MIT Open Access Articles

Active Subspace Methods in Theory and Practice: Applications to Kriging Surfaces

Citation	Constantine, Paul G., Eric Dow, and Qiqi Wang. "Active Subspace Methods in Theory and Practice: Applications to Kriging Surfaces." SIAM Journal on Scientific Computing 36, no. 4 (January 2014): A1500–A1524. © 2014 Society for Industrial and Applied Mathematics
As Published	http://dx.doi.org/10.1137/130916138
Publisher	Society for Industrial and Applied Mathematics
Version	Final published version
Citable link	http://hdl.handle.net/1721.1/92546
Terms of Use	Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.

ACTIVE SUBSPACE METHODS IN THEORY AND PRACTICE: APPLICATIONS TO KRIGING SURFACES[*]

PAUL G. CONSTANTINE[†], ERIC DOW[‡], AND QIQI WANG[‡]

Abstract. Many multivariate functions in engineering models vary primarily along a few directions in the space of input parameters. When these directions correspond to coordinate directions, one may apply global sensitivity measures to determine the most influential parameters. However, these methods perform poorly when the directions of variability are not aligned with the natural coordinates of the input space. We present a method to first detect the directions of the strongest variability using evaluations of the gradient and subsequently exploit these directions to construct a response surface on a low-dimensional subspace—i.e., the *active subspace*—of the inputs. We develop a theoretical framework with error bounds, and we link the theoretical quantities to the parameters of a kriging response surface on the active subspace. We apply the method to an elliptic PDE model with coefficients parameterized by 100 Gaussian random variables and compare it with a local sensitivity analysis method for dimension reduction.

Key words. active subspace methods, kriging, Gaussian process, uncertainty quantification, response surfaces

AMS subject classification. 65C60

DOI. 10.1137/130916138

1. Introduction and motivation. As computational models of physical systems become more complex, the need increases for uncertainty quantification (UQ) to enable defensible predictions. Monte Carlo methods are the workhorse of UQ, where model inputs are sampled according to a characterization of their uncertainty and the corresponding model outputs are treated as a data set for statistical analysis. However, the slow convergence of Monte Carlo methods coupled with the high computational cost of the models has led many to employ response surfaces trained on a few carefully selected runs in place of the full model. This strategy has had great success in forward [19, 26] and inverse uncertainty propagation problems [29, 4, 12] as well as optimization [24, 39]. However, most response surfaces suffer from the curse of dimensionality, where the cost of constructing an accurate surface increases exponentially as the dimension (i.e., the number of input parameters) increases.

To make construction tractable, one may first perform sensitivity analysis [36] to determine which variables have the most influence on the model predictions. With a ranking of the inputs, one may construct response surfaces that concentrate the approximation on the most influential variables, e.g., through a suitably anisotropic design; the same concept applies to mesh refinement strategies for solving PDEs. Methods for sensitivity analysis are typically classified as local perturbation or global methods. Local methods perturb the inputs—often along coordinate directions—around a nominal value and measure the effects on the outputs. Though relatively inexpensive, local methods are fraught with difficulties like sensitivity to noise and the
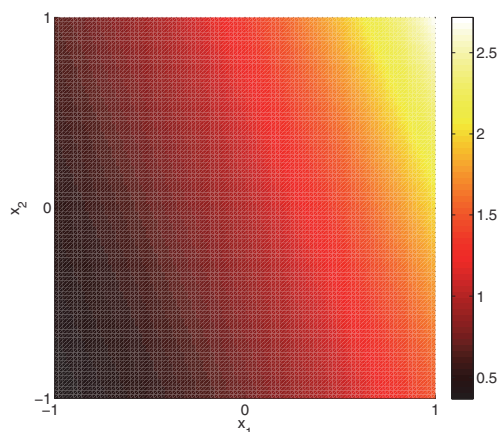
FIG. 1. *The function* $f(x_1, x_2) = \exp(0.7x_1 + 0.3x_2)$ *varies strongest along the direction* $[0.7, 0.3]$, *and it is flat in the direction* $[0.3, -0.7]$. *(Colors are visible in the electronic version.)*

choice of the perturbation step. Also, the local sensitivity measured at the nominal condition may be very different elsewhere in the parameter space. Global methods address these issues by providing integrated measures of the output's variability over the full range of parameters; consequently, they are computationally more expensive. Methods based on variance decompositions [30, 36] require approximating high-dimensional integrals in order to rank the inputs. Consider the simple function $f(x_1, x_2) = \exp(0.7x_1 + 0.3x_2)$ defined on $[-1, 1]^2$ plotted in Figure 1. A local perturbation method at the origin with a stepsize $\Delta x = 0.1$ reveals $f(\Delta x, 0) = 1.0725$ and $f(0, \Delta x) = 1.0305$ to four digits. The larger effect of the perturbation in x_1 designates x_1 more important than x_2. The global Sobol indices [30] for the main (univariate) effects of the ANOVA decomposition are $\sigma_1 = 0.1915$ and $\sigma_2 = 0.0361$, which yields a similar conclusion regarding x_1's importance.

Both classes of methods rank the coordinates of the inputs. However, some models may vary most prominently along directions of the input space that are not aligned with the coordinate system. The example f plotted in Figure 1 varies strongest along the direction $[0.7, 0.3]$, and it is flat along the direction $[-0.3, 0.7]$. This bivariate function is in effect univariate once the coordinate system has been rotated appropriately. This suggests an alternative form of dimension reduction: rotate the coordinates such that the directions of the strongest variation are aligned with the rotated coordinates, and construct a response surface using only the most important rotated coordinates.

We propose a method based on gradient evaluations for detecting and exploiting the directions of strongest variability of a given function to construct an approximation on a low-dimensional subspace of the function's inputs. Given continued interest in gradient computations based on adjoint methods [5, 22] and algorithmic differentiation [20], it is not unreasonable to assume that one has access to the gradient of the function. We detect the directions by evaluating the function's gradient at a set of input points and determining a rotation of the input space that separates the directions of relative variability from directions of relative flatness. We exploit these directions by first projecting the input space to the low-dimensional subspace that captures the function's variability and then approximating the function on the

subspace. Following Russi's 2010 Ph.D. thesis [34], we call this low-dimensional subspace the *active subspace*.

Subspace approximations are commonly used in optimization, where local quadratic models of a function are decomposed to reveal search directions [18]. They are also found in many areas of model reduction [1] and optimal control [41], where a high-dimensional state space vector is approximated by a linear combination of relatively few basis vectors. Common to both of these fields are methods for matrix factorizations and eigenvalue computations [35], which are replete with subspace oriented approaches. The use of subspace methods for approximating high-dimensional functions arising in science and engineering models appears rare by comparison. Recent work by Lieberman, Willcox, and Ghattas [27] describes a method for finding a subspace in a high-dimensional input space via a greedy optimization procedure. Russi [34] proposes a method for discovering the *active subspace* of a function using evaluations of the gradient and constructing a quadratic response surface on the subspace; his methodology is similar to ours in practice. Recently Fornasier, Schnass, and Vybiral [16] analyzed subspace approximation algorithms that do not need gradient evaluations but make strong assumptions on the function they are approximating; they take advantage of results from compressed sensing. Our previous work has applied the active subspace method to design optimization [8, 15], inverse analysis [12], and spatial sensitivity [13].

The contribution of this paper is twofold. First we provide a theoretical foundation for gradient-based dimension reduction and subspace approximation. We construct and factorize a covariance-like matrix of the gradient to determine the directions of variability. These directions define a new set of coordinates, which we separate into a set $\mathbf{y}$ along which the function varies the strongest and a set $\mathbf{z}$ along which the function varies relatively little on average. We then approximate the function by a sequence of three functions that are $\mathbf{z}$-*invariant*, i.e., that are functions of only the $\mathbf{y}$ coordinates. The first is a theoretical best approximation via conditional expectation. The second approximates the conditional expectation with a Monte Carlo method. The third builds a response surface on the $\mathbf{y}$ coordinates using a few evaluations of the Monte Carlo approximations. We provide error bounds for these approximations, and we examine the effects of using directions that are slightly perturbed. Second, we provide a bridge between the theoretical analysis and computational practice by (i) relating the derived error bounds to SVD-based approaches for discovering the active subspace and (ii) heuristically linking the theoretical quantities to the parameters of a kriging surface constructed on the active subspace. We apply this procedure to an elliptic PDE model with a 100-parameter model for the coefficients and a scalar quantity of interest. We compare the active subspace approach to a dimension reduction approach based on local sensitivity analysis.

2. Active subspaces and z-invariance. In this section, we describe the class of functions that vary primarily along a few directions of the input space. We characterize the active subspace and discuss a computational procedure for approximating its basis. We perform the analysis using tools from probability theory such as expectation $\mathbb{E}[\cdot]$, but we emphasize that there is nothing inherently stochastic about the functions or the approximations; the probability notation provides a convenient shorthand.

Consider a function f with m continuous inputs

$$(2.1) \qquad\qquad f = f(\mathbf{x}), \quad \mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^m,$$

where we assume without loss of generality that $\mathcal{X}$ is centered at the origin. Let $\mathcal{X}$ be equipped with a bounded probability density function $\rho : \mathbb{R}^m \to \mathbb{R}_+$, where

$$(2.2) \qquad \rho(\mathbf{x}) > 0, \mathbf{x} \in \mathcal{X} \quad \text{and} \quad \rho(\mathbf{x}) = 0, \mathbf{x} \notin \mathcal{X}.$$

We assume that f is absolutely continuous and square-integrable with respect to ρ. Denote the gradient of f by the column vector $\nabla_{\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_m} \end{bmatrix}^T$. Define the $m \times m$ matrix $\mathbf{C}$ by

$$(2.3) \qquad \mathbf{C} = \mathbb{E}\left[(\nabla_{\mathbf{x}} f)(\nabla_{\mathbf{x}} f)^T \right],$$

where we assume that f is such that $\mathbf{C}$ exists; in other words, the products of partial derivatives are integrable. $\mathbf{C}$ can be interpreted as the uncentered covariance of the gradient vector. Note that $\mathbf{C}$ is symmetric and positive semidefinite, so it admits a real eigenvalue decomposition,

$$(2.4) \qquad \mathbf{C} = \mathbf{W} \Lambda \mathbf{W}^T, \quad \Lambda = \text{diag}\,(\lambda_1, \ldots, \lambda_m), \quad \lambda_1 \geq \cdots \geq \lambda_m \geq 0.$$

The following lemma quantifies the relationship between the gradient of f and the eigendecomposition of $\mathbf{C}$.

LEMMA 2.1. *The mean-squared directional derivative of f with respect to the eigenvector $\mathbf{w}_i$ is equal to the corresponding eigenvalue,* $\mathbb{E}\left[((\nabla_{\mathbf{x}} f)^T \mathbf{w}_i)^2 \right] = \lambda_i$.

Proof. By the definition of $\mathbf{C}$,

$$(2.5) \qquad \lambda_i = \mathbf{w}_i^T \mathbf{C} \mathbf{w}_i = \mathbf{w}_i^T \left(\mathbb{E}\left[(\nabla_{\mathbf{x}} f)(\nabla_{\mathbf{x}} f)^T \right] \right) \mathbf{w}_i = \mathbb{E}\left[((\nabla_{\mathbf{x}} f)^T \mathbf{w}_i)^2 \right],$$

as required. $\square$

The eigenvectors $\mathbf{W}$ define a rotation of $\mathbb{R}^m$ and consequently the domain of f. With eigenvalues in decreasing order, we can separate components of the rotated coordinate system into a set that corresponds to greater average variation and a set corresponding to smaller average variation. The eigenvalues and eigenvectors are partitioned

$$(2.6) \qquad \Lambda = \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix}, \qquad \mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_2 \end{bmatrix},$$

where $\Lambda_1 = \text{diag}\,(\lambda_1, \ldots, \lambda_n)$ with $n < m$, and $\mathbf{W}_1$ is $m \times n$. Define the rotated coordinates $\mathbf{y} \in \mathbb{R}^n$ and $\mathbf{z} \in \mathbb{R}^{m-n}$ by

$$(2.7) \qquad \mathbf{y} = \mathbf{W}_1^T \mathbf{x}, \quad \mathbf{z} = \mathbf{W}_2^T \mathbf{x}.$$

Then we have the following lemma.

LEMMA 2.2. *The mean-squared gradients of f with respect to the coordinates $\mathbf{y}$ and $\mathbf{z}$ satisfy*

$$(2.8) \qquad \begin{aligned} \mathbb{E}\left[(\nabla_{\mathbf{y}} f)^T (\nabla_{\mathbf{y}} f) \right] &= \lambda_1 + \cdots + \lambda_n, \\ \mathbb{E}\left[(\nabla_{\mathbf{z}} f)^T (\nabla_{\mathbf{z}} f) \right] &= \lambda_{n+1} + \cdots + \lambda_m. \end{aligned}$$

Proof. First note that we can write

$$(2.9) \qquad f(\mathbf{x}) = f(\mathbf{W}\mathbf{W}^T \mathbf{x}) = f(\mathbf{W}_1 \mathbf{W}_1^T \mathbf{x} + \mathbf{W}_2 \mathbf{W}_2^T \mathbf{x}) = f(\mathbf{W}_1 \mathbf{y} + \mathbf{W}_2 \mathbf{z}).$$

By the chain rule, the gradient of f with respect to $\mathbf{y}$ can be written

$$(2.10) \quad \nabla_{\mathbf{y}} f(\mathbf{x}) = \nabla_{\mathbf{y}} f(\mathbf{W}_1 \mathbf{y} + \mathbf{W}_2 \mathbf{z}) = \mathbf{W}_1^T \nabla_{\mathbf{x}} f(\mathbf{W}_1 \mathbf{y} + \mathbf{W}_2 \mathbf{z}) = \mathbf{W}_1^T \nabla_{\mathbf{x}} f(\mathbf{x}).$$

Then

$$
\begin{aligned}
(2.11) \qquad \mathbb{E}\left[(\nabla_{\mathbf{y}} f)^T (\nabla_{\mathbf{y}} f) \right] &= \mathbb{E}\left[\operatorname{trace}\left((\nabla_{\mathbf{y}} f)(\nabla_{\mathbf{y}} f)^T \right) \right] \\
&= \operatorname{trace}\left(\mathbb{E}\left[(\nabla_{\mathbf{y}} f)(\nabla_{\mathbf{y}} f)^T \right] \right) \\
&= \operatorname{trace}\left(\mathbf{W}_1^T \mathbb{E}\left[(\nabla_{\mathbf{x}} f)(\nabla_{\mathbf{x}} f)^T \right] \mathbf{W}_1 \right) \\
&= \operatorname{trace}\left(\mathbf{W}_1^T \mathbf{C} \mathbf{W}_1 \right) \\
&= \operatorname{trace}\left(\Lambda_1 \right) \\
&= \lambda_1 + \cdots + \lambda_n,
\end{aligned}
$$

as required. The derivation for the $\mathbf{z}$ components is similar. $\square$

Lemma 2.2 motivates the use of the label *active subspace*. In particular, f varies more on average along the directions defined by the columns of $\mathbf{W}_1$ than along the directions defined by the columns of $\mathbf{W}_2$, as quantified by the eigenvalues of $\mathbf{C}$. When the eigenvalues $\lambda_{n+1}, \ldots, \lambda_m$ are all zero, Lemma 2.2 implies that the gradient $\nabla_{\mathbf{z}} f$ is zero everywhere in $\mathcal{X}$. We call such functions $\mathbf{z}$-*invariant*. The next proposition shows that $\mathbf{z}$-invariant functions have both linear contours and linear isoclines. Similar arguments can be used for higher-order derivatives when they exist.

PROPOSITION 2.3. *Let f be $\mathbf{z}$-invariant, i.e., $\lambda_{n+1} = \cdots = \lambda_m = 0$. Then for any two points $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$ such that $\mathbf{W}_1^T \mathbf{x}_1 = \mathbf{W}_1^T \mathbf{x}_2$, $f(\mathbf{x}_1) = f(\mathbf{x}_2)$ and $\nabla_{\mathbf{x}} f(\mathbf{x}_1) = \nabla_{\mathbf{x}} f(\mathbf{x}_2)$.*

Proof. The gradient $\nabla_{\mathbf{z}} f$ being zero everywhere in $\mathcal{X}$ implies that $f(\mathbf{x}_1) = f(\mathbf{x}_2)$. To show that the gradients are equal, assume that $\mathbf{x}_1$ and $\mathbf{x}_2$ are on the interior of $\mathcal{X}$. Then for arbitrary $\mathbf{c} \in \mathbb{R}^m$, define

$$(2.12) \qquad \mathbf{x}_1' = \mathbf{x}_1 + \varepsilon \mathbf{c}, \qquad \mathbf{x}_2' = \mathbf{x}_2 + \varepsilon \mathbf{c},$$

where $\varepsilon > 0$ is chosen so that $\mathbf{x}_1'$ and $\mathbf{x}_2'$ are in $\mathcal{X}$. Note that $\mathbf{W}_1^T \mathbf{x}_1' = \mathbf{W}_1^T \mathbf{x}_2'$ so $f(\mathbf{x}_1') = f(\mathbf{x}_2')$. Then

$$(2.13) \quad \mathbf{c}^T \left(\nabla_{\mathbf{x}} f(\mathbf{x}_1) - \nabla_{\mathbf{x}} f(\mathbf{x}_2) \right) = \lim_{\varepsilon \to 0} \frac{1}{\varepsilon} \left[(f(\mathbf{x}_1') - f(\mathbf{x}_1)) - (f(\mathbf{x}_2') - f(\mathbf{x}_2)) \right] = 0.$$

Simple limiting arguments can be used to extend this result to $\mathbf{x}_1$ or $\mathbf{x}_2$ on the boundary of $\mathcal{X}$. $\square$

2.1. Two special cases. We present two cases where the rank of $\mathbf{C}$ may be determined a priori. The first is a *ridge function* [11], which has the form $f(\mathbf{x}) = h(\mathbf{a}^T \mathbf{x})$, where h is a univariate function, and $\mathbf{a}$ is a constant m-vector. In this case, $\mathbf{C}$ is rank one, and the eigenvector defining the active subspace is $\mathbf{a}/\|\mathbf{a}\|$, which can be discovered by a single evaluation of the gradient anywhere in $\mathcal{X}$. The function shown in Figure 1 is an example of a ridge function.

The second special case is a function of the form $f(\mathbf{x}) = h(\mathbf{x}^T \mathbf{A} \mathbf{x})$, where h is a univariate function and $\mathbf{A}$ is a symmetric $m \times m$ matrix. In this case

$$(2.14) \qquad \mathbf{C} = 4 \, \mathbf{A} \, \mathbb{E}\left[(h')^2 \, \mathbf{x} \mathbf{x}^T \right] \, \mathbf{A}^T,$$

where $h' = h'(\mathbf{x}^T \mathbf{A} \mathbf{x})$ is the derivative of h. This implies that the null space of $\mathbf{C}$ is the null space of $\mathbf{A}$ provided that h' is nondegenerate.

2.2. Discovering the active subspace. We must compute the eigenvectors $\mathbf{W}$ and eigenvalues Λ of the matrix $\mathbf{C}$ from (2.3). We immediately encounter the obstacle of computing the elements of $\mathbf{C}$, which are integrals over the high-dimensional space $\mathcal{X}$. Thus, tensor product numerical quadrature rules are impractical. We opt for Monte Carlo integration, which will yield its own appealing interpretation. In particular, let

$$(2.15) \qquad \nabla_{\mathbf{x}} f_j = \nabla_{\mathbf{x}} f(\mathbf{x}_j), \qquad \mathbf{x}_j \in \mathcal{X}, \quad j = 1, \ldots, M,$$

be independently computed samples of the gradient vector, where $\mathbf{x}_j$ is drawn from the density ρ on $\mathcal{X}$. In practice computing the gradient at $\mathbf{x}_j$ typically involves first evaluating $f_j = f(\mathbf{x}_j)$; we will use these function evaluations when testing the response surface. With the samples of the gradient, we approximate

$$(2.16) \qquad \mathbf{C} \approx \tilde{\mathbf{C}} = \frac{1}{M} \sum_{j=1}^{M} (\nabla_{\mathbf{x}} f_j)(\nabla_{\mathbf{x}} f_j)^T$$

and compute the eigenvalue decomposition $\tilde{\mathbf{C}} = \tilde{\mathbf{W}} \tilde{\Lambda} \tilde{\mathbf{W}}^T$. The size of $\tilde{\mathbf{C}}$ is $m \times m$, where we expect m to be on the order of hundreds or thousands corresponding to the number of variables $\mathbf{x}$. Thus we anticipate no memory limitations when computing the complete eigendecomposition of $\tilde{\mathbf{C}}$ on a modern personal computer.

There is another interpretation of the sampling approach to approximate the eigenpairs of $\mathbf{C}$. We can write $\tilde{\mathbf{C}} = \mathbf{G}\mathbf{G}^T$, where the $m \times M$ matrix $\mathbf{G}$ is

$$(2.17) \qquad \mathbf{G} = \frac{1}{\sqrt{M}} \begin{bmatrix} \nabla_{\mathbf{x}} f_1 & \cdots & \nabla_{\mathbf{x}} f_M \end{bmatrix}.$$

If we compute the singular value decomposition (SVD) of $\mathbf{G}$, then with elementary manipulations,

$$(2.18) \qquad \mathbf{G} = \tilde{\mathbf{W}} \sqrt{\tilde{\Lambda}} \mathbf{V}^T.$$

This provides an alternative computational approach via the SVD. Again, we stress that the number of variables m and the number of gradient samples M are small enough in many applications of interest that the SVD can easily be computed on a modern personal computer. More importantly, the SVD shows that the rotation matrix $\tilde{\mathbf{W}}$ can be interpreted as the uncentered principal directions [23] from an ensemble of gradient evaluations.

It is natural to ask how large M must be for an accurate approximation of the eigenvectors; this is one focus of our current research efforts. If nothing is known a priori about $\mathbf{C}$, then at least m evaluations are necessary (though maybe not sufficient) to approximate a full rank $\mathbf{C}$. However, we hypothesize that the number of samples needed for accurate approximation may be related to the rank of $\mathbf{C}$. Loosely speaking, f must be very smooth for the Monte Carlo approximation to be effective. If f's variability is limited to a small subset of the high-dimensional domain, then the samples of the gradient may not reveal true directions of variability. We are currently exploring how to make such intuitive statements more precise and how to create robust sampling approaches for extreme cases, e.g., a step function in high dimensions.

In practice, we use the eigenpairs $\tilde{\mathbf{W}}$ and $\tilde{\Lambda}$ from the finite-sample approximation $\tilde{\mathbf{C}}$ in place of the true eigenpairs $\mathbf{W}$ and Λ of $\mathbf{C}$ from (2.3). There may be numerical integration methods that produce better approximations than simple Monte Carlo;

the latter merely offers an appealing interpretation in terms of the principle components of the gradients. Sequential sampling techniques [14] combined with a measure of the stability of the computed subspaces could be a powerful approach for accurate approximation with relatively few samples of the gradient. Alternatively, randomized algorithms for low-rank approximation offer promise for reducing the number of gradient samples [6, 21]. Quantifying the error in these finite-sample approximations is beyond the scope of this paper. However, in section 3.4 we will examine the effects of using the perturbed $\tilde{\mathbf{W}}$ to construct the response surface given an estimate of the perturbation.

3. Approximation in the active subspace. We assume that the number of variables m is too large to permit standard response surface constructions that suffer from the curse of dimensionality—such as regression or interpolation. The goal is to approximate the m-variate function f by a function that is $\mathbf{z}$-invariant. If f is nearly $\mathbf{z}$-invariant, then we expect a good approximation. A $\mathbf{z}$-invariant function only varies with changes in the $n < m$ coordinates $\mathbf{y}$. Therefore, we can build a response surface approximation using only the variables $\mathbf{y}$. Note that this requires (at least) two levels of approximation: (i) approximating f by a $\mathbf{z}$-invariant function and (ii) building a response surface of the n-variate approximation. In this section we develop the framework and error analysis for this type of approximation.

A few preliminaries: define the joint density function π of the coordinates $\mathbf{y}$ and $\mathbf{z}$ from (2.7) as

$$(3.1) \qquad \pi(\mathbf{y}, \mathbf{z}) = \rho(\mathbf{W}_1 \mathbf{y} + \mathbf{W}_2 \mathbf{z}).$$

With this definition, we can define marginal densities $\pi_Y(\mathbf{y})$, $\pi_Z(\mathbf{z})$ and conditional densities $\pi_{Y|Z}(\mathbf{y}|\mathbf{z})$, $\pi_{Z|Y}(\mathbf{z}|\mathbf{y})$ in the standard way. Next we define the domain of a function that only depends on $\mathbf{y}$. Define the set $\mathcal{Y}$ to be

$$(3.2) \qquad \mathcal{Y} = \left\{ \mathbf{y} : \mathbf{y} = \mathbf{W}_1^T \mathbf{x}, \mathbf{x} \in \mathcal{X} \right\} \subseteq \mathbb{R}^n.$$

Note that the marginal density $\pi_Y(\mathbf{y})$ defines a probability density on $\mathcal{Y}$. With these defined we can begin approximating.

3.1. Conditional expectation. For a fixed $\mathbf{y}$, the best guess one can make at the value of f is its average over all values of $\mathbf{x}$ that map to $\mathbf{y}$; this is precisely the conditional expectation of f given $\mathbf{y}$. Define the function G that depends on $\mathbf{y}$ by

$$(3.3) \qquad G(\mathbf{y}) = \mathbb{E}\left[f \,|\, \mathbf{y}\right] = \int_{\mathbf{z}} f(\mathbf{W}_1 \mathbf{y} + \mathbf{W}_2 \mathbf{z}) \, \pi_{Z|Y}(\mathbf{z}) \, d\mathbf{z}.$$

The second equality follows from the so-called law of the unconscious statistician. The domain of this function is $\mathcal{Y}$ from (3.2). Since G is a conditional expectation, it is the best mean-squared approximation of f given $\mathbf{y}$ [40, Chapter 9].

We can use G to approximate f at a given $\mathbf{x}$ with the following construction:

$$(3.4) \qquad f(\mathbf{x}) \approx F(\mathbf{x}) \equiv G(\mathbf{W}_1^T \mathbf{x}).$$

The next theorem provides an error bound for F in terms of the eigenvalues of $\mathbf{C}$ from (2.3).

THEOREM 3.1. *The mean-squared error of F defined in* (3.4) *satisfies*

$$(3.5) \qquad \mathbb{E}\left[(f - F)^2\right] \leq C_1(\lambda_{n+1} + \cdots + \lambda_m),$$

where C_1 is a constant that depends on the domain $\mathcal{X}$ and the weight function ρ.

Proof. Note that $\mathbb{E}\left[f - F \mid \mathbf{y}\right] = 0$ by definition (3.4). Thus,

$$(3.6) \qquad \mathbb{E}\left[(f - F)^2\right] = \mathbb{E}\left[\mathbb{E}\left[(f - F)^2 \mid \mathbf{y}\right]\right]$$

$$(3.7) \qquad \leq C_1 \, \mathbb{E}\left[\mathbb{E}\left[(\nabla_{\mathbf{z}} f)^T (\nabla_{\mathbf{z}} f) \mid \mathbf{y}\right]\right]$$

$$(3.8) \qquad = C_1 \, \mathbb{E}\left[(\nabla_{\mathbf{z}} f)^T (\nabla_{\mathbf{z}} f)\right]$$

$$(3.9) \qquad = C_1(\lambda_{n+1} + \cdots + \lambda_m).$$

Lines (3.6) and (3.8) are due to the tower property of conditional expectations. Line (3.7) is a Poincaré inequality, where the constant C_1 depends only on $\mathcal{X}$ and the density function ρ. Line (3.9) follows from Lemma 2.2. □

3.2. Monte Carlo approximation. The trouble with the approximation F from (3.4) is that each evaluation of F requires an integral with respect to the $\mathbf{z}$ coordinates. In other words, evaluating F requires high-dimensional integration. However, if f is nearly $\mathbf{z}$-invariant, then it is nearly constant along the coordinates $\mathbf{z}$. Thus, its variance along $\mathbf{z}$ will be very small, and we expect that simple numerical integration schemes to approximate the conditional expectation G will work well. We use simple Monte Carlo to approximate G and derive an error bound on such an approximation. The error bound validates the intuition that we need very few evaluations of f to approximate G if f is nearly $\mathbf{z}$-invariant.

Define the Monte Carlo estimate $\hat{G} = \hat{G}(\mathbf{y})$ by

$$(3.10) \qquad G(\mathbf{y}) \approx \hat{G}(\mathbf{y}) = \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{W}_1 \mathbf{y} + \mathbf{W}_2 \mathbf{z}_i),$$

where the $\mathbf{z}_i$ are drawn independently from the conditional density $\pi_{Z|Y}$. We approximate f as

$$(3.11) \qquad f(\mathbf{x}) \approx \hat{F}(\mathbf{x}) \equiv \hat{G}(\mathbf{W}_1^T \mathbf{x}).$$

Next we derive an error bound for this approximation.

THEOREM 3.2. *The mean-squared error of $\hat{F}$ defined in (3.11) satisfies*

$$(3.12) \qquad \mathbb{E}\left[(f - \hat{F})^2\right] \leq C_1 \left(1 + \frac{1}{N}\right)(\lambda_{n+1} + \cdots + \lambda_m),$$

where C_1 is from Theorem 3.1.

Proof. First define the conditional variance of f given $\mathbf{y}$ as $\sigma_{\mathbf{y}}^2 = \mathbb{E}\left[(f - F)^2 \mid \mathbf{y}\right]$, and note that the proof of Theorem 3.1 shows

$$(3.13) \qquad \mathbb{E}\left[\sigma_{\mathbf{y}}^2\right] \leq C_1 \left(\lambda_{n+1} + \cdots + \lambda_m\right).$$

Next note that the mean-squared error in the Monte Carlo approximation satisfies [31]

$$(3.14) \qquad \mathbb{E}\left[(F - \hat{F})^2 \mid \mathbf{y}\right] = \frac{\sigma_{\mathbf{y}}^2}{N},$$

so that

$$(3.15) \quad \mathbb{E}\left[(F - \hat{F})^2\right] = \mathbb{E}\left[\mathbb{E}\left[(F - \hat{F})^2 \mid \mathbf{y}\right]\right] = \frac{1}{N} \mathbb{E}\left[\sigma_{\mathbf{y}}^2\right] \leq \frac{C_1}{N}\left(\lambda_{n+1} + \cdots + \lambda_m\right).$$

Finally, using Theorem 3.1,

$$
(3.16) \qquad \mathbb{E}\left[(f - \hat{F})^2\right] \leq \mathbb{E}\left[(f - F)^2\right] + \mathbb{E}\left[(F - \hat{F})^2\right]
$$
$$
\leq C_1 \left(1 + \frac{1}{N}\right)(\lambda_{n+1} + \cdots + \lambda_m),
$$

as required. □

This bound shows that if $\lambda_{n+1}, \ldots, \lambda_m$ are sufficiently small, then the Monte Carlo estimate with small N (e.g., $N = 1$) will produce a very good approximation of f.

3.3. Response surfaces. We now reach the point where $n < m$ can reduce the cost of approximating f. Up to this point, there has been no advantage to using the conditional expectation F or its Monte Carlo approximation $\hat{F}$ to approximate f; each evaluation of $\hat{F}(\mathbf{x})$ requires at least one evaluation of $f(\mathbf{x})$. The real advantage of this method is that one can construct response surfaces with respect to the few variables $\mathbf{y} \in \mathbb{R}^n$ instead of f's natural variables $\mathbf{x} \in \mathbb{R}^m$. We will train a response surface on the domain $\mathcal{Y} \subseteq \mathbb{R}^n$ using a set of evaluations of $\hat{G} = \hat{G}(\mathbf{y})$.

Here we do not specify the form of the response surface; several are possible, and section 4 discusses applications using kriging. However, there is one important consideration before choosing a response surface method willy-nilly. If the eigenvalues $\lambda_{n+1}, \ldots, \lambda_m$ are not exactly zero, then evaluations of the Monte Carlo approximation $\hat{G}$ will contain *noise* due to the finite number of samples; in other words, the Monte Carlo estimate $\hat{G}$ is a random variable. This noise implies that $\hat{G}$ is not a smooth function of $\mathbf{y}$. Thus, we prefer smoothing, regression-based response surfaces over exact interpolation. In section 4, we characterize the noise and use it to tune the parameters of a kriging surface.

We construct a generic response surface for a function defined on $\mathcal{Y}$ from (3.2) as follows. Define the *design* on $\mathcal{Y}$ to be a set of points $\mathbf{y}_k \in \mathcal{Y}$ with $k = 1, \ldots, P$. The specific design will depend on the form of the response surface. Define $\hat{G}_k = \hat{G}(\mathbf{y}_k)$. Then we approximate

$$
(3.17) \qquad \hat{G}(\mathbf{y}) \approx \tilde{G}(\mathbf{y}) \equiv \mathcal{R}(\mathbf{y}; \hat{G}_1, \ldots, \hat{G}_P),
$$

where $\mathcal{R}$ is a response surface constructed with the training data $\hat{G}_1, \ldots, \hat{G}_P$. We use this response surface to approximate f as

$$
(3.18) \qquad f(\mathbf{x}) \approx \tilde{F}(\mathbf{x}) \equiv \tilde{G}(\mathbf{W}_1^T \mathbf{x}).
$$

To derive an error estimate for $\tilde{F}$, we assume the error in the response surface can be bounded as follows.

Assumption 1. Let $\mathcal{Z} = \{\mathbf{z} : \mathbf{z} = \mathbf{W}_2^T \mathbf{x}, \mathbf{x} \in \mathcal{X}\}$. Then there exists a constant C_2 such that

$$
(3.19) \qquad \mathbb{E}\left[(\hat{F} - \tilde{F})^2 \mid \mathbf{z}\right] \leq C_2 \delta
$$

for all $\mathbf{z} \in \mathcal{Z}$, where $\delta = \delta(\mathcal{R}, P)$ depends on the response surface method $\mathcal{R}$ and the number P of training data, and C_2 depends on the domain $\mathcal{X}$ and the probability density function ρ.

With Assumption 1, we have the following error estimate for $\tilde{F}$.

THEOREM 3.3. *The mean-squared error in $\tilde{F}$ defined in (3.18) satisfies*

$$(3.20) \qquad \mathbb{E}\left[(f - \tilde{F})^2\right] \leq C_1 \left(1 + \frac{1}{N}\right)(\lambda_{n+1} + \cdots + \lambda_m) + C_2\,\delta,$$

where C_1 is from Theorem 3.1, N is from Theorem 3.2, and C_2 and δ are from Assumption 1.

Proof. Note

$$(3.21) \qquad \mathbb{E}\left[(f - \tilde{F})^2\right] \leq \mathbb{E}\left[(f - \hat{F})^2\right] + \mathbb{E}\left[(\hat{F} - \tilde{F})^2\right].$$

Theorem 3.2 bounds the first summand. By the tower property and Assumption 1, the second summand satisfies

$$(3.22) \qquad \mathbb{E}\left[(\hat{F} - \tilde{F})^2\right] = \mathbb{E}\left[\mathbb{E}\left[(\hat{F} - \tilde{F})^2 \,|\, \mathbf{z}\right]\right] \leq C_2\delta,$$

as required. $\qquad\square$

The next equation summarizes the three levels of approximation:

$$(3.23) \qquad \begin{array}{ccccccc} f(\mathbf{x}) & \approx & F(\mathbf{x}) & \approx & \hat{F}(\mathbf{x}) & \approx & \tilde{F}(\mathbf{x}) \\ & & \| & & \| & & \| \\ & & G(\mathbf{W}_1^T\mathbf{x}) & \approx & \hat{G}(\mathbf{W}_1^T\mathbf{x}) & \approx & \tilde{G}(\mathbf{W}_1^T\mathbf{x}). \end{array}$$

The conditional expectation G is defined in (3.3), its Monte Carlo approximation $\hat{G}$ is defined in (3.10), and the response surface $\tilde{G}$ is defined in (3.17). The respective error estimates are given in Theorems 3.1, 3.2, and 3.3.

3.4. Using perturbed directions. Up to this point, we have assumed that we have the exact eigenvectors $\mathbf{W}$. However, as discussed in section 2.2, in practice we have a perturbed version $\tilde{\mathbf{W}}$—although both the true $\mathbf{W}$ and the perturbed $\tilde{\mathbf{W}}$ are orthonormal. In this section we examine the effects of this perturbation on the approximation of f. We assume the following characterization of the perturbation.

Assumption 2. Given $\mathbf{W}$ from (2.4), let $\tilde{\mathbf{W}}$ be a perturbed version of $\mathbf{W}$ that satisfies the following two conditions: (i) the sign of $\tilde{\mathbf{w}}_i$, the ith column of $\tilde{\mathbf{W}}$, is chosen to minimize $\|\mathbf{w}_i - \tilde{\mathbf{w}}_i\|$; (ii) there is an $\varepsilon > 0$ such that the perturbation satisfies $\|\mathbf{W} - \tilde{\mathbf{W}}\| \leq \varepsilon$ in the matrix 2-norm.

LEMMA 3.4. *Given the partition of $\mathbf{W}$ and a comparable partition $\tilde{\mathbf{W}} = \begin{bmatrix}\tilde{\mathbf{W}}_1 & \tilde{\mathbf{W}}_2\end{bmatrix}$, $\|\mathbf{W}_2^T\tilde{\mathbf{W}}_2\| \leq 1$ and $\|\mathbf{W}_1^T\tilde{\mathbf{W}}_2\| \leq \varepsilon$ in the matrix 2-norm.*

Proof. The orthogonality of the columns of $\mathbf{W}_2$ and $\tilde{\mathbf{W}}_2$ implies $\|\mathbf{W}_2^T\tilde{\mathbf{W}}_2\| \leq \|\mathbf{W}_2^T\|\|\tilde{\mathbf{W}}_2\| = 1$. The second inequality follows from

$$(3.24) \qquad \|\mathbf{W}_1^T\tilde{\mathbf{W}}_2 - \mathbf{0}\| = \|\mathbf{W}_1^T(\tilde{\mathbf{W}}_2 - \mathbf{W}_2)\| = \|\tilde{\mathbf{W}}_2 - \mathbf{W}_2\| \leq \varepsilon,$$

where the last relation follows from Assumption 2. $\qquad\square$

The perturbed $\tilde{\mathbf{W}}$ defines perturbed coordinates $\tilde{\mathbf{y}} = \tilde{\mathbf{W}}_1^T\mathbf{x}$ and $\tilde{\mathbf{z}} = \tilde{\mathbf{W}}_2^T\mathbf{x}$. The joint density $\tilde{\pi}(\tilde{\mathbf{y}}, \tilde{\mathbf{z}}) = \rho(\tilde{\mathbf{W}}_1\tilde{\mathbf{y}} + \tilde{\mathbf{W}}_2\tilde{\mathbf{z}})$ begets marginal densities $\tilde{\pi}_{\tilde{Y}}(\tilde{\mathbf{y}})$, $\tilde{\pi}_{\tilde{Z}}(\tilde{\mathbf{z}})$ and conditional densities $\tilde{\pi}_{\tilde{Y}|\tilde{Z}}(\tilde{\mathbf{y}}|\tilde{\mathbf{z}})$, $\tilde{\pi}_{\tilde{Z}|\tilde{Y}}(\tilde{\mathbf{z}}|\tilde{\mathbf{y}})$. The domain of the perturbed approximations is

$$(3.25) \qquad \tilde{\mathcal{Y}} = \{\tilde{\mathbf{y}} : \tilde{\mathbf{y}} = \tilde{\mathbf{W}}_1^T\mathbf{x}, \mathbf{x} \in \mathcal{X}\}.$$

We construct the same sequence of approximations of f using these perturbed coordinates. We denote the perturbed versions of the approximations with a subscript ε. The conditional expectation approximation of f becomes $F_\varepsilon(\mathbf{x}) \equiv G_\varepsilon(\tilde{\mathbf{W}}_1^T \mathbf{x})$, where

$$(3.26) \qquad G_\varepsilon(\tilde{\mathbf{y}}) = \mathbb{E}\left[f \mid \tilde{\mathbf{y}}\right] = \int_{\tilde{\mathbf{z}}} f(\tilde{\mathbf{W}}_1 \tilde{\mathbf{y}} + \tilde{\mathbf{W}}_2 \tilde{\mathbf{z}}) \, \tilde{\pi}_{\tilde{Z}|\tilde{Y}}(\tilde{\mathbf{z}}|\tilde{\mathbf{y}}) \, d\tilde{\mathbf{z}}.$$

Then we have the following error estimate

THEOREM 3.5. *The mean-squared error in the conditional expectation F_ε using the perturbed eigenvectors $\tilde{\mathbf{W}}$ satisfies*

$$(3.27) \qquad \mathbb{E}\left[(f - F_\varepsilon)^2\right] \leq C_1 \left(\varepsilon\,(\lambda_1 + \cdots + \lambda_n)^{\frac{1}{2}} + (\lambda_{n+1} + \cdots + \lambda_m)^{\frac{1}{2}}\right)^2,$$

where C_1 is from Theorem 3.1.

Proof. Following the same reasoning as the proof of Theorem 3.1 using the Poincaré inequality,

$$(3.28) \qquad \mathbb{E}\left[(f - F_\varepsilon)^2\right] \leq C_1 \mathbb{E}\left[(\nabla_{\tilde{\mathbf{z}}} f)^T (\nabla_{\tilde{\mathbf{z}}} f)\right].$$

Using the chain rule, $\nabla_{\tilde{\mathbf{z}}} f = \mathbf{W}_2^T \tilde{\mathbf{W}}_2 \nabla_{\mathbf{z}} f + \mathbf{W}_1^T \tilde{\mathbf{W}}_2 \nabla_{\mathbf{y}} f$. Then

$$(3.29)$$

$$\mathbb{E}\left[(\nabla_{\tilde{\mathbf{z}}} f)^T (\nabla_{\tilde{\mathbf{z}}} f)\right] \leq \mathbb{E}\left[(\nabla_{\mathbf{z}} f)^T (\nabla_{\mathbf{z}} f)\right] + 2\varepsilon\,\mathbb{E}\left[(\nabla_{\mathbf{z}} f)^T (\nabla_{\mathbf{y}} f)\right] + \varepsilon^2 \mathbb{E}\left[(\nabla_{\mathbf{y}} f)^T (\nabla_{\mathbf{y}} f)\right]$$

$$(3.30) \qquad \leq \left(\mathbb{E}\left[(\nabla_{\mathbf{z}} f)^T (\nabla_{\mathbf{z}} f)\right]^{\frac{1}{2}} + \varepsilon\,\mathbb{E}\left[(\nabla_{\mathbf{y}} f)^T (\nabla_{\mathbf{y}} f)\right]^{\frac{1}{2}}\right)^2$$

$$(3.31) \qquad \leq \left(\varepsilon\,(\lambda_1 + \cdots + \lambda_n)^{\frac{1}{2}} + (\lambda_{n+1} + \cdots + \lambda_m)^{\frac{1}{2}}\right)^2.$$

Line (3.29) follows from Lemma 3.4. Line (3.30) follows from the Cauchy–Schwarz inequality. Line (3.31) follows from Lemma 2.2   □

Notice how the eigenvalues $\lambda_1, \ldots, \lambda_n$ contribute to the error estimate given the perturbation bound ε on the eigenvectors. This contribution persists in error estimates for the Monte Carlo approximation and the response surface using the perturbed eigenvectors.

Let $\hat{F}_\varepsilon(\mathbf{x}) \equiv \hat{G}_\varepsilon(\tilde{\mathbf{W}}^T \mathbf{x})$, where

$$(3.32) \qquad \hat{G}_\varepsilon(\tilde{\mathbf{y}}) = \frac{1}{N} \sum_{i=1}^{N} f(\tilde{\mathbf{W}}_1 \tilde{\mathbf{y}} + \tilde{\mathbf{W}}_2 \tilde{\mathbf{z}}_i),$$

where $\tilde{\mathbf{z}}_i$ are drawn from the conditional density $\tilde{\pi}_{\tilde{Z}|\tilde{Y}}(\tilde{\mathbf{z}}|\tilde{\mathbf{y}})$. Then we have the following error estimate, whose derivation follows the proof of Theorem 3.2 using the perturbed coordinates and the reasoning from the proof of Theorem 3.5.

THEOREM 3.6. *The mean-squared error in the Monte Carlo approximation $\hat{F}_\varepsilon$ using the perturbed eigenvectors $\tilde{\mathbf{W}}_1$ satisfies*

$$(3.33) \quad \mathbb{E}\left[(f - \hat{F}_\varepsilon)^2\right] \leq C_1 \left(1 + \frac{1}{N}\right) \left(\varepsilon\,(\lambda_1 + \cdots + \lambda_n)^{\frac{1}{2}} + (\lambda_{n+1} + \cdots + \lambda_m)^{\frac{1}{2}}\right)^2,$$

where C_1 and N are the quantities from Theorem 3.2.

The response surface approximation using the perturbed eigenvectors is $\tilde{F}_\varepsilon(\mathbf{x}) \equiv \tilde{G}_\varepsilon(\tilde{\mathbf{W}}_1^T \mathbf{x})$, where

$$(3.34) \qquad \tilde{G}_\varepsilon(\tilde{\mathbf{y}}) = \mathcal{R}(\tilde{\mathbf{y}}; \hat{G}_{\varepsilon,1}, \ldots, \hat{G}_{\varepsilon,P})$$

for a chosen response surface method $\mathcal{R}$. The $\hat{G}_{\varepsilon,k}$ are evaluations of $\hat{G}_\varepsilon$ at the design points $\tilde{\mathbf{y}}_k \in \tilde{\mathcal{Y}}$. We have the following error estimate; again, its derivation follows the proof of Theorem 3.3 using the reasoning from the proof of Theorem 3.5.

THEOREM 3.7. *Under the assumptions of Theorem* 3.3, *the mean-squared error in the response surface approximation* $\tilde{F}_\varepsilon$ *satisfies*

$$(3.35)$$
$$\mathbb{E}\left[(f - \tilde{F}_\varepsilon)^2\right] \leq C_1 \left(1 + \frac{1}{N}\right) \left(\varepsilon \left(\lambda_1 + \cdots + \lambda_n\right)^{\frac{1}{2}} + \left(\lambda_{n+1} + \cdots + \lambda_m\right)^{\frac{1}{2}}\right)^2 + C_2\, \delta,$$

where C_1, N, C_2, *and* δ *are the quantities from Theorem* 3.3.

We summarize the three levels of approximation using the perturbed eigenvectors,

$$(3.36) \qquad \begin{array}{ccccccc} f(\mathbf{x}) & \approx & F_\varepsilon(\mathbf{x}) & \approx & \hat{F}_\varepsilon(\mathbf{x}) & \approx & \tilde{F}_\varepsilon(\mathbf{x}) \\ & & \| & & \| & & \| \\ & & G_\varepsilon(\tilde{\mathbf{W}}_1^T \mathbf{x}) & \approx & \hat{G}_\varepsilon(\tilde{\mathbf{W}}_1^T \mathbf{x}) & \approx & \tilde{G}_\varepsilon(\tilde{\mathbf{W}}_1^T \mathbf{x}). \end{array}$$

The conditional expectation G_ε is defined in (3.26), its Monte Carlo approximation $\hat{G}_\varepsilon$ is defined in (3.32), and the response surface $\tilde{G}_\varepsilon$ is defined in (3.34). The respective error estimates are given in Theorems 3.5, 3.6, and 3.7.

4. Heuristics for kriging surfaces. In this section, we detail a heuristic procedure to construct a kriging surface [25] (also known as Gaussian process approximation [33] and closely related to radial basis approximation [38]) on the n-dimensional reduced domain $\mathcal{Y}$ from (3.2) defined by the left singular vectors $\tilde{\mathbf{W}}$ from the samples of the gradient of f. To keep the notation clean, we use $\mathbf{W}$ instead of $\tilde{\mathbf{W}}$, and we do not explore the effects of the perturbed directions for this particular heuristic.

We must first choose the dimension n of the subspace. Many covariance-based reduction methods (e.g., the proper orthogonal decomposition [37]) use the magnitude of the λ_i to define n, e.g., so that $\lambda_1 + \cdots + \lambda_n$ exceeds some proportion of $\lambda_1 + \cdots + \lambda_m$. We are bound instead by more practical considerations, such as choosing n small enough to construct a reasonable design (e.g., a mesh) for the kriging surface. The trailing eigenvalues $\lambda_{n+1}, \ldots, \lambda_m$ then inform a noise model as discussed in section 4.2. A rapid decay in the λ_i implies that the low-dimensional approximation is relatively less noisy.

4.1. Design on reduced domain. We need to choose the points $\mathbf{y}_k$ on the reduced domain $\mathcal{Y}$ where we evaluate $\hat{G}$ and construct the kriging surface. We restrict our attention to $\mathcal{Y}$ derived from two particular choices of the domain $\mathcal{X}$ and the density function ρ that are often found in practice: a Gaussian density on $\mathbb{R}^m$ and a uniform density on a hypercube.

4.1.1. The Gaussian case. The first case we consider is when the domain $\mathcal{X}$ is $\mathbb{R}^m$ and $\rho(\mathbf{x})$ is a Gaussian density with mean zero and an identity covariance. In this case, the reduced domain $\mathcal{Y}$ is $\mathbb{R}^n$, and the marginal density $\pi_Y(\mathbf{y})$ is also a zero-mean Gaussian with an identity covariance, since $\mathbf{y} = \mathbf{W}_1^T \mathbf{x}$ and $\mathbf{W}_1^T \mathbf{W}_1 = \mathbf{I}$. We choose a simple tensor product design (i.e., a grid or lattice) on $\mathcal{Y}$ such that each

univariate design covers three standard deviations. For example, a nine-point design in $\mathbb{R}^2$ would use the points $\{-3, 0, 3\} \times \{-3, 0, 3\}$. This is the approach we will take in the numerical experiments in section 5.

4.1.2. The uniform case. Next, assume that $\mathcal{X} = [-1, 1]^m$, which we write equivalently as $-1 \leq \mathbf{x} \leq 1$, and the density ρ is uniform over $[-1, 1]^m$. The reduced domain becomes

$$(4.1) \qquad \mathcal{Y} = \left\{ \mathbf{y} : \mathbf{y} = \mathbf{W}_1^T \mathbf{x}, -1 \leq \mathbf{x} \leq 1 \right\}.$$

In general, $\mathcal{Y}$ will not be a hypercube in $\mathbb{R}^n$—only if $\mathbf{W}_1$ contains only columns of the identity matrix. However, $\mathcal{Y}$ will be a convex polytope in $\mathbb{R}^n$ whose vertices are a subset of vertices of $[-1, 1]^m$ projected to $\mathbb{R}^n$. For example, if $m = 3$ and $n = 2$, then one can imagine taking a photograph of a rotated cube; six of the cube's eight vertices define the polytope in $\mathbb{R}^2$. These projections of hypercubes are called *zonotopes*, and there exist polynomial time algorithms for discovering the vertices that define the convex hull [17]. In principle, one can give these vertices to a mesh generator (e.g., [32]) to create a design on $\mathcal{Y}$. The marginal density $\pi_Y(\mathbf{y})$ is challenging to compute; our current research efforts include this task.

4.2. Training the kriging surface. In section 3.2, we described approximating the conditional expectation $G(\mathbf{y})$ by its Monte Carlo estimate $\hat{G}(\mathbf{y})$ from (3.10). The error bound in Theorem 3.2 shows that if the trailing eigenvalues $\lambda_{n+1}, \ldots, \lambda_m$ are small enough, then the number N of samples needed for the Monte Carlo estimate can be very small—even $N = 1$. However, the proof of the theorem used the error measure (3.14), which assumes that $\mathbf{z}_i$ are drawn independently from the conditional density $\pi_{Z|Y}$. In practice, we can use a Metropolis–Hasting [10] scheme to sample from $\pi_{Z|Y}$ since it is proportional to the given $\rho(\mathbf{W}_1\mathbf{y} + \mathbf{W}_2\mathbf{z})$. But these samples are correlated, and the error bound in (3.14) does not strictly apply [7].

In practice, we have had success using only a single evaluation of f in the computation of $\hat{G}$, as suggested by Theorem 3.2. Therefore, we do not need to draw several samples from $\pi_{Z|Y}$; we need only a single $\mathbf{x}_k \in \mathcal{X}$ such that $\mathbf{W}_1^T \mathbf{x}_k = \mathbf{y}_k$ for each $\mathbf{y}_k$ in the design on $\mathcal{Y}$. We can easily find such an $\mathbf{x}_k$ for each of the two cases discussed in the previous section.

- If $\mathcal{X} = \mathbb{R}^m$ and ρ is a Gaussian density, then $\mathbf{x}_k = \mathbf{W}_1\mathbf{y}_k$.
- If $\mathcal{X} = [-1, 1]^m$ and ρ is a uniform density, then $\mathbf{x}_k$ must satisfy $\mathbf{W}_1^T \mathbf{x}_k = \mathbf{y}_k$ and $-1 \leq \mathbf{x}_k \leq 1$. Thus, $\mathbf{x}_k$ can be found using Phase 1 of a linear program [28].

With $\mathbf{x}_k$, we compute $\hat{G}_k = \hat{G}(\mathbf{y}_k) = f(\mathbf{x}_k)$; the set $\{(\mathbf{y}_k, \hat{G}_k)\}$ comprises the training data for the kriging surface.

We assume that the function we are trying to approximate is smooth with respect to the coordinates $\mathbf{y}$. However, since the training data $\hat{G}_k$ are not exactly equal to the conditional expectation $G(\mathbf{y}_k)$, we do not want to force the kriging surface to interpolate the training data. Instead, we want to build a model for the noise in the training data that is motivated by Theorem 3.1 and incorporate it into the kriging surface. We choose the correlation matrix of the training data to come from a product-type squared exponential kernel with an additional diagonal term to represent the noise,

$$(4.2) \qquad \mathrm{Cov}\left[\tilde{G}(\mathbf{y}_{k_1}), \tilde{G}(\mathbf{y}_{k_2})\right] = K\left(\mathbf{y}_{k_1}, \mathbf{y}_{k_2}\right) + \eta^2 \, \delta(k_1, k_2),$$

where $\delta(k_1, k_2)$ is 1 if $k_1 = k_2$ and zero otherwise, and

$$(4.3) \qquad K\left(\mathbf{y}_{k_1}, \mathbf{y}_{k_2}\right) = \exp\left(-\sum_{i=1}^{n} \frac{(y_{k_1,i} - y_{k_2,i})^2}{2\ell_i^2}\right).$$

Along with the correlation function, we choose a quadratic mean term, which will add $\binom{n+2}{n}$ polynomial basis functions to the kriging approximation. Note that this imposes a restriction that the design must be poised for quadratic approximation.

We are left to determine the parameters of the kriging surface, including the correlation lengths ℓ_i from (4.3) and the parameter η^2 of the noise model from (4.2). Given values for these parameters, the coefficients of both the polynomial bases and the linear combination of the training data are computed in the standard way [25, 33]. We could use a standard maximum likelihood method to compute ℓ_i and η^2. However, we can inform these parameters using the quantities from Lemma 2.1 and Theorem 3.1.

Toward this end, we approximate the directional derivative of f along $\mathbf{w}_i$ with a finite difference,

$$(4.4) \qquad \nabla_{\mathbf{x}} f(\mathbf{x})^T \mathbf{w}_i \approx \frac{1}{\delta}\left(f(\mathbf{x} + \delta\mathbf{w}_i) - f(\mathbf{x})\right),$$

which is valid for small δ. Decompose $f(\mathbf{x}) = f_0 + f'(\mathbf{x})$, where $f_0 = \mathbb{E}[f]$, so that f' has zero mean. By Lemma 2.1,

$$(4.5) \qquad \begin{aligned} \lambda_i &= \mathbb{E}\left[((\nabla_{\mathbf{x}} f)^T \mathbf{w}_i)^2\right] \\ &\approx \frac{1}{\delta^2}\mathbb{E}\left[(f(\mathbf{x} + \delta\mathbf{w}_i) - f(\mathbf{x}))^2\right] \\ &= \frac{1}{\delta^2}\mathbb{E}\left[(f'(\mathbf{x} + \delta\mathbf{w}_i) - f'(\mathbf{x}))^2\right] \\ &= \frac{2}{\delta^2}\left(\sigma^2 - \mathrm{Cov}\left[f'(\mathbf{x} + \delta\mathbf{w}_i),\, f'(\mathbf{x})\right]\right), \end{aligned}$$

where $\sigma^2 = \mathrm{Var}[f]$. Rearranged, we have

$$(4.6) \qquad \mathrm{Corr}\left[f'(\mathbf{x} + \delta\mathbf{w}_i),\, f'(\mathbf{x})\right] = \frac{1}{\sigma^2}\mathrm{Cov}\left[f'(\mathbf{x} + \delta\mathbf{w}_i),\, f'(\mathbf{x})\right] = 1 - \frac{\lambda_i \delta^2}{2\sigma^2}.$$

This implies that the correlation function along $\mathbf{w}_i$ is locally quadratic near the origin with coefficient $\lambda_i/2\sigma^2$. A univariate squared exponential correlation function with correlation length parameter ℓ has a Taylor series about the origin

$$(4.7) \qquad \exp\left(-\frac{\delta^2}{2\ell^2}\right) = 1 - \frac{\delta^2}{2\ell^2} + \cdots.$$

Comparing these terms to the locally quadratic approximation of the correlation function, we can use a univariate Gaussian with correlation length

$$(4.8) \qquad \ell_i^2 = \frac{\sigma^2}{\lambda_i}$$

for the reduced coordinate y_i. In other words, the correlation length parameter ℓ_i corresponding to y_i in the correlation kernel will be inversely proportional to the square root of the eigenvalue λ_i. Thus we need to approximate the variance σ^2.

Applying Theorem 3.1 with $n = 0$, we have

$$\sigma^2 = \text{Var}\,[f] \leq C_1(\lambda_1 + \cdots + \lambda_m). \tag{4.9}$$

Unfortunately, the Poincaré inequality is a notoriously loose bound, so we are reluctant to simply plug in an estimate of C_1 to approximate σ^2. Instead, we posit that for some constant α,

$$\sigma^2 = \alpha(\lambda_1 + \cdots + \lambda_m), \tag{4.10}$$

where $\alpha \leq C_1$, and we can employ an estimate of C_1. For the case when $\mathcal{X} = \mathbb{R}^m$ and ρ is a standard normal, $C_1 \leq 1$ [9]. If $\mathcal{X} = [-1, 1]^m$ and ρ is uniform, then $C_1 \leq 2\sqrt{m}/\pi$ [3]. Other cases can use comparable estimates, if available. To get a rough lower bound on α, we use the biased estimator of σ^2 from the samples f_j computed in section 2.2,

$$\hat{\sigma}^2 = \frac{1}{M} \sum_{j=1}^{M} (f_j - \hat{f}_0)^2, \tag{4.11}$$

where $\hat{f}_0$ is the empirical mean of the f_j. Since M will generally be small due to limited function evaluations, we expect the bias to be significant enough to justify the bound

$$\hat{\sigma}^2 \leq \alpha(\lambda_1 + \cdots + \lambda_m), \tag{4.12}$$

so that

$$\frac{\hat{\sigma}^2}{\lambda_1 + \cdots + \lambda_m} \leq \alpha \leq C_1. \tag{4.13}$$

From here, we treat α as a hyperparameter for the correlation kernel (4.3), and we can use a maximum likelihood approach to set it. Note that the number of variables in the maximum likelihood is one, in contrast to $n + 1$ variables for the standard approach. Given a value for α that achieves the maximum likelihood, we set σ^2 by (4.10) and

$$\eta^2 = \alpha(\lambda_{n+1} + \cdots + \lambda_m) \tag{4.14}$$

in (4.2). We now have all necessary quantities to build the kriging surface on the active subspace.

4.3. A step-by-step algorithm. We summarize this section with an algorithm incorporating the previously described computational procedures given a function $f = f(\mathbf{x})$ and its gradient $\nabla_{\mathbf{x}} f = \nabla_{\mathbf{x}} f(\mathbf{x})$ defined on $\mathcal{X}$ with probability density function ρ. Portions of this algorithm are specific to the Gaussian and uniform density cases discussed in section 4.1.

1. Initial sampling. Choose a set of M points $\mathbf{x}_j \in \mathcal{X}$ according to the measure $\rho(\mathbf{x})$. For each $\mathbf{x}_j$, compute $f_j = f(\mathbf{x}_j)$ and $\nabla_{\mathbf{x}} f_j = \nabla_{\mathbf{x}} f(\mathbf{x}_j)$. Compute the sample variance $\hat{\sigma}^2$.
2. Gradient analysis. Compute the SVD of the matrix

$$\mathbf{G} = \frac{1}{\sqrt{M}} \begin{bmatrix} \nabla_{\mathbf{x}} f_1 & \cdots & \nabla_{\mathbf{x}} f_M \end{bmatrix} = \mathbf{W} \Sigma \mathbf{V}^T, \tag{4.15}$$

and set $\Lambda = \Sigma^2$. Choose a reduced dimension $n < m$ according to practical considerations and the decay of λ_i. Partition $\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_2 \end{bmatrix}$.

3. Reduced domain. If $\mathcal{X} = \mathbb{R}^m$ and ρ is a standard Gaussian density, then the reduced domain $\mathcal{Y} = \mathbb{R}^n$. If $\mathcal{X} = [-1, 1]^m$ and ρ is a uniform density, then use the method described in [17] to determine the vertices of the zonotope in $\mathbb{R}^n$.

4. Design on reduced domain. Choose a set of P points $\mathbf{y}_k \in \mathcal{Y}$. In the Gaussian case, choose $\mathbf{x}_k = \mathbf{W}_1 \mathbf{y}_k$. In the uniform case, use a linear program solver to find an $\mathbf{x}_k \in [-1, 1]^m$ that satisfies $\mathbf{y}_k = \mathbf{W}_1^T \mathbf{x}_k$.

5. Train the response surface. For each $\mathbf{y}_k$, set $\hat{G}_k = \hat{G}(\mathbf{y}_k) = f(\mathbf{x}_k)$. Use a maximum likelihood method to find the hyperparameter α with bounds from (4.13). Set the noise model parameter η^2 as in (4.14) and correlation lengths

$$(4.16) \qquad \ell_i^2 = \frac{\alpha}{\lambda_i}(\lambda_1 + \cdots + \lambda_m)$$

for the product squared exponential correlation kernel on $\bar{\mathcal{Y}}$. Apply standard kriging with the training data $\{(\mathbf{y}_k, \hat{G}_k)\}$.

6. Evaluate the response surface. For a point $\mathbf{x} \in \mathcal{X}$, compute

$$(4.17) \qquad f(\mathbf{x}) \approx \tilde{F}(\mathbf{x}) = \tilde{G}(\mathbf{W}_1^T \mathbf{x}),$$

where $\tilde{G}$ is the trained kriging surface.

We conclude this section with summarizing remarks. First, the choice of n requires some interaction from the user. We advocate such interaction since we have found that one can uncover insights into the function f by examining the λ_i and the elements of $\mathbf{w}_i$. Second, as written, there is substantial freedom in choosing both the design sites on the reduced domain and the response surface. This was intentional. For our purposes, it suffices to use the gradient analysis to construct an approximation on the active subspace, but the details of the approximation will require many more practical considerations than we can address here. We have chosen kriging primarily because of (i) the natural fit of the computed λ_i to the correlation length parameters and training data noise model, and (ii) its flexibility with scattered design sites. However, many other options for approximation are possible, including global polynomials, regression splines, or finite element approximations; Russi advocates a global quadratic polynomial [34] on the subspace.

Third, with the gradient available for f, one could use (2.10) to obtain gradients with respect to the reduced coordinates. This could then be used to improve the response surface on the active subspace [25]. As also mentioned in [25], since we know a great deal about our correlation function, we could create designs that satisfy optimality criteria such as maximum entropy.

Finally, we note that the function evaluations f_j could be better used to construct the response surface on the subspace. We have intentionally avoided proposing any strategies for such use; we prefer instead to use them as a testing set for the response surface as detailed in the next section.

5. Numerical example. In this numerical exercise, we study an elliptic PDE with a random field model for the coefficients. Such problems are common test cases for methods in UQ [2]. MATLAB codes for this study can be found at https://bitbucket.org/paulcon/active-subspace-methods-in-theory-and-practice. They require MATLAB's PDE Toolbox, the random field simulation code at http://www.mathworks.com/matlabcentral/fileexchange/27613-random-field-simulation, and the Gaussian process regression codes at http://www.gaussianprocess.org/gpml/code/matlab/doc/.

5.1. Forward and adjoint problem. Consider the following linear elliptic PDE. Let $u = u(\mathbf{s}, \mathbf{x})$ satisfy

$$(5.1) \qquad -\nabla_{\mathbf{s}} \cdot (a \, \nabla_{\mathbf{s}} u) = 1, \qquad \mathbf{s} \in [0, 1]^2.$$

We set homogeneous Dirichlet boundary conditions on the left, top, and bottom of the spatial domain; denote this boundary by Γ_1. The right side of the spatial domain denoted Γ_2 has a homogeneous Neumann boundary condition. The log of the coefficients $a = a(\mathbf{s}, \mathbf{x})$ of the differential operator are given by a truncated Karhunen–Loeve (KL) type expansion

$$(5.2) \qquad \log(a(\mathbf{s}, \mathbf{x})) = \sum_{i=1}^{m} x_i \, \gamma_i \, \phi_i(\mathbf{s}),$$

where the x_i are independent, identically distributed standard normal random variables, and the $\{\phi_i(\mathbf{s}), \gamma_i\}$ are the eigenpairs of the correlation operator

$$(5.3) \qquad \mathcal{C}(\mathbf{s}, \mathbf{t}) = \exp\left(\beta^{-1} \|\mathbf{s} - \mathbf{t}\|_1\right).$$

We will study the quality of the active subspace approximation for two correlation lengths, $\beta = 1$ and $\beta = 0.01$. We choose a truncation of the field $m = 100$, which implies that the parameter space $\mathcal{X} = \mathbb{R}^{100}$ with ρ a standard Gaussian density function. Define the linear function of the solution

$$(5.4) \qquad \frac{1}{|\Gamma_2|} \int_{\Gamma_2} u(\mathbf{s}, \mathbf{x}) \, d\mathbf{s}.$$

This is the function we will study with the active subspace method.

Given a value for the input parameters $\mathbf{x}$, we discretize the elliptic problem with a standard linear finite element method using MATLAB's PDE Toolbox. The discretized domain has 34,320 triangles and 17,361 nodes; the eigenfunctions ϕ_i from (5.2) are approximated on this mesh. The matrix equation for the discrete solution $\mathbf{u} = \mathbf{u}(\mathbf{x})$ at the mesh nodes is

$$(5.5) \qquad \mathbf{K}\mathbf{u} = \mathbf{f},$$

where $\mathbf{K} = \mathbf{K}(\mathbf{x})$ is symmetric and positive definite for all $\mathbf{x} \in \mathcal{X}$. We can approximate the linear functional as

$$(5.6) \qquad f(\mathbf{x}) = \mathbf{c}^T \mathbf{M} \mathbf{u}(\mathbf{x}) \approx \frac{1}{|\Gamma_2|} \int_{\Gamma_2} u(\mathbf{s}, \mathbf{x}) \, d\mathbf{s},$$

where $\mathbf{M}$ is the symmetric mass matrix, and the components of $\mathbf{c}$ corresponding to mesh nodes on Γ_2 are equal to one with the rest equal to zero.

Since the quantity of interest can be written as a linear functional of the solution, we can define adjoint variables that enable us to compute $\nabla_{\mathbf{x}} f$,

$$(5.7) \qquad f = \mathbf{c}^T \mathbf{M} \mathbf{u} = \mathbf{c}^T \mathbf{M} \mathbf{u} - \mathbf{y}^T (\mathbf{K}\mathbf{u} - \mathbf{f}),$$

for any constant vector $\mathbf{y}$. Taking the derivative of (5.7) with respect to the input x_i, we get

$$(5.8)$$
$$\frac{\partial f}{\partial x_i} = \mathbf{c}^T \mathbf{M} \left(\frac{\partial \mathbf{u}}{\partial x_i}\right) - \mathbf{y}^T \left(\frac{\partial \mathbf{K}}{\partial x_i} \mathbf{u} + \mathbf{K} \frac{\partial \mathbf{u}}{\partial x_i}\right) = (\mathbf{c}^T \mathbf{M} - \mathbf{y}^T \mathbf{K}) \left(\frac{\partial \mathbf{u}}{\partial x_i}\right) - \mathbf{y}^T \left(\frac{\partial \mathbf{K}}{\partial x_i}\right) \mathbf{u}.$$

TABLE 1

This table compares the normalized KL singular values from (5.2) with the normalized singular values from the active subspace analysis, i.e., the singular values of $\mathbf{G}$ from (2.18). Two values of β are the correlation length parameters for the random field model of the elliptic coefficients from (5.3).

KL, $\beta = 1$	ASM, $\beta = 1$	KL, $\beta = 0.01$	ASM, $\beta = 0.01$
1.0000	1.0000	1.0000	1.0000
0.0978	0.0010	0.9946	0.0055
0.0975	0.0006	0.9873	0.0047
0.0282	0.0005	0.9836	0.0046
0.0280	0.0002	0.9774	0.0042

If we choose $\mathbf{y}$ to solve the adjoint equation $\mathbf{K}^T\mathbf{y} = \mathbf{M}^T\mathbf{c}$, then

$$(5.9) \qquad \frac{\partial f}{\partial x_i} = -\mathbf{y}^T \left(\frac{\partial \mathbf{K}}{\partial x_i} \right) \mathbf{u}.$$

To approximate the gradient $\nabla_{\mathbf{x}} f$ at the point $\mathbf{x}$, we compute the finite element solution with (5.5), solve the adjoint problem, and compute the components with (5.9). The derivative of $\mathbf{K}$ with respect to x_i is straightforward to compute from the derivative of $a(\mathbf{s}, \mathbf{x})$ and the same finite element discretization.

5.2. Dimension reduction study. Next we apply the dimension reduction method to the quantity of interest f from (5.6). We compare two cases: (i) the random field model for a has a long correlation length ($\beta = 1$ in (5.3)), which corresponds to rapidly decaying KL singular values γ_i in (5.2), and (ii) a has a short correlation length ($\beta = 0.01$), which corresponds to slowly decaying KL singular values. The number of terms in the KL series is often chosen according to the decay of the singular values, e.g., to capture some proportion of the energy in a; we choose $m = 100$ in both cases for illustration purposes. Thus, the linear function f of the solution to the PDE u is parameterized by the $m = 100$ parameters characterizing the elliptic coefficients a.

We use the finite element model and its adjoint to compute f and $\nabla_{\mathbf{x}} f$ at $M = 300$ points drawn from a m-variate normal distribution with zero mean and identity covariance. Table 1 lists the first five singular values from the SVD of the matrix of gradient samples (labeled ASM for *active subspace method*) for both $\beta = 1$ and $\beta = 0.01$. They are normalized so that the first singular value is 1. We compare these with normalized versions of the KL singular values for both correlation lengths. Notice that the slow decay of the KL singular values for $\beta = 0.01$ suggests that little dimension reduction is possible. However, the singular values of $\mathbf{G}$ decay very rapidly, which suggests dimension reduction will be effective—assuming that the Monte Carlo approximation of the matrix $\mathbf{C}$ is sufficiently accurate. The comparison is meaningful only in the sense of dimension reduction; the singular values of $\mathbf{G}$ are with respect to a specific scalar quantity of interest, while the KL singular values are for the spatially varying field.

Figure 2 plots the components of the first two eigenvectors from the active subspace analysis—i.e., the first two columns of $\mathbf{W}$ from (2.18)—for both correlation lengths. Notice that the mass is more evenly distributed for the short correlation length. This is not surprising. The magnitude of the components of the first eigenvector are a measure of sensitivity of the function f to perturbations in the parameters. The relative clustering of large values toward smaller indices (the left side of the plot) for the longer correlation length implies that the coefficients x_i in the KL series with
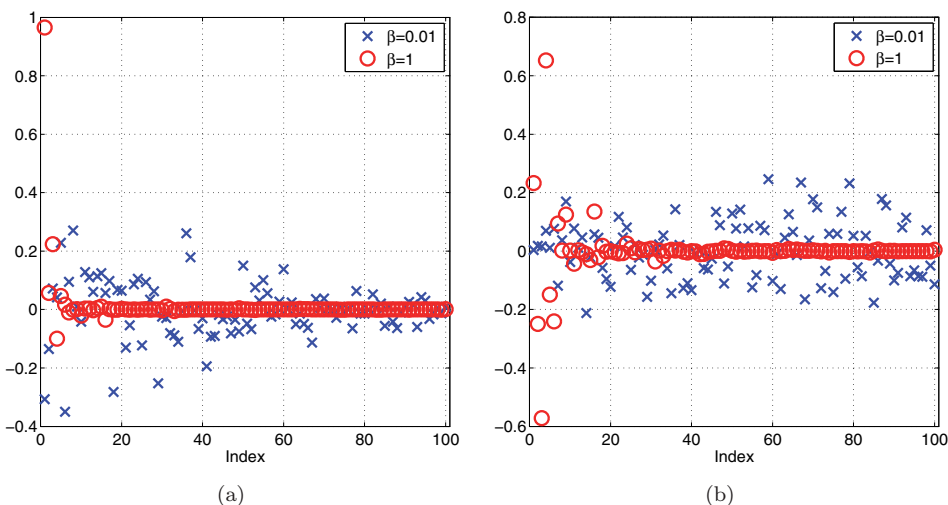
(a)   (b)

FIG. 2. *The left figure shows the components of the first eigenvector from the active subspace analysis for both long ($\beta = 1$) and short ($\beta = 0.01$) correlation lengths. The right figure shows the components of the second eigenvector. (Colors are visible in the electronic version.)*

TABLE 2
This table shows the average relative error of the kriging surface at 300 *testing sites as the dimension of the active subspace increases. We report this approximation error for $\beta = 1$ and $\beta = 0.01$ in* (5.3). *Note the relatively slow decay of the error justifies our attention on active subspaces for $n = 1$ and $n = 2$.*

n	$\beta = 0.01$	$\beta = 1$
1	7.88e-3	1.78e-1
2	7.82e-3	1.49e-1
3	7.57e-3	1.88e-1
4	6.75e-3	1.22e-1
5	6.61e-3	1.10e-1

larger singular values contribute the most to the variability in f. But this relationship relaxes for shorter correlation lengths.

Table 2 studies the approximation quality as the dimension of the low-dimensional subspace increases for both choices of the correlation length parameter β. The numbers represent the average relative error in the active subspace method's approximation for the $M = 300$ testing evaluations. We trained the kriging surfaces using a five-point tensor product design on the reduced domain; the univariate designs use the points $\{-3, -1.5, 0, 1.5, 3\}$ to cover three standard deviations in the input space. We use the heuristic described in section 4.2 for choosing the kriging hyperparameters. The average error does not decrease rapidly for increasing n. Therefore, we will consider only the active subspace approximations for $n = 1$ and $n = 2$.

We compare the accuracy of the kriging surface on the active subspace with a kriging surface on the one- and two-dimensional coordinate subspaces defined by the largest-in-magnitude components of the gradient $\nabla_{\mathbf{x}} f$ evaluated at the origin. In other words, we compare the approximation on the active subspace to an approximation using local sensitivity analysis to reduce the number of parameters. For the long correlation length $\beta = 1$, the two most important coordinates are x_1 and x_3. For the

short correlation length $\beta = 0.01$, the two most important coordinates are x_6 and x_1. For the coordinate dimension reduction, we use the maximum likelihood method implemented in the GPML code [33] with a maximum of 500 function evaluations to tune the hyperparameters of an isotropic squared-exponential covariance kernel and a quadratic polynomial basis.

Figure 3 shows the one-dimensional projections. In all subfigures, the solid black line is the mean kriging prediction, and the gray shaded region is the two-standard-deviation confidence interval. The blue dots show the 300 evaluations of f computed while studying the gradients projected onto the subspaces; we use these evaluations as testing data. Figures 3(a) and 3(c) show the approximation for the long
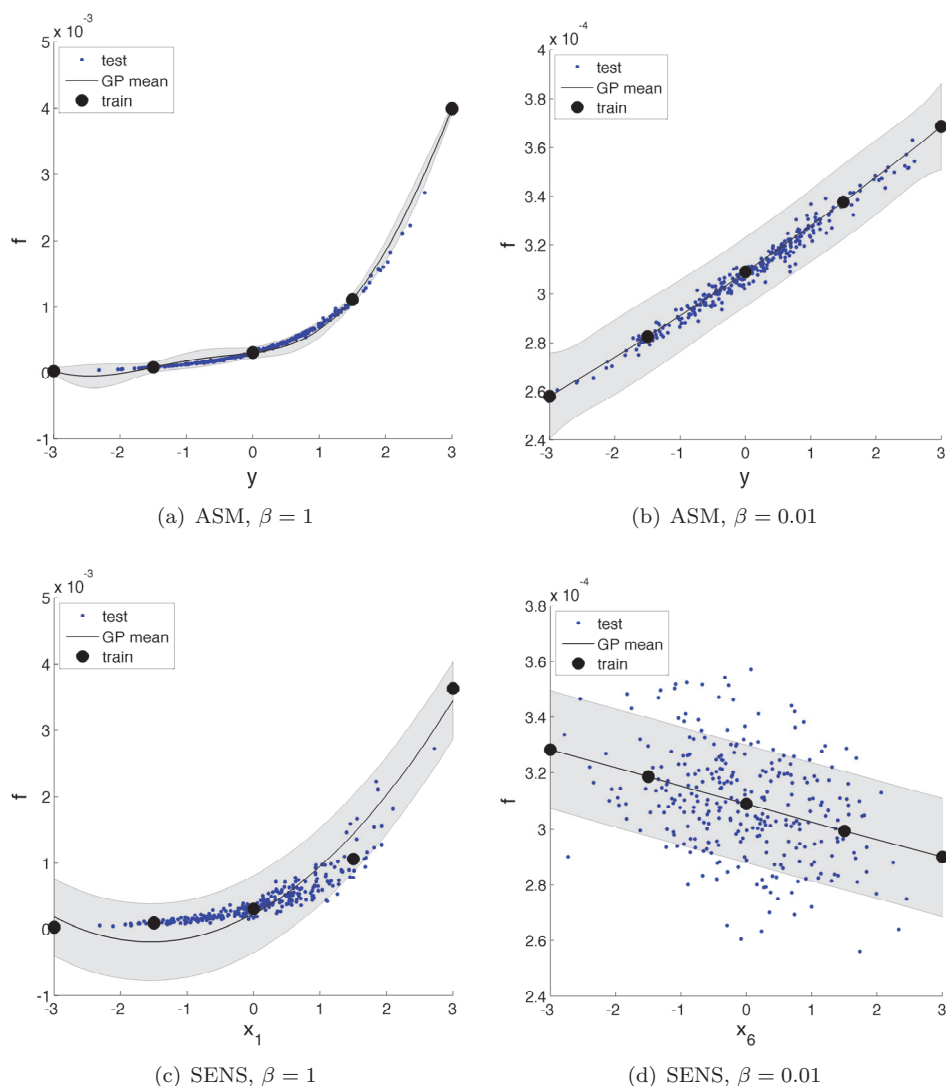

(a) ASM, $\beta = 1$　　　　　　　　　(b) ASM, $\beta = 0.01$

(c) SENS, $\beta = 1$　　　　　　　　　(d) SENS, $\beta = 0.01$

FIG. 3. *Comparing the kriging surfaces constructed on the one-dimensional subspaces defined by the active subspaces, (a) and (b), and a local sensitivity analysis at the origin, (c) and (d). (Colors are visible in the electronic version.)*

correlation length. Notice how the function evaluations cluster more tightly around the mean prediction for the active subspace. Loosely speaking, this means the active subspace has found the right angle from which to view the high-dimensional data to uncover its one-dimensional character. Figures 3(b) and 3(d) show the same plots for the shorter correlation length. Notice how for both methods the spread of the function evaluations is larger. However, the active subspace is again able to uncover a strongly dominant direction; when viewed from this direction, the data is essentially one-dimensional. The local sensitivity method reveals no such trend for the shorter correlation length. Figure 4 shows the same plots for the two-dimensional subspaces without the shaded regions for the confidence intervals. The conclusions drawn from the one-dimensional plots are the same for the two-dimensional plots.

Figure 5 shows histograms of the log of the relative error in the testing data for the two correlation lengths on both the one- and two-dimensional subspace approximations. For each case, the histogram of the testing error in the active subspace approach is compared with the coordinate reduction approach. In all cases, the active subspace approach performs better, as indicated by the leftward shift in the histogram, which corresponds to smaller error.
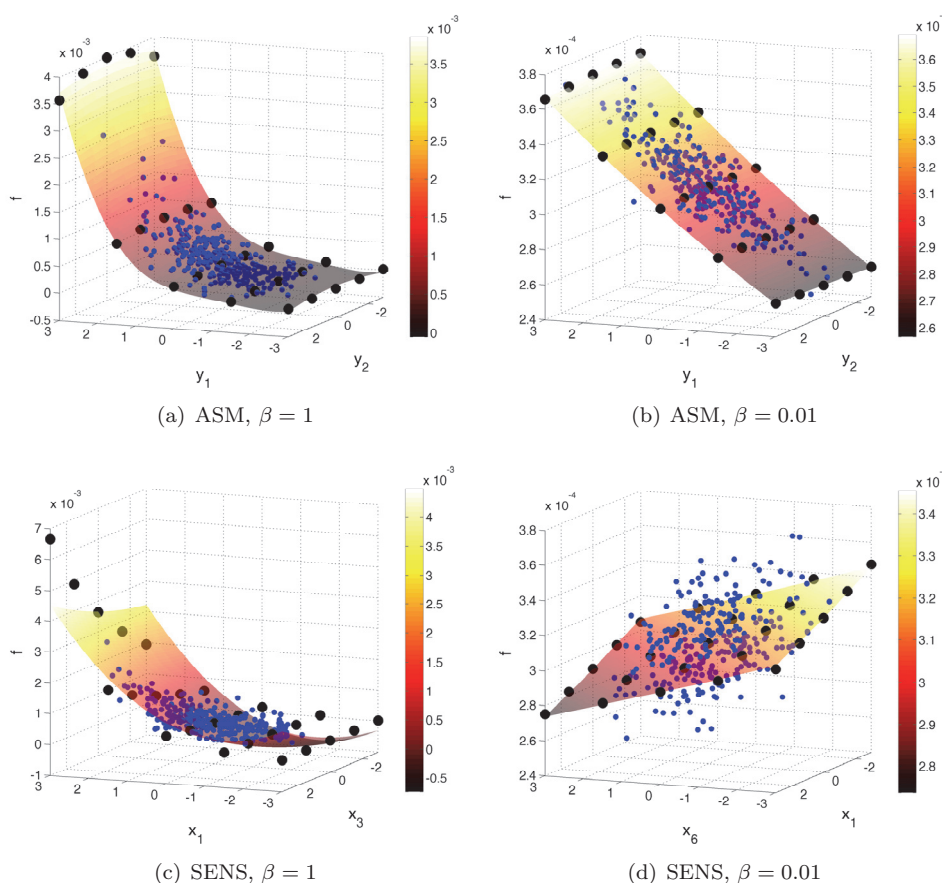

(a) ASM, $\beta = 1$ (b) ASM, $\beta = 0.01$

(c) SENS, $\beta = 1$ (d) SENS, $\beta = 0.01$

FIG. 4. *Kriging surfaces constructed on the two-dimensional subspaces defined by the active subspaces, (a) and (b), and a local sensitivity analysis at the origin, (c) and (d). (Colors are visible in the electronic version.)*
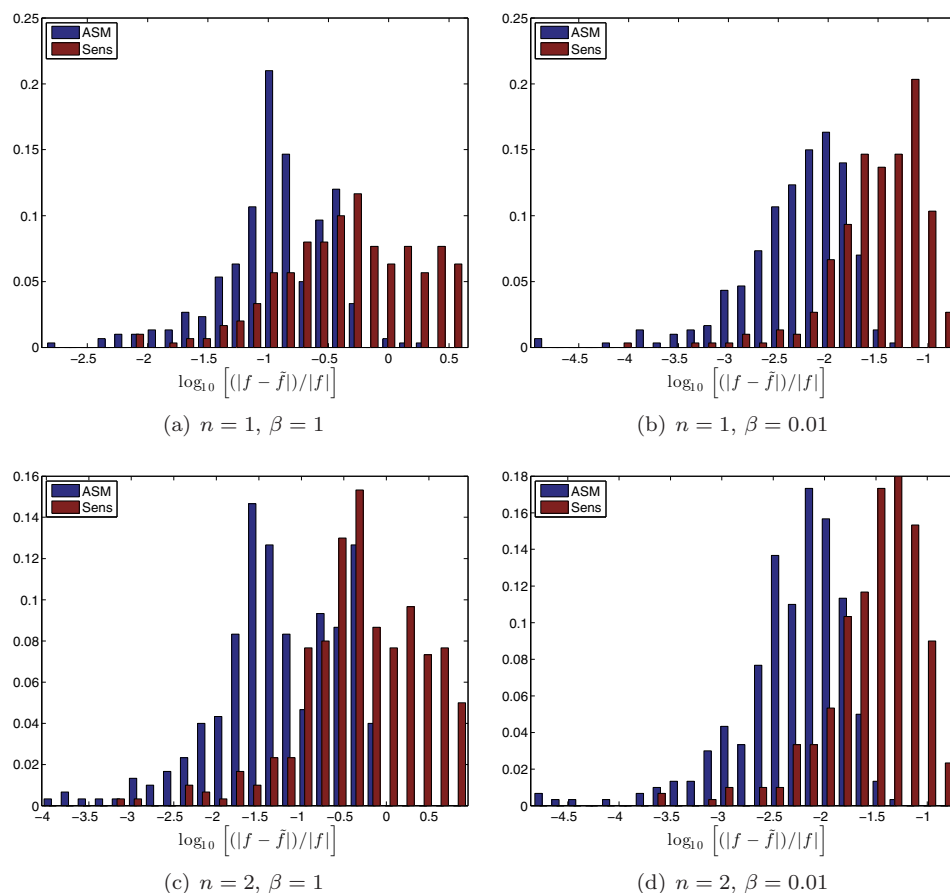
FIG. 5. *Histograms of the log of the relative error in the testing data for kriging surfaces constructed on the one- and two-dimensional subspaces for the two correlation lengths. (Colors are visible in the electronic version.)*

We mention the costs of these two approaches in terms of the number of function and gradient evaluations. The active subspace method used $M = 300$ samples of the gradient to approximate the covariance matrix. Since we had no prior knowledge that **C** would be low-rank, we chose $M = 300$ as a $3\times$ oversampling rate given the $m = 100$ variables. However, if we had suspected rapid decay in the singular values—and given that we would use at most only a two-dimensional subspace—we could have used many fewer gradient samples. With each gradient evaluation, we also get a function evaluation that we can use for testing the approximation. Given the eigenvectors defining the active subspace, we evaluated the function $P = 5$ or 25 more times for the one- and two-dimensional subspaces, respectively. We then tested the kriging surface on the subspace using the testing set computed along with the gradients.

The local sensitivity method used one gradient evaluation to find the first and second most important input variable. It also used 5 or 25 additional function evaluations to train a kriging surface on one- and two-dimensional coordinate subspaces. It then used the same 300 function evaluations as testing data. Thus, the local method was significantly cheaper but substantially less accurate.
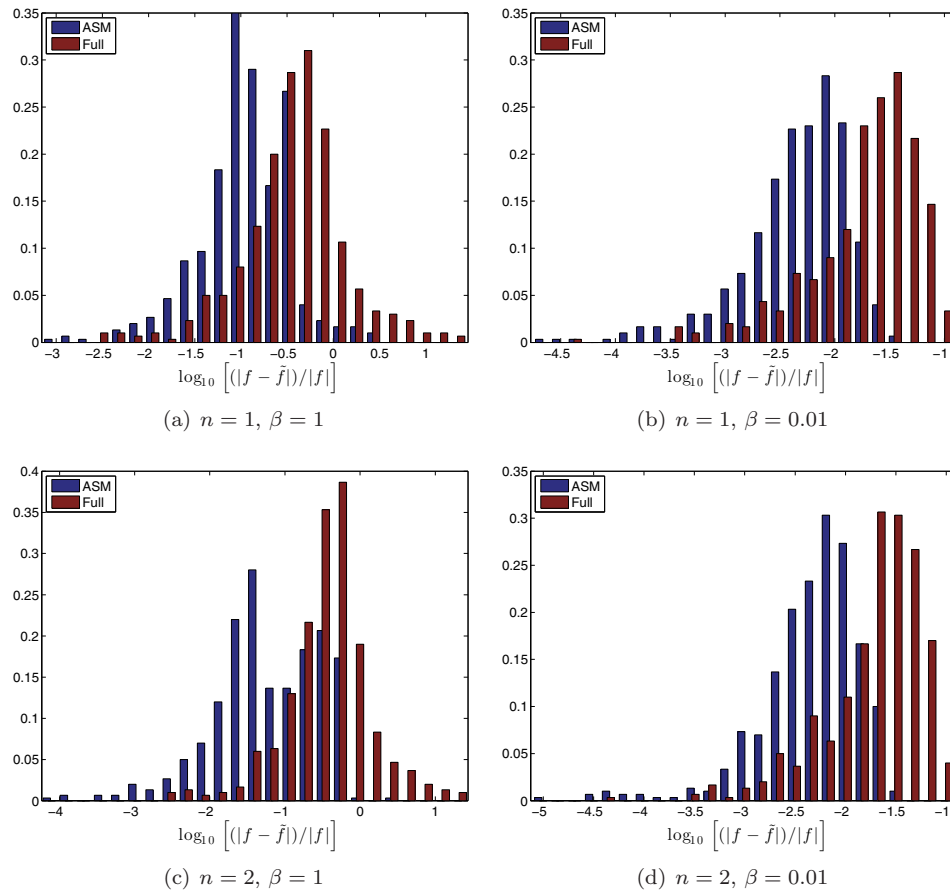
FIG. 6. *Histograms of the log of the relative error in the testing data for kriging surfaces constructed on the one- and two-dimensional subspaces for the two correlation lengths. ASM is the kriging surface using the active subspace, and Full is the kriging surface on the full space. (Colors are visible in the electronic version.)*

5.3. Comparison with kriging on $\mathcal{X}$. Finally, we compare the kriging surface constructed on the low-dimensional domain $\mathcal{Y}$ using the active subspace with a kriging surface on the full domain $\mathcal{X}$. The cost of computing the gradient $\nabla_{\mathbf{x}} f$ via adjoint computations is roughly twice the cost of computing the function f for a particular $\mathbf{x}$. Thus, the cost of constructing the active subspace approximation is roughly $3M + P$ function evaluations, where M is the number of gradient samples, and P is the number of evaluations for the design on $\mathcal{Y}$.

For a fair comparison, we build a kriging surface on the m-dimensional space $\mathcal{X}$ using $3M + P$ function evaluations. In this case $M = 300$, $P = 5$ for the one-dimensional subspace, and $P = 25$ for the two-dimensional subspace. We evaluate f at 500 additional points to create an independent testing set. Histograms of the testing errors are shown in Figure 6. For the same cost, the relative focus of the active subspace method produces a more accurate approximation than the response surface in $m = 100$ dimensions.

6. Summary and conclusions. Active subspace methods enable response surface approximations of a multivariate function on a low-dimensional subspace of the domain. We have analyzed a sequence of approximations that exploits the active subspace: a best approximation via conditional expectation, a Monte Carlo approximation of the best approximation, and a response surface trained with a few Monte Carlo estimates. We have used these analyses to motivate a computational procedure for detecting the directions defining the subspace and constructing a kriging surface on the subspace. We have applied this procedure to an elliptic PDE problem with a random field model for the coefficients. We compared the active subspace method with an approach based on the local sensitivity analysis and showed the superior performance of the active subspace method.

Loosely speaking, active subspace methods are appropriate for certain classes of functions that vary primarily in low-dimensional subspaces of the input. If there is no decay in the eigenvalues of $\mathbf{C}$, then the methods will perform poorly; constructing such functions is not difficult. However, we have found many high-dimensional applications in practice where the eigenvalues do decay quickly, and the functions respond well to active subspace methods [8, 15, 12, 13]. Most of those applications look similar to the one presented in section 5, where uncertainty in some spatially varying physical input can be represented by a series expansion, and the coefficients of the expansion are treated as random variables; such models arise frequently in UQ.

The computational method we have proposed is ripe for improvements and extensions. We have mentioned many such possibilities in section 4.3, and we are particularly interested in methods for using fewer evaluations of the gradient to compute the directions defining the active subspace. We will also pursue strategies that make better use of the function evaluations acquired during the gradient sampling.

REFERENCES

[1] A. C. ANTOULAS, *Approximation of Large-Scale Dynamical Systems*, Adv. Des. Control 6, SIAM, Philadelphia, 2005.

[2] I. BABUŠKA, R. TEMPONE, AND G. E. ZOURARIS, *Galerkin finite element approximations of stochastic elliptic partial differential equations*, SIAM J. Numer. Anal., 42 (2004), pp. 800–825.

[3] M. BEBENDORF, *A note on the Poincaré inequality for convex domains*, Z. Anal. Anwend., 22 (2003), pp. 751–756.

[4] N. BLIZNYUK, D. RUPPERT, C. SHOEMAKER, R. REGIS, S. WILD, AND P. MUGUNTHAN, *Bayesian calibration and uncertainty analysis for computationally expensive models using optimization and radial basis function approximation*, J. Comput. Graph. Statist., 17 (2008).

[5] A. E. BRYSON AND Y. C. HO, *Applied Optimal Control: Optimization, Estimation, and Control*, Hemisphere, London, 1975.

[6] J. F. CAI, E. J. CANDES, AND Z. SHEN, *A singular value thresholding algorithm for matrix completion*, SIAM J. Optim., 20 (2010), pp. 1956–1982.

[7] K. S. CHAN AND C. J. GEYER, *Discussion: Markov chains for exploring posterior distributions*, Ann. Statist., 22 (1994), pp. 1747–1758.

[8] H. CHEN, Q. WANG, R. HU, AND P. CONSTANTINE, *Conditional sampling and experiment design for quantifying manufacturing error of a transonic airfoil*, in Proceedings of the 49th Aerospace Sciences Meeting, 2011.

[9] L. H. CHEN, *An inequality for the multivariate normal distribution*, J. Multivariate Anal., 12 (1982), pp. 306–315.

[10] S. CHIB AND E. GREENBERG, *Understanding the Metropolis-Hastings algorithm*, Amer. Statist., 49 (1995), pp. 327–335.

[11] A. COHEN, I. DAUBECHIES, R. DEVORE, G. KERKYACHARIAN, AND D. PICARD, *Capturing ridge functions in high dimensions from point queries*, Constr. Approx., 35 (2012), pp. 225–243.

[12] P. G. CONSTANTINE, Q. WANG, A. DOOSTAN, AND G. IACCARINO, *A surrogate-accelerated Bayesian inverse analysis of the HyShot II flight data*, in Proceedings of the 49th Aerospace Sciences Meeting, 2011.

[13] P. G. Constantine, Q. Wang, and G. Iaccarino, *A Method for Spatial Sensitivity Analysis*, Center for Turbulence Research, Stanford, CA, 2012.

[14] A. Doucet, *Sequential Monte Carlo Methods*, John Wiley & Sons, New York, 2001.

[15] E. Dow and Q. Wang, *Output based dimensionality reduction of geometric variability in compressor blades*, in Proceedings of the 51st Aerospace Sciences Meeting, 2013.

[16] M. Fornasier, K. Schnass, and J. Vybiral, *Learning functions of few arbitrary linear parameters in high dimensions*, Found. Comput. Math., 12 (2012), pp. 229–262.

[17] K. Fukuda, *From the zonotope construction to the minkowski addition of convex polytopes*, J. Symbolic Comput., 38 (2004), pp. 1261–1272.

[18] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, New York, 1981.

[19] A. A. Giunta, J. M. McFarland, L. P. Swiler, and M. S. Eldred, *The promise and peril of uncertainty quantification using response surface approximations*, Struct. Infrastructure Engrg., 2 (2006), pp. 175–189.

[20] A. Griewank, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, SIAM, Philadelphia, 2000.

[21] N. Halko, P. G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288.

[22] A. Jameson, *Aerodynamic design via control theory*, J. Sci. Comput., 3 (1988), pp. 233–260.

[23] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed., Springer-Verlag, New York, 2002.

[24] D. R. Jones, *A taxonomy of global optimization methods based on response surfaces*, J. Global Optim., 21 (2001), pp. 345–383.

[25] J. R. Koehler and A. B. Owen, *Computer Experiments*, Handbook of Statist. 13, Elsevier, Amsterdam, 1996, pp. 261–308.

[26] J. Li and D. Xiu, *Evaluation of failure probability via surrogate models*, J. Comput. Phys., 229 (2010), pp. 8966–8980.

[27] C. Lieberman, K. Willcox, and O. Ghattas, *Parameter and state model reduction for large-scale statistical inverse problems*, SIAM J. Sci. Comput., 32 (2010), pp. 2523–2542.

[28] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, Springer, New York, 2008.

[29] Y. M. Marzouk, H. N. Najm, and L. A. Rahn, *Stochastic spectral methods for efficient Bayesian solution of inverse problems*, J. Comput. Phys., 224 (2007), pp. 560–586.

[30] A. B. Owen, *Variance components and generalized Sobol' indices*, SIAM/ASA J. Uncertainty Quantification, 1 (2013), pp. 19–41.

[31] A. B. Owen, *Monte Carlo Theory, Methods and Examples*, http://www-stat.stanford.edu/~owen/mc (2013).

[32] P. O. Persson and G. Strang, *A simple mesh generator in matlab*, SIAM Rev., 46 (2004), pp. 329–345.

[33] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*, vol. 1, MIT Press, Cambridge, MA, 2006.

[34] T. M. Russi, *Uncertainty Quantification with Experimental Data and Complex System Models*, Ph.D. thesis, UC Berkeley, 2010.

[35] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, UK, 1992.

[36] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Global Sensitivity Analysis: The Primer*, Wiley-Interscience, New York, 2008.

[37] L. Sirovich, *Turbulence and the dynamics of coherent structures,* i-iii, Quart. Appl. Math., 45 (1987), pp. 561–590.

[38] H. Wendland, *Scattered Data Approximation*, vol. 2, Cambridge University Press, Cambridge, UK, 2005.

[39] S. M. Wild, R. G. Regis, and C. A. Shoemaker, *Orbit: Optimization by radial basis function interpolation in trust-regions*, SIAM J. Sci. Comput., 30 (2008), pp. 3197–3219.

[40] D. Williams, *Probability with Martingales*, Cambridge University Press, Cambridge, UK, 1991.

[41] K. Zhou, J. C. Doyle, K. Glover, et al., *Robust and Optimal Control*, Prentice Hall, Englewood Cliffs, NJ, 1996.