

DATA-DRIVEN POLYNOMIAL RIDGE APPROXIMATION USING VARIABLE PROJECTION*

JEFFREY M. HOKANSON[†] AND PAUL G. CONSTANTINE[†]

Abstract. Inexpensive surrogates are useful for reducing the cost of science and engineering studies involving large-scale, complex computational models with many input parameters. A ridge approximation is one class of surrogate that models a quantity of interest as a nonlinear function of a few linear combinations of the input parameters. When used in parameter studies (e.g., optimization or uncertainty quantification), ridge approximations allow the low-dimensional structure to be exploited, reducing the effective dimension. We introduce a new, fast algorithm for constructing a ridge approximation where the nonlinear function is a polynomial. This polynomial ridge approximation is chosen to minimize least squares mismatch between the surrogate and the quantity of interest on a given set of inputs. Naively, this would require optimizing both the polynomial coefficients and the linear combination of weights, the latter of which define a low-dimensional subspace of the input space. However, given a fixed subspace the optimal polynomial can be found by solving a linear least-squares problem. Hence using *variable projection* the polynomial can be implicitly defined, leaving an optimization problem over the subspace alone. Here we develop an algorithm that finds this polynomial ridge approximation by minimizing over the Grassmann manifold of low-dimensional subspaces using a Gauss–Newton method. Our Gauss–Newton method has superior theoretical guarantees and faster convergence on our numerical examples than the alternating approach for polynomial ridge approximation earlier proposed by Constantine, Eftekhari, Hokanson, and Ward [*Comput. Methods Appl. Mech. Engrg.*, 326 (2017), pp. 402–421] that alternates between (i) optimizing the polynomial coefficients given the subspace and (ii) optimizing the subspace given the coefficients.

Key words. active subspaces, emulator, Grassmann manifold, response surface, ridge function, variable projection

AMS subject classifications. 49M15, 62J02, 90C53

DOI. 10.1137/17M1117690

1. Introduction. Many problems in uncertainty quantification [49, 50] and design [48, 55] involve a scalar quantity of interest derived from a complex model’s output that depends on the model inputs. Here we denote the map from model inputs to the quantity of interest as $f : \mathcal{D} \subseteq \mathbb{R}^m \rightarrow \mathbb{R}$. In many cases evaluating the quantity of interest is too expensive to permit the number of evaluations needed for the design or uncertainty study—e.g., estimating the distribution of f given a probability distribution for $\mathbf{x} \in \mathcal{D}$ or optimizing f over design variables $\mathbf{x} \in \mathcal{D}$. One approach to reduce the parameter study’s cost is to construct an inexpensive surrogate, also known as a *response surface* [31, 37] or an *emulator* [46], using pairs of inputs $\{\mathbf{x}_i\}_{i=1}^M \subset \mathcal{D}$ and outputs $\{f(\mathbf{x}_i)\}_{i=1}^M \subset \mathbb{R}$. In high dimensions (i.e., when m is large), building this surrogate is challenging since the number of parameters in the surrogate often grows exponentially in the input dimension; for example, a polynomial surrogate of total

*Submitted to the journal’s Methods and Algorithms for Scientific Computing section February 21, 2017; accepted for publication (in revised form) January 26, 2018; published electronically June 5, 2018.

<http://www.siam.org/journals/sisc/40-3/M111769.html>

Funding: This work was supported by Department of Defense, Defense Advanced Research Project Agency’s program Enabling Quantification of Uncertainty in Physical Systems. The second author’s work was partly supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under award DE-SC-0011077.

[†]Department of Computer Science, University of Colorado Boulder, 1111 Engineering Drive, Boulder, CO 80309 (Jeffrey.Hokanson@colorado.edu, Paul.Constantine@colorado.edu).

degree p has $\mathcal{O}(p^m)$ parameters (polynomial coefficients) as $m \rightarrow \infty$. In such cases, an exponential number of samples are required to yield a well-posed problem for constructing the surrogate. One workaround is to use surrogate models where the number of parameters grows more slowly; however, to justify such low-dimensional models, we must assume the function being approximated admits the related low-dimensional structure. Here we assume that the quantity of interest f varies primarily along $n < m$ directions in its m -dimensional input space—that is, f has an n -dimensional *active subspace* [8]. Such a structure has been demonstrated in quantities of interest arising in a wide range of computational science applications: integrated hydrologic models [29, 30], a solar cell circuit model [13], a subsurface permeability model [21], a lithium-ion battery model [9], a magnetohydrodynamics power generation model [22], a hypersonic scramjet model [12], an annular combustor model [4], models of turbomachinery [47], satellite system models [27], in-host HIV models [33], and computational models of aerospace vehicles [34, 35] and automobiles [39].

For functions with this low-dimensional structure an appropriate surrogate model is a *ridge function* [43]: the composition of a linear map from \mathbb{R}^m to \mathbb{R}^n with a nonlinear function g of n variables:

$$(1) \quad f(\mathbf{x}) \approx g(\mathbf{U}^\top \mathbf{x}), \quad \text{where } \mathbf{U} \in \mathbb{R}^{m \times n} \text{ and } \mathbf{U}^\top \mathbf{U} = \mathbf{I}.$$

In this paper we consider *polynomial ridge approximation* [11], where g is a multivariate polynomial of total degree p , and construct an efficient algorithm for *data-driven polynomial ridge approximation* that chooses g and \mathbf{U} to minimize the 2-norm misfit on a training set $\{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^M$:

$$(2) \quad \underset{\substack{g \in \mathbb{P}^p(\mathbb{R}^n) \\ \text{Range } \mathbf{U} \in \mathbb{G}(n, \mathbb{R}^m)}}{\text{minimize}} \quad \sum_{i=1}^M [f(\mathbf{x}_i) - g(\mathbf{U}^\top \mathbf{x}_i)]^2,$$

where $\mathbb{P}^p(\mathbb{R}^n)$ denotes the set of polynomials on \mathbb{R}^n of total degree p and $\mathbb{G}(n, \mathbb{R}^m)$ denotes the Grassmann manifold of n dimensional subspaces of \mathbb{R}^m . Figure 1 provides an example of this approximation on a toy problem. By exploiting ridge structure, fewer samples of the quantity of interest are required to make this approximation problem overdetermined. For example, whereas a polynomial of total degree p on \mathbb{R}^m requires $\binom{m+p}{p}$ samples, a polynomial ridge approximation of total degree p on an n -dimensional subspace requires only $\binom{n+p}{p} + mn$.

In the remainder of this paper we develop an efficient algorithm for solving the least squares polynomial ridge approximation problem (2) by exploiting its inherent structure. To begin, we first review the existing literature from the applied math and statistics communities on ridge functions in section 2. Then to start building our algorithm for polynomial ridge approximation, we show how *variable projection* [23] can be used to implicitly construct the polynomial approximation given a subspace defined by the range of \mathbf{U} in section 3. We further address the numerical issues inherent in polynomial approximation by using a Legendre basis and employing shifting and scaling of the projected coordinates $\mathbf{U}^\top \mathbf{x}_i$. Then, with an optimization problem posed over the subspace spanned by \mathbf{U} alone, we use techniques for optimization on the Grassmann manifold developed by Edelman, Arias, and Smith [17]. Due to the structure of the Jacobian, we are able to develop an efficient Gauss–Newton algorithm as described in section 4. We compare the performance of our data-driven polynomial ridge approximation algorithm to the alternating approach described by Constantine, Eftekhari, Hokanson, and Ward [11, Alg. 2]. Their algorithm constructs the ridge approximation by alternating between minimizing the polynomial given a fixed subspace

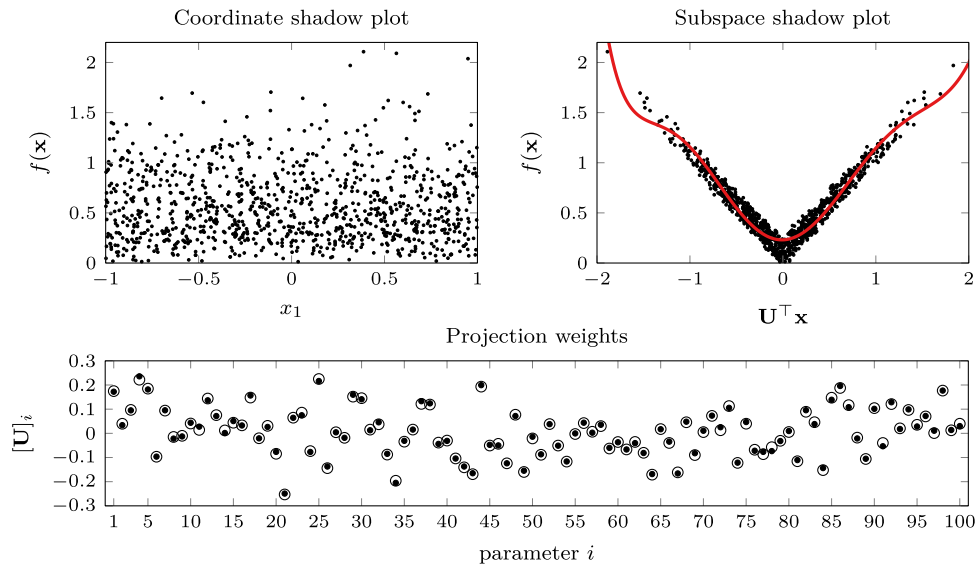


FIG. 1. Ridge approximations reveal structure not present in coordinate perspectives. In this toy example, $f : \mathcal{D} := [-1, 1]^{100} \rightarrow \mathbb{R}$ with $f(\mathbf{x}) = |\hat{\mathbf{u}}^\top \mathbf{x}| + 0.1(\sin(1000[\mathbf{x}]_2) + 1)$, where $\hat{\mathbf{u}}$ has been sampled uniformly on the unit sphere and the sine term simulates deterministic noise. Viewed from a coordinate perspective on the upper left, no structure is present. However, after fitting a ridge function to $N = 1000$ samples of f with polynomial degree $p = 7$ and subspace dimension $n = 1$ using our Algorithm 1, the structure of f is revealed by looking along the recovered subspace \mathbf{U} . On the bottom, we see that the coefficients of \mathbf{U} in the ridge approximation, \bullet , closely match the coefficients of $\hat{\mathbf{u}}$, \circ .

and then minimizing the subspace given a fixed polynomial. Our algorithm provides improved performance due to the faster convergence of Gauss–Newton-like algorithms and a more careful implementation that exploits the structure of the ridge approximation problem during the subspace optimization step. The improved performance can aid model selection studies—e.g., cross-validation—that find the best parameters of the approximation, such as the subspace dimension n and the polynomial degree p ; addressing this issue is beyond the scope of the present work. In section 5 we provide examples comparing the Gauss–Newton method to the alternating approach on toy problems and then further demonstrate the effectiveness of our algorithm by constructing polynomial ridge approximations of two f ’s from application problems: an 18-dimensional airfoil model [8, sect. 5.3.1] and a 100-dimensional elliptic PDE problem [10]. For completeness, we compare the performance of the polynomial ridge approximation to alternative surrogate models (Gaussian processes and sparse polynomial approximations) using a testing set of samples from the physics-based models. However, we emphasize that our goal is not necessarily to claim that ridge approximation is always superior to alternative models—only that ridge approximation is appropriate for functions that vary primarily along a handful of directions in their domain.

2. Related ideas and literature. Approximating multivariate functions by ridge functions has been studied under various names by different applied mathematics and statistics subcommunities. Pinkus’ monograph [43] surveys the approximation theory related to ridge functions. Hastie, Tibshirani, and Friedman [25, Chap. 3.5] review the general idea of building regression models on a few derived variables that are linear combinations of the predictors and discuss options for choosing the linear

combination weights, e.g., principal components or partial least squares. In what follows, we review related ideas across the literature that we are aware of; some exposition mirrors the review in Constantine, Eftekhari, Hokanson, and Ward [11]. Although our nonlinear least squares approach may benefit from ideas embedded in these approaches, their precise application is outside the scope of this paper.

2.1. Projection pursuit regression. In the context of statistical regression, Friedman and Stuetzle [19] proposed *projection pursuit regression*, which models the quantity of interest as a sum of one-dimensional ridge functions:

$$(3) \quad y_i = \sum_{k=1}^r g_k(\mathbf{u}_k^T \mathbf{x}_i) + \varepsilon_i, \quad \mathbf{u}_k \in \mathbb{R}^m,$$

where \mathbf{x}_i 's are samples of the predictors, y_i 's are the associated responses, and ε_i 's model random noise—all standard elements of statistical regression [56]. The g_k 's are smooth univariate functions (e.g., splines), and the \mathbf{u}_k 's are the directions of the ridge approximation. To fit the projection pursuit regression model, one minimizes the mean-squared error over the directions $\{\mathbf{u}_k\}$ and the parameters of $\{g_k\}$. Motivated by the projection pursuit regression model, Diaconis and Shahshahani [15] studied the approximation properties of nonlinear functions (g_k in (3)) of linear combinations of the variables ($\mathbf{u}_k^T \mathbf{x}$ in (3)). Huber [28] surveyed a wide class of projection pursuit approaches across an array of multivariate problems; by his terminology, *ridge approximation* could be called *projection pursuit approximation*. Chapter 11 of Hastie, Tibshirani, and Friedman [25] links projection pursuit regression to neural networks, which uses ridge functions with particular choices for the g_k 's (e.g., the sigmoid function). Although the optimization problem may be similar, the statistical regression context is different from the approximation context, since there is no inherent randomness (e.g., ε_i in (3)) in the approximation problem.

2.2. Sufficient dimension reduction. In the context of statistical regression there is also a vast body of literature that devises methods for finding a low-dimensional linear subspace of the predictor space that is statistically sufficient to characterize the predictor/response relationship; see the well-known text by Cook [14] and the more modern review by Adraghi and Cook [2]. From this literature, the minimum average variance estimate (MAVE) method [59] uses an optimization formulation similar to (2) to identify the dimension reduction subspace. However, the space of functions for ridge approximation in this approach is local linear models—as opposed to a global polynomial of degree p —where locality is imposed by kernel-based weights in the objective function centered around each data point. A similar approach was used in Xia's multiple-index model for regression [58].

2.3. Gaussian processes with low-rank correlation models. In Gaussian process regression [44], the conditional mean of the Gaussian process model given data (e.g., $\{y_i\}$ as in (3)) is the model's prediction. This conditional mean is a linear combination of radial basis functions with centers at a set of points $\{\mathbf{x}_i\}$, where the form of the basis function is related to the Gaussian process' assumed correlation. Vivarelli and Williams [54] proposed a correlation model of the form

$$(4) \quad C(\mathbf{x}, \mathbf{x}') \propto \exp \left[-\frac{1}{2} (\mathbf{x} - \mathbf{x}')^T \mathbf{U} \mathbf{U}^T (\mathbf{x} - \mathbf{x}') \right],$$

where \mathbf{U} is a tall matrix. In effect, the resulting conditional mean is a function of linear combinations of the predictors, $\mathbf{U}^T \mathbf{x}$ —i.e., a ridge function. A maximum

likelihood estimate of \mathbf{U} is the minimizer of an optimization problem similar to (2). Tripathy, Bilonis, and Gonzalez [52] use a related approach from a Bayesian perspective in the context of uncertainty quantification, where the subspace defined by \mathbf{U} enables powerful dimension reduction. And Liu and Guillas [32] develop a related low-dimensional model with linear combinations of predictors for Gaussian processes; their approach leverages gradient-based dimension reduction proposed by Fukumizu and Leng [20] to find the linear combination weights.

2.4. Ridge function recovery. Recent work in constructive approximation seeks to recover the parameters of a ridge function from point queries [7, 18, 53]; that is, determine \mathbf{U} in the ridge function $f(\mathbf{x}) = g(\mathbf{U}^\top \mathbf{x})$ using pairs $\{\mathbf{x}_i, f(\mathbf{x}_i)\}$. Algorithms for identifying \mathbf{U} (e.g., Algorithm 2 in [18]) are quite different than optimizing a ridge approximation over \mathbf{U} . However, the recovery problem is similar in spirit to the ridge approximation problem.

3. Separable reformulation. The key to efficiently constructing a data-driven polynomial ridge approximation is exploiting structure to reduce the effective number of parameters. As stated in (2), this approximation problem requires minimizing over two sets of variables: the polynomial g and the subspace spanned by \mathbf{U} . Defining g through its expansion in a basis $\{\psi_j\}_{j=1}^N$ of polynomials of total degree p on \mathbb{R}^n , $\mathbb{P}^p(\mathbb{R}^n)$, we write g as the sum

$$(5) \quad g(\mathbf{y}) := \sum_{j=1}^N c_j \psi_j(\mathbf{y}), \quad \mathbf{y} \in \mathbb{R}^n, \quad N := \binom{n+p}{p},$$

where the coefficients c_j are entries of a vector $\mathbf{c} \in \mathbb{R}^N$ specifying the polynomial. Using this expansion, evaluating the ridge approximation at the inputs $\{\mathbf{x}_i\}_{i=1}^M$ is equivalent to the product of a Vandermonde-like matrix $\mathbf{V}(\mathbf{U})$ and the coefficients \mathbf{c} :

$$(6) \quad g(\mathbf{U}^\top \mathbf{x}_i) = [\mathbf{V}(\mathbf{U})\mathbf{c}]_i, \quad [\mathbf{V}(\mathbf{U})]_{i,j} := \psi_j(\mathbf{U}^\top \mathbf{x}_i), \quad \mathbf{V} : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{M \times N},$$

where $[\mathbf{V}(\mathbf{U})]_{i,j}$ denotes the i th row and j th column of $\mathbf{V}(\mathbf{U})$. The matrix-vector product (6) allows us to restate the polynomial ridge approximation problem (2) in terms of coefficients $\mathbf{c} \in \mathbb{R}^N$ rather than the polynomial $g \in \mathbb{P}^p(\mathbb{R}^n)$:

$$(7) \quad \underset{\substack{g \in \mathbb{P}^p(\mathbb{R}^n) \\ \text{Range } \mathbf{U} \in \mathbb{G}(n, \mathbb{R}^m)}}{\text{minimize}} \sum_{i=1}^M [f(\mathbf{x}_i) - g(\mathbf{U}^\top \mathbf{x}_i)]^2 \iff \underset{\substack{\mathbf{c} \in \mathbb{R}^N \\ \text{Range } \mathbf{U} \in \mathbb{G}(n, \mathbb{R}^m)}}{\text{minimize}} \|\mathbf{f} - \mathbf{V}(\mathbf{U})\mathbf{c}\|_2^2,$$

where $\mathbf{f} \in \mathbb{R}^M$ holds the values of f at \mathbf{x}_i , $[\mathbf{f}]_i := f(\mathbf{x}_i)$. This new formulation reveals that the polynomial ridge approximation problem is a *separable nonlinear least-squares problem*, as for a fixed \mathbf{U} , \mathbf{c} is easily found by solving a least-squares problem. This structure allows us to use *variable projection* [23] to define an equivalent optimization problem over \mathbf{U} alone and construct its Jacobian as described in subsection 3.2. Then in subsection 3.3 we prove that slices of the Jacobian are orthogonal to the range of \mathbf{U} , allowing us to reduce the cost of computing the Grassmann–Gauss–Newton step given in section 4. However, first we address the choice of polynomial basis $\{\psi_j\}_{j=1}^N$ as this choice has a profound influence on the conditioning of the approximation problem.

3.1. Choice of basis. Polynomial approximation has a well-deserved reputation for being ill-conditioned [26, Chap. 22]. For example, for a one-dimensional ridge function constructed in the monomial basis the matrix $\mathbf{V}(\mathbf{U})$ is a Vandermonde matrix,

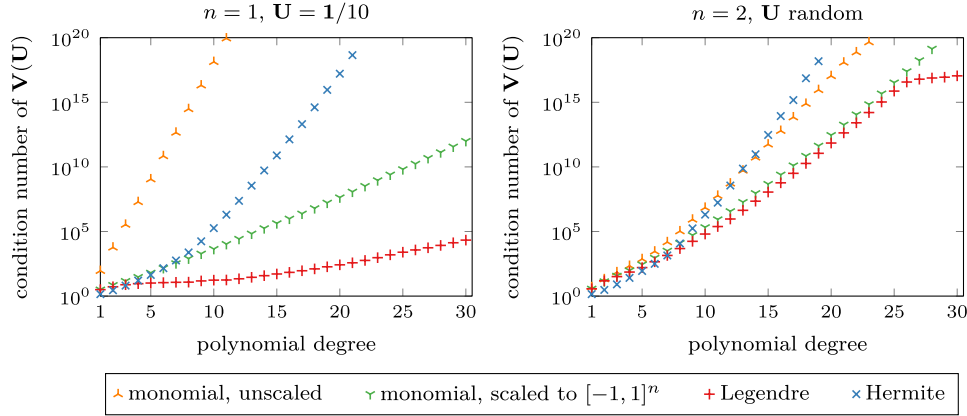


FIG. 2. The condition number of the matrix $\mathbf{V}(\mathbf{U}) \in \mathbb{R}^{M \times N}$ based on different polynomial bases generated from $M = 1000$ random samples drawn with uniform probability from $\mathcal{D} = [0, 1]^{100}$. On the left, with the subspace chosen to equally weight each component, we might expect the Hermite basis to be well conditioned as low-dimensional projections of high-dimensional data are often Gaussian [15]. On the right we show an example with a two-dimensional subspace drawn randomly with uniform probability on the sphere.

$$(8) \quad \mathbf{V}(\mathbf{U}) = \begin{bmatrix} 1 & (\mathbf{U}^\top \mathbf{x}_1) & \cdots & (\mathbf{U}^\top \mathbf{x}_1)^p \\ 1 & (\mathbf{U}^\top \mathbf{x}_2) & \cdots & (\mathbf{U}^\top \mathbf{x}_2)^p \\ \vdots & \vdots & & \vdots \\ 1 & (\mathbf{U}^\top \mathbf{x}_M) & \cdots & (\mathbf{U}^\top \mathbf{x}_M)^p \end{bmatrix} \in \mathbb{R}^{M \times (p+1)}, \quad n = 1, \quad \psi_j(y) = y^{j-1}.$$

Unless the sample points $\mathbf{y}_i = \mathbf{U}^\top \mathbf{x}_i \in \mathbb{R}$ are uniformly distributed on the complex unit circle, the condition number of this matrix grows exponentially in polynomial degree p [40]. As we assume we are given the sample points, our only hope for controlling the condition number comes from our choice of basis $\{\psi_j\}_{j=1}^N$. Here we invoke two strategies to control this condition number: shifting and scaling the projected points \mathbf{y}_i and choosing an appropriate orthogonal basis $\{\psi_j\}_{j=1}^N$.

We are free to shift and scale the projected points \mathbf{y}_i as any polynomial basis of total degree p is still a basis for this space when composed with an affine transformation. Namely, if $\boldsymbol{\eta} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the affine transformation

$$(9) \quad \boldsymbol{\eta}(\mathbf{y}) := \mathbf{a} + \mathbf{D}\mathbf{y}$$

and $\{\psi_j\}_{j=1}^N$ is a basis for $\mathbb{P}^p(\mathbb{R}^n)$, then $\{\psi_j \circ \boldsymbol{\eta}\}_{j=1}^N$ is also a basis for $\mathbb{P}^p(\mathbb{R}^n)$. By careful choice of $\boldsymbol{\eta}$ we can significantly decrease the condition number. For example, as shown on the left of Figure 2, shifting and scaling the projected points to the interval $[-1, 1]$ drastically decreases the condition number in the monomial basis. Similar results are seen for the other bases.

The second strategy for controlling the condition number is to change the polynomial basis. Ideally, we would choose a basis orthonormal under the weighted inner product induced by the projected points,

$$(10) \quad \int \psi_j(\mathbf{y}) \psi_k(\mathbf{y}) d\mu(\mathbf{y}) = \sum_{i=1}^M \psi_j(\mathbf{y}_i) \psi_k(\mathbf{y}_i) = \begin{cases} 1, & j = k; \\ 0, & j \neq k; \end{cases} \quad \mu(\mathbf{y}) = \sum_{i=1}^M \delta(\mathbf{y} - \mathbf{y}_i).$$

With this choice, $\mathbf{V}(\mathbf{U})$ would have a condition number of one. However, constructing this basis for an arbitrary set of points is fraught with the same ill-conditioning issues as $\mathbf{V}(\mathbf{U})$ itself. Instead, a reasonable heuristic is to choose a basis $\{\psi_j\}_{j=1}^M$ which is orthogonal under a weight μ that approximates the distribution of $\{\mathbf{y}_i\}_{i=1}^M$. For example, in one dimension, the Legendre polynomials are orthogonal with respect to the uniform measure on $[-1, 1]$ and the Hermite polynomials are orthogonal with respect to $\mu(y) = e^{-y^2/2}$. In Figure 2, we compare the conditioning of three different bases applied to 1000 points \mathbf{x}_i uniformly randomly chosen from the cube $[0, 1]^{100}$ and projected onto a one-dimensional subspace spanned by the ones vector. By the law of large numbers, the projected points $\{\mathbf{U}^\top \mathbf{x}_i\}_{i=1}^M$ are approximately normally distributed. After shifting and scaling these points to have zero mean and unit standard deviation, $\{\mathbf{y}_i = \mathbf{U}^\top \mathbf{x}_i\}_{i=1}^M$ will approximately sample the normal distribution, and hence we might expect the Hermite basis to be well conditioned. Although the Hermite basis is well conditioned for low degree polynomials, the condition number rapidly grows for high degree polynomials which is consistent with the observations by Hampton and Doostan [24]. In contrast, the Legendre polynomial basis when scaled and shifted to $[-1, 1]$ provides a well conditioned basis even when the degree is high.

When we seek a ridge approximation with two or more dimensions, we can form the basis $\{\psi_j\}_{j=1}^M$ for total degree polynomials in \mathbb{R}^n from a tensor product of one-dimensional polynomials. If this one-dimensional basis $\{\varphi_k\}_{k=0}^p \subset \mathbb{P}^p(\mathbb{R})$ has the property that φ_k is degree k , then the basis $\{\psi_j\}_{j=1}^N$ has elements

$$(11) \quad \psi_j(\mathbf{y}) = \prod_{k=1}^n \varphi_{[\alpha_j]_k}([\mathbf{y}]_k), \quad \alpha_j \in \mathbb{N}^n, \quad |\alpha_j| := \sum_{k=1}^n [\alpha_j]_k \leq p,$$

where $\{\alpha_j\}_{j=1}^N$ is an enumeration of the multi-indices satisfying $|\alpha_j| \leq p$. For the Legendre basis, this tensor product is still an orthogonal basis with respect to the uniform measure on $[-1, 1]^n$. However, the condition number of the matrix $\mathbf{V}(\mathbf{U})$ built from the Legendre basis, as well as the other bases, grows rapidly despite the scaling and shifting. Since the tensor product Legendre basis is least ill-conditioned, we use this basis in the remainder of this paper.

3.2. Variable projection. The key insight of variable projection is that if the nonlinear parameters \mathbf{U} are fixed, the linear parameters \mathbf{c} are easily recovered using the Moore–Penrose pseudoinverse, denoted by $^+$:

$$(12) \quad \underset{\mathbf{c}}{\text{minimize}} \quad \|\mathbf{f} - \mathbf{V}(\mathbf{U})\mathbf{c}\|_2 \quad \Rightarrow \quad \mathbf{c} = \mathbf{V}(\mathbf{U})^+ \mathbf{f}.$$

Then, replacing \mathbf{c} with $\mathbf{V}(\mathbf{U})^+ \mathbf{f}$ in (7) reveals an optimization problem over \mathbf{U} alone:

$$(13) \quad \underset{\text{Range } \mathbf{U} \in \mathbb{G}(n, \mathbb{R}^m)}{\text{minimize}} \quad \|\mathbf{f} - \mathbf{V}(\mathbf{U})\mathbf{V}(\mathbf{U})^+ \mathbf{f}\|_2^2.$$

Recognizing $\mathbf{V}(\mathbf{U})\mathbf{V}(\mathbf{U})^+$ as an orthogonal projector onto the range of $\mathbf{V}(\mathbf{U})$, we can rewrite the minimization problem as

$$(14) \quad \underset{\text{Range } \mathbf{U} \in \mathbb{G}(n, \mathbb{R}^m)}{\text{minimize}} \quad \|\mathbf{P}_{\mathbf{V}(\mathbf{U})}^\perp \mathbf{f}\|_2^2,$$

where $\mathbf{P}_{\mathbf{V}(\mathbf{U})}^\perp$ is the orthogonal projector onto the complement of the range of $\mathbf{V}(\mathbf{U})$.

Golub and Pereyra provide a formula for the derivative of the residual $\mathbf{r}(\mathbf{U}) := \mathbf{P}_{\mathbf{V}(\mathbf{U})}^\perp \mathbf{f}$ [23, eq. (5.4)] with respect to \mathbf{U} . Denoting the derivative of the residual as a tensor $\mathcal{J}(\mathbf{U}) \in \mathbb{R}^{M \times m \times n}$, where $[\mathcal{J}(\mathbf{U})]_{i,j,k} = \partial[\mathbf{r}(\mathbf{U})]_i / \partial[\mathbf{U}]_{j,k}$, then

$$(15) \quad [\mathcal{J}(\mathbf{U})]_{\cdot,j,k} = - \left[\left(\mathbf{P}_{\mathbf{V}(\mathbf{U})}^\perp \frac{\partial \mathbf{V}}{\partial [\mathbf{U}]_{j,k}}(\mathbf{U}) \mathbf{V}(\mathbf{U})^- \right) + \left(\mathbf{P}_{\mathbf{V}(\mathbf{U})}^\perp \frac{\partial \mathbf{V}}{\partial [\mathbf{U}]_{j,k}}(\mathbf{U}) \mathbf{V}(\mathbf{U})^- \right)^\top \right] \mathbf{f},$$

where $^-$ denotes a *least-squares pseudoinverse* such that $\mathbf{V}\mathbf{V}^-\mathbf{V} = \mathbf{V}$ and $\mathbf{V}\mathbf{V}^- = (\mathbf{V}\mathbf{V}^-)^\top$, a weaker pseudoinverse than the Moore–Penrose pseudoinverse [6, Chap. 6]. Then, from the definition of $\mathbf{V}(\mathbf{U})$ in (6) its derivative is

$$(16) \quad \left[\frac{\partial \mathbf{V}}{\partial [\mathbf{U}]_{k,\ell}}(\mathbf{U}) \right]_{i,j} = [\mathbf{x}_i]_k \frac{\partial \psi_j(\mathbf{y})}{\partial [\mathbf{y}]_\ell} \Big|_{\mathbf{y}=\mathbf{U}^\top \mathbf{x}_i}.$$

In particular, for a tensor product basis $\{\psi_j(\mathbf{y}) = \prod_{k=1}^n \varphi_{[\alpha_j]_k}(\boldsymbol{\eta}([\mathbf{y}]_k))\}_{j=1}^N$ composed with an affine transformation $\boldsymbol{\eta}(\mathbf{y}) = \mathbf{a} + \text{diag}(\mathbf{d})\mathbf{y}$, we have

$$(17) \quad \left[\frac{\partial \mathbf{V}}{\partial [\mathbf{U}]_{k,\ell}}(\mathbf{U}) \right]_{i,j} = [\mathbf{d}]_\ell [\mathbf{x}_i]_k \varphi'_{[\alpha_j]_\ell}([\boldsymbol{\eta}(\mathbf{U}^\top \mathbf{x}_i)]_\ell) \prod_{\substack{q=1 \\ q \neq \ell}}^n \varphi_{[\alpha_j]_q}([\boldsymbol{\eta}(\mathbf{U}^\top \mathbf{x}_i)]_q).$$

3.3. Orthogonality. In addition to constructing an explicit formula for the Jacobian of $\mathbf{r}(\mathbf{U})$, we also show that slices of this Jacobian $\mathcal{J}(\mathbf{U})$ are orthogonal to \mathbf{U} ; that is, $\mathbf{U}^\top [\mathcal{J}(\mathbf{U})]_{i,\cdot,\cdot} = \mathbf{0}$. This emerges due to the use of variable projection and the use of a polynomial basis for the columns of $\mathbf{V}(\mathbf{U})$ so that when \mathbf{U}^\top is multiplied by slices of the Jacobian, this derivative is mapped back into the range of the polynomial basis for $\mathbf{V}(\mathbf{U})$. Later, in subsection 4.2, we exploit this structure to reduce the computational burden during optimization.

THEOREM 1. *If $\mathcal{J}(\mathbf{U})$ is defined as in (15), then $\mathbf{U}^\top [\mathcal{J}(\mathbf{U})]_{i,\cdot,\cdot} = \mathbf{0} \ \forall i$.*

Proof. First note that

$$\begin{aligned} [\mathbf{U}^\top [\mathcal{J}(\mathbf{U})]_{i,\cdot,\cdot}]_{j,k} &= \sum_{\ell=1}^m [\mathbf{U}]_{\ell,j} [\mathcal{J}(\mathbf{U})]_{i,\ell,k} \\ &= - \sum_{\ell=1}^m [\mathbf{U}]_{\ell,j} \left[\mathbf{P}_{\mathbf{V}(\mathbf{U})}^\perp \frac{\partial \mathbf{V}(\mathbf{U})}{\partial [\mathbf{U}]_{\ell,k}} \mathbf{V}(\mathbf{U})^- \mathbf{f} + \mathbf{V}(\mathbf{U})^{-\top} \frac{\partial \mathbf{V}(\mathbf{U})}{\partial [\mathbf{U}]_{\ell,k}}^\top \mathbf{P}_{\mathbf{V}(\mathbf{U})}^\perp \mathbf{f} \right]_i. \end{aligned}$$

By linearity, we take the sum inside each term, leaving

$$(18) \quad = - \left[\mathbf{P}_{\mathbf{V}(\mathbf{U})}^\perp \left[\sum_{\ell=1}^m [\mathbf{U}]_{\ell,j} \frac{\partial \mathbf{V}(\mathbf{U})}{\partial [\mathbf{U}]_{\ell,k}} \right] \mathbf{V}(\mathbf{U})^- \mathbf{f} + \mathbf{V}(\mathbf{U})^{-\top} \left[\sum_{\ell=1}^m [\mathbf{U}]_{\ell,j} \frac{\partial \mathbf{V}(\mathbf{U})}{\partial [\mathbf{U}]_{\ell,k}}^\top \right] \mathbf{P}_{\mathbf{V}(\mathbf{U})}^\perp \mathbf{f} \right]_i.$$

Defining this interior sum as the matrix $\mathbf{W}^{(j,k)}$,

$$(19) \quad \mathbf{W}^{(j,k)} := \sum_{\ell=1}^m [\mathbf{U}]_{\ell,j} \frac{\partial \mathbf{V}(\mathbf{U})}{\partial [\mathbf{U}]_{\ell,k}},$$

we can show $\mathbf{W}^{(j,k)}$ is in the range of $\mathbf{V}(\mathbf{U})$. Without loss of generality, we work in the monomial tensor product basis where $\psi_j(\mathbf{y}) = \prod_{k=1}^n [\mathbf{y}]_k^{[\alpha_j]_k}$, where

$$(20) \quad [\mathbf{V}(\mathbf{U})]_{i,j} = \prod_{k=1}^n [\mathbf{U}^\top \mathbf{x}_i]_k^{[\alpha_j]_k},$$

$$(21) \quad \left[\frac{\partial \mathbf{V}}{\partial [\mathbf{U}]_{k,\ell}}(\mathbf{U}) \right]_{i,j} = [\alpha_j]_\ell [\mathbf{x}_i]_k [\mathbf{U}^\top \mathbf{x}_i]_\ell^{[\alpha_j]_\ell - 1} \prod_{\substack{q=1 \\ q \neq \ell}}^n [\mathbf{U}^\top \mathbf{x}_i]_q^{[\alpha_j]_q}.$$

Then the interior sum encodes the inner product $[\mathbf{U}^\top \mathbf{x}_r]_j$,

$$\begin{aligned} [\mathbf{W}^{(j,k)}]_{r,q} &= \left[\sum_{\ell} [\mathbf{U}]_{\ell,j} \frac{\partial \mathbf{V}(\mathbf{U})}{\partial [\mathbf{U}]_{\ell,k}} \right]_{r,q} = \sum_{\ell} [\mathbf{U}]_{\ell,j} [\alpha_q]_k [\mathbf{U}^\top \mathbf{x}_r]_k^{[\alpha_q]_k - 1} [\mathbf{x}_r]_\ell \prod_{\substack{s=1 \\ s \neq k}}^n [\mathbf{U}^\top \mathbf{x}_r]_s^{[\alpha_q]_s} \\ &= [\alpha_q]_k [\mathbf{U}^\top \mathbf{x}_r]_k^{[\alpha_q]_k - 1} [\mathbf{U}^\top \mathbf{x}_r]_j \prod_{\substack{s=1 \\ s \neq k}}^n [\mathbf{U}^\top \mathbf{x}_r]_s^{[\alpha_q]_s}. \end{aligned}$$

The term on the right is a polynomial of total degree at most p in $\mathbf{U}^\top \mathbf{x}_r$, since although the power on the k th term of $\mathbf{U}^\top \mathbf{x}_r$ has decreased by one, the j th term has increased by one, leaving the total degree of this term the same, namely less than or equal to p . Thus, as $\mathbf{W}^{(j,k)} \in \text{Range}(\mathbf{V}(\mathbf{U}))$ then $\mathbf{P}_{\mathbf{V}(\mathbf{U})}^\perp \mathbf{W}^{(j,k)} = \mathbf{0}$. As this product appears in both terms of (18), we conclude $\mathbf{U}^\top [\mathcal{J}]_{i,\cdot} = \mathbf{0}$. \square

4. Optimization on the Grassmann manifold. Having implicitly found the polynomial g using variable projection in the previous section, we now develop an algorithm for solving the ridge approximation problem posed over \mathbf{U} alone:

$$(22) \quad \underset{\text{Range } \mathbf{U} \in \mathbb{G}(n, \mathbb{R}^m)}{\text{minimize}} \quad \|\mathbf{P}_{\mathbf{V}(\mathbf{U})}^\perp \mathbf{f}\|_2^2.$$

This optimization problem over the Grassmann manifold of all n -dimensional subspaces of \mathbb{R}^m is more complicated than optimization on Euclidean space. Here we follow the approach of Edelman, Arias, and Smith [17], where the subspace is parameterized by a matrix $\mathbf{U} \in \mathbb{R}^{m \times n}$ with orthonormal columns; i.e., \mathbf{U} satisfies the constraint $\mathbf{U}^\top \mathbf{U} = \mathbf{I}$. We first review Newton's method on the Grassmann manifold following their construction before modifying their approach to construct a Gauss–Newton method on the Grassmann manifold for the data-driven polynomial ridge approximation problem (2). In the process we note that the orthogonality of \mathbf{U} to slices of the Jacobian as proved in Theorem 1 allows many terms to drop, simplifying the optimization problem. An alternative approach would be to follow the Gauss–Newton approach of Absil, Mahony, and Sepulchre [1, sect. 8.4.1] which removes the orthogonality constraint. However, by working in the framework of Edelman, Arias, and Smith we are able to show the additional terms in the Hessian drop due to the orthogonality result from Theorem 1.

4.1. Newton's method on the Grassmann manifold. To begin, we first review Newton's method on the Grassmann manifold, following Edelman, Arias, and Smith [17]. There are two key properties we consider: how to update \mathbf{U} given a search direction Δ and how to choose the search direction Δ using Newton's method.

When optimizing in a Euclidean space, given a search direction \mathbf{d} and an initial point \mathbf{x}_0 , the next iterate is chosen along the trajectory $\mathbf{x}(t) = \mathbf{x}_0 + t\mathbf{d}$ for $t \in (0, \infty)$, where t is selected to ensure convergence. However, if we were to apply the same search strategy to our parameterization of the Grassmann manifold, this would result in a point that does not obey the orthogonality constraint: $\mathbf{U}^\top \mathbf{U} = \mathbf{I}$. Instead, we replace the linear trajectory with a geodesic (a contour with constant derivative). Following [17, Thm. 2.3], if $\Delta \in \mathbb{R}^{m \times n}$ is the search direction that is tangent to the current estimate \mathbf{U}_0 , $\mathbf{U}_0^\top \Delta = \mathbf{0}$, then we choose $\mathbf{U}(t)$ on the geodesic

$$(23) \quad \mathbf{U}(t) = \mathbf{U}_0 \mathbf{Z} \cos(\Sigma t) \mathbf{Z}^\top + \mathbf{Y} \sin(\Sigma t) \mathbf{Z}^\top,$$

where $\Delta = \mathbf{Y} \Sigma \mathbf{Z}^\top$ is the short form singular value decomposition.

In addition to changing the search trajectories, optimization on the Grassmann manifold changes the gradient and Hessian. Here we consider an arbitrary function ϕ of a matrix $\mathbf{U} \in \mathbb{R}^{m \times n}$ with orthonormal columns. Following [17, eqs. (2.52) and (2.56)], the first and second derivatives of ϕ with respect to the entries of \mathbf{U} are

$$(24) \quad [\phi_{\mathbf{U}}]_{i,j} := \frac{\partial \phi}{\partial [\mathbf{U}]_{i,j}}, \quad \phi_{\mathbf{U}} \in \mathbb{R}^{m \times n};$$

$$(25) \quad [\phi_{\mathbf{U}\mathbf{U}}]_{i,j,k,\ell} := \frac{\partial^2 \phi}{\partial [\mathbf{U}]_{i,j} \partial [\mathbf{U}]_{k,\ell}}, \quad \phi_{\mathbf{U}\mathbf{U}} \in \mathbb{R}^{m \times n \times m \times n}.$$

With these definitions, the gradient of ϕ on the Grassmann manifold that is tangent to \mathbf{U} is [17, eq. (2.70)]

$$(26) \quad \text{grad } \phi = \phi_{\mathbf{U}} - \mathbf{U} \mathbf{U}^\top \phi_{\mathbf{U}} = \mathbf{P}_{\mathbf{U}}^\perp \phi_{\mathbf{U}} \in \mathbb{R}^{m \times n}.$$

The Hessian of ϕ on the Grassmann manifold, $\text{Hess } \phi \in \mathbb{R}^{m \times n \times m \times n}$, is defined by its action on two test matrices $\Delta, \mathbf{X} \in \mathbb{R}^{m \times n}$ [17, eq. (2.71)]:

$$(27) \quad \text{Hess } \phi(\Delta, \mathbf{X}) = \sum_{i,j,k,\ell} [\phi_{\mathbf{U}\mathbf{U}}]_{i,j,k,\ell} [\Delta]_{i,j} [\mathbf{X}]_{k,\ell} - \text{Tr}(\Delta^\top \mathbf{X} \mathbf{U}^\top \phi_{\mathbf{U}}).$$

Using these definitions, Newton's method on the Grassmann manifold at \mathbf{U} chooses the tangent search direction $\Delta \in \mathbb{R}^{m \times n}$ satisfying [17, eq. (2.58)]:

$$(28) \quad \mathbf{U}^\top \Delta = \mathbf{0} \quad \text{and} \quad \text{Hess } \phi(\Delta, \mathbf{X}) = -\langle \text{grad } \phi, \mathbf{X} \rangle \quad \forall \mathbf{X} \text{ such that } \mathbf{U}^\top \mathbf{X} = \mathbf{0},$$

where the inner product $\langle \cdot, \cdot \rangle$ on this space is [17, eq. (2.69)]

$$(29) \quad \langle \mathbf{X}, \mathbf{Y} \rangle = \text{Tr } \mathbf{X}^\top \mathbf{Y} = (\text{vec } \mathbf{X})^\top (\text{vec } \mathbf{Y})$$

and vec maps the matrix $\mathbb{R}^{m \times n}$ into the vector \mathbb{R}^{mn} .

4.2. Grassmann Gauss–Newton. The challenge with Newton's method is it requires second derivative information which is difficult to obtain for our problem. Here we replace the true Hessian with the Gauss–Newton approximation, yielding a Grassmann Gauss–Newton method. To summarize the key points of the following argument, the orthogonality of slices of the Jacobian to \mathbf{U} established in Theorem 1 allows us to drop the second term in the Hessian (27) and replace the normal equations (28) with a better conditioned least-squares problem (44) analogous to the Gauss–Newton method in Euclidean space [38, eq. (10.26)]. A similar Gauss–Newton

method is given in [1, sect. 8.4] where the subspace is parameterized by a matrix that is not necessarily orthogonal and uses a different geodesic step.

The objective function for data-driven polynomial ridge approximation is

$$(30) \quad \phi(\mathbf{U}) = \frac{1}{2} \|\mathbf{P}_{\mathbf{V}(\mathbf{U})}^\perp \mathbf{f}\|_2^2 = \frac{1}{2} \|\mathbf{r}(\mathbf{U})\|_2^2,$$

where first and second derivatives of ϕ are

$$(31) \quad \phi_{\mathbf{U}} = \sum_{i=1}^M [\mathcal{J}(\mathbf{U})]_{i,\cdot,\cdot} [\mathbf{r}(\mathbf{U})]_i;$$

$$(32) \quad [\phi_{\mathbf{U}\mathbf{U}}]_{i,j,k,\ell} = \sum_q [\mathcal{J}(\mathbf{U})]_{q,i,j} [\mathcal{J}(\mathbf{U})]_{q,k,\ell} + \sum_q [\mathbf{r}(\mathbf{U})]_q \frac{[\mathbf{r}(\mathbf{U})]_q}{\partial[\mathbf{U}]_{i,j} \partial[\mathbf{U}]_{k,\ell}}.$$

Invoking the Gauss–Newton approximation, we drop the second term above:

$$(33) \quad [\phi_{\mathbf{U}\mathbf{U}}]_{i,j,k,\ell} \approx [\tilde{\phi}_{\mathbf{U}\mathbf{U}}]_{i,j,k,\ell} := \sum_q [\mathcal{J}(\mathbf{U})]_{q,i,j} [\mathcal{J}(\mathbf{U})]_{q,k,\ell}.$$

Then replacing $\phi_{\mathbf{U}\mathbf{U}}$ with $\tilde{\phi}_{\mathbf{U}\mathbf{U}}$ in the Hessian (27) yields the approximate Hessian

$$(34) \quad \widetilde{\text{Hess}} \phi(\Delta, \mathbf{X}) := \sum_{i,j,k,\ell,q} [\mathcal{J}(\mathbf{U})]_{q,i,j} [\Delta]_{i,j} [\mathcal{J}(\mathbf{U})]_{q,k,\ell} [\mathbf{X}]_{k,\ell} - \text{Tr}(\Delta^\top \mathbf{X} \mathbf{U}^\top \phi_{\mathbf{U}}).$$

Immediately, we note that by Theorem 1 $\mathbf{U}^\top [\mathcal{J}(\mathbf{U})]_{i,\cdot,\cdot} = \mathbf{0}$ and hence

$$(35) \quad \mathbf{U}^\top \phi_{\mathbf{U}} = \sum_i \mathbf{U}^\top [\mathcal{J}(\mathbf{U})]_{i,\cdot,\cdot} [\mathbf{r}(\mathbf{U})]_i = \mathbf{0}.$$

Thus the second term drops out of the approximate Hessian (34), leaving

$$(36) \quad \widetilde{\text{Hess}} \phi(\Delta, \mathbf{X}) = \sum_{i,j,k,\ell,q} [\mathcal{J}(\mathbf{U})]_{q,i,j} [\Delta]_{i,j} [\mathcal{J}(\mathbf{U})]_{q,k,\ell} [\mathbf{X}]_{k,\ell}.$$

This summation can be rearranged to look like the more familiar Hessian approximation $\mathbf{J}^\top \mathbf{J}$ when the Jacobian \mathbf{J} is a matrix. Defining the vectorization operator for tensors that maps $\mathcal{J}(\mathbf{U}) \in \mathbb{R}^{M \times m \times n}$ to a matrix in $\mathbb{R}^{M \times mn}$, this product above is

$$(37) \quad \widetilde{\text{Hess}} \phi(\Delta, \mathbf{X}) = (\text{vec } \mathbf{X})^\top (\text{vec } \mathcal{J}(\mathbf{U}))^\top (\text{vec } \mathcal{J}(\mathbf{U})) (\text{vec } \Delta).$$

With this familiar expression for the approximate Hessian, we now seek to rework the Newton step from a square linear system into an overdetermined least-squares problem. First, invoking Theorem 1 via (35), we note that the gradient automatically satisfies the orthogonality constraint since $\mathbf{U}^\top \phi_{\mathbf{U}} = \mathbf{0}$:

$$(38) \quad \text{grad } \phi(\mathbf{U}) = \phi_{\mathbf{U}} - \mathbf{U} \mathbf{U}^\top \phi_{\mathbf{U}} = \phi_{\mathbf{U}} = (\text{vec } \mathcal{J}(\mathbf{U}))^\top \mathbf{r}(\mathbf{U}).$$

Then, examining the right-hand side of the Newton step (28),

$$(39) \quad \langle \text{grad } \phi, \mathbf{X} \rangle = (\text{vec } \mathbf{X})^\top (\text{vec } \mathcal{J}(\mathbf{U}))^\top \mathbf{r}(\mathbf{U}).$$

Thus, the Gauss–Newton step $\Delta \in \mathbb{R}^{m \times n}$ is the matrix satisfying $\mathbf{U}^\top \Delta = \mathbf{0}$ and

$$(40) \quad (\text{vec } \mathbf{X})^\top (\text{vec } \mathcal{J}(\mathbf{U}))^\top (\text{vec } \mathcal{J}(\mathbf{U})) (\text{vec } \Delta) = -(\text{vec } \mathbf{X})^\top (\text{vec } \mathcal{J}(\mathbf{U}))^\top \mathbf{r}(\mathbf{U}),$$

for all test matrices \mathbf{X} such that $\mathbf{U}^\top \mathbf{X} = \mathbf{0}$. To convert this into a least-squares problem, we first replace the constraint $\mathbf{U}^\top \mathbf{X} = \mathbf{0}$ by substituting \mathbf{X} by $\mathbf{P}_\mathbf{U}^\perp \mathbf{X}$, where $\mathbf{P}_\mathbf{U}^\perp$ is the orthogonal projector onto the complement of the range of \mathbf{U} , $\mathbf{P}_\mathbf{U}^\perp = \mathbf{I} - \mathbf{U}\mathbf{U}^\top$. This leaves a linear system of equations over all test matrices $\mathbf{X} \in \mathbb{R}^{m \times n}$:

$$(41) \quad (\text{vec } \mathbf{P}_\mathbf{U}^\perp \mathbf{X})^\top (\text{vec } \mathcal{J}(\mathbf{U}))^\top (\text{vec } \mathcal{J}(\mathbf{U})) (\text{vec } \Delta) = -(\text{vec } \mathbf{P}_\mathbf{U}^\perp \mathbf{X})^\top (\text{vec } \mathcal{J}(\mathbf{U}))^\top \mathbf{r}(\mathbf{U}).$$

On the left, the projector $\mathbf{P}_\mathbf{U}^\perp$ vanishes by Theorem 1:

$$(42) \quad \begin{aligned} [(\text{vec } \mathcal{J}(\mathbf{U}))(\text{vec } \mathbf{P}_\mathbf{U}^\perp \mathbf{X})]_i &= \text{Tr}[(\mathbf{P}_\mathbf{U}^\perp \mathbf{X})^\top [\mathcal{J}(\mathbf{U})]_{i, \cdot, \cdot}] = \text{Tr}[\mathbf{X}^\top (\mathbf{I} - \mathbf{U}\mathbf{U}^\top) [\mathcal{J}(\mathbf{U})]_{i, \cdot, \cdot}] \\ &= \text{Tr}[\mathbf{X}^\top [\mathcal{J}(\mathbf{U})]_{i, \cdot, \cdot}] = (\text{vec } \mathcal{J}(\mathbf{U}))(\text{vec } \mathbf{X}). \end{aligned}$$

Then, using the coordinate matrices $\mathbf{e}_i \mathbf{e}_j^\top$ as a basis for $\mathbf{X} \in \mathbb{R}^{m \times n}$, we then recover the normal equations

$$(43) \quad (\text{vec } \mathcal{J}(\mathbf{U}))^\top (\text{vec } \mathcal{J}(\mathbf{U})) (\text{vec } \Delta) = (\text{vec } \mathcal{J}(\mathbf{U}))^\top \mathbf{r}(\mathbf{U}).$$

Hence as in a Euclidean space (cf. [38, eq. (10.26)]), the Gauss–Newton step Δ is the solution to the linear least-squares problem

$$(44) \quad \underset{\substack{\Delta \in \mathbb{R}^{m \times n} \\ \mathbf{U}^\top \Delta = \mathbf{0}}}{\text{minimize}} \|\text{vec } \mathcal{J}(\mathbf{U}) \text{vec } \Delta - \mathbf{r}(\mathbf{U})\|_2^2.$$

Finally, we note that $\text{vec } \mathcal{J}(\mathbf{U})$ has a nullspace such that when (44) is solved using the pseudoinverse, the step Δ will automatically satisfy the constraint $\mathbf{U}^\top \Delta = \mathbf{0}$. Using (42) we can insert the projector $\mathbf{P}_\mathbf{U}^\perp$ into the Gauss–Newton step (44):

$$(45) \quad \text{vec } \mathcal{J}(\mathbf{U}) \text{vec } \Delta = \text{vec } \mathcal{J}(\mathbf{U}) \text{vec } (\mathbf{P}_\mathbf{U}^\perp \Delta) = \text{vec } \mathcal{J}(\mathbf{U}) [\mathbf{I}_n \otimes \mathbf{P}_\mathbf{U}^\perp] \text{vec } \Delta,$$

where \otimes is the Kronecker product. Thus $\text{vec } \mathcal{J}(\mathbf{U})$ has a nullspace of dimension n^2 which contains $\mathbf{I}_n \otimes \mathbf{U}\mathbf{U}^\top$, and hence if (44) is solved via the pseudoinverse, Δ will automatically obey the constraint $\mathbf{U}^\top \Delta = \mathbf{0}$. This reveals our Gauss–Newton step:

$$(46) \quad \text{vec } \Delta = -[\text{vec } \mathcal{J}(\mathbf{U})]^\dagger \mathbf{r}(\mathbf{U}) = -[[\mathcal{J}(\mathbf{U})]_{\cdot, \cdot, 1} \quad \dots \quad [\mathcal{J}(\mathbf{U})]_{\cdot, \cdot, n}]^\dagger \mathbf{r}(\mathbf{U}).$$

This pseudoinverse solution has a similar asymptotic cost to the normal equations: an $\mathcal{O}(M(mn)^2)$ operation SVD where $M > mn$ compared to an $\mathcal{O}((mn)^3)$ dense linear solve. However, the pseudoinverse solution is better conditioned, avoiding the squaring of the condition number in the normal equations [5, sect. 2.3.3].

4.3. Algorithm. We now combine the Gauss–Newton step (46) with backtracking along the geodesic (23) to construct a convergent data-driven polynomial ridge approximation algorithm. The complete algorithm is given in Algorithm 1, using a pseudoinverse to construct the Gauss–Newton step as in (46). We ensure convergence by inserting a check on line 14 to ensure Δ is always a descent direction, and if not, replacing it with the negative gradient. Then the sequence of Δ is a *gradient related sequence* [1, Def. 4.2.1] and the iterates \mathbf{U} converge to a stationary point where $\text{grad } \phi(\mathbf{U}) = \mathbf{0}$ by [1, Cor. 4.3.2] since the Grassmann manifold is compact [57, sect. 9].

When working with a relatively high order polynomial basis, we have found a small Armijo tolerance β , such as $\beta = 10^{-6}$, is necessary. As the polynomial degree increases, the rapid oscillation of these polynomials causes the gradient to grow rapidly, and without a small tolerance, all but the tiniest steps are rejected.

Algorithm 1. Variable projection polynomial ridge approximation.

Input : Sample points $\mathbf{X} \in \mathbb{R}^{M \times m}$; function values $\mathbf{f} \in \mathbb{R}^M$, $[\mathbf{f}]_i = f([\mathbf{X}]_{i,\cdot})$;
subspace dimension n ; polynomial degree p (if $p = 1$, then $n = 1$);
step length reduction factor $\gamma \in (0, 1)$; Armijo tolerance $\beta \in (0, 1)$.

Output : Active subspace $\mathbf{U} \in \mathbb{R}^{m \times n}$; polynomial coefficients $\mathbf{c} \in \mathbb{R}^N$, $N = \binom{n+p}{p}$.

- 1 Sample entries of $\mathbf{Z} \in \mathbb{R}^{m \times n}$ from a normal distribution;
- 2 Compute short form QR $\mathbf{U}\mathbf{R} \leftarrow \mathbf{Z}$ hence \mathbf{U} uniformly samples $\mathbb{G}(n, \mathbb{R}^m)$
- 3 **repeat**
- 4 Compute $\{\mathbf{y}_i = \mathbf{U}^\top \mathbf{x}_i\}_{i=1}^M$;
- 5 Construct affine transformation $\boldsymbol{\eta}$ of $\{\mathbf{y}_i\}_{i=1}^M$ to $[-1, 1]^n$;
- 6 Build $\mathbf{V}(\mathbf{U})$ using tensor product Legendre basis composed with $\boldsymbol{\eta}$ via (6):
 $\mathbf{V} \leftarrow \mathbf{V}(\mathbf{U})$;
- 7 Compute polynomial coefficients $\mathbf{c} \leftarrow \mathbf{V}^+ \mathbf{f}$;
- 8 Compute the residual: $\mathbf{r} \leftarrow \mathbf{r}(\mathbf{U}) = \mathbf{f} - \mathbf{V}\mathbf{c}$;
- 9 Build the Jacobian (15): $\mathcal{J} \leftarrow \mathcal{J}(\mathbf{U}) \in \mathbb{R}^{M \times m \times n}$;
- 10 Build the gradient (38): $\mathbf{G} \leftarrow \mathbf{G}(\mathbf{U}) = \sum_{i=1}^M [\mathcal{J}]_{i,\cdot,\cdot} [\mathbf{r}]_i$;
- 11 Compute the short form SVD: $\mathbf{Y}\boldsymbol{\Sigma}\mathbf{Z}^\top \leftarrow \text{vec } \mathcal{J}$;
- 12 Compute the Gauss–Newton step (46): $\text{vec } \boldsymbol{\Delta} \leftarrow -\sum_{i=1}^{mn-n^2} [\boldsymbol{\Sigma}]_{i,i}^{-1} [\mathbf{Z}]_{\cdot,i} [\mathbf{Y}^\top \mathbf{r}]_i$;
- 13 Compute slope along Gauss–Newton step: $\alpha \leftarrow \text{Tr } \mathbf{G}^\top \boldsymbol{\Delta} = (\text{vec } \mathbf{G})^\top (\text{vec } \boldsymbol{\Delta})$;
- 14 **if** $\alpha \geq 0$ **then** $\boldsymbol{\Delta}$ is not a descent direction
- 15 $\boldsymbol{\Delta} \leftarrow -\mathbf{G}$;
- 16 $\alpha \leftarrow \text{Tr } \mathbf{G}^\top \boldsymbol{\Delta}$;
- 17 Compute the short form SVD: $\mathbf{Y}\boldsymbol{\Sigma}\mathbf{Z}^\top \leftarrow \boldsymbol{\Delta}$;
- 18 **for** $t = \gamma^0, \gamma^1, \gamma^2, \dots$ **do** backtracking line search
- 19 Compute new step (23): $\mathbf{U}_+ \leftarrow \mathbf{U}\mathbf{Z} \cos(\boldsymbol{\Sigma}t)\mathbf{Z}^\top + \mathbf{Y} \sin(\boldsymbol{\Sigma}t)\mathbf{Z}^\top$;
- 20 Compute new residual: $\mathbf{r}_+ \leftarrow \mathbf{f} - \mathbf{V}(\mathbf{U}_+)\mathbf{V}(\mathbf{U}_+)^+ \mathbf{f}$;
- 21 **if** $\|\mathbf{r}_+\|_2 \leq \|\mathbf{r}\|_2 + \alpha\beta t$ **then** Armijo condition satisfied
- 22 **break**;
- 23 Update the estimate $\mathbf{U} \leftarrow \mathbf{U}_+$;
- 24 **until** \mathbf{U} converges;

In addition, our implementation equips this algorithm with three different convergence criteria: small change between \mathbf{U} and \mathbf{U}_+ as measured by the smallest canonical subspace angle, small change in the norm of the residual, and small gradient norm. Further, the algorithm places a feasibility constraint on the polynomial degree and subspace dimension. A linear polynomial ($p = 1$) with any dimensional subspace is equivalent to a ridge function on a one-dimensional subspace; namely, if $\mathbf{U} \in \mathbb{R}^{m \times n}$ spans an n -dimensional subspace, then

$$(47) \quad g(\mathbf{U}^\top \mathbf{x}) = c_0 + c_1 \mathbf{U}_{\cdot,1}^\top \mathbf{x} + \dots + c_n \mathbf{U}_{\cdot,n}^\top \mathbf{x} = c_0 + \left(\sum_{k=1}^n c_k \mathbf{U}_{\cdot,k} \right) \mathbf{x} = c_0 + \hat{c}_1 \hat{\mathbf{U}}^\top \mathbf{x},$$

where $\hat{\mathbf{U}} \in \mathbb{R}^{m \times 1}$ spans a one-dimensional subspace.

5. Examples. To demonstrate the effectiveness of our algorithm for ridge approximation we apply it to a mixture of synthetic and application problems. Code generating these examples along with an implementation of Algorithm 1 are available at <https://github.com/jeffrey-hokanson/varproridge>. First we compare the proposed algorithm to the alternating approach of [11] on a problem where the ridge function is

known a priori and examine both convergence and wall clock time. Next, we demonstrate that even though finding the ridge approximation is a nonconvex problem, the proposed algorithm frequently finds the global minimizer from a random initialization. Then we study how rapidly the ridge approximation identifies the active subspace of a test problem as the number of samples grows and compare these results to gradient based approaches for estimating the active subspace. Finally, we apply the proposed algorithm to two application problems: modeling lift and drag from a NACA0012 airfoil with 18 parameters and modeling mean solution on the Neumann boundary of an elliptic PDE with 100-random coefficients. We compare the accuracy of our ridge approximation to both Gaussian process and sparse surrogates and find that polynomial ridge approximations provide more accurate approximations on these application problems which exhibit ridge structure.

5.1. Convergence of the optimization iteration. As a first example, we compare the convergence of our proposed Gauss–Newton based algorithm to the alternating approach [11] on examples with and without a zero-residual solution. The alternating approach switches between optimizing for the polynomial coefficients \mathbf{c} and the subspace defined by \mathbf{U} at each iteration; i.e.,

$$(48) \quad \mathbf{c}_k \leftarrow \mathbf{V}(\mathbf{U}_{k-1})^+ \mathbf{f},$$

$$(49) \quad \mathbf{U}_k \leftarrow \underset{\text{Range } \mathbf{U} \in \mathbb{G}(m, \mathbb{R}^n)}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{f} - \mathbf{V}(\mathbf{U})\mathbf{c}_k\|_2^2.$$

The implementation of Constantine et al. allows the nonlinear optimization problem for \mathbf{U}_k to be solved using multiple steps of steepest descent using `pymanopt` [51].

Existing results describe the convergence of both of these algorithms for both the zero-residual and nonzero-residual cases. Iterates of a Gauss–Newton method converging to a zero-residual solution do so quadratically and those that converge to a nonzero-residual solution do so only superlinearly [38, sect. 10.3]. Following Ruhe and Wedin [45, section 3], variable projection on a separable problem (in their notation, Algorithm I) converges at the same rate as Gauss–Newton. Hence, our ridge approximation algorithm should converge quadratically for zero-residual problems and superlinearly otherwise. However, an alternating approach (in their notation, Algorithm III), such as the alternating approach for ridge approximation of Constantine et al., should converge at best only linearly. Our numerical experiments support this analysis. Figure 3 compares the convergence of the alternating and Gauss–Newton approaches. In this example f is taken to be a cubic ridge function on a two-dimensional subspace:

$$(50) \quad f : \mathcal{D} \subset \mathbb{R}^{10} \rightarrow \mathbb{R}; \quad f(\mathbf{x}) = (\mathbf{e}_1^\top \mathbf{x})^2 + (\mathbf{1}^\top \mathbf{x}/10)^3 + 1; \quad \mathcal{D} = [-1, 1]^{10}.$$

Sampling $M = 1000$ points uniformly over the domain, Figure 3 shows the per-iteration convergence history for 10 different initializations of each algorithm. As expected in the case with a zero-residual solution, our proposed Gauss–Newton approach converges quadratically while the alternating approach converges only linearly. The bottom row of this figure shows the case where Gaussian random noise with unit variance has been added to the function values $f(\mathbf{x}_i)$ ensuring there is not a zero-residual solution. Although our method no longer converges quadratically, our method converges in fewer iterations on average than the alternating approach.

Figure 3 also exposes two interesting features of the optimization. First, for both algorithms there is a plateau in the convergence history at around 10^{-2} in the zero-residual case and around $3 \cdot 10^{-2}$ in the nonzero-residual case. For both algorithms,

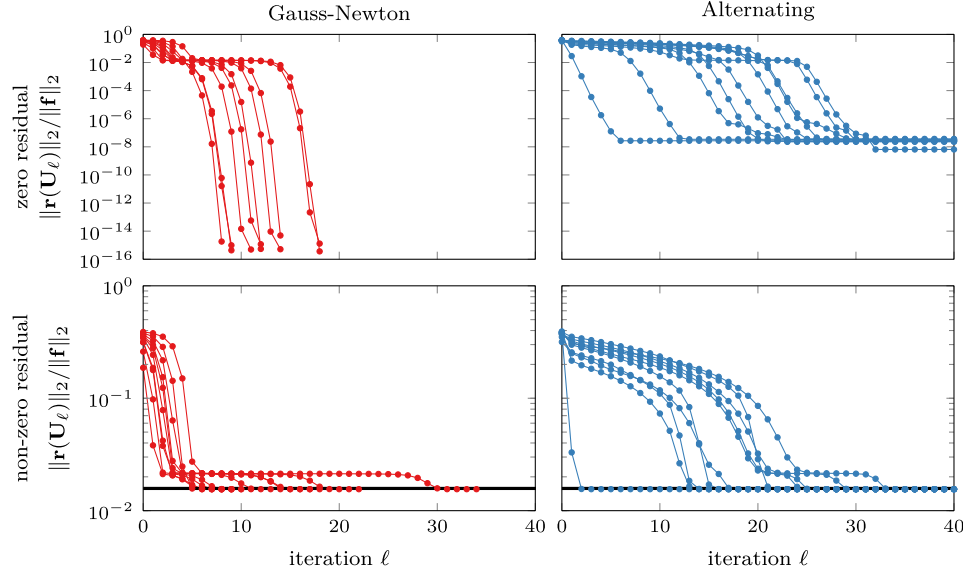


FIG. 3. A comparison of the per-iteration performance of our Gauss-Newton method (Algorithm 1) and the alternating approach of Constantine, Eftekhari, Hokanson, and Ward [11, Alg. 2] using 100 steepest descent iterations per alternating iteration. The top rows compare these algorithms when finding a two variable cubic ridge function approximation from $M = 1000$ uniform random samples of (50). The bottom row shows the performance when independent and identically distributed Gaussian random noise was added to each sample of f to provide a nonzero-residual solution; the black line indicates the norm of the noise introduced.

TABLE 1

Per-iteration cost of Gauss-Newton and alternating based approaches, including only M -dependent steps and treating evaluating the polynomial ψ as an $\mathcal{O}(1)$ cost.

Gauss-Newton		Alternating	
Step	Cost	Step	Cost
Fitting polynomial (SVD)	$\mathcal{O}(MN^2)$	Fitting polynomial (QR)	$\mathcal{O}(MN^2)$
Constructing Jacobian	$\mathcal{O}(MNmn)$	Constructing Jacobian	$\mathcal{O}(MNmn)$
Gauss-Newton direction	$\mathcal{O}(M(mn)^2)$	Steepest descent direction	$\mathcal{O}(MNmn)$
Step acceptance	$\mathcal{O}(MN^2)$	Step acceptance	$\mathcal{O}(MN)$

this happens when one of the two directions in the active subspace has been found. Second, the alternating algorithm on the zero-residual case stagnates with a residual around 10^{-8} ; however, the residual should be able to converge to an error of 10^{-14} . We suspect this is due to numerical issues in the implementation as the residual floor does not decrease as the termination criteria are made more strict.

5.2. Cost comparison. Comparing the cost of our variable projection Gauss-Newton based approach to the alternating approach of Constantine, Eftekhari, Hokanson, and Ward [11] is not simple. Although both algorithms scale linearly with the number of samples M , the constant multiplying M depends on the dimension of the ridge approximation n , the polynomial degree p , the dimension of the polynomial basis $N = \binom{n+p}{p}$, and the dimension of the input space m . Table 1 lists the dominant costs in each algorithm. This analysis motivates using multiple steepest descent steps with a fixed polynomial per alternating iteration as this step is cheap compared to

TABLE 2

Median wall clock time in seconds from ten replicates of each algorithm applied to identical zero-residual data consisting of $M = 1000$ uniform samples of $f_{n,p}$ from (51), initialized with the same random subspace \mathbf{U} , and stopped when the normalized residual reached 10^{-5} , which both algorithms achieved in Figure 3. The alternating method times represent the shortest time when using 1, 10, or 100 steps of steepest descent per alternating iteration. Experiments were conducted on a 2013 Mac Pro with a six-core Intel Xeon CPU E5-1650 v2 clocked at 3.50GHz with 16GB RAM.

p	Gauss–Newton					Alternating				
	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$
2	0.012	0.029	0.071	0.107	0.130	4.0	26.4	33.6	40.0	50.8
3	0.013	0.068	0.129	0.492	2.023	5.8	32.4	45.9	68.1	109.5
4	0.015	0.107	0.187	1.901	2.985	7.5	13.7	69.6	131.1	250.9
5	0.016	0.112	0.364	11.759	7.331	8.7	17.5	41.7	113.2	282.9

the cost of the polynomial fitting step. This cost analysis also suggests that for high-dimensional input spaces, the alternating approach may be cheaper as computing the descent direction scales like m^2 in the Gauss–Newton approach vs m in the steepest descent approach. However, which algorithm is faster for a given set of m , n , and p depends on the constants associated with each algorithm, which this analysis neglects.

To get a sense of which algorithm is faster in practice, we compare the wall clock time of each algorithm on a 10-dimensional test function

$$(51) \quad f_{n,p} : \mathcal{D} \subset \mathbb{R}^{10} \rightarrow \mathbb{R}; \quad f_{n,p}(\mathbf{x}) = (\mathbf{1}^\top \mathbf{x})^p + \sum_{j=1}^{n-1} (\mathbf{e}_j^\top \mathbf{x})^{p-1}; \quad \mathcal{D} = [-1, 1]^{10}$$

fitting $M = 1000$ points for a variety of polynomial degrees p and subspace dimensions n as shown in Table 2. As these results show, our proposed algorithm is significantly faster when measured in wall clock time. Part of this improvement is due to the quadratic convergence of our Gauss–Newton method that requires fewer iterations to terminate compared to the linearly convergent alternating method. However, a nontrivial portion of performance difference is due to our more careful implementation that is tightly coupled to the Grassmann optimization and exploits the structure of the Jacobian revealed in Theorem 1.

5.3. Convergence to the global minimizer. As the polynomial ridge approximation problem (2) is not necessarily a convex problem, one concern is that our proposed algorithm might converge to a spurious local minimizer rather than the global minimizer. However, our numerical results suggest our algorithm frequently converges to the global minimizer regardless of the initial subspace estimate. As an example, we consider a quadratic ridge approximation on an n -dimensional subspace built from $M = 1000$ samples of

$$(52) \quad f_n : \mathcal{D} \subset \mathbb{R}^{10} \rightarrow \mathbb{R}; \quad f_n(\mathbf{x}) = \sum_{j=1}^n (\mathbf{e}_j^\top \mathbf{x})^2; \quad \mathcal{D} = [-1, 1]^{10}.$$

As this function is a polynomial ridge function, we can assess if the algorithm has correctly converged if the residual is near zero. Table 3 shows the frequency with which our proposed algorithm terminated with an incorrect subspace. In general, this frequency is low and so it should be sufficient in many practical cases to try multiple random initializations, taking the best to ensure convergence to an approximate global minimizer.

TABLE 3

Probability of not finding the global minimizer given a random initial subspace of dimension n . This probability was estimated from 1000 trials of Algorithm 1 fitting a quadratic ridge function of n variables to $M = 1000$ samples of $f(\mathbf{x})$ from (52) taken uniformly over the domain.

$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$	$n = 10$
0.0%	10.8%	16.2%	15.2%	7.3%	7.6%	10.6%	7.1%	3.4%	0.0%

5.4. Convergence to the active subspace with increasing samples. As one goal of building a ridge approximation is to construct an inexpensive surrogate of an expensive function f , an important feature of this approximation is how many samples of this function are required to construct a good surrogate. Here we compare the ridge subspace found by our polynomial ridge approximation and the active subspace computed from the outer product of (approximate) gradients [8] to the true active subspace of a toy problem consisting of a one-dimensional quadratic ridge function plus low-amplitude sinusoidal oscillations:

$$(53) \quad f : [-1, 1]^m \rightarrow \mathbb{R}, \quad f(\mathbf{x}) = \frac{1}{2}(\mathbf{1}^\top \mathbf{x})^2 + \alpha \sum_{j=1}^m \cos(\beta \pi [\mathbf{x}]_j), \quad \alpha > 0, \quad \beta \in \mathbb{N}.$$

These oscillations are necessary so that f does not have an exact one-dimensional ridge structure, in which case both the polynomial ridge approximation and the active subspace approach will correctly estimate the active subspace with only $M = m + 3$ samples unless $\{\mathbf{x}_i\}_{i=1}^M$ is adversely chosen. For this toy problem, the outer product of gradients matrix has a closed form expression:

$$(54) \quad \mathbf{C} := \int_{\mathcal{D}} (\nabla f(\mathbf{x}))(\nabla f(\mathbf{x}))^\top d\mu(\mathbf{x}) = \mathbf{1}\mathbf{1}^\top + (\alpha\beta\pi)^2 \mathbf{I}.$$

The leading eigenvector of this matrix is always the ones vector for any value of α and β , but α and β affect each approach differently. By increasing α , the “noise”—the response that cannot be explained by a one-dimensional ridge approximation—increases. By increasing β , the frequency of oscillations increases, increasing the oscillations in the gradients appearing in \mathbf{C} . Together, α and β determine the first and second eigenvalues of \mathbf{C} , namely $m + (\alpha\beta\pi)^2$ and $(\alpha\beta\pi)^2$, and the relative eigenvalue gap determines the convergence of the Monte Carlo estimate of \mathbf{C} , $\hat{\mathbf{C}}$ [8, Cor. 3.10]:

$$(55) \quad \hat{\mathbf{C}} := \frac{1}{L} \sum_{i=1}^L (\nabla f(\mathbf{x}_i))(\nabla f(\mathbf{x}_i))^\top.$$

To provide a fair comparison in terms of function evaluations, we compare our polynomial ridge approximation approach to the Monte Carlo estimate of the active subspace using a finite difference approximation of the gradient:

$$(56) \quad \tilde{\mathbf{C}} := \frac{1}{L} \sum_{i=1}^L (\tilde{\nabla} f(\mathbf{x}_i))(\tilde{\nabla} f(\mathbf{x}_i))^\top, \quad [\tilde{\nabla} f(\mathbf{x}_i)]_j = \frac{f(\mathbf{x}_i + h\mathbf{e}_j) - f(\mathbf{x}_i)}{h}.$$

Although a finite difference gradient is used here, f is smooth so the error introduced by the finite difference gradient is negligible. As seen in Figure 4, the Monte Carlo estimate converges at the expected $\mathcal{O}(M^{-1/2})$ rate. When the relative gap remains the same, as in $\alpha = 0.02$, $\beta = 1$ and $\alpha = 0.004$, $\beta = 5$ cases of (53), the convergence of the gradient based active subspace estimate is similar.

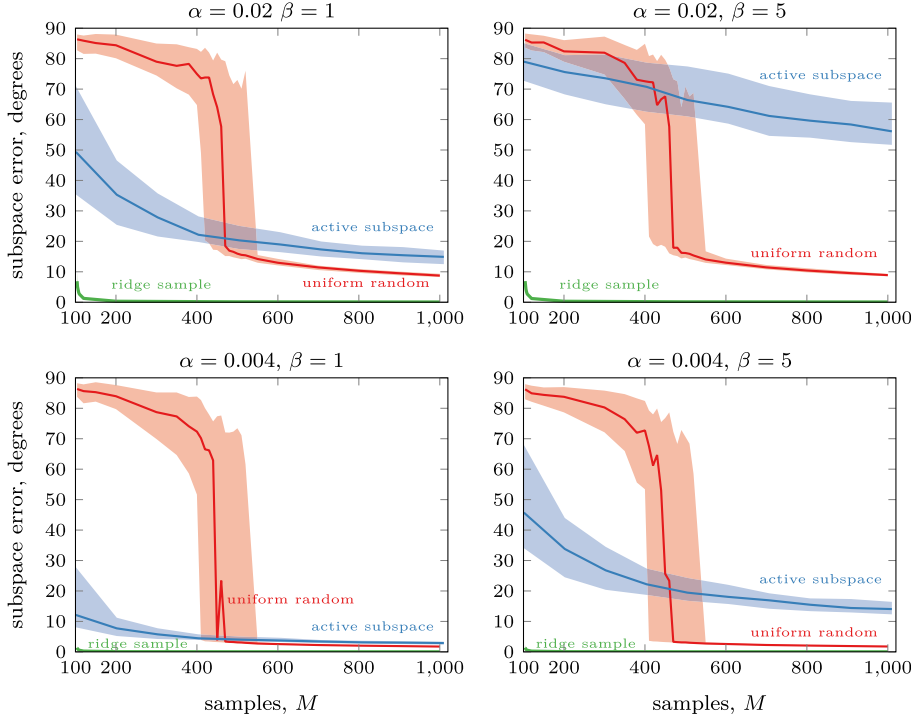


FIG. 4. The subspace angle between the active subspace estimate \mathbf{U} and its true value $\hat{\mathbf{U}} = \mathbf{1}/10$ using M samples with various methods for multiple random samples. In this example we use f given in (53) on $\mathcal{D} = [-1, 1]^{100}$. For each method, the median error is shown by a thick line and the 50% interval by the shaded region. The active subspace estimate was computed using a one-sided finite difference from points randomly selected in the domain. The other two lines show the polynomial ridge approximation of degree 2 constructed from different sampling schemes. The uniform random scheme chose points on \mathcal{D} with uniform weight, whereas the ridge sample scheme chose points evenly spaced along the true active subspace and randomly with uniform probability in the orthogonal complement.

In contrast to the Monte Carlo estimate of the active subspace, the polynomial ridge approximation displays an interesting plateau, followed by apparent $\mathcal{O}(M^{-1/2})$ convergence. During the plateau, the large angle with the true subspace is not (primarily) an artifact of a local minima; even initializing with the true active subspace yields a subspace with large angles to the true active subspace. This suggests that this plateau is likely due, loosely, to the information contained in the samples. The later $\mathcal{O}(M^{-1/2})$ convergence we conjecture is due to the interpretation of the discrete least-squares problem as a Monte Carlo approximation of the *continuous* least-squares problem:

$$(57) \quad \underset{\substack{g \in \mathbb{P}^p(\mathbb{R}^n) \\ \text{Range } \mathbf{U} \in \mathbb{G}(n, \mathbb{R}^m)}}{\text{minimize}} \quad \int_{\mathcal{D}} |f(\mathbf{x}) - g(\mathbf{U}^\top \mathbf{x})|^2 d\mu(\mathbf{x}).$$

The second ridge approximation built from samples uniformly along the true active subspace suggests that with a better sampling scheme the ridge approximation can converge more rapidly.

5.5. NACA airfoil. As a first demonstration of our algorithm on an application problem, we consider an 18-parameter model of a NACA0012 airfoil with two

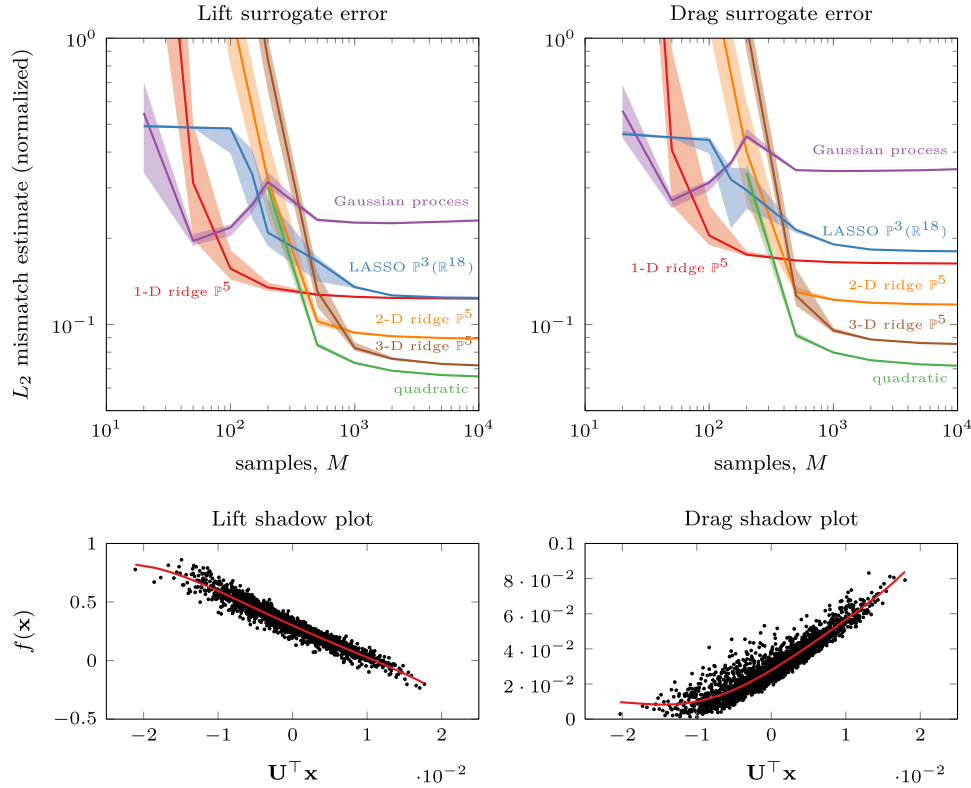


FIG. 5. The top two plots show the estimated L_2 error using Monte Carlo integration for several different surrogate models applied to the lift and drag of a NACA0012 airfoil as described in subsection 5.5. The solid line indicates the median mismatch and the shaded region encloses the 25th to 75th percentile of mismatch from 100 fits using randomly selected samples. The bottom two plots show the shadow of these high-dimensional points onto the 1D ridge subspace fit with a 5th degree polynomial.

quantities of interest: the nondimensionalized lift and drag coefficients. This model from [8, sect. 5.3.1] depends on 18 Hicks–Henne parameters that modify the airfoil geometry, and both lift and drag are computed using the Stanford University Unstructured (SU2) computational fluid dynamics code [16]. Figure 5 shows the estimated L_2 mismatch normalized by the L_2 norm of f ; both integrals are estimated using Monte Carlo, (cf. (57)). In addition to polynomial ridge approximations of various subspace dimensions, this example compares two other surrogate models: a Gaussian process model using `sklearn`’s `GaussianProcessRegressor` and a global cubic model with a sparsity encouraging ℓ_1 penalty using `LassoCV` which includes cross-validation to pick the regularization parameter [41]. As these results show, a 1-D polynomial ridge approximation does well with limited samples, providing a better surrogate than either a Gaussian process or a sparse approximation. However, increasing the subspace dimension does not significantly improve the fit, and the quadratic polynomial using all the input coordinates provides the best surrogate with a large number of samples.

5.6. Elliptic PDE. As a final demonstration of our algorithm, we consider a 2D elliptic PDE from [10] with $m = 100$ input parameters. These parameters $\mathbf{x} \in \mathbb{R}^{100}$ characterize the coefficient $a = a(\mathbf{s}, \mathbf{x})$ in the differential equation

$$(58) \quad -\nabla_{\mathbf{s}} \cdot (a \nabla_{\mathbf{s}} u) = 1, \quad \mathbf{s} \in [0, 1]^2,$$

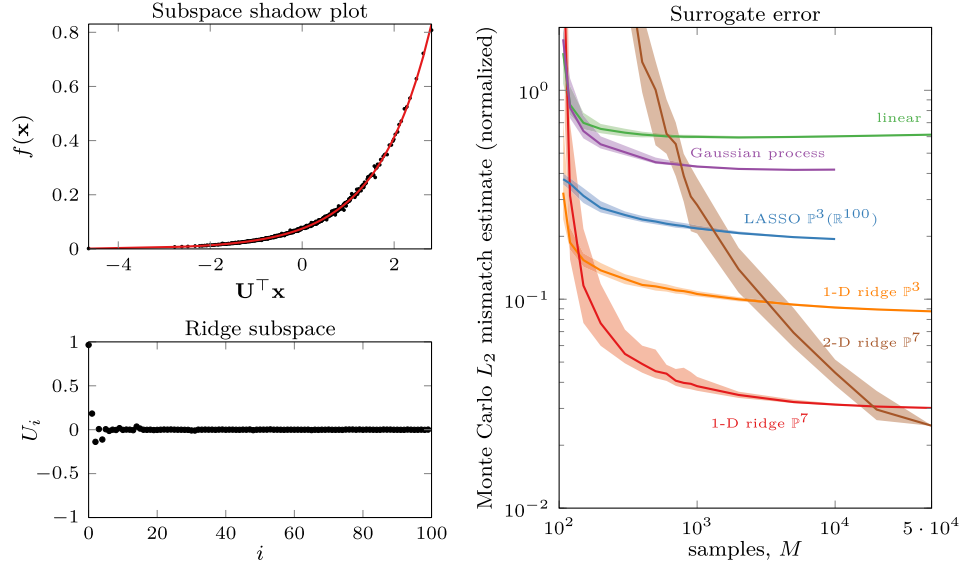


FIG. 6. The top left plot shows the 1-D ridge function approximation (solid line) along with the projected points $y_i = \mathbf{U}^T \mathbf{x}_i \in \mathbb{R}$ (dots). The bottom left plot shows the entries in the matrix $\mathbf{U} \in \mathbb{R}^{100 \times 1}$. The right plot shows the estimated L_2 error using Monte Carlo integration for several different surrogate models applied to the elliptic PDE model described in subsection 5.6 where the solid line indicates the median mismatch and the shaded region encloses the 25th to 75th percentile mismatch from 100 fits with different random samples.

where the inputs x_j are the coefficients of a truncated Karhunen–Loève expansion of $\log(a)$ with a correlation function,

$$(59) \quad \text{Corr}(\mathbf{s}_1, \mathbf{s}_2) = \exp\left(\frac{-\|\mathbf{s}_1 - \mathbf{s}_2\|_1}{2\ell}\right),$$

where ℓ is a correlation length parameter. The boundary conditions are homogeneous Neumann conditions on the right boundary and homogeneous Dirichlet conditions on the other boundaries; the quantity of interest is the spatial average of the solution u on the right boundary. Figure 6 shows the shadow plot for this quantity of interest, which exhibits a strong, 1-D ridge structure where the coordinates of the active subspace spanned by \mathbf{U} are approximately sparse. As this figure illustrates, by exploiting the approximate ridge structure present in this quantity of interest, we can form an accurate surrogate using relatively few samples.

6. Summary and discussion. Here we have derived a structure exploiting algorithm to efficiently solve the data-driven polynomial ridge approximation problem (2). The key feature of this algorithm is exploiting the separable structure which allows us to optimize over the subspace alone by implicitly solving for the polynomial approximation using variable projection. This allows our Gauss–Newton based optimization over the Grassmann manifold to display superior convergence properties compared to the alternating method of Constantine, Eftekhari, Hokanson, and Ward [11] and allows us to reduce computational costs by exploiting the orthogonality properties of the Jacobian revealed in Theorem 1.

This combination of variable projection and manifold optimization can likely be used to accelerate optimization of other surrogate model classes. For example, we

could replace the polynomial model of total degree p with a tensor product spline model which also yields a linear least-squares problem to recover g . However, as this spline model is not rotationally invariant, the optimization would have to be with respect to the Stiefel manifold. Or, we could build g using a Gaussian process model on the projected points $\{\mathbf{y}_i = \mathbf{U}^\top \mathbf{x}_i\}_{i=1}^M$ and apply a regularization technique to ensure we do not obtain an interpolant for any choice of \mathbf{U} . However, in this case we could no longer use variable projection as the analog of $\mathbf{V}(\mathbf{U})$ would be a square, full rank matrix. Similar techniques could also be extended to the projection pursuit model (3), but then optimization would be over the product of n one-dimensional Grassmann manifolds.

Beyond the scope of this work are two more fundamental questions about constructing surrogates, and in particular, ridge approximations: (i) how do we select the “best” choice of polynomial degree p and subspace dimension n and (ii) how do we choose our samples $\{\mathbf{x}_i\}_{i=1}^M \subset \mathcal{D}$ to maximize the accuracy of our ridge approximation? When $f(\mathbf{x})$ contains random noise with a known distribution, as is the case in statistical regression, there are existing approaches to answer both these questions. The hyperparameters n and p can be chosen using a number of techniques such as the Akaike information criterion (AIC) [3] or cross-validation [42]. With this assumption of noise, we can also invoke traditional experimental design techniques to choose points $\{\mathbf{x}_i\}_{i=1}^M$ that minimize the variance of our parameter estimates \mathbf{U} and g [46, Chap. 6]. However, the quantities of interest that often appear in uncertainty quantification do not have statistical noise but instead often display structured artifacts due to mesh discretizations and solver tolerances [36]. But, as Figure 4 suggests, if a good sequential point selection heuristic can be determined, this would enable the construction of a better ridge approximation with fewer function evaluations. This remains an active area of research.

Acknowledgment. The authors would like to thank Akil Narayan for his suggestion to consider orthogonal polynomial bases, such as the Legendre basis, and Henry Yuchi for catching a typo in an early draft.

REFERENCES

- [1] P.-A. ABSIL, R. MAHONY, AND R. SEPULCHRE, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, NJ, 2008.
- [2] K. P. ADRAGNI AND R. D. COOK, *Sufficient dimension reduction and prediction in regression*, Philos. Trans. Roy. Soc. A, 367 (2009), pp. 4385–4405, <https://doi.org/10.1098/rsta.2009.0110>.
- [3] H. AKAIKE, *A new look at the statistical model identification*, IEEE Trans. Automat. Control, 19 (1974), pp. 716–723, <https://doi.org/10.1109/TAC.1974.1100705>.
- [4] M. BAUERHEIM, A. NDIAYE, P. CONSTANTINE, S. MOREAU, AND F. NICLOUD, *Symmetry breaking of azimuthal thermoacoustic modes: The UQ perspective*, J. Fluid Mech., 789 (2016), pp. 534–566, <https://doi.org/10.1017/jfm.2015.730>.
- [5] Å. BJÖRCK, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
- [6] S. L. CAMPBELL AND C. D. MEYER, JR., *Generalized Inverses of Linear Transformations*, Pitman, Boston, 1979.
- [7] 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, <https://doi.org/10.1007/s00365-011-9147-6>.
- [8] P. G. CONSTANTINE, *Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies*, SIAM, Philadelphia, 2015.
- [9] P. G. CONSTANTINE AND A. DOOSTAN, *Time-dependent global sensitivity analysis with active subspaces for a lithium ion battery model*, Stat. Anal. Data Min., 10 (2017), pp. 243–262, <https://doi.org/10.1002/sam.11347>.

- [10] P. G. CONSTANTINE, E. DOW, AND Q. WANG, *Active subspace methods in theory and practice: Applications to kriging surfaces*, SIAM J. Sci. Comput., 36 (2014), pp. A1500–A1524, <https://doi.org/10.1137/130916138>.
- [11] P. G. CONSTANTINE, A. EFTEKHARI, J. HOKANSON, AND R. A. WARD, *A near-stationary subspace for ridge approximation*, Comput. Methods Appl. Mech. Engrg., 326 (2017), pp. 402–421, <https://doi.org/10.1016/j.cma.2017.07.038>.
- [12] P. G. CONSTANTINE, M. EMORY, J. LARSSON, AND G. IACCARINO, *Exploiting active subspaces to quantify uncertainty in the numerical simulation of the HyShot II scramjet*, J. Comput. Phys., 302 (2015), pp. 1–20, <https://doi.org/10.1016/j.jcp.2015.09.001>.
- [13] P. G. CONSTANTINE, B. ZAHARATOS, AND M. CAMPANELLI, *Discovering an active subspace in a single-diode solar cell model*, Stat. Anal. Data Min., 8 (2015), pp. 264–273, <https://doi.org/10.1002/sam.11281>.
- [14] R. D. COOK, *Regression Graphics: Ideas for Studying Regressions Through Graphics*, John Wiley and Sons, Hoboken, 1998, <https://doi.org/10.1002/9780470316931>.
- [15] P. DIACONIS AND M. SHAHSHAHANI, *On nonlinear functions of linear combinations*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 175–191, <https://doi.org/10.1137/0905013>.
- [16] T. D. ECONOMON, F. PALACIOS, S. R. COPELAND, T. W. LUKACZYK, AND J. J. ALONSO, *SU2: An open-source suite for multiphysics simulation and design*, AIAA J., 54 (2016), pp. 828–846, <https://doi.org/10.2514/1.J053813>.
- [17] A. EDELMAN, T. A. ARIAS, AND S. T. SMITH, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353, <https://doi.org/10.1137/S0895479895290954>.
- [18] 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, <https://doi.org/10.1007/s10208-012-9115-y>.
- [19] J. H. FRIEDMAN AND W. STUETZLE, *Projection pursuit regression*, J. Amer. Statist. Assoc., 76 (1981), pp. 817–823, <http://www.jstor.org/stable/2287576>.
- [20] K. FUKUMIZU AND C. LENG, *Gradient-based kernel dimension reduction for regression*, J. Amer. Statist. Assoc., 109 (2014), pp. 359–370, <https://doi.org/10.1080/01621459.2013.838167>.
- [21] J. M. GILBERT, J. L. JEFFERSON, P. G. CONSTANTINE, AND R. M. MAXWELL, *Global spatial sensitivity of runoff to subsurface permeability using the active subspace method*, Adv. Water Res., 92 (2016), pp. 30–42, <https://doi.org/10.1016/j.advwatres.2016.03.020>.
- [22] A. GLAWS, P. G. CONSTANTINE, J. N. SHADID, AND T. M. WILDEY, *Dimension reduction in magnetohydrodynamics power generation models: Dimensional analysis and active subspaces*, Stat. Anal. Data Min., 10 (2017), pp. 312–325, <https://doi.org/10.1002/sam.11355>.
- [23] G. H. GOLUB AND V. PEREYRA, *The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate*, SIAM J. Numer. Anal., 10 (1973), pp. 413–432.
- [24] J. HAMPTON AND A. DOOSTAN, *Coherence motivated sampling and convergence analysis of least squares polynomial chaos regression*, Comput. Methods Appl. Mech. Engrg., 290 (2015), pp. 73–97, <https://doi.org/10.1016/j.cma.2015.02.006>.
- [25] T. HASTIE, R. J. TIBSHIRANI, AND J. FRIEDMAN, *Elements of Statistical Learning*, 2nd ed., Springer Science+Business Media, New York, 2009, <https://doi.org/10.1007/978-0-387-84858-7>.
- [26] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 2002.
- [27] X. HU, G. T. PARKS, X. CHEN, AND P. SESHADRI, *Discovering a one-dimensional active subspace to quantify multidisciplinary uncertainty in satellite system design*, Adv. Space Res., 57 (2016), pp. 1268–1279, <https://doi.org/10.1016/j.asr.2015.11.001>.
- [28] P. J. HUBER, *Projection pursuit*, Ann. Statist., 13 (1985), pp. 435–475, <http://www.jstor.org/stable/2241175>.
- [29] J. L. JEFFERSON, J. M. GILBERT, P. G. CONSTANTINE, AND R. M. MAXWELL, *Active subspaces for sensitivity analysis and dimension reduction of an integrated hydrologic model*, Comput. Geosci., 83 (2015), pp. 127–138, <https://doi.org/10.1016/j.cageo.2015.11.002>.
- [30] J. L. JEFFERSON, R. M. MAXWELL, AND P. G. CONSTANTINE, *Exploring the sensitivity of photosynthesis and stomatal resistance parameters in a land surface model*, J. Hydrometeorol., 18 (2017), pp. 897–915, <https://doi.org/10.1175/JHM-D-16-0053.1>.
- [31] D. JONES, *A taxonomy of global optimization methods based on response surfaces*, J. Global Optim., 21 (2001), pp. 345–383, <https://doi.org/10.1023/A:1012771025575>.
- [32] X. LIU AND S. GUILLAS, *Dimension reduction for Gaussian process emulation: An application to the influence of bathymetry on tsunami heights*, SIAM/ASA J. Uncertain. Quantif., 5 (2017), pp. 787–812, <https://doi.org/10.1137/16M1090648>.
- [33] T. LOUDON AND S. PANKAVICH, *Mathematical analysis and dynamic active subspaces for a long term model of HIV*, Math. Biosci. Eng., 14 (2017), pp. 709–733, <https://doi.org/10.3934/mbe.2017040>.

- [34] T. W. LUKACZYK, *Surrogate Modeling and Active Subspaces for Efficient Optimization of Supersonic Aircraft*, Ph.D. thesis, Stanford University, 2015, <https://purl.stanford.edu/xx611nd3190>.
- [35] T. W. LUKACZYK, P. CONSTANTINE, F. PALACIOS, AND J. J. ALONSO, *Active subspaces for shape optimization*, in 10th AIAA Multidisciplinary Design Optimization Conference, National Harbor, MD, AIAA SciTech Forum, 2014, <https://doi.org/10.2514/6.2014-1171>.
- [36] J. J. MORÉ AND S. M. WILD, *Estimating computational noise*, SIAM J. Sci. Comput., 33 (2011), pp. 1292–1314.
- [37] R. H. MYERS AND D. C. MONTGOMERY, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, John Wiley and Sons, New York, 1995.
- [38] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer, New York, 2006.
- [39] C. OTHMER, T. W. LUKACZYK, P. CONSTANTINE, AND J. J. ALONSO, *On active subspaces in car aerodynamics*, in 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Washington, DC, AIAA AVIATION Forum, 2016, <https://doi.org/10.2514/6.2016-4294>.
- [40] V. Y. PAN, *How bad are Vandermonde matrices?*, SIAM J. Matrix Anal. Appl., 37 (2016), pp. 676–694.
- [41] F. PEDREGOSA, G. VAROQUAUX, A. GRAMFORT, V. MICHEL, B. THIRION, O. GRISEL, M. BLONDEL, P. PRETTENHOFER, R. WEISS, V. DUBOURG, J. VANDERPLAS, A. PASSOS, D. COURNAPEAU, M. BRUCHER, M. PERROT, AND E. DUCHESNAY, *Scikit-learn: Machine learning in Python*, J. Mach. Learn. Res., 12 (2011), pp. 2825–2830.
- [42] R. R. PICARD AND R. D. COOK, *Cross-validation of regression models*, J. Amer. Statist. Assoc., 79 (1984), pp. 575–583, <https://doi.org/10.1080/01621459.1984.10478083>.
- [43] A. PINKUS, *Ridge Functions*, Cambridge University Press, Cambridge, 2015, <https://doi.org/10.1017/CBO9781316408124>.
- [44] C. E. RASMUSSEN AND C. K. I. WILLIAMS, *Gaussian Processes for Machine Learning*, The MIT Press, Boston, 2006, <http://www.gaussianprocess.org/gpml/>.
- [45] A. RUHE AND P. A. WEDIN, *Algorithms for separable nonlinear least squares problems*, SIAM Rev., 22 (1980), pp. 318–337, <https://doi.org/10.1137/1022057>.
- [46] T. J. SANTNER, B. J. WILLIAMS, AND W. I. NOTZ, *The Design and Analysis of Computer Experiments*, Springer, New York, 2003, <https://doi.org/10.1007/978-1-4757-3799-8>.
- [47] P. SESHADRI, S. SHAHPAR, P. CONSTANTINE, G. PARKS, AND M. ADAMS, *Turbomachinery active subspace performance maps*, in ASME Turbo Expo 2017: Turbomachinery Technical Conference and Exposition, no. 50787, Charlotte, NC, International Gas Turbine Institute, 2017, p. V02AT39A034, <https://doi.org/10.1115/GT2017-64528>.
- [48] S. SHAN AND G. G. WANG, *Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions*, Struct. Multidiscip. Optim., 41 (2010), pp. 291–241, <https://doi.org/10.1007/s00158-009-0420-2>.
- [49] R. C. SMITH, *Uncertainty Quantification: Theory, Implementation, and Applications*, SIAM, Philadelphia, 2013.
- [50] T. J. SULLIVAN, *Introduction to Uncertainty Quantification*, Springer, New York, 2015, <https://doi.org/10.1007/978-3-319-23395-6>.
- [51] J. TOWNSEND, N. KOEP, AND S. WEICHWALD, *Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation*, J. Mach. Learn. Res., 17 (2016), pp. 1–5, <http://jmlr.org/papers/v17/16-177.html>.
- [52] R. TRIPATHY, I. BILIONIS, AND M. GONZALEZ, *Gaussian processes with built-in dimensionality reduction: Applications to high-dimensional uncertainty propagation*, J. Comput. Phys., 321 (2016), pp. 191–223, <https://doi.org/10.1016/j.jcp.2016.05.039>.
- [53] H. TYAGI AND V. CEVHER, *Learning non-parametric basis independent models from point queries via low-rank methods*, Appl. Comput. Harmon. Anal., 37 (2014), pp. 389–412, <https://doi.org/10.1016/j.acha.2014.01.002>.
- [54] F. VIVARELLI AND C. K. I. WILLIAMS, *Discovering hidden features with gaussian processes regression*, in Advances in Neural Information Processing Systems 11, M. J. Kearns, S. A.olla, and D. A. Cohn, eds., MIT Press, Cambridge, MA, 1999, pp. 613–619, <http://papers.nips.cc/paper/1579-discovering-hidden-features-with-gaussian-processes-regression.pdf>.
- [55] G. G. WANG AND S. SHAN, *Review of metamodeling techniques in support of engineering design optimization*, J. Mech. Design, 129 (2006), pp. 370–380, <https://doi.org/10.1115/1.2429697>.
- [56] S. WEISBERG, *Applied Linear Regression*, 3rd ed., John Wiley and Sons, Hoboken, 2005, <https://doi.org/https://doi.org/10.1002/0471704091>.

- [57] Y.-C. WONG, *Differential geometry of Grassmann manifolds*, Proc. Nat. Acad. Sci. USA, 57 (1967), pp. 589–594, <https://doi.org/10.1073/pnas.57.3.589>.
- [58] Y. XIA, *A multiple-index model and dimension reduction*, J. Amer. Statist. Assoc., 103 (2008), pp. 1631–1640, <http://www.jstor.org/stable/27640210>.
- [59] Y. XIA, H. TONG, W. K. LI, AND L.-X. ZHU, *An adaptive estimation of dimension reduction space*, J. R. Stat. Soc. Ser. B. Stat. Methodol., 64 (2002), pp. 363–410, <https://doi.org/10.1111/1467-9868.03411>.