen the joint approach mainly on the grounds that it is computationally efficient. In realistic applications, the outputs can be separated into groups according to their nature and a different MRVM can be used on each group. Extending our methodology to this case is straightforward.

Remark 2. *Posterior independence of the weights.* As observed above, for each given output $r = 1, \ldots, M$, the vector of weights $\mathbf{w}_r$ has nondiagonal covariance given by $\boldsymbol{\Sigma}$. On the other hand, the vectors of weights corresponding to different outputs are a posteriori independent. This is expected, since we have made no attempt to capture the correlation between the various outputs. On one hand, this is a drawback of the proposed model since it will definitely lead to suboptimal use of the available observations. On the other hand, it is a silently made assumption in practically all UQ methodologies with which we are familiar. We have chosen to address this issue in our work in [2].

2.2. Maximization of the marginal likelihood. We develop a maximization framework for the marginal likelihood with respect to the hyperparameters α and β. In [37], it is shown that the logarithm of the marginal likelihood of the rth output is given by

$$\mathcal{L}(\boldsymbol{\alpha}|\mathcal{D}_{\text{sc},r}) := \log p(\mathcal{D}_{\text{sc},r}|\boldsymbol{\alpha},\beta) = -\frac{1}{2}\left[N\log 2\pi + \log|\mathbf{C}| + \mathbf{z}_r^T\mathbf{C}^{-1}\mathbf{z}_r\right], \tag{2.23}$$

where the $N \times N$ matrix $\mathbf{C}$ is defined by

$$\mathbf{C} = \beta^{-1}\mathbf{I} + \boldsymbol{\Phi}\,\text{diag}(\boldsymbol{\alpha})^{-1}\boldsymbol{\Phi}^T, \tag{2.24}$$

with $\mathbf{I}$ being the identity matrix and $|\cdot|$ the determinant operator. Under the assumptions of the previous section, the logarithm of the marginal likelihood of all the data $\mathcal{L}(\boldsymbol{\alpha},\beta|\mathcal{D}_{\text{sc}})$ is given by the sum of $\mathcal{L}(\boldsymbol{\alpha},\beta|\mathcal{D}_{\text{sc},r})$ for $r = 1, \ldots, M$. We will be working with a normalized version of $\mathcal{L}(\boldsymbol{\alpha},\beta|\mathcal{D}_{\text{sc}})$, which we call the *evidence*:

$$\begin{aligned}\mathcal{E}(\boldsymbol{\alpha}|\mathcal{D}_{sc}) &:= \frac{1}{MN}\mathcal{L}(\boldsymbol{\alpha},\beta|\mathcal{D}_{\text{sc}}) \\ &= -\frac{1}{2}\log 2\pi - \frac{1}{2N}\log|\mathbf{C}| - \frac{1}{2MN}\sum_{r=1}^{M}\mathbf{z}_r^T\mathbf{C}^{-1}\mathbf{z}_r.\end{aligned} \tag{2.25}$$

The evaluation of this quantity is based on the generalized singular value decomposition (GSVD) [18] of the matrices $\mathbf{A} = \beta^{1/2}\,\mathrm{diag}(\boldsymbol{\alpha})^{1/2}$ and $\boldsymbol{\Phi}$ and is discussed in Appendix D. Fix the hyperparameters $\boldsymbol{\alpha}$ and let $\boldsymbol{\alpha}_{-s}$ denote the S-dimensional vector with $\alpha_s = +\infty$, that is, with the sth basis function removed. An important observation about the evidence $\mathcal{E}(\boldsymbol{\alpha})$ is that it can be decomposed into two terms (see Appendix A):

$$\mathcal{E}(\boldsymbol{\alpha}) = \mathcal{E}(\boldsymbol{\alpha}_{-s}) + \epsilon(\alpha_s), \tag{2.26}$$

where $\mathcal{E}(\boldsymbol{\alpha}_{-s})$ is the evidence of the model without the sth basis function and $\epsilon(\alpha_s)$ (A.1) depends only on α_s and the sufficient statistics h_s and q_{rs}, $s = 1, \ldots, S, r = 1, \ldots, M$, defined in (A.2). A study of the stationary points of $\epsilon(\alpha_s)$ subject to $\alpha_s > 0$ is carried out in Appendix B and reveals that there exist two distinct possibilities depending on the value of a statistic θ_s (B.2): (1) If $\theta_s > 0$, there exists a finite $\alpha_s^{\mathrm{new}} > 0$, that is, a given unique global maximum (B.3). (2) If $\theta_s \leq 0$, $\epsilon(\alpha_s)$ is maximized at $\alpha_s = +\infty$, that is, when the sth basis function is removed. This observation suggests an iterative algorithmic procedure that is guaranteed to increase the evidence at each step converging to a local maximum. We start with a single basis function, for example, randomly chosen, setting $\alpha_s = +\infty$ for the rest. At each step, there are three possible actions: (1) Add a new basis function. (2) Reestimate the hyperparameters of an existing basis function. (3) Remove a basis function. The action that results in the maximum change in evidence is selected. In Appendix C we explicitly discuss each possible action and how to calculate the change in evidence.

As is also noted by other authors [20], the choice of β can have a critical impact on the predictive capabilities of a computer surrogate. Since β corresponds to the inverse noise of the model, one would expect it to be relatively large, indicating the very low (if existent) noise of computer experiments. However, in the case of limited data and/or an inadequate basis set (e.g., polynomials of a given degree), a very big β might lead to severe overfitting. In these cases, including a finite amount of noise penalizes too-complex models and can improve the predictive performance of the surrogate. A natural way to choose β is to maximize the evidence with respect to it as well. Optimizing the evidence jointly for $\boldsymbol{\alpha}$ and β would make the model computationally intractable. For this reason, we have devised the following heuristic, which results in a local maximum of the evidence. We first optimize the evidence with respect to $\boldsymbol{\alpha}$ for a fixed β with the procedure outlined in the previous paragraph. Then we keep $\boldsymbol{\alpha}$ fixed and maximize the evidence with respect to β. The latter optimization involves a function of a single variable and can be easily carried out by utilizing a golden section–based algorithm. We iterate between the two optimizations until a desired tolerance in the evidence has been reached. Algorithm 1 outlines the maximization procedure. For further details on how to evaluate the various quantities that are involved, consult Appendix D.

2.3. On the choice of the basis functions. The framework developed thus far is independent of the choice of the basis function $\phi_s(\mathbf{x})$. The only requirement is that each element $\mathbf{X}^i$ have its own set of locally defined basis functions. In the numerical examples section, we investigate the performances of two possible choices:

1. SE kernels: Here, we choose $\phi_s(\mathbf{x})$ to be kernel functions centered on top of the observed data points. Assume that we are working on element $\mathbf{X}^i$ and that we have observed $\mathcal{D}$ as in (2.6). Then, we define $\phi_s(\mathbf{x})$ to be $\phi_s(\mathbf{x}) := \phi(\mathbf{x}, \mathbf{x}^{(s)}), s = 1, \ldots, N$, where $\phi(\cdot, \cdot)$ is a kernel function. That is, $S = N$. A usual choice of $\phi(\cdot, \cdot)$ is the SE kernel $\phi(\mathbf{x}_1, \mathbf{x}_2) = \exp\{-\frac{1}{2}\sum_{k=1}^{K} \frac{(x_{1k}-x_{2k})^2}{\ell_k^2}\}$.

Algorithm 1 MRVM training procedure.

Step 1: Initialize $\boldsymbol{\alpha}$ by including the single basis function maximizing the evidence (i.e., all α_s's are infinity except for one), and pick a starting value for β (we use $\beta = 100$ in all numerical experiments).

Step 2: Compute all the statistics (H_m, Q_{rm}), (h_m, q_{rm}), and θ_m, $s = 1, \ldots, S$, and $r = 1, \ldots, M$, using (A.3), (A.4), and (B.2), respectively.

Step 3: Iterate through all basis functions and calculate the change in evidence that results from their optimal action (add, reestimate, and remove), as discussed in Appendix C. Select the basis function that results in the maximum change in evidence.

Step 4: If the maximum change in evidence is less than a threshold (in the numerical examples we use 10^{-6}), then go to Step 5. Otherwise, perform the selected action and go to Step 2.

Step 5: Find the maximum of the evidence with respect to β while keeping $\boldsymbol{\alpha}$ fixed. If the change in evidence is less than a threshold, then stop. Otherwise, update the value of β and go to Step 2.

The parameters ℓ_k have the interpretation of the length scale of each particular dimension. They can be selected by a cross-validation procedure or by maximizing the evidence with respect to them. To keep our framework as simple as possible, we fix the length scale on each element to be proportional to the standard deviation of the observed input dimensions, i.e., $\ell_k := (\frac{1}{N}\sum_{n=1}^{N}(x_k^{(n)} - \bar{x}_k)^2)^{\frac{1}{2}}$, where $\bar{x}_k = \frac{1}{N}\sum_{n=1}^{N} x_k^{(n)}$.

2. Optimal orthogonal polynomials: One can choose the $\phi_s(\mathbf{x})$'s as in [40] to be the locally defined set of optimal polynomials with respect to the conditional density $p^i(\mathbf{x})$ of the element $\mathbf{X}^i$. By the independence assumption, $p^i(\mathbf{x}) = \prod_{k=1}^{K} p_k^i(x_k)$, where $p_k^i(x_k)$ is the conditional distribution pertaining to the kth dimension. Using the Lanczos procedure of [12], we construct the one-dimensional orthogonal polynomials with respect to $p_k^i(x_k)$ for all $k = 1, \ldots, K$ and then multiply them to obtain K-variate orthogonal polynomials with respect to $p^i(\mathbf{x})$ up to a given degree. In that case, the RVM framework is used to decide how many and which of these polynomials should be kept. As will be discussed in section 2.4, the optimal orthogonal polynomials have the nice property of yielding analytic estimates for the mean and the variance without the need to fit the second power of the response.

2.4. Calculation of the local statistics. As before, the focus is again on a specific element $\mathbf{X}^i$. All quantities are again local to $\mathbf{X}^i$. In order to keep notational complexity to a minimum, we do not explicitly show this dependence. We will derive point estimates for the mean and the higher moments of the response based on the linear point estimator of $\mathbf{f}$ over $\mathbf{X}^i$ given in (2.21). To be exact, we are interested in estimating all (local) moments $\mathbf{m}^q = (m_1^q, \ldots, m_M^q)$, where

$$m_r^q = \int_{\mathbf{X}^i} f_r^q(\mathbf{x}) p^i(\mathbf{x}) d\mathbf{x}, \tag{2.27}$$

$q \geq 1$, and $p^i(\mathbf{x})$ is the conditional probability distribution of $\mathbf{X}^i$ given in (2.7). For the case of nonorthogonal basis functions (i.e., SEs), we can derive semianalytic estimates by keeping concurrent MRVM estimates of the response raised to the qth

power. In particular, the qth power of the response is treated as an MRVM with its own hyperparameters $\boldsymbol{\alpha}^q$. Let us denote the predictive mean for the qth power of the response at $\mathbf{x} \in \mathbf{X}^i$ by

$$\mu_{f_r^q}(\mathbf{x}; \boldsymbol{\alpha}^q) = \sigma^q_{\text{obs},r} \left(\boldsymbol{\mu}_r^q\right)^T \boldsymbol{\phi}(\mathbf{x}) + \mu^q_{\text{obs},r},$$

where $\mu^q_{\text{obs},r}$ and $\sigma^q_{\text{obs},r}$ are defined as in (2.9) and (2.10), respectively, using the qth power of the observed response, and $\boldsymbol{\mu}^q$ is the posterior mean of the weights of the qth power of the scaled response (see (2.18)). The q-moment can be approximated by

$$\hat{m}_r^q = \int_{\mathbf{X}^i} \mu_{f_r^q}(\mathbf{x}; \boldsymbol{\alpha}^q) p^i(\mathbf{x}) d\mathbf{x}. \tag{2.28}$$

Fortunately, the integrals involved can be expressed in terms of expectations of the basis functions with respect to the conditional input distribution. Due to the independence assumption of the input random variables and the special choice of basis functions made in section 2.3, these expectations can be numerically evaluated using a quadrature rule[1] in $O(K)$ time (K being the number of input dimensions) for special choices of basis functions $\phi_s(\mathbf{x})$. This is possible for the exponential kernels chosen in this work (see section 2.3) and for any polynomial basis. In particular, for the exponential kernels it is straightforward to show that

$$\hat{m}_r^q = \sigma^q_{\text{obs},r} \sum_{s=1}^{S} \left[\mu^q_{rs} \prod_{k=1}^{K} \left(\int_{a_k^i}^{b_k^i} \exp\left\{ -\frac{\left(x_k - x_k^{(s)}\right)^2}{2\ell_k^2} \right\} p_k^i(x_k) dx_k \right) \right] + \mu^q_{\text{obs},r}, \tag{2.29}$$

where $p_k^i(x_k)$ is the marginal conditional distribution of $\mathbf{X}^i = \times_{k=1}^K [a_k^i, b_k^i]$ ($-\infty \leq a_k^i < x_k < b_k^i \leq +\infty$) along dimension k:

$$p_k^i(\mathbf{x}) := \int_{\times_{\ell \neq k}[a_\ell^i, b_\ell^i]} p^i(\mathbf{x}) dx_1 \dots dx_{k-1} dx_{k+1} \dots dx_K = \frac{p_k(x_k) 1_{[a_k^i, b_k^i]}(x_k)}{\int_{a_k^i}^{b_k^i} p_k(x_k) dx_k}. \tag{2.30}$$

The $S \times K$ one-dimensional integrals are the ones that are evaluated numerically. In case the support of $p_k^i(x_k)$ is infinite (or semi-infinite), the integrand is mapped—using a suitable transformation—to $[-1, 1]$. For arbitrary basis functions, one would have to use an MC-based integration technique, which would also be computationally efficient since the evaluation of the local surrogates is extremely cheap compared to the underlying deterministic solver.

When optimal orthogonal polynomials are used as basis functions, then it is trivial to derive analytic estimates for the moments by exploiting their orthogonality [40] without requiring an estimate of the second power of the response. Assuming they are normalized, the first two moments are given by

$$\hat{m_r}^1 = \sigma^1_{\text{obs},r} \mu^1_{r1} + \mu^1_{\text{obs},r} \text{ and } \hat{m_r}^2 = (\sigma^1_{\text{obs,r}})^2 \sum_{s=1}^{S} (\mu^1_{rs})^2,$$

where μ^1_{r1} is the coefficient of the constant polynomial.

[1] In our numerical examples, we used the QAGS algorithm of QUADPACK as implemented in GSL. This is based on an adaptive bisection scheme combined with the Wynn epsilon-algorithm that utilizes a Gauss–Kronrod 21-point rule (see [33]). This algorithm is obviously overkill for the task under consideration, but its cost is negligible compared to a single run of the deterministic code.

2.5. From local to global statistics. In the same spirit as the ME methods [39, 40, 8], we combine the statistics over each stochastic element in order to obtain the global analogues. Since we now work over the whole domain, we will explicitly mark the dependence of the underlying quantities on the element $\mathbf{X}^i, i = 1, \ldots, I$. Let $\hat{m}_r^{q,i}$ be the estimate for the local q-moment that pertains to the element $\mathbf{X}^i$. An estimate of the global q-moment is provided by

$$\hat{m}_r^q = \sum_{i=1}^{I} \hat{m}_r^{q,i} P(\mathbf{X}^i), \tag{2.31}$$

where $P\left(\mathbf{X}^i\right)$ was defined in (2.8) and can easily be numerically evaluated due to the independence assumption. Finally, an estimate of the variance of the response is obtained via

$$\hat{v_r} = \hat{m}_r^2 - \left(\hat{m}_r^1\right)^2. \tag{2.32}$$

2.6. Quantifying the uncertainty of the predicted statistics. The Bayesian nature of the underlying regression model induces a probability distribution on the space of possible surrogates and, in turn, on the predicted statistics. In order to make this connection clear, one may think of the space of possible surrogates as being parametrized by the weights. Then, the posterior of the weights $\mathbf{w}_r$ of each element (see (2.17)) spawns a posterior distribution on the space of surrogates. The weight of this posterior is directly associated with the epistemic uncertainty due to the limited number of observations. In this section, we describe a procedure that yields samples from the posterior distribution of the surrogates. Given a sample surrogate we can compute its statistics, quantifying thereby the impact that the epistemic uncertainty has on them. In what follows, we work within a specific element $\mathbf{X}^i$ even though this is not explicitly denoted, in order to limit the notational burden.

Let $\boldsymbol{\alpha}$ denote the hyperparameters and $p(\mathbf{w}_r|\mathcal{D}_{\mathrm{sc},r}, \boldsymbol{\alpha})$ the posterior of the weights, as implied by (2.17). We obtain a sample of the qth moment in the following way:

1. Sample $\mathbf{w}_{\mathrm{sample},r}$ from $p(\mathbf{w}_r|\mathcal{D}_{\mathrm{sc},r}, \boldsymbol{\alpha})$ for $r = 1, \ldots, M$ (2.17). This is a normal distribution and can be sampled directly.
2. Then, a sample of $f_r(\mathbf{x}), r = 1, \ldots, M$ is given by

$$f_{\mathrm{sample},r}(\mathbf{x}) = \sigma_{\mathrm{obs},r} \left(\mathbf{w}_{\mathrm{sample},r}\right)^T \boldsymbol{\phi}(\mathbf{x}) + \mu_{\mathrm{obs},r}. \tag{2.33}$$

3. Finally, use (2.33) to obtain a sample of the local qth moment:

$$\hat{m}^q_{\mathrm{sample},r} = \int_{\mathbf{X}^i} f^q_{\mathrm{sample},r}(\mathbf{x}) p^i(\mathbf{x}) d\mathbf{x}. \tag{2.34}$$

In order to obtain a sample of the global qth moment, the above-mentioned procedure is repeated on each element and the results are combined in the spirit of section 2.5. For the case of optimal orthogonal polynomials, step 3 can be carried out analytically as discussed in section 2.4. For the case of the SE basis functions, step 3 has to be carried out numerically. If one is also fitting the qth power of the response in order to make use of the semianalytic formulas of (2.29), then one has to modify the above procedure so that it samples the posterior of the weights $\mathbf{w}_r^q$ pertaining to that particular power.

2.7. Adaptivity. In this section, we develop an iterative procedure to adaptively decompose the stochastic space in small elements. The initial step of this procedure starts by considering a single element, that is, the whole stochastic space $\mathbf{X}$. Here, we assume that we are already given a decomposition of the domain as well as a local surrogate model on each element. The decision we wish to make is whether or not to refine a given element and in which way. We develop refinement criteria that are based solely on information gathered by the current surrogate model, and no further calls to the deterministic solver are required. The Bayesian predictive variance equation (2.22) is used to define a measure of our uncertainty about the prediction over the whole domain $\mathbf{X}$. We show how this measure can be broken down into contributions coming from each element. Based on this observation, we derive a criterion that suggests refinement of an element if its contribution to the global uncertainty is larger than a prespecified threshold. For the sake of simplicity, we only consider rectangular elements and refine them by splitting them perpendicular to the dimension of greater importance. The importance of a particular dimension is characterized by the variability of the predicted response function and biased by the underlying input probability distribution.

Suppose that we already have a decomposition of the stochastic domain $\mathbf{X}$ in *rectangular* elements $\mathbf{X}^i$, e.g., $\mathbf{X}^i = [a_1^i, b_1^i] \times \cdots \times [a_K^i, b_K^i]$, with $-\infty \leq a_k^i < b_k^i \leq +\infty, k = 1, \ldots, K, i = 1, \ldots, I$, such that (2.4) holds. Furthermore, assume that we have already learned the local surrogates on each element $\mathbf{X}^i$. Let $\sigma^2_{f_r^i}(\mathbf{x})$ be the predictive variance of the rth output of the local surrogate of $\mathbf{f}^i$ at $\mathbf{x} \in \mathbf{X}^i$ (2.22). By the conditional independence assumption for the predictive distribution over each element and (2.5), the predictive variance of the rth output of the global surrogate $\sigma^2_{f_r}(\mathbf{x})$ at $\mathbf{x} \in \mathbf{X}$ is given by

$$\sigma^2_{f_r}(\mathbf{x}) = \sum_{i=1}^{I} \sigma^2_{f_r^i}(\mathbf{x}) 1_{\mathbf{X}^i}(\mathbf{x}). \tag{2.35}$$

Taking the expectation of this quantity with respect to the input probability density $p(\mathbf{x})$ and averaging over $r = 1, \ldots, M$, we obtain

$$\sigma^2_{\mathbf{f},p} := \int_{\mathbf{X}} \sigma^2_{\mathbf{f}}(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} := \int_{\mathbf{X}} \frac{1}{M} \sum_{r=1}^{M} \sigma^2_{f_r}(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}. \tag{2.36}$$

This quantity is a measure of our uncertainty about our prediction over the whole domain $\mathbf{X}$. Notice that, in $\sigma^2_{\mathbf{f},p}$, the uncertainty of the model at $\mathbf{x}$ is weighted by its probability of occurrence $p(\mathbf{x})$. Intuitively speaking, we are willing to accept a somewhat less accurate surrogate in regions of the space occurring with lower probability. Using (2.35), it is straightforward to see that

$$\sigma^2_{\mathbf{f},p} = \sum_{i=1}^{I} \sigma^2_{\mathbf{f},p^i} P(\mathbf{X}^i) \tag{2.37}$$

where

$$\sigma^2_{\mathbf{f},p^i} := \int_{\mathbf{X}^i} \sigma^2_{\mathbf{f}}(\mathbf{x}) p^i(\mathbf{x}) d\mathbf{x} := \int_{\mathbf{X}} \frac{1}{M} \sum_{r=1}^{M} \sigma^2_{f_r}(\mathbf{x}) p^i(\mathbf{x}) d\mathbf{x}$$

is the uncertainty of our prediction over the element $\mathbf{X}^i$. We refine the element $\mathbf{X}^i$ if the contribution to the global uncertainty coming from it is greater than a certain threshold $\delta > 0$; i.e., we refine $\mathbf{X}^i$ if

$$\sigma^2_{\mathbf{f},p^i} P(\mathbf{X}^i) > \delta. \tag{2.38}$$

As already mentioned, we refine elements by cutting them perpendicular to the most "important" dimension. At this point, we attempt to give a precise meaning to the concept of "the most important dimension." The sensitivity of a real function f with respect to each dimension is captured by the partial derivatives $\frac{\partial f}{\partial x_k}$. Generally speaking, the derivative is significantly different than zero in regions of space where the function varies the most. A quantity that measures the variability of the prediction along the kth dimension is

$$V_k^i := \left(\frac{1}{M} \sum_{r=1}^{M} \int_{\mathbf{X}^i} \left(\frac{\partial \mu_{f_r}(\mathbf{x})}{\partial x_k} \right)^2 p^i(\mathbf{x}) d\mathbf{x} \right)^{1/2}.$$

V_k^i can be thought as the weighted norm of the sensitivity functions for each output (the weight having been specified by the conditional probability density $p^i(\mathbf{x})$). Furthermore, let us introduce the probability P_k^i that the kth dimension x_k of a random input point $\mathbf{x} \in \mathbf{X}$ falls inside $\mathbf{X}^i$:

$$P_k^i := \int_{a_k^i}^{b_k^i} \left(\int_{\times_{\ell \neq k} [a_\ell^i, b_\ell^i]} p(\mathbf{x}) dx_1 \dots dx_{k-1} dx_{k+1} \dots dx_K \right) dx_k = \int_{a_k^i}^{b_k^i} p_k(x_k) dx_k, \tag{2.39}$$

where the last equality is a consequence of the independence assumption. We define the *importance* I_k^i of the dimension k of the element $\mathbf{X}^i$ to be

$$I_k^i = V_k^i P_k^i. \tag{2.40}$$

Intuitively, the importance of a particular dimension is proportional to the variability observed along that dimension and to the probability mass along that dimension trapped within the stochastic element. Thus, if $\mathbf{X}^i$ needs refinement (i.e., satisfies (2.38)) we cut it perpendicular to the most important dimension k^*, given by

$$k^* = \arg\max_k I_k^i. \tag{2.41}$$

In order to have two new elements with the same probabilities, the splitting point is given by the median of the marginal conditional distribution of $\mathbf{X}^i$ along dimension k given in (2.30).

2.8. A complete view of the framework. In this final subsection, we put together the building blocks of our scheme and discuss the algorithmic details and parallelization strategies. The basic input required is the number of observations per element N and the tolerance $\delta > 0$ used for the refinement criterion (2.38). Algorithm 2 provides a serial implementation of the scheme.

The scheme works in one-element cycles that consist of collecting observations (random samples from the conditional distribution $p^i(\mathbf{x})$ of the element), fitting (sections 2.1 and 2.2), and adapting (section 2.7). Let us denote with $\mathbf{X}^i$ a stochastic element, $\mathcal{D}^i$ the observations made on $\mathbf{X}^i$, and $\mathcal{M}^i$ the MRVM fitted over $\mathbf{X}^i$ using $\mathcal{D}^i$. Let $\mathcal{C}$ be the set of triplets $\mathbf{X}^i, \mathcal{D}^i, \mathcal{M}^i$ for which the refinement criterion (2.38) is not satisfied. We will refer to $\mathcal{C}$ as the set of *completed triplets*. The rest of the triplets are put in $\mathcal{U}$, called the set of *uncompleted triplets*. With $|\mathcal{D}^i|$ we denote the number of observations inside $\mathcal{D}^i$.

Algorithm 2 The complete surrogate building framework.

$\mathcal{U} \leftarrow \{(\mathbf{X}, \emptyset, \emptyset)\}$.
$\mathcal{C} \leftarrow \emptyset$.
while $\mathcal{U} \neq \emptyset$ **do**
  Remove $(\mathbf{X}^i, \mathcal{D}^i, \mathcal{M}^i)$ from $\mathcal{U}$.
  if $\mathcal{M}_i = \emptyset$ **then**
    Observe N random points drawn from $p^i(\mathbf{x})$ (2.7).
  else
    Observe $N - |\mathcal{D}^i|$ random points drawn from $p^i(\mathbf{x})$ and add them to $\mathcal{D}^i$.
  end if
  Fit $\mathcal{M}^i$ using only the data in $\mathcal{D}^i$ (sections 2.1 and 2.2).
  if the refinement criterion of (2.38) is satisfied for δ **then**
    Split $\mathbf{X}^i$ in $\mathbf{X}^{i,1}$ and $\mathbf{X}^{i,2}$ according to (2.41).
    Let $\mathcal{D}^{i,1}$ and $\mathcal{D}^{i,2}$ be the set observations in $\mathcal{D}^i$ whose inputs live in $\mathbf{X}^{i,1}$ and $\mathbf{X}^{i,2}$, respectively.
    $\mathcal{U} \leftarrow \mathcal{U} \cup \{(\mathbf{X}^{i,1}, \mathcal{D}^{i,1}, \mathcal{M}^i), (\mathbf{X}^{i,2}, \mathcal{D}^{i,2}, \mathcal{M}^i)\}$.
  else
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{(\mathbf{X}^i, \mathcal{D}^i, \mathcal{M}^i)\}$.
  end if
end while

Parallelization of Algorithm 2 is relatively easy. Each node p has its own set of completed, $\mathcal{C}_p$, and uncompleted, $\mathcal{U}_p$, elements. Initially the root node $p = 0$ starts as in Algorithm 2, and the rest with $\mathcal{U}_p = \emptyset, \mathcal{C}_p = \emptyset, p \neq 0$. Then, everything proceeds as in Algorithm 2 with load rebalancing at the end of each outer iteration (uncompleted elements are sent to processors with $\mathcal{U}_p = \emptyset$).

3. Numerical examples. All examples are run on massively parallel computers at the National Energy Research Scientific Computing Center. The parallelization strategy is straightforward: each processor is assigned to work with a single element. The communication burden between the processes is minimal. Our implementation utilizes the Trilinos library [24] and GSL [10] extensively.

The ultimate goal of the numerical examples is to demonstrate that the method can

1. learn nonstationary surfaces,
2. deal with discontinuities,
3. identify localized features of the response, and
4. reduce sampling frequency on unimportant input dimensions.

In section 3.1 we apply our method to the Krainchnan–Orszag ODE system with random initial conditions. Section 3.2 examines the classical stochastic elliptic problem. Finally, in section 3.3, we solve a stochastic flow through porous media problem. In all problems, the underlying input probability distribution $p(\mathbf{x})$ is explicitly stated. All tasks start with a single element (the input domain itself) and N random samples drawn from the input distribution. N is also the maximum number of samples taken within an element and is different for each example. From that point, the algorithm proceeds until a prespecified tolerance $\delta > 0$ is reached. The refinement criterion is given by (2.38). The only parameters of the method are N and δ. RVM-SE stands for RVM using SE kernels and RVM-GPC stands for RVM using optimal orthogonal polynomials. For RVM-GPC, all orthogonal polynomials up to a given degree P are constructed on the fly for each element.

3.1. Krainchnan–Orszag three-mode problem. Consider the system of ordinary differential equations [39]

$$\frac{dy_1}{dt} = y_1 y_3,$$
$$\frac{dy_2}{dt} = -y_2 y_3,$$
$$\frac{dy_3}{dt} = -y_1^2 + y_2^2,$$

subject to random initial conditions at $t = 0$. This dynamical system is particularly interesting because the response has a discontinuity at the planes $y_1(0) = 0,\ y_2(0) = 0$. We solve the system for the time interval $[0, 10]$ and record the response at time step intervals of $\Delta t = 0.01$. This results in a total of $M = 300$ outputs (100 for each of the three dimensions of the response). We will consider two different cases of increasing difficulty with two and three input dimensions and various input distributions. The results we obtain will be compared to MC estimates with 10^6 samples. Let the MC mean and variance be $m_{r,\mathrm{MC}}$ and $v_{r,\mathrm{MC}}$, respectively, for $r = 1, \ldots, 300$. The error of the statistics will be evaluated using the (normalized) L_2 *norm of the error in variance* defined by

$$\mathrm{E}_{L_2} = \frac{1}{M} \sum_{r=1}^{M} \left(v_{r,\mathrm{MC}} - \hat{v}_r \right)^2, \tag{3.1}$$

where $\hat{v}_r$ is given by (2.32).

We start by considering the two-dimensional problem (KO-2) with stochastic initial conditions defined by

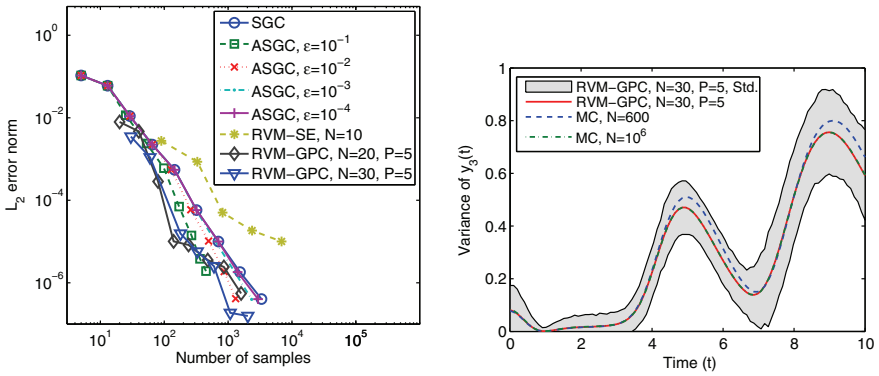
$$y_1(0) = 1,\ y_2(0) = 0.1(2x_1 - 1),\ y_3(0) = 2x_2 - 1,$$

where $x_k, k = 1, 2$, are random variables. We will examine two cases of input distributions, as follows:

1. Uniform input: $\mathbf{X} = [0, 1]^2$ and

$$p_k(x_k) = 1,\ k = 1, 2.$$

Figure 3.1(a) shows the evolution of the L_2 norm of the error in variance for RVM-SE ($N = 10$) as well as RVM-GPC ($N = 20$) and RMV-GPC ($N = 30$) for maximum polynomial degree $P = 5$ as a function of the number of calls to the deterministic solver. For comparison, we include in the same plot the performance of SGC and ASGC. The $\epsilon > 0$ of ASGC is a parameter specifying the sensitivity of the adaptation criterion (see [30]); that is, collocation points with surpluses larger than ϵ spawn new points around them. Therefore, for $\epsilon = 0$, one obtains the normal SGC method and, as ϵ is increased, fewer and fewer collocation points are taken into consideration. The data for the SG-based methods are collected as follows: (1) A maximum collocation level is specified (here it is 7). (2) An ϵ is specified ($\epsilon = 0$ for SGC). (3) We add collocation points until the maximum level has been reached or there are no more collocation points with surpluses greater than ϵ. (4) Each time an interpolation level is crossed, we use the number of samples gathered and measure the L_2 error in variance. We observe that RVM-SE is the slowest, with performance an order of magnitude worse than SGC. However, RVM-GPC seems to perform at least as well as the fastest ASGC run ($\epsilon = 10^{-1}$).



(a) L_2 error (b) Variance Samples

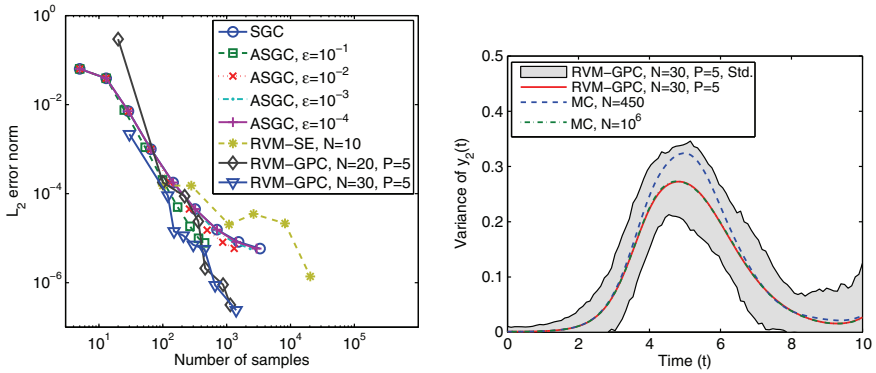
FIG. 3.1. *KO-2 (Uniform input):* (a) *L_2 error norm in variance for RVM-SE ($N = 10$), RVM-GPC ($N = 20$), and RVM-GPC ($N = 30$) with maximum polynomial degree $P = 5$; SGC and ASGC with $\epsilon = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$.* (b) *The predictive variance of $y_3(t)$ for RVM-GPC ($N = 30$) with maximum polynomial degree $P = 5$ for a tolerance of $\delta = 10^{-6}$ (red solid line). The total number of observations used by RVM-GPC for this case is* 600. *The gray area denotes plus/minus one standard deviation of predicted variance estimated from* 100 *samples of the weights. For comparison, we have included an MC estimator using* 600 *(dashed blue line) and one using* 10^6 *observations (dot-dashed green line).*

We believe that the poor performance of RVM-SE is due to the poor choice of the kernels and that a scheme that would optimize the evidence with respect to the length scales on each element would perform better. Figure 3.1(b) shows the predicted variance of $y_3(t)$ for RVM-GPC ($N = 30$ with maximum polynomial degree $P = 5$ and a tolerance of $\delta = 10^{-6}$), along with plus/minus one standard deviation estimated from 100 samples as described in section 2.6. The number of observations gathered for that particular case is 600, and for comparison we have included the results of an MC estimate using the same number of samples as well as the reference MC estimator based on 10^6 samples. It is apparent that the RVM estimate is much more accurate than the MC result using the same number of samples. However, the error bars are clearly larger than necessary. We believe that this unwanted uncertainty is induced by the underlying assumption of independence of the various output dimensions. The same error bars for the mean are significantly smaller, and we have numerically observed that they shrink as the desired tolerance δ decreases, albeit always remaining much wider than the true error.

2. Beta input: $\mathbf{X} = [0, 1]^2$ and

$$p_k(x_k) = \frac{\Gamma(\alpha + \beta)}{2\Gamma(\alpha)\Gamma(\beta)} x_k^{\alpha-1} (1 - x_k)^{\beta-1}, \; k = 1, 2,$$

where $\Gamma(z)$ is the gamma function. We use $\alpha = 2$ and $\beta = 5$.
Figure 3.2(a) shows the evolution of the L_2 norm of the error in variance for RVM-SE ($N = 10$) as well as RVM-GPC ($N = 20$) and RMV-GPC ($N = 30$) for maximum polynomial degree $P = 5$ as a function of the number of calls to the deterministic solver. In this convergence test, we see again that RVM-SE ($N = 10$) has the worst performance. Notice that it furthermore exhibits some instabilities in the sense that the error is slightly increasing as the number of samples increases as we pass from tolerance $\delta = 10^{-4}$ to $\delta = 10^{-5}$. This is a numerical artifact created by the combination of a poor choice of

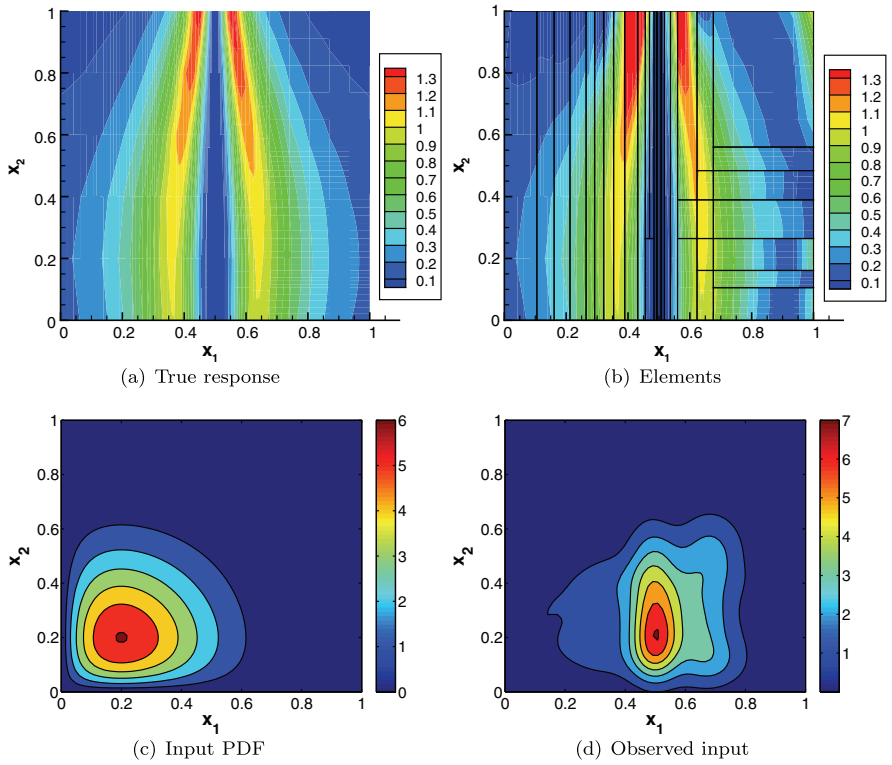


(a) L_2 error (b) Variance Samples

FIG. 3.2. *KO-2 (Beta input):* (a) L_2 *error norm in variance for RVM-SE* ($N = 10$)*, RVM-GPC* ($N = 20$)*, and RVM-GPC* ($N = 30$) *with maximum polynomial degree* $P = 5$*; SGC and ASGC with* $\epsilon = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$. (b) *Predictive variance of* $y_2(t)$ *for RVM-GPC* ($N = 30$) *with maximum polynomial degree* $P = 5$ *for a tolerance of* $\delta = 10^{-6}$ *(red solid line). The total number of observations used by RVM-GPC for this case is* 450*. The gray area denotes plus/minus one standard deviation of predicted variance estimated from* 100 *samples of the weights. For comparison, we have included an MC estimator using* 450 *(dashed blue line) and one using* 10^6 *observations (dot-dashed green line).*

the kernels and of N. On the other hand, RVM-GPC converges quite fast with the $N = 30$ case clearly outperforming ASGC. Figure 3.2(b) shows the predicted variance of $y_2(t)$ for RVM-GPC ($N = 30$ with maximum polynomial degree $P = 5$ and a tolerance of $\delta = 10^{-6}$), along with plus/minus one standard deviation estimated from 100 samples as described in section 2.6. The number of observations gathered for that particular case is 450, and for comparison we have included the results of an MC estimate using the same number of samples as well as the reference MC estimator based on 10^6 samples. Again, the RVM estimate is significantly more accurate than the MC result using the same number of samples, with the error bars, however, clearly overestimating the true error. For the same RVM-GPC case, Figure 3.3 depicts the stochastic elements (b) and a kernel density estimator of the probability density of the observed input (d). For a complete comparison, Figure 3.3(b) also includes a contour of the prediction at $y_2(t = 10)$, while (a) and (c) show the true response of the same output variable and the original input probability density, respectively. Notice how the input probability density affects the decomposition of the stochastic space and, as a result, the selection of the observed samples. The discontinuity is partly resolved. Parts of the response that reside in low-probability regions do not have to be fully resolved for an accurate calculation of the statistics.

Remark 3. *Time-dependent discontinuity.* In the examples presented above the discontinuity occurs at the same points for all time instants. Since our scheme decomposes the stochastic space based on information coming from all time instants, it will probably lead to very fine decompositions in case the discontinuity moves significantly as a function of time. If one wishes to capture such a situation, we suggest adding time as one more variable of the surrogate and performing the decomposition in the extended stochastic-time domain. However, this case goes beyond the scope of the current work.



(a) True response (b) Elements

(c) Input PDF (d) Observed input

FIG. 3.3. *KO-2 (Beta input):* (a) *True response* $y_2(t = 10)$ *as a function of the initial conditions.* (b) *The discovered elements for RVM-GPC* ($N = 30$) *for maximum polynomial degree* $P = 5$ *at a tolerance of* $\delta = 10^{-6}$ *along with the mean prediction.* (c) *The original input probability density.* (d) *A kernel density estimator of the actually observed input points.*

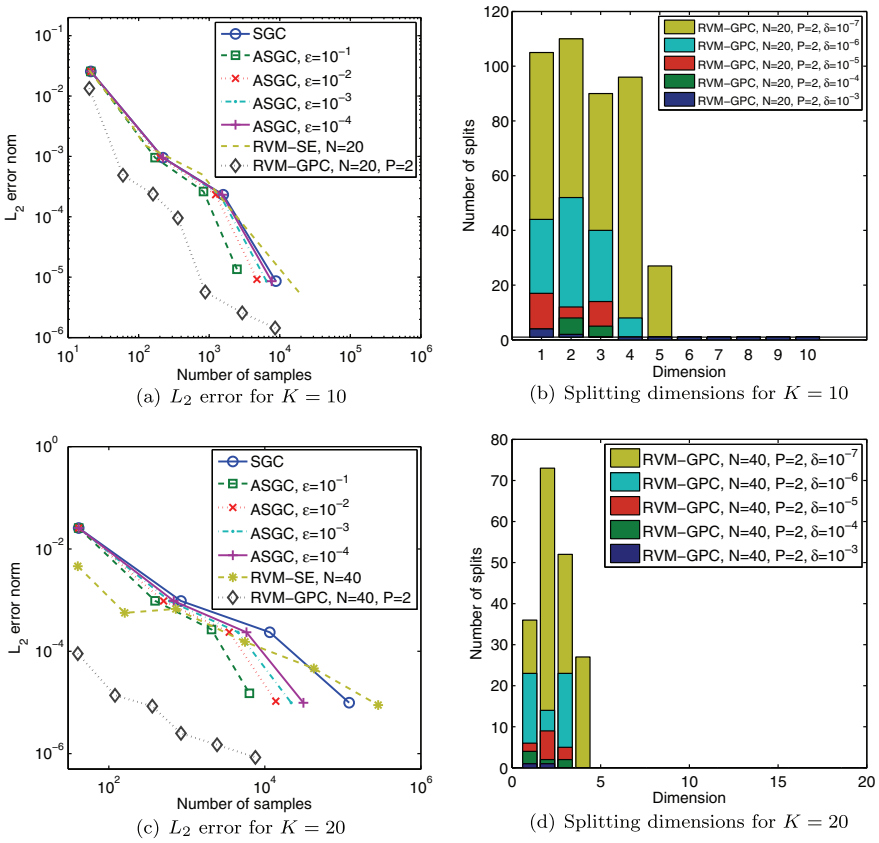
3.2. Elliptic problem. In this section, we consider a simple stochastic elliptic problem [32]. Consider the stochastic partial differential equation

$$\begin{aligned} -\nabla \cdot (a_K(\boldsymbol{\omega}, \cdot)\nabla u(\boldsymbol{\omega}, \cdot)) &= f(\cdot) \ \text{ in } D, \\ u(\boldsymbol{\omega}, \cdot) &= 0 \ \text{ on } \partial D, \end{aligned}$$

where the physical domain is $D = [0, 1]^2$. In order to avoid confusion with the physical dimension $\mathbf{x}$, we have chosen to denote the random variables with $\boldsymbol{\omega}$ instead of $\mathbf{x}$. We choose a smooth *deterministic* load $f(x, y) = 100\cos(x)\sin(y)$ and work with homogeneous boundary conditions. The deterministic problem is solved with the finite element method using 400 (20×20 grid) bilinear quadrilateral elements. The random diffusion coefficient $a_K(\boldsymbol{\omega}, x)$ is constructed to have a one-dimensional dependence $\log(a_K(\boldsymbol{\omega}, x, y) - 0.5) = 1 + \omega_1 \left(\frac{\sqrt{\pi}L}{2}\right)^{1/2} + \sum_{k=2}^{K} \xi_k \phi_k(x)\omega_k$, where $\xi_k := (\sqrt{\pi}L)^{1/2} \exp\left(\frac{-(\lfloor \frac{k}{2} \rfloor \pi L)^2}{8}\right)$ for $k \geq 2$ and

$$\phi_k(x) := \begin{cases} \sin\left(\frac{\lfloor \frac{k}{2} \rfloor \pi x}{L_p}\right) & \text{if } k \text{ is even}, \\ \cos\left(\frac{\lfloor \frac{k}{2} \rfloor \pi x}{L_p}\right) & \text{if } k \text{ is odd}. \end{cases}$$

We choose the $\omega_k, k = 1, \ldots, K$, to be independent identically distributed random variables $\omega_k \sim U([-\sqrt{3}, \sqrt{3}])$. Hence, the stochastic input space is $\boldsymbol{\Omega} = [-\sqrt{3}, \sqrt{3}]^K$.



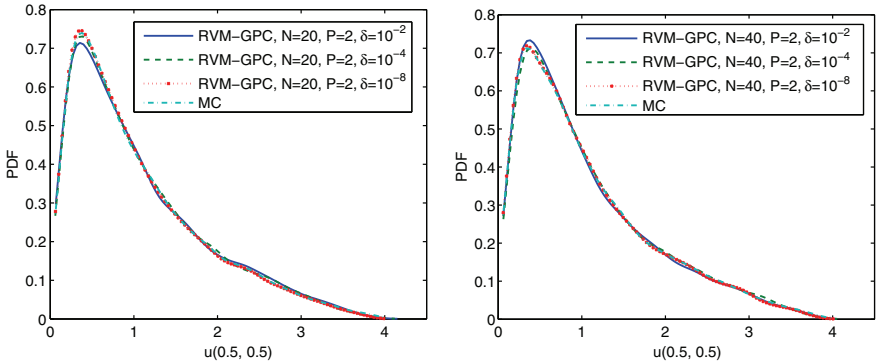
(a) L_2 error for $K = 10$ (b) Splitting dimensions for $K = 10$

(c) L_2 error for $K = 20$ (d) Splitting dimensions for $K = 20$

FIG. 3.4. *The left column shows L_2 error norm in variance for $K = 10$ (a) and $K = 20$ (c) for RVM-SE and RVM-GPC with maximum polynomial degree $P = 2$, SGC and ASGC with $\epsilon = 10^{-1}, 10^{-2}, 10^{-3}$, and 10^{-4}. The right column shows the number of splits per dimension performed by RVM-GPC for $K = 10$ (b) and $K = 20$ (d).*

Finally, we set $L_p = \max\{1, 2L_c\}$ and $L = \frac{L_c}{L_p}$, where L_c is called the *correlation length*.

In this study, we set the correlation length to $L_c = 0.6$ and test the convergence of our method for $K = 10$ and 20 input dimensions. The $K = 10$ and 20 cases are solved using RVM-SE and RVM-GPC up to maximum polynomial degree $P = 2$ for $N = 20$ and 40, respectively, up to a tolerance of 10^{-8}. Figure 3.4(a) and (c) compares the convergence of our method to ASGC for $K = 10$ and $K = 20$, respectively. The reference variance was calculated using an MC estimator with 10^6 samples. We observe that RVM-SE performs similarly to ASGC, while RVM-GPC seems to converge much faster, particularly for the high-dimensional case. In Figure 3.4(b) and (d) we show the number of splits per dimension for various tolerances δ of the RVM-GPC case for $K = 10$ and $K = 20$ input dimensions, respectively. It is clearly seen that the method identifies only the first few dimensions as important, while completely ignoring the rest. Finally, we calculate the predictive PDFs for selected outputs for RVM-GPC. The results are computed as follows: At a given tolerance, we draw 10,000 random input samples and propagate them through the surrogate. Based on these samples, we fit a kernel density estimator (see [35]) and compare it to the kernel density estimator obtained by using the full deterministic solver instead of the surrogate (this



(a) PDF of $u(0.5, 0.5)$ for $K = 10$ (b) PDF of $u(0.5, 0.5)$ for $K = 20$

FIG. 3.5. *Elliptic problem: Comparison of the predicted PDF of* $u(0.5, 0.5)$ *using various RVM-GPC surrogates with the MC predictions.* (a) $K = 10$ *input dimensions.* (b) $K = 20$ *input dimensions.*

is referred to as "MC" on figure legends). Because of the limited number of samples used for the kernel density estimation, a small variation on the outcome is expected. Figure 3.5(a) and (b) depicts the PDFs of $u(0.5, 0.5)$ for $K = 10$ and $K = 20$ input dimensions, respectively. As a general rule, a smaller tolerance is required to capture a qualitatively similar PDF than is required to capture the correct variance. At this point, we must mention that the PDFs produced by RVM-SE are not quite as satisfactory. This is due to the known limitations of the SE kernels: (1) It is very difficult to identify the right length scales. (2) In high dimensions, the SE kernel is not a good similarity measure between input points [9].

3.3. Flow through porous media. Consider a bounded two-dimensional spatial domain $D \subset \mathbb{R}^2$ with smooth boundary ∂D. The governing equations for immiscible and incompressible two-phase flow in porous media consist of an elliptic equation for fluid pressure and a transport equation for the movement of fluid phases. For simplicity, we neglect the effects from gravity and capillary forces and assume that the porosity is a constant. The two phases will be referred to as water and oil, denoted by w and o, respectively. The total Darcy velocity $\mathbf{u}$ and the pressure p satisfy [26]

$$\nabla \cdot \mathbf{u} = 0, \ \mathbf{u} = -\alpha(\mathbf{x}, \boldsymbol{\omega})\lambda_t \nabla p \quad \forall x \in D, \tag{3.2}$$

with the following boundary conditions:

$$p = \bar{p} \text{ on } \partial D_p \text{ and } \mathbf{u} \cdot \mathbf{n} = \bar{\mathbf{u}} \text{ on } \partial D_u. \tag{3.3}$$

The total velocity $u = u_o + u_w$ is the sum of the velocities of oil, $\mathbf{u}_o$, and water, $\mathbf{u}_w$. The random permeability $a(\mathbf{x}, \cdot)$ is assumed to be diagonal and uniformly positive definite. In addition, we will assume that $a(\mathbf{x}, \cdot)$ is a stochastic scalar function. The total mobility is given by $\lambda_t = \lambda_w + \lambda_o$, where λ_i models the reduced mobility of phase i due to the presence of the other phase. Without loss of generality, we assume that the boundary conditions are deterministic and that $\bar{\mathbf{u}} \cdot \mathbf{n} = 0$ on ∂D_u, where $\mathbf{n}$ is the unit normal of ∂D_u. Furthermore, we use the unit mobility displacement model, i.e., $\lambda_w = S, \lambda_o = 1 - S$, and hence $\lambda_t = 1$, where S is the water saturation. Under these assumptions, the water saturation equation is given by

$$\frac{\partial S(\mathbf{x}, t, \boldsymbol{\omega})}{\partial t} + \mathbf{u} \cdot \nabla S(\mathbf{x}, t, \boldsymbol{\omega}) = 0 \ \ \forall x \in D, t \in [0, T]. \tag{3.4}$$

Geostatistical models often suggest that the permeability field is a weakly stationary second-order random field such that the mean log-permeability $G(\mathbf{x}, \cdot) = \ln a(\mathbf{x}, \cdot)$ is constant and its covariance function only depends on the relative distance of two points [5]:

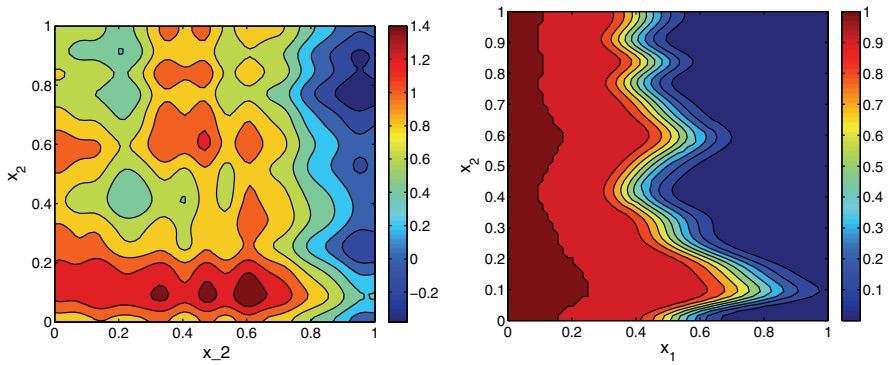
$$\mathrm{Cov}(\mathbf{x}, \mathbf{y}) = \exp\left\{-\frac{|x_1 - y_1|}{L_1} - \frac{|x_2 - y_2|}{L_2}\right\},$$

where $L_i, i = 1, 2$, are the correlation lengths. Employing the finite-dimensional noise assumption [43] and the Karhunen–Loève (KL) expansion [15], we approximate the log permeability field via a finite-dimensional representation

$$G(\mathbf{x}, \boldsymbol{\omega}) = \sum_{k=1}^{K} \sqrt{\lambda_k}\phi_k(\mathbf{x})\omega_k, \tag{3.5}$$

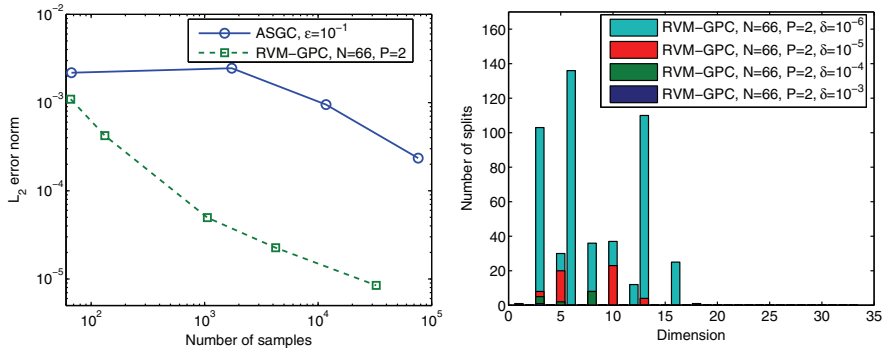
where ω_k are uncorrelated random variables, while $\phi_k(\mathbf{x})$ and λ_k are the eigenfunctions and eigenvalues of the covariance function, respectively, which are analytically available [27]. According to the KL expansion, $\omega_k \sim \mathcal{N}(0, 1)$. However, we may assume that $\omega_k \sim U(-1, 1)$ without losing the main features of the output uncertainty [28]. Note that such a restriction is not necessary for our approach (we could just as well use the Gaussian distribution), but we use it so that we can solve the same problem using ASGC. The deterministic problem for the velocity defined in (3.3) is solved with a mixed finite element method [11, 31] on the spatial domain $D = [0, 1]^2$ utilizing a 64×64 fine grid. The boundary conditions are set by fixing the pressure to 1 on the left boundary and 0 on the right boundary and using $\bar{\mathbf{u}} \cdot \mathbf{n} = 0$ on the top and the bottom. Given the velocity field $\mathbf{u}$, we solve the saturation equation (3.4) following a discontinuous Galerkin approach with piecewise-constant elements [7] coupled with a simple Euler scheme. The initial saturation is set to 0, while it is kept fixed to 1 on the left side of the boundary. Our C++ solver is built upon FEniCS [29]. The response is taken to be the value of the saturation S at time $t = 0.5$ PVI (see [31] for the definition of PVI) on each finite element node; that is, the problem has $M = 64 \times 64 = 4096$ output dimensions.

The stochastic problem is solved for correlation lengths $L_i = 1$ $(i = 1, 2)$, requiring $K = 33$ input dimensions to account for 95% of the field's energy. In Figure 3.6, we plot a sample permeability field along with the corresponding saturation. In this



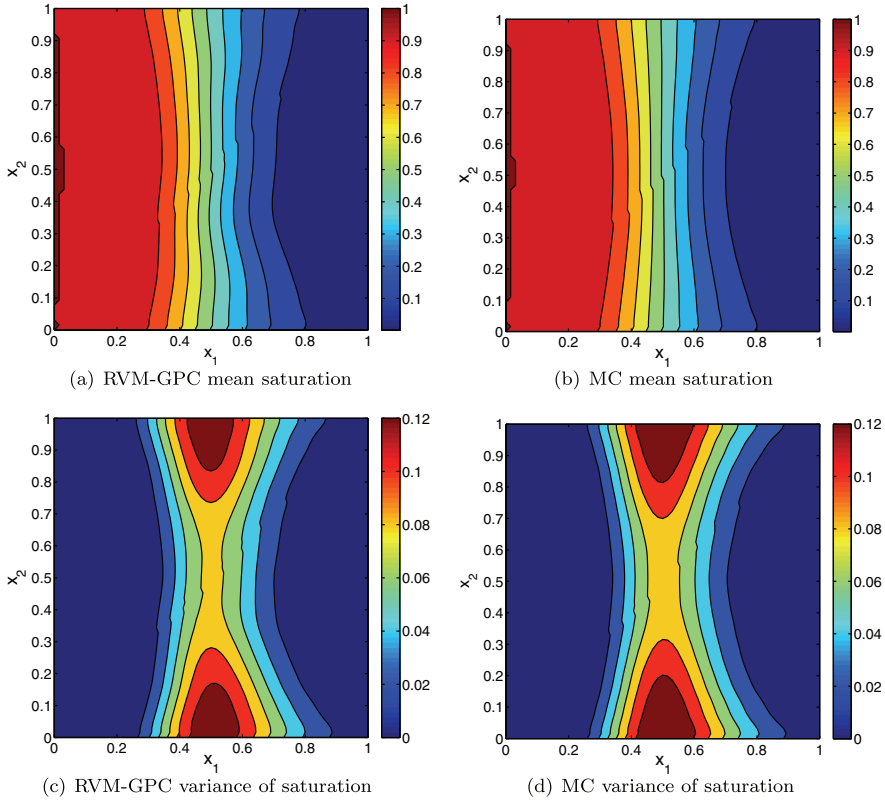
(a) Sample log permeability (b) Sample saturation

FIG. 3.6. *Saturation field:* (a) *A random sample of the permeability.* (b) *The corresponding saturation at* $t = 0.5$ *PVI.*



(a) L_2 error (b) Splitting dimensions

FIG. 3.7. *Saturation field:* (a) *The L_2 error in variance for RVM-GPC ($N = 66$) with maximum polynomial degree $P = 2$ and ASGC.* (b) *The number of splits per dimension performed by RVM for a tolerance of $\delta = 10^{-6}$.*



(a) RVM-GPC mean saturation (b) MC mean saturation

(c) RVM-GPC variance of saturation (d) MC variance of saturation

FIG. 3.8. *Saturation field: The left column shows the statistics predicted by RVM-GPC ($N = 33$ with maximum polynomial degree $P = 2$ at a tolerance $\delta = 10^{-6}$), while the right column shows the corresponding results using* 860,160 *MC samples. The top row (*(a) *and* (b)*) shows the mean and the bottom row (*(c) *and* (d)*) the variance of the saturation.*

case, we use only RVM-GPC with $N = 66$ samples per element and polynomials of maximum degree $P = 2$. In Figure 3.7(a) we plot the L_2 error in variance as a function of the number of observations and compare it to ASGC with $\epsilon = 10^{-1}$.

The reference variance was calculated using an MC estimator based on 860,160 samples. It is clear that RVM-GPC outperforms ASGC by at least two orders of magnitude. Figure 3.7(b) shows the number of splits per dimension performed by RVM-GPC. As in the results of the elliptic problem, we observe that RVM-GPC puts more weight on the first few dimensions while ignoring the rest. Finally, Figure 3.8 compares the predicted statistics with the corresponding MC estimates.

4. Conclusions. We have developed a UQ framework based on local Bayesian regression models. The stochastic space is adaptively decomposed in fine elements based solely on information conveyed by the local regressors. The adaptivity criteria developed are applicable to arbitrary input probability distributions and any local Bayesian regression model. The Bayesian regression we used was based on an extension of the RVM model that accounts for the multiple dimensions of the output (MRVM). A fast algorithm was developed to train MRVM on a given data set that does not require matrix inversions. The Bayesian nature of the scheme allowed us to sample from the predictive distribution of the desired statistics, thereby quantifying the epistemic uncertainty introduced by replacing the deterministic solver with a surrogate. The scheme was demonstrated through various numerical examples, and its ability to capture discontinuities was verified. In high-dimensional input settings ($K > 20$), the optimal orthogonal polynomial basis performed much better, especially in correctly identifying the PDFs of the various outputs. In the future, we plan to investigate the way the choice of N affects the results and identify ways for an automatic optimal choice for it.

Appendix A. Splitting the evidence. What follows is basically a generalization to the multioutput case of the ideas found in [37]. By making repeated use of the matrix determinant lemma [23] and the Woodbury matrix identity [22], it is possible to show that

$$\begin{aligned}\mathcal{E}(\boldsymbol{\alpha}|\mathcal{D}_{\mathrm{sc}}) = &-\frac{1}{2}\log 2\pi \\ &-\frac{1}{2N}\left(\log|\mathbf{C}_{-s}| - \log\alpha_s + \log|\alpha_s + \phi_s^T\mathbf{C}_{-s}^{-1}\boldsymbol{\phi}_s|\right) \\ &-\frac{1}{2MN}\sum_{r=1}^{M}\mathbf{z}_r^T\left(\mathbf{C}_{-s}^{-1} - \frac{\mathbf{C}_{-s}^{-1}\boldsymbol{\phi}_s\boldsymbol{\phi}_s^T\mathbf{C}_{-s}^{-1}}{\alpha_s + \boldsymbol{\phi}_s^T\mathbf{C}_{-s}^{-1}\boldsymbol{\phi}_s}\right)\mathbf{z}_r \\ = &\,\mathcal{E}(\boldsymbol{\alpha}_{-s}) + \epsilon(\alpha_s),\end{aligned}$$

where $\mathcal{E}(\boldsymbol{\alpha}_{-s})$ is the evidence with α_s removed and $\epsilon(\alpha_s)$ is given by

$$\epsilon(\alpha_s) = \frac{1}{2N}\log\alpha_s - \frac{1}{2N}\log|\alpha_s + h_s| + \frac{1}{2MN}\frac{\sum_{r=1}^{M} q_{rs}^2}{\alpha_s + h_s}, \tag{A.1}$$

where we have introduced the intermediate statistics

$$h_s = \boldsymbol{\phi}_s^T\mathbf{C}_{-s}^{-1}\boldsymbol{\phi}_s \ \text{ and } \ q_{rs} = \boldsymbol{\phi}_s^T\mathbf{C}_{-s}^{-1}\mathbf{z}_r. \tag{A.2}$$

In practice, it is more convenient to keep track of the following statistics:

$$H_s = \boldsymbol{\phi}_s^T\mathbf{C}^{-1}\boldsymbol{\phi}_s \ \text{ and } \ Q_{rs} = \boldsymbol{\phi}_s^T\mathbf{C}^{-1}\mathbf{z}_r, \tag{A.3}$$

since

$$h_s = \frac{\alpha_s H_s}{\alpha_s - H_s} \ \text{ and } \ q_{rs} = \frac{\alpha_s Q_{rs}}{\alpha_s - H_s}. \tag{A.4}$$

Appendix B. Stationary points of $\epsilon(\alpha_s)$. The derivative of $\epsilon(\alpha_s)$ is

$$\frac{\partial \epsilon(\alpha_s)}{\partial \alpha_s} = \frac{h_s^2 - \alpha_s \left(\frac{1}{M} \sum_{r=1}^{M} q_{rs}^2 - h_s \right)}{2\alpha_s N(\alpha_s + h_s)^2} = \frac{h_s^2 - \alpha_s \theta_s}{2\alpha_s N(\alpha_s + h_s)^2}, \tag{B.1}$$

where

$$\theta_s = \frac{1}{M} \sum_{r=1}^{M} q_{rs}^2 - h_s. \tag{B.2}$$

We are basically interested in maximizing $\epsilon(\alpha_s)$ with respect to $\alpha_s > 0$. We observe that there exist two possible cases:

1. If $\theta_s > 0$, then $\epsilon(\alpha_s)$ has a unique stationary point at

   $$\alpha_s = \frac{h_s^2}{\theta_s}. \tag{B.3}$$

   By evaluating the second derivative of $\epsilon(\alpha_s)$ at $\alpha_s = h_s^2/\theta_s$, it is straightforward to check that this is indeed the maximum of the function.
2. If $\theta_s \leq 0$, then $\epsilon(\alpha_s)$ is maximized at

   $$\alpha_s = +\infty, \tag{B.4}$$

   since $\partial\epsilon/\partial\alpha_s$ is positive for all $\alpha_s > 0$. This effectively removes the sth basis function from the model, since its corresponding weight is identically set to zero.

Appendix C. Possible actions. Whenever some $\alpha_s = +\infty$, the corresponding basis functions can be thought of as being out of the model. Assume that currently there are $1 \leq S_{\text{in}} \leq S$ basis functions in the model. The design matrix $\boldsymbol{\Phi}$ is built only upon the basis functions $\boldsymbol{\phi}_s$ for which $\alpha_s < \infty$, i.e., it is an $N \times S_{\text{in}}$ matrix. The mean of the weights $\boldsymbol{\mu}_r$ and their covariance matrix $\boldsymbol{\Sigma}$ defined in (2.18) are an S_{in}-dimensional vector and an $S_{\text{in}} \times S_{\text{in}}$ matrix, respectively. Also, assume that the statistics H_s, Q_{rs} given in (A.3) are already calculated for all $s = 1, \ldots, S$. At each step of the algorithm we update a single α_s. There are three possible actions: (1) *add* a new basis function to the model, (2) *reestimate* the hyperparameter of an existing basis function, and (3) *remove* a basis function from the model. The action that results in the maximum change in the evidence $\Delta\mathcal{E}^{\text{action}} = \mathcal{E}(\boldsymbol{\alpha}^{\text{new}}) - \mathcal{E}(\boldsymbol{\alpha})$ is selected. For completeness, we list how the possible actions can be distinguished, how the change in evidence can be calculated, and how the various statistics can be updated iteratively:

1. If $\theta_s > 0$ and $\alpha_s = \infty$, then $\boldsymbol{\phi}_s$ is a candidate for addition. The value for α_s maximizing the evidence is

   $$\alpha_s^{\text{new}} = \frac{h_s^2}{\theta_s},$$

   yielding a change in evidence

   $$\Delta\mathcal{E}^{\text{add}} = \epsilon(\alpha_s^{\text{new}}).$$
2. If $\theta_s > 0$ and $\alpha_s < \infty$, then $\boldsymbol{\phi}_s$ is a candidate for reestimation. The value for α_s maximizing the evidence is

   $$\alpha_s^{\text{new}} = \frac{h_s^2}{\theta_s},$$

   yielding a change in evidence

$$\Delta\mathcal{E}^{\text{reestimate}} = \epsilon(\alpha_s^{\text{new}}) - \epsilon(\alpha_s).$$

3. If $\theta_s \leq 0$ and $\alpha_s < \infty$, then $\boldsymbol{\phi}_s$ is a candidate for removal. The value for α_s maximizing the evidence is

$$\alpha_s^{\text{new}} = \infty,$$

yielding a change in evidence

$$\Delta\mathcal{E}^{\text{reestimate}} = -\epsilon(\alpha_s).$$

Appendix D. Implementation details. Let us consider again the matrix $\mathbf{C}$ defined in (2.24). As before, let $\boldsymbol{\Phi}$ be the design matrix of all relevant basis functions (the ones for which $\alpha_s < \infty$). These are the only ones contributing to $\mathbf{C}$, and without loss of generality we may assume that the rest are not present in the model. Using the Woodbury matrix identity [22], we can show that

$$\mathbf{C}^{-1} = \beta\mathbf{I} - \beta\boldsymbol{\Phi}\left(\beta^{-1}\operatorname{diag}(\boldsymbol{\alpha}) + \boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T,$$

where, of course, $\boldsymbol{\alpha}$ is the vector of only the finite hyperparameters. Define the $S_{\text{in}} \times S_{\text{in}}$ matrix $\mathbf{A}$ by

$$\mathbf{A} = \beta^{-1/2}\operatorname{diag}(\boldsymbol{\alpha})^{1/2},$$

and notice that the part that needs to be inverted has the familiar form $\mathbf{A}^T\mathbf{A} + \boldsymbol{\Phi}^T\boldsymbol{\Phi}$ found in regularized least squares problems. This suggests using the GSVD of the pair $(\mathbf{A}, \boldsymbol{\Phi})$ in order to carry out the computations. The GSVD is given by

$$\boldsymbol{\Phi} = \mathbf{U}\boldsymbol{\Sigma}_1[0\ \mathbf{R}]\mathbf{Q}^T \text{ and } \mathbf{A} = \mathbf{V}\boldsymbol{\Sigma}_2[0\ \mathbf{R}]\mathbf{Q}^T,$$

where $\mathbf{U} \in \mathbb{R}^{N\times N}$, $\mathbf{V} \in \mathbb{R}^{S_{\text{in}}\times S_{\text{in}}}$, and $\mathbf{Q} \in \mathbb{R}^{S_{\text{in}}\times S_{\text{in}}}$ are orthogonal; $\boldsymbol{\Sigma}_1 \in \mathbb{R}^{N\times r}$ and $\boldsymbol{\Sigma}_2 \in \mathbb{R}^{S_{\text{in}}\times r}$ are zero except for the diagonal and have the property

$$\boldsymbol{\Sigma}_1^T\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2^T\boldsymbol{\Sigma}_2 = I;$$

$\mathbf{R} \in \mathbb{R}^{r\times r}$ is upper triangular and nonsingular; and r is the rank of $[\boldsymbol{\Phi}^T\ \mathbf{A}^T]$. When the number of samples is greater than or equal to the relevant basis functions, i.e., $N \geq S_{\text{in}}$, then $r = S_{\text{in}}$, since $\mathbf{A}$ is nonsingular. We will always assume that this is the case, since it does not make sense to include in the model more basis functions than observations. We want to make clear at this point the distinction between the pool of basis functions from which we choose and the relevant basis functions. The former can be as large as we want, but the latter should be less than or equal to the number of available observations. In this case, the matrix of zeros disappears and we obtain

$$\boldsymbol{\Phi} = \mathbf{U}\boldsymbol{\Sigma}_1\mathbf{R}\mathbf{Q}^T \text{ and } \mathbf{A} = \mathbf{V}\boldsymbol{\Sigma}_2\mathbf{R}\mathbf{Q}^T.$$

We can now notice that

$$\mathbf{A}^T\mathbf{A} + \boldsymbol{\Phi}^T\boldsymbol{\Phi} = \mathbf{Q}\mathbf{R}^T\mathbf{R}\mathbf{Q}^T,$$

and as a result

$$\mathbf{C}^{-1} = \beta\left(\mathbf{I} - \left(\boldsymbol{\Sigma}_1^T\mathbf{U}^T\right)^T\left(\boldsymbol{\Sigma}_1^T\mathbf{U}^T\right)\right).$$

The H_s statistic now takes the form

$$H_s = \beta \left(\| \boldsymbol{\phi}_s \|_2^2 - \| \boldsymbol{\Sigma}_1^T \mathbf{U}^T \boldsymbol{\phi}_s \|_2^2 \right). \tag{D.1}$$

Similarly, the Q_{rs} statistic becomes

$$Q_{rs} = \beta \left(\phi_s^T \mathbf{z}_r - \left(\boldsymbol{\Sigma}_1^T \mathbf{U}^T \boldsymbol{\phi}_s \right)^T \left(\boldsymbol{\Sigma}_1^T \mathbf{U}^T \mathbf{z}_r \right) \right), \tag{D.2}$$

while the $\mathbf{z}_r^T \mathbf{C}^{-1} \mathbf{z}_r$ part of the evidence becomes

$$\mathbf{z}_r^T \mathbf{C}^{-1} \mathbf{z}_r = \beta \left(\| \mathbf{z}_r \|_2^2 - \| \boldsymbol{\Sigma}_1^T \mathbf{U}^T \mathbf{z}_r \|_2^2 \right). \tag{D.3}$$

The only part that remains in order to finalize the computation of the likelihood is $|\mathbf{C}|$. For this, we use the matrix-determinant lemma to obtain

$$\begin{aligned} |\mathbf{C}| &= |\beta^{-1}\mathbf{I} + \boldsymbol{\Phi} \operatorname{diag} \boldsymbol{\alpha}^{-1} \boldsymbol{\Phi}^T| \\ &= |\operatorname{diag} \boldsymbol{\alpha}^{-1}| \cdot |\beta^{-1} \operatorname{diag} \boldsymbol{\alpha} + \boldsymbol{\Phi}^T \boldsymbol{\Phi}| \beta^{S_{\text{in}} - N} \\ &= |\operatorname{diag} \boldsymbol{\alpha}^{-1}| \cdot |\mathbf{Q}\mathbf{R}^T \mathbf{R}\mathbf{Q}| \beta^{S_{\text{in}} - N} \\ &= |\operatorname{diag} \boldsymbol{\alpha}^{-1}| \cdot |\mathbf{R}|^2 \beta^{S_{\text{in}} - N}, \end{aligned}$$

since $\mathbf{Q}$ is orthogonal. This gives us

$$\log |\mathbf{C}| = -\sum_{i=1}^{S_{\text{in}}} \log \alpha_i + 2 \sum_{i=1}^{S_{\text{in}}} \log |R_{ii}| + (S_{\text{in}} - N) \log \beta. \tag{D.4}$$

Finally, predictions can be carried out easily by noticing that

$$\boldsymbol{\Sigma}^{-1} = \beta \left(\mathbf{A}^T \mathbf{A} + \boldsymbol{\Phi}^T \boldsymbol{\Phi} \right) = \beta \mathbf{Q}\mathbf{R}^T \mathbf{R}\mathbf{Q}^T.$$

For example, the mean of the weights can be found by solving the system

$$(\mathbf{R}\mathbf{Q}^T) \boldsymbol{\mu}_r = \boldsymbol{\Sigma}_1^T \mathbf{U}^T \mathbf{z}_r,$$

which is a simple operation since $\mathbf{R}$ is upper triangular and $\mathbf{Q}$ is orthogonal.

REFERENCES

[1] I. Babuška, F. Nobile, and R. Tempone, *A stochastic collocation method for elliptic partial differential equations with random input data*, SIAM J. Numer. Anal., 45 (2007), pp. 1005–1034.

[2] I. Bilionis, N. Zabaras, B. A. Konomi, and G. Lin, *Multi-output Gaussian process: Towards an efficient, fully Bayesian paradigm for uncertainty quantification*, J. Comput. Phys., 2012 submitted.

[3] I. Bilionis and N. Zabaras, *Multi-output local Gaussian process regression: Applications to uncertainty quantification*, J. Comput. Phys., 231 (2012), pp. 5718–5746.

[4] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag, New York, 2006.

[5] Z. Chen and T. Y. Hou, *A mixed multiscale finite element method for elliptic problems with oscillating coefficients*, Math. Comput., 72 (2003), pp. 541–576.

[6] C. Cortes and V. Vapnik, *Support-vector networks*, Machine Learning, 20 (1995), pp. 273–297.

[7] D. A. Di Pietro, S. Lo Forte, and N. Parolini, *Mass preserving finite element implementations of the level set method*, Appl. Numer. Math., 56 (2006), pp. 1179–1195.

[8] J. Foo, X. Wan, and G. E. Karniadakis, *The Multi-element probabilistic collocation method (ME-PCM): Error analysis and applications*, J. Comput. Phys., 227 (2008), pp. 9572–9595.
[9] D. Francois, V. Wertz, and M. Verleysen, *About the locality of kernels in high-dimensional spaces*, in Proceedings of the International Symposium on Applied Stochastic Models and Data Analysis, Brest, France, 2005, pp. 238–245.
[10] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, and F. Rossi, *GNU Scientific Library Reference Manual*, http://WWW.gnn.org/software/gsl/manual/html-node (2009).
[11] B. Ganapathysubramanian and N. Zabaras, *A stochastic multiscale framework for modeling flow through heterogeneous porous media*, J. Comput. Phys., 228 (2009), pp. 591–618.
[12] W. Gautschi, *Algorithm 726: ORTHPOL—A package of routines for generating orthogonal polynomials and Gauss-type quadrature rules*, ACM Transa. Math. Software, 20 (1994), pp. 21–62.
[13] T. Gerstner and M. Griebel, *Numerical integration using sparse grids*, Numer. Algorithms, 18 (1998), pp. 209–232.
[14] R. G. Ghanem and A. Doostan, *On the construction and analysis of stochastic models: Characterization and propagation of the errors associated with limited data*, J. Comput. Phys., 217 (2006), pp. 63–81.
[15] R. G. Ghanem and P. D. Spanos, *Stochastic Finite Elements: A Spectral Approach*, Dover Publications, New York, 2003.
[16] M. B. Giles, *Multilevel Monte Carlo Path Simulation*, Technical report, Oxford University Computing Laboratory, Oxford, UK, 2006.
[17] M. B. Giles, *Improved multilevel Monte Carlo convergence using the Milstein scheme*, in Monte Carlo and Quasi-Monte Carlo Methods 2006, A. Keller, S. Heinrich, and H. Niederreiter, eds., Springer-Verlag, Berlin, 2008, pp. 343–358.
[18] G. H. Golub and C. F. Van Loan, *Matrix Computations,* 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
[19] R. B. Gramacy and H. K. H. Lee, *Bayesian treed Gaussian process models with an application to computer modeling*, J. Amer. Statist. Assoc., 103 (2008), pp. 1119–1130.
[20] R. B. Gramacy and H. K. Lee, *Cases for the nugget in modeling computer experiments*, Statist. Comput., 22 (2012), pp. 713–722.
[21] M. Griebel and G. W. Zumbusch, *Adaptive sparse grids for hyperbolic conservation laws*, in Hyperbolic Problems: Theory, Numerics, Applications, 7th International Conference in Zürich, February 1998, M. Fey and R. Jeltsch, eds., Internat. Ser. Numer. Math. 129, Birkhäuser, Basel, Switzerland, 1999, pp. 411–422.
[22] W. W. Hager, *Updating the inverse of a matrix*, SIAM Rev., 31 (1989), pp. 221–239.
[23] D. A. Harville, *Matrix Algebra from a Statistician's Perspective*, Springer-Verlag, New York, 1997.
[24] M. A. Heroux and J. M. Willenbring, *Trilinos Users Guide*, Technical report, Sandia National Laboratories, 2003.
[25] R. L. Iman and W. J. Conover, *Small sample sensitivity analysis techniques for computer models, with an application to risk assessment*, Comm. Statist. Theory Methods, 9 (1980), pp. 1749–1842.
[26] V. Kippe, J. E. Aarnes, and K. A. Lie, *A comparison of multiscale methods for elliptic problems in porous media flow*, Comput. Geosci., 12 (2008), pp. 377–398.
[27] G. Lin and A. Tartakovsky, *An efficient, high-order probabilistic collocation method on sparse grids for three-dimensional flow and solute transport in randomly heterogeneous porous media*, Advances in Water Resources, 32 (2009), pp. 712–722.
[28] G. Lin and A. Tartakovsky, *Numerical studies of three-dimensional stochastic Darcy's equation and stochastic advection-diffusion-dispersion equation*, J. Sci. Comput., 43 (2010), pp. 92–117.
[29] A. Logg, K.-A. Mardal, and G. N. Wells, eds., *Automated Solution of Differential Equations by the Finite Element Method*, Lect. Notes Comput. Sci. Eng. 84, Springer-Verlag, New York, 2012.
[30] X. Ma and N. Zabaras, *An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations*, J. Comput. Phys., 228 (2009), pp. 3084–3113.
[31] X. Ma and N. Zabaras, *A stochastic mixed finite element heterogeneous multiscale method for flow in porous media*, J. Comput. Phys., 230 (2011), pp. 4696–4722.
[32] F. Nobile, R. Tempone, and C. G. Webster, *A sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM J. Numer. Anal., 46 (2008), pp. 2309–2345.

[33] R. Piessens, E. de Doncker-Kapenga, C. W. Ueberhuber, and D. K. Kahaner, *QUADPACK: A Subroutine Package for Automatic Integration*, Springer-Verlag, New York, 1983.
[34] M. Rosenblatt, *Remarks on a multivariate transformation*, The Annals of Mathematical Statistics, 23 (1952), pp. 470–472.
[35] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman & Hall/CRC, London, 1998.
[36] S. A. Smolyak, *Quadrature and interpolation formulas for tensor products of certain classes of functions*, Dokl. Akad. Nauk SSSR, 4 (1963), p. 123.
[37] M. E. Tipping and A. Faul, *Fast marginal likelihood maximisation for sparse Bayesian models*, in Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, vol. 1, 2003, pp. 1–13.
[38] M. E. Tipping, *Sparse Bayesian learning and the relevance vector machine*, J. Mach. Learn. Res., 1 (2001), pp. 211–245.
[39] X. Wan and G. E. Karniadakis, *An adaptive multi-element generalized polynomial chaos method for stochastic differential equations*, J. Comput. Phys., 209 (2005), pp. 617–642.
[40] X. Wan and G. E. Karniadakis, *Multi-element generalized polynomial chaos for arbitrary probability measures*, SIAM J. Sci. Comput., 28 (2006), pp. 901–928.
[41] D. Xiu and J. S. Hesthaven, *High-order collocation methods for differential equations with random inputs*, SIAM J. Sci. Comput., 27 (2005), pp. 1118–1139.
[42] D. Xiu and G. E. Karniadakis, *The Wiener–Askey polynomial chaos for stochastic differential equations*, SIAM J. Sci. Comput., 24 (2002), pp. 619–644.
[43] D. Xiu, *High-order collocation methods for differential equations with random inputs*, SIAM J. Sci. Comput., 27 (2005), pp. 1118–1139.
[44] D. Xiu, *Efficient collocational approach for parametric uncertainty analysis*, Commun. Comput. Phys., 2 (2007), pp. 293–309.