SIAM J. SCI. COMPUT.
Vol. 34, No. 3, pp. A1729–A1752

STOCHASTIC COLLOCATION METHODS ON UNSTRUCTURED GRIDS IN HIGH DIMENSIONS VIA INTERPOLATION*

AKIL NARAYAN† AND DONGBIN XIU†

Abstract. In this paper we propose a method for conducting stochastic collocation on arbitrary sets of nodes. To accomplish this, we present the framework of *least orthogonal interpolation*, which allows one to construct interpolation polynomials based on arbitrarily located grids in arbitrary dimensions. These interpolation polynomials are constructed as a subspace of the family of orthogonal polynomials corresponding to the probability distribution function on stochastic space. This feature enables one to conduct stochastic collocation simulations in practical problems where one cannot adopt some popular node selections such as sparse grids or cubature nodes. We present in detail both the mathematical properties of the least orthogonal interpolation and its practical implementation algorithm. Numerical benchmark problems are also presented to demonstrate the efficacy of the method.

Key words. stochastic collocation, polynomial chaos, interpolation, orthogonal polynomials

AMS subject classifications. 65D05, 41A63, 41A10

DOI. 10.1137/110854059

1. Introduction.

Stochastic computation has received an enormous amount of attention recently, due to the pressing need to conduct uncertainty quantification (UQ) in scientific computing. Many numerical techniques have been developed to make UQ computation feasible for large and complex systems. Among them, a widely used method is based on generalized polynomial chaos (gPC) [22], an extension of the classical polynomial chaos (PC) [15]. In the gPC methodology, one seeks to construct an orthogonal polynomial approximation for the solution of the stochastic system over the entire random space, defined by the random inputs. When the solution dependence on the random inputs is smooth, such an expansion exhibits fast convergence, is highly accurate, and results in more efficient algorithms than other traditional methods such as perturbation methods and moment equation methods.

Numerical algorithms for obtaining gPC expansions are usually divided into either Galerkin or collocation approaches. The former is "intrusive" and requires one to modify existing deterministic codes; the latter is "nonintrusive," imposing a minimal amount of coding effort, as one can repetitively use existing deterministic codes. In cases when dependence on the random inputs is smooth, then one may obtain exponential accuracy for both methods.

Owing to its nonintrusive nature of implementation, stochastic collocation methods have become popular in many practical simulations. In stochastic collocation, one first places a set of nodes in random space, and then conducts individual deterministic simulations at each of the nodes to obtain a solution ensemble. Finally a polynomial approximation of the solution based on the ensemble is constructed. The

*Submitted to the journal's Methods and Algorithms for Scientific Computing section November 7, 2011; accepted for publication (in revised form) March 23, 2012; published electronically June 26, 2012. This work was partially supported by AFOSR award FA9550-11-1-0310, DOE award DE-FC52-08NA28617, and NSF award DMS-0645035.
http://www.siam.org/journals/sisc/34-3/85405.html
†Department of Mathematics, Purdue University, West Lafayette, IN 47907 (akil@purdue.edu, dxiu@purdue.edu).

development of stochastic collocation algorithms became very active after the introduction of high-order algorithms using sparse grids [21] and produced many different techniques; cf. [1, 2, 3, 4, 6, 10, 12, 13, 16, 17, 19], to name a few. The construction of the polynomial approximation usually falls into one of the following categories: interpolation, regression, and discrete projection. While the latter two do not require the polynomial reconstruction to pass through the solution data, the interpolation approach requires that the polynomial approximation match the solution precisely at the nodes. For more detailed exposition of these methods, including their properties, strengths, and weaknesses, see [20].

In this paper we focus on the *interpolation approach* for stochastic collocation. The difficulty is the dimensionality of the random space, which is determined by the number of (usually independent) random variables (d) used to characterize the random inputs. For many systems, this number can be large, $d \gg 1$. In high-dimensional spaces, polynomial interpolation becomes a difficult problem, with many issues remaining open; see [14] for a summary of approaches. One issue that is still under active research is how to construct an interpolation basis on an arbitrarily given set of nodes to interpolate any data. To this end, a common approach is to employ well-studied interpolation schemes in one dimension, e.g., polynomial interpolation on the real line, and then progressively fill up the entire random space in a dimension-by-dimension manner. By doing so, the grids are usually structured and more susceptible to the curse of dimensionality. Typical examples of this kind are the tensor grids and sparse grids. Another common approach is to directly invert the Vandermonde-like interpolation matrix. This requires one to choose the polynomial interpolation basis first. Hence the number of nodes must be equal to (or larger than) the dimensionality of the interpolation space and cannot be arbitrary. Also for any given set of nodes, they may lie on some complicated hypersurface and render the Vandermonde-like matrix singular.

In this paper we study the stochastic collocation based on interpolation on *unstructured grids*. The use of unstructured grids is largely motivated by a practical concern: in many simulations of complex systems, one does not have complete control over the choice of simulation samples—either the locations of the samples are subject to certain simulations/experimental constraints, or the total number of samples is dictated by a computational or an experimental resource but not by an accuracy requirement. Furthermore, an effective polynomial approximation based on unstructured grids could potentially allow one to design highly efficient adaptive methods. We remark that one can always resort to regression methods to construct polynomial approximations on unstructured grids. These can be overdetermined least-squares type or underdetermined ℓ_1-minimization type [9]. Here we focus on the interpolation approach, which requires the polynomial approximation to match the simulation samples precisely at the nodes. This represents a very different approximation principle from that of regression types. We do not attempt to judge the relative merits of interpolation versus regression in this paper.

The method proposed here is termed *least orthogonal interpolation*. It is closely related to the least interpolation method proposed by de Boor and Ron [7, 8], where a framework of polynomial interpolation using arbitrary grids was proposed. Our proposed method of least orthogonal interpolation is a notable generalization, where the interpolation is tailored to a particular probability distribution and utilizes the corresponding orthogonal polynomials. This is particularly desirable for stochastic computations. In fact, it can be shown that the original least interpolation of [7, 8] is a special case of the proposed least orthogonal interpolation, in the sense that

it coincides with the least orthogonal interpolant when the probability density is a Gaussian and Hermite polynomials are employed.

The least orthogonal interpolant possesses several desirable properties. Let X be a finite collection of unique points in d-dimensional Euclidean space. Given some probability density function ω over this space, we will introduce $\Pi_{X,\omega}$, the *least orthogonal space*. It is a space of polynomials for which interpolation for data on X can be accomplished in a bijective manner. Some properties of this space and the resulting interpolant are

- connection to probability distribution: given an input probability measure ω, one can generate X using statistical sampling based on ω and construct $\Pi_{X,\omega}$ using corresponding orthogonal polynomials;
- monotonicity: if $X_1 \subset X_2$, then $\Pi_{X_1,\omega} \subset \Pi_{X_2,\omega}$;
- formulaic construction: the space $\Pi_{X,\omega}$ can be constructed, and we show that for certain cases the construction using orthogonal polynomials is better conditioned than the original least interpolant;
- continuity for nondegenerate cases: for certain collections of nodes X, the resulting space $\Pi_{X,\omega}$ varies continuously. More precisely, $\Pi_{X,\omega}$ varies continuously with X when, for some s, $\Pi_{X,\omega}$ is a superset of all degree-s polynomials, but a subset of all degree-$(s+1)$ polynomials;
- generalization of the original least interpolant: we show that if ω corresponds to the d-dimensional standard multivariate normal distribution, the least orthogonal interpolant coincides with the original least interpolant [7, 8].

Finally we remark that interpolation in high dimensions has been an active and challenging field. The ability to construct an interpolation polynomial does not always mean that the reconstruction is faithful away from the interpolation points. The accuracy of an interpolation polynomial largely depends on, among other things, the distribution of the grids. Furthermore, to this end, there does not seem to exist rigorous methods on determining the "good" set of grids in high dimensions, though heuristic choices do exist in practical applications. In general, one might consider the construction of an approximant and the choice of interpolation nodes as coupled issues. However, in this work we focus on constructing multidimensional interpolations for an arbitrarily *given* set of grids, and we leave the choice of optimal grids as a separate and independent issue to be studied elsewhere.

In section 2 we discuss the general stochastic computing setup for gPC collocation methods applied to differential equations, and we discuss multivariate orthogonal polynomial families, which form the basis for our methods. Section 3 presents the theoretical framework for the least orthogonal polynomial space, and section 4 follows up with the algorithmic method of construction. Finally, section 5 presents some example interpolation problems to illustrate the use of the method.

2. Generalized polynomial chaos and orthogonal polynomials. This section defines the stochastic problems we study and frames them in the context of the collocation approach for gPC. We also introduce the notation for orthogonal polynomials that we use in the remainder of this article.

2.1. Generalized polynomial chaos and stochastic collocation. We will work on a complete probability space $(\Xi, \mathcal{F}, P)$. In UQ computations, one is often concerned with propagating uncertain inputs through a given system, e.g., a partial differential equation (PDE). Let $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, $d \geq 1$, be a set of random variables modeling the random inputs for a PDE, and let $y \in \mathbb{R}^k$ for $k = 1, 2, 3$ represent the spatial variable. It is common in practical simulations to assume that

the x_i's are independent. We consider PDEs of the form

$$\begin{cases} u_t(y,t;x) = \mathcal{L}(u), & D \times (0,T] \times \Omega, \\ \mathcal{B}(u) = 0, & \partial D \times [0,T] \times \Omega, \\ u = u_0, & D \times \{t=0\} \times \Omega, \end{cases} \tag{1}$$

where $\Omega \subseteq \mathbb{R}^d$, $d \geq 1$, is the range of the random variable x; $D \in \mathbb{R}^k$, $k = 1, 2$, or 3, is the physical domain; and $T > 0$. Here $\mathcal{L}$ stands for a (nonlinear) differential operator and $\mathcal{B}$ a boundary condition operator. We equip x with a distribution function $\nu(a) = P(x \leq a) = P(x_1 \leq a_1, \ldots, x_d \leq a_d)$, where $a \in \mathbb{R}^d$. We assume that the induced probability density $\omega(x) = \frac{\partial^n \nu}{\partial x_1 \cdots \partial x_n}$ is a Lebesgue-measurable function. If the x_i are all independent, then $\omega(x) = \prod_{i=1}^d \frac{\partial \nu_i}{\partial x_i}$, where $\nu_i(x_i)$ is the marginal distribution function for $x_i \in \mathbb{R}$.

Following the standard exposition, we will abandon the notation of y and t hereafter, with the understanding that all discussions are for fixed spatial and temporal locations. In gPC methods, one seeks to construct a degree-K polynomial $u_K(x)$ to approximate the solution $u(x)$; it is computationally convenient to construct u_K by specifying its coordinates in a basis of polynomials orthogonal under the weight ω: this is the gPC basis. It is well known that there exist different gPC orthogonal bases corresponding to different probability distributions. For example, Hermite polynomials correspond to the Gaussian distribution, Legendre polynomials pair with the uniform distribution, etc. The advantages of using the orthogonal polynomial family that corresponds to the probability distribution to construct u_K are manifold; for details, see [22].

The construction of a polynomial approximation relies on the original system (1), and in stochastic collocation methods, one uses samples of the solutions of (1). This is accomplished by first choosing a set of grid points $\{x^{(j)}\}_{j=1}^N \subset \Omega$, where $N \geq 1$ is the number of nodes. And then for each $j = 1, \ldots, N$, one solves a *deterministic problem* of (1) at the node $x^{(j)}$ to obtain solution $u^{(j)}$. Finally, once the pairings $\left(x^{(j)}, u^{(j)}\right)$, $j = 1, \ldots, N$, are obtained, the task is to construct a polynomial approximation $u_K(x)$ such that $u_K(x) \approx u(x)$ in a proper sense. Note that in high dimensions, there does not exist a straightforward relation between the number of nodes N and the achievable polynomial degree K.

In the construction of the approximating function $u_K(x)$, the pairing information can be enforced exactly by requiring $u_K(x^{(j)}) = u^{(j)}$ for all $j = 1, \ldots, N$. This usually leads to a (polynomial) interpolation problem. Alternatively, one can adopt a regression-type approach which does not require precise matching of the function values at each node. In this paper, we seek to construct exact polynomial interpolants. In multiple dimensions this process is not straightforward, and the formulation we present is the major contribution of this paper. We present one example that compares exact interpolation to a regression approach, but in general we do not attempt to compare the methods, or to advocate one approach over the other, as the approaches have distinctly different mathematical and numerical properties and serve different purposes in practice.

2.2. Notation and setup. In d-dimensional Euclidean space, we adopt the standard multi-index notation: $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_d)$ with $|\alpha| = \|\alpha\|_{\ell^1} = \sum_j |\alpha_j|$. If $x \in \mathbb{R}^d$, then $x^\alpha = \prod_{j=1}^d x_j^{\alpha_j}$. Admissible multi-indices lie in the set $\mathbb{N}_0^d = \{\alpha : \alpha_j \in \mathbb{N} \cup \{0\}\}$. All d-variate polynomials p have the monomial expansion $p = \sum_\alpha c_\alpha x^\alpha$ for some coefficients c_α. We say that p has total degree k if $k = \max\{|\alpha| : c_\alpha \neq 0\}$.

We will order the space of multi-indices by the ℓ^1 norm: $\alpha \leq \beta$ if $|\alpha| \leq |\beta|$, and the corresponding definition for $<$. This does not define a total ordering of $\mathbb{N}_0^d$ as we do not specify ordering of the set satisfying $|\alpha| = k \in \mathbb{N}_0$. For most of our discussion, such a total ordering is unnecessary, and any choice in this matter may be made. For example, one ordering of $\mathbb{N}_0^3$ respecting the ℓ^1 norm is total-degree reverse lexicographic ordering:

$$\begin{array}{cccccccc} \alpha & (0,0,0) & (1,0,0) & (0,1,0) & (0,0,1) & (2,0,0) & (1,1,0) & \ldots, \\ |\alpha| & 0 & 1 & 1 & 1 & 2 & 2 & \ldots \end{array}$$

Let Π^d be the space of all d-variate polynomials. Π_k^d is the space of polynomials of degree less than or equal to $k \in \mathbb{N}_0$, and $\Pi_{=k}^d$ and $\Pi_{<k}^d$ are subspaces of Π_k^d corresponding to polynomials of total degree exactly equal to k and strictly less than k, respectively.

2.3. Orthogonal polynomials. We let $\Omega \subseteq \mathbb{R}^d$ be a connected region in d-dimensional space whose interior has nonvanishing Lebesgue measure, along with a measurable function $\omega : \Omega \to \mathbb{R}$ that is positive on the interior of Ω and semipositive on $\partial\Omega$. If $\int_\Omega x^{2\alpha}\omega(x)\mathrm{d}x < \infty$ for all multi-indices α, then a Gram–Schmidt orthogonalization argument shows that there exists a family of polynomial subspaces $\mathcal{V}_k^\omega$ for $k \in \mathbb{N}_0$, defined by

$$\mathcal{V}_k^\omega = \left\{ p \in \Pi_k^d : \langle p, q\rangle_\omega = 0 \; \forall \; q \in \Pi_{<k}^d \right\}. \tag{2}$$

with $\mathcal{V}_{-1}^\omega = \{0\}$ the trivial subspace. With ω specified, $\mathcal{V}_k^\omega$ is unique. The assumptions on ω and Ω imply that the inner product $\langle f, g\rangle_\omega = \int_\Omega f\, g\, \mathrm{d}\nu$ induces a proper norm $\|f\|_\omega^2 = \langle f, f\rangle_\omega$, and therefore $\dim \mathcal{V}_k^\omega = \dim \Pi_{=k}^d$. We will endow $\mathcal{V}_k^\omega$ with a basis $\{\Phi_\alpha\}_{|\alpha|=k}$ of polynomials that are ω-orthonormal:

$$\int_\Omega \Phi_\alpha(x)\Phi_\beta(x)\omega(x)\mathrm{d}x = \langle \Phi_\alpha, \Phi_\beta\rangle_\omega = \delta_{\alpha,\beta}, \tag{3}$$

where $\delta_{\alpha,\beta}$ is the Kronecker delta. Note that any orthogonal transformation of $\{\Phi_\alpha\}_{|\alpha|=k}$ is also an orthonormal basis for $\mathcal{V}_k^\omega$. Under such a (nonpermutation) transformation, these two bases are distinct whenever $d > 1$ and $k > 0$. Thus, the specification of Φ_α is in general not unique. For example, there are different classical definitions for the family $\{\Phi_\alpha\}$ that is orthogonal on the unit sphere in $\mathbb{R}^d$ [23].

Construction of the subspaces $\mathcal{V}_k^\omega$ and their basis representations Φ is not a straightforward undertaking for general ω and Ω. When Ω has a simple shape, various classical families have been derived. For example, if Ω is a hypercube and $\omega(x) = \prod_j \omega_j(x_j)$ for some measurable functions ω_j, then one can construct the family Φ by taking products of univariate orthogonal polynomial families associated with each of the ω_j. (This is the case for independent random variables $\{x_i\}_{i=1}^d$.) Generating univariate families is a well-studied problem, and various constructive methods exist for nonclassical weight functions. In general our discussion does not revolve around construction of the family Φ: we will assume that, given ω and Ω, evaluation of Φ_α is feasible practically.

Let L_ω^2 be the space of d-variate functions whose squares have finite norm $\|\cdot\|_\omega$. L_ω^2 is a Hilbert space with orthogonal elements Φ_α; the issue of whether the collection $\{\Phi_\alpha\}$ spans the space is a more subtle issue and is not guaranteed. For example, the orthogonal polynomial family derived from a univariate lognormal random variable is not complete in the associated L^2 space. Examples of sufficient conditions ensuring

L^2_ω completeness of $\{\Phi_\alpha\}$ include boundedness of Ω, or $\mathbb{E}\exp(a\|x\|_2) < \infty$ for some $a > 0$. For more intricate conditions and discussions, we refer the reader to [11]. In the remainder of this work we will assume that Φ_α forms an L^2_ω-complete family. This assumption implies that any $f \in L^2_\omega$ has the L^2_ω-convergent expansion

$$f = \sum_{\alpha \in \mathbb{N}_0^d} \hat{f}_\alpha \Phi_\alpha, \tag{4}$$

with $\hat{f}_\alpha = \langle f, \Phi_\alpha \rangle_\omega$. We introduce the L^2_ω-orthogonal projection operator $\mathscr{P}_k$ for $k \in \mathbb{N}_0$ onto Π^d_k:

$$\mathscr{P}_k f = \sum_{|\alpha| \le k} \hat{f}_\alpha \Phi_\alpha \in \Pi^d_k. \tag{5}$$

Although the operators $\mathscr{P}_k$ depend on ω, we suppress such notation. Various relations connect the projection operators $\mathscr{P}_k$, the orthogonal subspaces $\mathcal{V}^\omega_k$, and the standard vector polynomial spaces Π^d_k (cf. [23]):

$$\dim \Pi^d_k = \begin{pmatrix} d+k \\ k \end{pmatrix}, \qquad \dim \Pi^d_{=k} = \dim \mathcal{V}^\omega_k = \begin{pmatrix} d+k-1 \\ k \end{pmatrix},$$

$$\Pi^d_k = \bigoplus_{0 \le l \le k} \mathcal{V}^\omega_l, \qquad \mathscr{P}_k \Pi^d_l = \Pi^d_{\min\{k,l\}},$$

$$\ker{}_{\Pi^d_k} \mathscr{P}_{k-1} = \mathcal{V}^\omega_k, \qquad \mathscr{P}_k \mathcal{V}^\omega_l = \begin{cases} \mathcal{V}^\omega_l & \text{if} \quad l \le k, \\ \{0\} & \text{if} \quad l > k, \end{cases}$$

where we define $\mathscr{P}_{-1} f = 0$ for completeness.

Given a point $x^{(0)} \in \Omega$, we formally define the generalized function

$$\delta_{x^{(0)}}(\cdot) = \sum_\alpha \Phi_\alpha(x^{(0)}) \Phi_\alpha(\cdot), \tag{6}$$

and will regularly identify the algebraic functional action of $\delta_{x^{(0)}}$ on polynomials as point-evaluation:[1]

$$\delta^*_{x^{(0)}}(f) \triangleq \langle \delta_{x^{(0)}}, f \rangle_\omega = \sum_\alpha \hat{f}_\alpha \Phi_\alpha(x^{(0)}) = f(x^{(0)}) \qquad \forall\, f \in \Pi^d. \tag{7}$$

We note in particular that $(\mathscr{P}_{k+1} - \mathscr{P}_k)\delta_{x^{(0)}} \in \mathcal{V}^\omega_{k+1}$ is the same polynomial regardless of the orthonormal basis chosen for $\mathcal{V}^\omega_k$. Thus, so long as we restrict our discussion of $\delta_{x^{(0)}}$ to its images under $\mathscr{P}_k$, the choice of Φ_α is irrelevant. For a finite collection of N nodes X, $\delta(X)$ is the subspace of algebraic functionals spanned by δ_x for $x \in X$. A simple but necessary observation is that this space has dimension N and the functionals $\{\delta_{x^{(n)}}\}$ form a basis.

LEMMA 2.1. *Given N unique points $\{x^{(n)}\}_{n=1}^N$ in d dimensions, the space of generalized functions*

$$\delta(X) = \operatorname{span}\{\delta_{x^{(n)}} : n = 1, 2, \ldots, N\} \tag{8}$$

has dimension N, and therefore $\delta_{x^{(n)}}$ is a basis.

[1] One way of formally stating this is, for example, to extend $\langle \cdot, \cdot \rangle_\omega$ to $\Pi^d \times (\Pi^d)^*$, where $(\Pi^d, L^2_\omega, (\Pi^d)^*)$ forms a Gelfand triple.

Proof. We show that the polynomials $\mathscr{P}_{N-1}\delta_{x^{(n)}}$ are linearly independent. Let the matrix $\mathbf{V}$ have entries $(\mathbf{V})_{n,\alpha} = \Phi_\alpha(x^{(n)})$ for all $|\alpha| \leq N-1$, where we assume any ℓ^1 norm respecting ordering for multi-indices α. There is an invertible matrix $\mathbf{C}$ such that $\mathbf{VC}$ has entries $(\mathbf{VC})_{n,\alpha} = \left(x^{(n)}\right)^\alpha$ for $|\alpha| \leq N-1$. The matrix $\mathbf{VC}$ has full rank N; see, e.g., [8]. Since $\mathbf{VC}$ has the same rank as $\mathbf{V}$, the rows of $\mathbf{V}$ are linearly independent. ∎

3. Framework: The least orthogonal interpolant. Our proposed interpolation method can be divided into two components: first we identify a polynomial space that is "amenable" for interpolation, and we subsequently use the provided data to extract an interpolant from this space via straightforward linear algebra. In this section we describe the abstract construction and the resulting theoretical properties for these two components. A constructive algorithm for computing the interpolant is presented in the next section.

We show here that for any distribution of finitely many unique points X on the domain Ω, the polynomial family Φ induced from ω defines a polynomial space of smallest degree that can uniquely interpolate data on X. The abstract construction makes it clear that this space depends nontrivially on ω.

We formally state our problem of interpolation:

(9) Let ω be a weight function positive on Ω that defines a family of orthogonal polynomials, and let X be a given set of N unique interpolation nodes. Our task is to find a polynomial space $\Pi_{X,\omega}$ such that the map $p : \mathbb{R}^N \to \Pi_{X,\omega}$ between the space of interpolation data on $\delta(X)$ and the space of polynomials $\Pi_{X,\omega}$ is isomorphic.

We reiterate that this problem already has one solution provided by the framework of [7], where multivariate monomials are used, without the consideration of ω. Our goal is to extend that framework for applicability in stochastic computing by using orthogonal polynomials and, perhaps more importantly, by explicitly incorporating different (probability) densities ω. We will show that the earlier framework of [7] becomes a special case of the new framework with ω being a Gaussian measure.

3.1. The least orthogonal space. In the original least interpolation formulation [7, 8], one utilizes the smallest-degree nonvanishing polynomial term of an analytic function's Taylor expansion about the origin. In our least orthogonal interpolation framework, we consider the smallest-degree projection of an L^2_ω function: for any $f \in L^2_\omega$, define

$$f_{\downarrow,\omega} \triangleq \mathscr{P}_k f, \qquad k = \min\{j : \mathscr{P}_j f \neq 0\}. \tag{10}$$

Then $f_{\downarrow,\omega} \in \mathcal{V}^\omega_k$ since $\mathscr{P}_l f = 0$ for all $l < k$. Similar to the notation in [7, 8], $f_{\downarrow,\omega}$ is read "f least ω". $f_{\downarrow,\omega}$ measures the smallest-degree orthogonal contribution to an expansion of f. This definition depends on the choice of ω. For example, in one dimension on $\Omega = (-1, 1)$, let $f = \sin(\pi x)$. If $\omega = 1/2$ (a uniform distribution), then $f_{\downarrow,\omega} = (3/\pi)x$ with $k = 1$, but if $\omega = (1 + x)/2$ (an asymmetric Beta distribution), then $f_{\downarrow,\omega} = 1/\pi$ with $k = 0$.

Given our finite-dimensional vector space of algebraic functionals $\delta(X)$ defined in (8), we define

$$\delta(X)_{\downarrow,\omega} = \operatorname{span}\{g_{\downarrow,\omega} : g \in \delta(X)\}, \tag{11}$$

where $g_{\downarrow,\omega}$ is interpreted from the distributional expansion (6). $\delta(X)_{\downarrow,\omega}$ is a polynomial subspace of finite dimension, and it turns out that $\delta(X)_{\downarrow,\omega}$ is our sought-after polynomial space for interpolation.

THEOREM 3.1. *The space of polynomials $\delta(X)_{\downarrow,\omega}$, defined in* (11)*, is minimally total for the interpolation conditions $\delta(X)$: data on $\delta(X)$ is isomorphic to polynomials in $\delta(X)_{\downarrow,\omega}$.*

Proof. To compare the dimensions of the subspaces $\delta(X)$ and $\delta(X)_{\downarrow,\omega}$, let $\ker_{\delta(X)}\mathscr{P}_j$ denote the kernel of $\mathscr{P}_j$ in $\delta(X)$. If $f \in \delta(X)$ and $f_{\downarrow,\omega} = \mathscr{P}_k f$ for some $k \geq 0$, this implies that $f_{\downarrow,\omega} \in \mathscr{P}_k\left(\ker_{\delta(X)}\mathscr{P}_{k-1}\right)$. Also, for all j, we have $\ker_{\delta(X)}\mathscr{P}_{j-1} \supseteq \ker_{\delta(X)}\mathscr{P}_j$ for $j \geq 0$, with $\delta(X) = \ker_{\delta(X)}\mathscr{P}_{-1}$. These two observations imply that

$$\dim \mathscr{P}_j(\ker_{\delta(X)}\mathscr{P}_{j-1}) = \dim\ker_{\delta(X)}\mathscr{P}_{j-1} - \dim\ker_{\delta(X)}\mathscr{P}_j.$$

We have already noted that if $g_{\downarrow,\omega} = \mathscr{P}_k f$, then $g_{\downarrow,\omega} \in \mathcal{V}_k^\omega$. Let $\mathcal{W}_k^\omega = \mathscr{P}_k(\ker_{\delta(X)}\mathscr{P}_{k-1}) \subset \mathcal{V}_k^\omega$. We have

$$\dim\ (\mathcal{W}_k^\omega) = \dim\ \left(\ker_{\delta(X)}\mathscr{P}_{k-1}\right) - \dim\ \left(\ker_{\delta(X)}\mathscr{P}_k\right)$$

and

$$\delta(X)_{\downarrow,\omega} = \bigoplus_{j=0}^{\infty} \mathcal{W}_k^\omega.$$

This results in the telescoping series

$$\begin{aligned}\dim\ \left(\delta(X)_{\downarrow,\omega} \cap \Pi_k^d\right) &= \dim\ \left(\bigoplus_{j=0}^{k} \mathcal{W}_k^\omega\right) = \sum_{j=0}^{k} \dim\ \left(\mathcal{W}_j^\omega\right) \\ &= \sum_{j=0}^{k} \left(\dim\ker_{\delta(X)}\mathscr{P}_{j-1} - \dim\ker_{\delta(X)}\mathscr{P}_j\right) \\ &= \dim\delta(X) - \dim\ker_{\delta(X)}\mathscr{P}_k = \dim\mathscr{P}_k(\delta(X)). \end{aligned} \tag{12}$$

Taking limits in k shows that $\dim\delta(X)_{\downarrow,\omega} = \dim\delta(X)$. This also shows that $\left(\Pi_k^d\right)_{\downarrow,\omega} = \Pi_k^d$ and $\left(\mathcal{V}_k^\omega\right)_{\downarrow,\omega} = \mathcal{V}_k^\omega$ for all k. Now if $u \in \delta(X)$ with $u \neq 0$, then $\langle u, u_{\downarrow,\omega}\rangle_\omega > 0$, and thus if $u \in \delta(X)$ satisfies $\langle u, v_{\downarrow,\omega}\rangle_\omega = 0$ for all $v \in \delta(X)_{\downarrow,\omega}$, then $u = 0$. Therefore, the space of polynomials $\delta(X)_{\downarrow,\omega}$ is total for the interpolation conditions $\delta(X)$. However, since $\dim\delta(X)_{\downarrow,\omega} = \dim\delta(X)$, it is also minimally total (meaning an isomorphism between data from $\delta(X)$ and functions on $\delta(X)_{\downarrow,\omega}$ exists). This shows that the space $\delta(X)_{\downarrow,\omega}$ is indeed our sought-after space of polynomials that is isomorphic to data on $\delta(X)$. ∎

We thus identify $\delta(X)_{\downarrow,\omega}$ as the polynomial space $\Pi_{X,\omega}$ for the interpolation problem (9). If two sets of nodes X_1 and X_2 are nested, $X_1 \subset X_2$, then the monotonicity property $\Pi_{X_1,\omega} \subset \Pi_{X_2,\omega}$ is a straightforward consequence of the definition (11).

The space $\delta(X)_{\downarrow,\omega}$ also measures how well the functionals $\delta(X)$ can be used to measure elements in Π_k^d. First we need a measure of how well a space can represent polynomials in the L^2_ω sense. We choose to make this measure the largest polynomial degree that the projected expansions of $\delta(X)$ can approximate.

DEFINITION 3.2. *For $\delta(X)$, the ω-orthogonal approximation order from $\delta(X)$ is the largest $s \in \mathbb{N}_0$ such that $\Pi_s^d = \mathscr{P}_s\delta(X)$.*

A result we will soon derive shows that the above definition is actually independent of ω. The ω-orthogonal approximation order of $\delta(X)$ can be measured by the least orthogonal space $\delta(X)_{\downarrow,\omega}$.

PROPOSITION 3.3. *The ω-orthogonal approximation order from $\delta(X)$ is the largest integer s such that $\Pi_s^d \subseteq \delta(X)_{\downarrow,\omega}$.*

Proof. Let s be the ω-orthogonal approximation order from $\delta(X)$. This means that for any $u \in \Pi_s^d$, there exists $h \in \delta(X)$ such that $\mathscr{P}_s h = u$. Then for all $|\alpha| \leq s$, choose $h_\alpha \in \delta(X)$ such that $\mathscr{P}_s h_\alpha = \Phi_\alpha$. Then we also have $(h_\alpha)_{\downarrow,\omega} = \mathscr{P}_s h_\alpha = \Phi_\alpha$. Therefore there exist $h_\alpha \in \delta(X)$ with $(h_\alpha)_{\downarrow,\omega} = \Phi_\alpha$. Since $\{\Phi_\alpha\}_{|\alpha|\leq s}$ forms a basis for $\bigoplus_{k=0}^s \mathcal{V}_k^\omega$, and $\bigoplus_{k=0}^s \mathcal{V}_k^\omega$ spans Π_s^d, we have that $\Pi_s^d \subset \delta(X)_{\downarrow,\omega}$.

Now assume that $\Pi_s^d \subset \delta(X)_{\downarrow,\omega}$; we first show that there exists a projector $\mathscr{P}_{s,\delta(X)}$ from L^2_ω into $\delta(X)$ that is "finer" than $\mathscr{P}_s$, i.e., $\mathscr{P}_s = \mathscr{P}_s \mathscr{P}_{s,\delta(X)}$. There exist elements $h_\alpha \in \delta(X)$ such that $(h_\alpha)_{\downarrow,\omega} = \Phi_\alpha$ for all $|\alpha| \leq s$. This implies that $(\langle h_\alpha, \Phi_\beta \rangle_\omega)_{|\alpha|,|\beta|\leq s}$ is a triangular matrix with unity diagonal and hence is invertible. Therefore, given $u \in \Pi_s^d$ with expansion $u = \sum_{|\alpha|\leq s} \hat{u}_\alpha \Phi_\alpha$, there are coefficients $\hat{v}_\alpha$ such that $v = \sum_{|\alpha|\leq s} \hat{v}_\alpha h_\alpha \in \delta(X)$ satisfies $\mathscr{P}_s v = u$. Similar arguments show that if we can accomplish the same task for any $u \in \Pi_{s+1}^d$, then s is not the largest integer such that $\Pi_s^d \subset \delta(X)_{\downarrow,\omega}$. Thus we have shown that that the L^2_ω approximation order from $\delta(X)$ is s. ∎

Finally, a simple argument shows that $\delta(X)_{\downarrow,\omega}$ is of smallest degree in the following sense.

PROPOSITION 3.4. *If P is any polynomial subspace that is minimally total for interpolation on X, then*

$$\dim\left(\delta(X)_{\downarrow,\omega} \cap \Pi_j^d\right) \geq \dim\left(P \cap \Pi_j^d\right)$$

for all j.

Proof. Let B be any basis of $P \cap \Pi_j$. Define B^* as the collection of b^* for all $b \in B$, where b^* is the L^2_ω dual functional corresponding to b. By definition of minimal totality, the functionals B^* are linearly independent on $\delta(X)$. In addition, for any $p \in \Pi_j^d$, $p^*(\cdot) = \langle \cdot, p \rangle_\omega = \langle \mathscr{P}_j(\cdot), p \rangle_\omega = p^* \circ \mathscr{P}_j$. Therefore, the basis B^* is also linearly independent of $\mathscr{P}_j \delta(X)$. Using (12) we then have

$$\dim\left(\delta(X)_{\downarrow,\omega} \cap \Pi_j^d\right) = \dim\left(\mathscr{P}_j \delta(X)\right) \geq |B| = \dim\left(P \cap \Pi_j^d\right). \qquad ∎$$

The original least interpolant [7] has a similar characterization of the local approximation order of a space of functionals. Since the least interpolant is degree minimal and the ω-least interpolant is also degree minimal, we immediately obtain the result that our definition of the ω-orthogonal approximation order of a space of functionals is identical to the space's local approximation order.

COROLLARY 3.5. *For any ω, the ω-orthogonal approximation order s of X is identical to its local approximation order.*

Thus s is actually independent of ω. We conclude this section by identifying the equivalence of the Hermite polynomial orthogonal least interpolant and the least interpolant by [7].

THEOREM 3.6. *Let $\omega = (\frac{1}{\sqrt{\pi}})^d \exp\left(-\|x\|_2^2\right)$ be the Hermite density function in d dimensions. Given a collection of nodes X, it follows that $\Pi_{X,\omega} = \Pi_X$, where Π_X is the polynomial space corresponding to the original least interpolant from* [7].

Our proof of this result requires details of the algorithm used to compute the space $\Pi_{X,\omega} = \delta(X)_{\downarrow,\omega}$, so we postpone presentation of the arguments. The essential

idea is the realization that the exponential generating function for the univariate Hermite polynomial family is itself an exponential function. That the exponential family is the cornerstone for the traditional least interpolant allows one to equate the two interpolants. Details are provided in section 4.5 after presentation of the interpolation algorithm.

3.2. The least orthogonal interpolation polynomial. We have shown that $\delta(X)_{\downarrow,\omega}$ is our sought polynomial space $\Pi_{\downarrow,\omega}$, isomorphic to data for interpolation on X. We now turn to the task of constructing an interpolant.

THEOREM 3.7. *Let $\{h_n\}_{n=1}^N \subset \delta(X)$ be a basis for $\delta(X)$ such that*

$$\langle h_n, (h_m)_{\downarrow,\omega}\rangle_\omega = \delta_{m,n}. \tag{13}$$

Then given a continuous function f, let $p \in \Pi^d$ be defined by

$$p = \sum_{n=1}^N \langle h_n, f\rangle_\omega \, (h_n)_{\downarrow,\omega}.$$

Then p interpolates the data $\delta_{x^{(n)}}(f)$ for $x^{(n)} \in X$.

Proof. To see this, let

$$\delta_{x^{(n)}} = \sum_{m=1}^N c_{n,m} h_m$$

define the (invertible) relationship between the functionals $\delta_{x^(n)}$ and the basis h_n of $\delta(X)$. Then

$$\begin{aligned} p(x_n) = \delta^*_{x^{(n)}}(p) &= \sum_{m=1}^N \langle h_m, f\rangle_\omega \langle (h_m)_{\downarrow,\omega}, \delta_{x^{(n)}}\rangle_\omega \\ &= \sum_{m=1}^N \langle h_m, f\rangle_\omega \left\langle (h_m)_{\downarrow,\omega}, \sum_{j=1}^N c_{n,j} h_j \right\rangle_\omega \\ &= \sum_{m=1}^N c_{n,m} \langle h_m, f\rangle_\omega \\ &= \langle \delta_{x^{(n)}}, f\rangle_\omega = \delta^*_{x^{(n)}}(f) = f(x_n). \qquad \square \end{aligned}$$

The coefficients of the interpolant $\langle h_n, f\rangle_\omega$ are computable from the data $f(x^{(n)})$. Since h_n is a linear combination of $\delta_{x^{(j)}}$, it follows that $\langle h_n, f\rangle_\omega$ is a linear combination of $f(x^{(j)})$. The precise transformation will be given in the next section.

4. Algorithm: Construction of the interpolant. We now tackle the problem of constructing the polynomial space $\delta(X)_{\downarrow,\omega}$ and the resulting interpolant given data. The main idea behind the algorithm can be understood from the proof of Theorem 3.1: we will search for the linear spaces $\mathcal{W}_k^\omega$, which are the ω-projection of $\delta(X)_{\downarrow,\omega}$ onto $\mathcal{V}_k^\omega$. We will see that this search involves only some standard numerical linear algebra, specifically LU and QR decompositions. Our method is quite similar to the "block LU" decomposition already described in [8] for the original least interpolation.[2] The algorithm can be divided into three stages:

[2] This "block LU" decomposition is not directly related to algorithmic divide-and-conquer strategies for distributing the LU factorization of a large matrix across an array of processors via block decompositions.

1. basis determination—identification of basis h_n from $\delta(X)$ satisfying (13),
2. coefficient determination—determine interpolatory coefficients on basis elements $(h_n)_{\downarrow,\omega}$,
3. connection problem—translating coefficients on $(h_n)_{\downarrow,\omega}$ to coefficients on Φ_α.

Given $\delta(X)$, we assume that for any $\lambda \in \delta(X)$, evaluation of $\lambda(\Phi_\alpha)$ for all multiindices α is (easily) computable. For example, if $\lambda = \delta_x$ for some $x \in \Omega$, then $\lambda(\Phi_\alpha) = \Phi_\alpha(x)$.

Given any basis λ_n of $\delta(X)$, we seek a transformation to a basis h_n satisfying (13). (Clearly it is simplest to choose $\lambda_n = \delta_{x^{(n)}}$.) We form the (infinite) Vandermonde-like matrix $\mathbf{V}$ with entries

$$V_{n,\alpha} = \lambda_n(\Phi_\alpha), \tag{14}$$

where we order the columns of α with any ℓ^1-respecting total ordering on $\mathbb{N}_0^d$. The columns are putatively divided into blocks defined by the values $|\alpha|$. Row n of $\mathbf{V}$ is denoted $\mathbf{V_n}$. The row vector $\mathbf{V_{n,m}}$ for nonnegative integer m denotes the block of columns in row n with $|\alpha| = m$; i.e., the block $\mathbf{V_{n,m}}$ contains expansion coefficients of λ_n in $\mathcal{V}_m^\omega$. We will use an intuitive notation to specify submatrices. For example, $\mathbf{V_{2:5,0:3}}$ is the submatrix of $\mathbf{V}$ corresponding to rows 2–5 and column blocks $m = |\alpha| = 0, 1, 2, 3$.

Remark 4.1. The proof of Lemma 2.1 shows that $\delta(X)_{\downarrow,\omega} \subseteq \Pi_{N-1}^d$, where $N = |X|$. Thus an infinite-size Vandermonde matrix is unnecessary: we need only consider all columns up to $|\alpha| < N$. This also implies that we need not consider the span of δ_x as given by (6). Instead, we need only consider the span of the x-centered L^2_ω-reproducing kernels K_{N-1} of Π_{N-1}^d:

$$K_{N-1}(x, \cdot) = \sum_{|\alpha|<N} \Phi_\alpha(x)\Phi_\alpha(\cdot).$$

4.1. Basis determination. $\mathbf{V_n}$ represents expansion coefficients of λ_n in the Φ_α; cf. (6). Note that with this layout, if $h \in \delta(X)$ is the functional defined by the expansion coefficients in $\mathbf{V_n}$, then $h_{\downarrow,\omega}$ is easily identifiable as the polynomial corresponding to the first nonzero column block of that row. In an abuse of notation, we shall also write $(\mathbf{V_n})_{\downarrow,\omega}$ to mean the polynomial $h_{\downarrow,\omega}$ corresponding to the functional h whose expansion coefficients are given by $\mathbf{V_n}$. Then in order to find basis elements satisfying (13), we now need only perform appropriate row operations. We will store a diary of these operations in the $N \times N$ matrix $\mathbf{L}$. In addition, we will store some additional orthogonalization information in an $N \times N$ matrix $\mathbf{U}$.

Let us start at row $n = 1$ and column block $m = 0$. We seek to orthogonalize $\mathbf{V_k}$ for $k > n$ against $(\mathbf{V_n})_{\downarrow,\omega}$. Therefore, let $\mathbf{R}^T\mathbf{Q}$ be the LQ decomposition of $\mathbf{V_{n:N,m}}$; the decomposition will find $r_m = \operatorname{rank}\mathbf{R}$ linearly independent rows of $\mathbf{V_{n:N,m}}$; we want to allocate r_m functionals to this degree m and then move on to higher degrees. To do this, set the first r_m rows of $\mathbf{V_{n:N,m}}$ to be the first r_m rows of $\mathbf{Q}$, and let $\mathbf{L_{n:N,n:(n+r_m-1)}}$ be the first r_m rows of $\mathbf{R}^T$. In addition, perform the same row operations on all columns of $\mathbf{V}$ with degrees $> m$. Now enforce orthogonality to all rows with indices $< n$: we already have r_m orthogonal rows from $\mathbf{Q}$—we need only take inner products between the first r_m rows of $\mathbf{Q}$ and all rows $\mathbf{V_k}$ with $k < n$ and store this information in $\mathbf{U}$. Now set $m \leftarrow m + 1$ and $n \leftarrow n + r_m$. Repeat this paragraph until $n = N + 1$.

Now the matrix $\mathbf{L}$ is a lower-triangular matrix containing a diary of the row operations committed by the successive LQ factorizations on column blocks; i.e., these

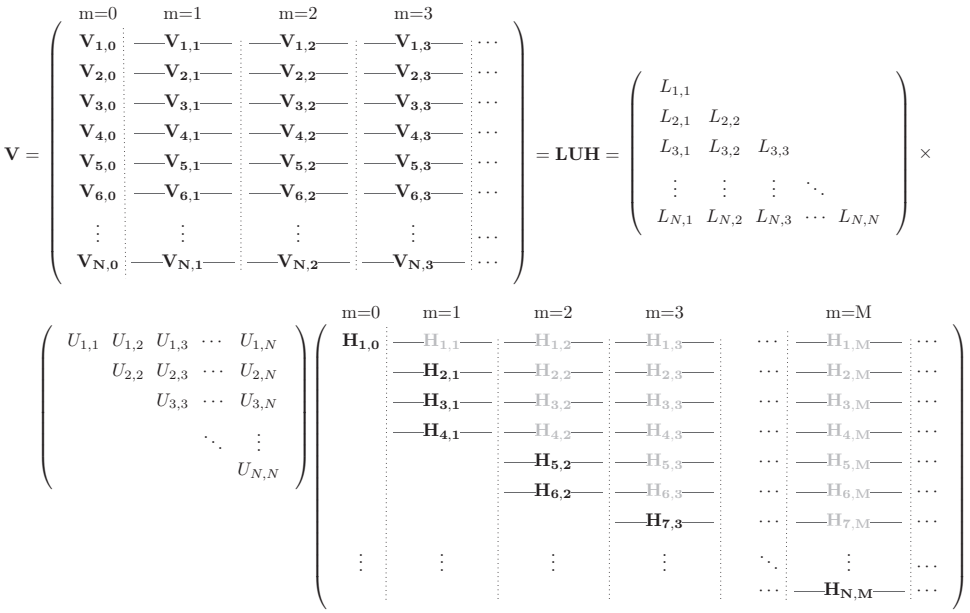

FIG. 1. *Schematic of the "block LU" factorization starting with the Vandermonde-like matrix* $\mathbf{V}$. *For* $\mathbf{V}$ *and* $\mathbf{H}$, *entry* $\mathbf{V_{n,m}}$ *is a row vector of length* $\dim \Pi^d_{=m}$. *In this example,* $r_0 = \dim \mathcal{W}_0^\omega = 1$, $r_1 = \dim \mathcal{W}_1^\omega = 3$, *and* $r_2 = \dim \mathcal{W}_2^\omega = 2$. *The maximum polynomial degree in* $\delta(X)_{\downarrow,\omega}$ *is* M. *Standard bold entries* $\mathbf{H_{n,m}}$ *are basis coefficients in the* Φ_α *for* $(h_n)_{\downarrow,\omega} \in \mathcal{W}_m^\omega$. *Faded entries indicate potentially nonvanishing elements that may be discarded.*

are orthogonalization steps on rows "below." The matrix $\mathbf{U}$ similarly contains inner product information to complete the orthogonalization; i.e., these are orthogonalization steps on rows "above." If we left-multiply the original $\mathbf{V}$ with $\mathbf{U}^{-1}\mathbf{L}^{-1}$, we obtain a matrix $\mathbf{H}$, which is "block" diagonal—block m contains r_m nonvanishing rows and $\dim \Pi^d_{=m}$ columns. All rows in each column block of $\mathbf{H}$ are mutually orthogonal and contain at least r_m nonvanishing rows. We thus obtain the decomposition $\mathbf{V} = \mathbf{LUH}$, which is shown in Figure 1. We truncate the matrix $\mathbf{H}$ at column block M, where M is the maximum degree used. Then $\mathbf{H}$ has orthogonal rows.

Relating the algorithmic variables to the abstract construction, $r_m = \dim \mathcal{W}_k^\omega$, where $\mathcal{W}_k^\omega$ is the space from the proof of Theorem 3.1 and represents $\delta(X)_{\downarrow,\omega} \cap \mathcal{V}_k^\omega$. Let $m(n)$ denote the degree corresponding to $(\mathbf{H_n})_{\downarrow,\omega}$, i.e., $(h_n)_{\downarrow,\omega} \in \mathcal{V}_m^\omega$. Then the "diagonal" elements of $\mathbf{H}$ are the row vectors $\{\mathbf{H_{n,m(n)}}\}_{n=1}^N$ and contain the Φ_α-expansion coefficients of $(h_n)_{\downarrow,\omega}$ from (13). For any fixed m, the coefficients $\{\mathbf{H_{n,m}}\}$ for all n satisfying $m(n) = m$ define the subspace $\mathcal{W}_m^\omega$.

The complexity of this step can be considerable: we require a QR-type decomposition of size $\dim \mathcal{V}_m^\omega$ for every degree m. In high dimensions it is unlikely that m will be very large, but in the "worst" case where we require a unique order m for each functional, the size of the column blocks can grow very large (though the number of rows is bounded by N). This is not necessarily a weakness of the algorithm but is a fundamental challenge in the task of computing $\delta(X)_{\downarrow,\omega}$. However, we must emphasize that this is a "preprocessing" step in the sense that the determination of the basis depends *only* on the node locations X and not on the function data. Therefore, no simulation of the underlying physical system is required. And once this

step is completed, the resulting interpolation basis can be used for any function data obtained on the same nodes X.

The basis determination step is the most complicated step. Further improvements include performing row permutations and storing only the diagonal elements of $\mathbf{H}$. We give a more detailed version of the algorithm in Algorithm 1. Although Algorithm 1 details usage of a permutation matrix $\mathbf{P}$ such that $\mathbf{PV} = \mathbf{LUH}$, we will omit explicit use of the permutation matrix in the text for simplicity.

ALGORITHM 1. The block LU factorization for basis pursuit (independent of interpolation data $\mathbf{f}$).

Input: N functionals λ_n on Π^d, orthogonal polynomials Φ_α
Initialize: $n = 1$, $m = 0$, vector $\mathbf{r}$, identity arrays $\mathbf{L}$, $\mathbf{P}$, $\mathbf{U}$
Initialize: Vandermonde matrix $\mathbf{V}$, $V_{n,\alpha} = \lambda_n(\Phi_\alpha)$
while $n \le N$ **do**
  $\mathbf{V_{1:N,m}} \leftarrow (\mathbf{L_{1:N,1:N}})^{-1}\mathbf{V_{1:N,m}}$
  Do $\mathbf{Q}, \mathbf{R}, \mathbf{T} = \texttt{QR\_factorization}((\mathbf{V_{n:N,m}})^T)$.
  $rk \leftarrow \text{rank}(\mathbf{R})$
  Permute rows $n, \ldots, N$ of $\mathbf{L}, \mathbf{V}$, and $\mathbf{P}$ according to $\mathbf{T}$
  $\mathbf{L_{n:N,n:n+rk-1}} \leftarrow (\mathbf{R_{1:rk,:}})^T$
  $\mathbf{V_{n:(n+rk-1),m}} \leftarrow (\mathbf{Q_{:,1:rk}})^T$
  $\mathbf{U_{1:(n-1),m}} \leftarrow \mathbf{V_{1:(n-1),m}}\mathbf{Q_{:,1:rk}}$
  $r_{m+1} \leftarrow rk$
  $n \leftarrow n + rk$, $m \leftarrow m + 1$
end while
Output: $\mathbf{P}$, $\mathbf{L}$, $\mathbf{U}$, $\mathbf{r}$, "diagonal" elements of $\mathbf{V}$ (named $\widetilde{\mathbf{H}}$ in text)

4.2. Coefficient determination. We have computed the decomposition

$$\mathbf{V} = \mathbf{LUH}$$

in the basis determination section. The matrix $\mathbf{H}$ contains coefficients for the basis elements of $\delta(X)_{\downarrow,\omega}$ on its "diagonal." Therefore, given interpolation data $\mathbf{f} \in \mathbb{R}^N$, the solution to the system

$$\mathbf{LUu} = \mathbf{f} \tag{15}$$

defines the coefficients $\mathbf{u}$ in the ω-orthogonal basis elements (13). Since both $\mathbf{L}$ and $\mathbf{U}$ are triangular and invertible, this operation is a routine linear solve of size N—the same as any other linear interpolation problem.

4.3. Connection problem. Finally, we seek a method for translating the h_n coefficients $\mathbf{u}$ into the Φ_α coefficients $\mathbf{c}$. Since we have the diagonal elements of $\mathbf{H}$, this is not difficult:

$$\sum_{n=1}^{N} u_n h_n = \sum_{n=1}^{N} u_n \left(\sum_{|\alpha = m(n)|} H_{n,\alpha}\Phi_\alpha \right) = \sum_{|\alpha| \le m(N)} \left(\sum_{n=1}^{N} u_n H_{n,\alpha} \right) \Phi_\alpha.$$

Therefore,

$$c_\alpha = \sum_{n=1}^{N} u_n H_{n,\alpha} = \sum_{n:|\alpha|=m(n)} u_n H_{n,\alpha}. \tag{16}$$

This step obviates the fact that storage of the entire matrix $\mathbf{H}$ is entirely unnecessary—we need only the diagonal entries: for any row n we need only the column block m satisfying $m = m(n)$. Therefore, during the basis determination step, once we find $\mathbf{L}$ and $\mathbf{U}$, we need only store the "diagonal" of the matrix $\mathbf{U}^{-1}\mathbf{L}^{-1}\mathbf{V}$ (which is computed on-the-fly anyway) and then discard $\mathbf{V}$. A more practical way to do this is to generate only column block $|\alpha| = m$ of $\mathbf{V}$ at each step. For large-dimensional problems this is a significant savings of memory since we need only store a portion of the Vandermonde matrix at any given time. These improvements are not listed in Algorithm 1.

An alternative way to apply the procedure (16) is as follows: let $\widetilde{\mathbf{H}}$ denote the matrix $\mathbf{H}$ where the "off-diagonal" entries are set to 0. (This is tantamount to applying the operation $(\cdot)_{\downarrow,\omega}$ on each row.) Then $\mathbf{c} = \widetilde{\mathbf{H}}^{\mathbf{T}}\mathbf{u}$, where $\mathbf{c}$ is a vector containing the c_α. This allows us to mathematically characterize the action of the algorithm. The map $\mathbf{u} \mapsto \mathbf{c}$ is effected by the matrix $\widetilde{\mathbf{H}}^T$, and so

$$\mathbf{c} = \widetilde{\mathbf{H}}^T\mathbf{u} = \widetilde{\mathbf{H}}^T\mathbf{U}^{-1}\mathbf{L}^{-1}\mathbf{f} \triangleq \mathbf{V}^+\mathbf{f}.$$

The matrix $\mathbf{V}^+$ is a right-inverse of $\mathbf{V}$: $\mathbf{V}\mathbf{V}^+ = \mathbf{I}$. It is in general not the Moore–Penrose pseudoinverse.

We have completed the interpolation problem: the (unique) polynomial from the space $\delta(X)_{\downarrow,\omega}$ that interpolates the data $\mathbf{f}$ is given by $\sum_\alpha c_\alpha \Phi_\alpha$.

4.4. Adding nodes to X. The previous sections dealt with computing the dimension-N space $\Pi_{X,\omega}$, along with the information necessary to compute interpolants. We now consider the case of adding a point $x^{(N+1)} \in \mathbb{R}^d$ to X. That is, assume that we have all the decomposition information from Algorithm 1 for X, and we now wish to add $x^{(N+1)} \notin X$ to the space of nodes and update all the matrices accordingly. An efficient way to accomplish this will be useful for adaptive approaches.

Due to the monotonicity of the least orthogonal interpolation space, this task can be accomplished by simply adding an extra basis function to $\Pi_{X,\omega}$. Therefore, we can take advantage of a significant computational savings in computing the basis (the most expensive part of the algorithm). As usual, we will need the expansion coefficients of $\delta_{x^{(N+1)}}$. Instead of calling it row $N+1$ of $\mathbf{V}$, in this subsection we will denote this row vector $\mathbf{x}$. Given $\mathbf{x}$, we will update row $N+1$ of $\mathbf{L}$, and column $N+1$ of $\mathbf{U}$.

Let the L^2_ω approximation order of $\Pi_{X,\omega}$ be $s \geq 0$. This means that $\Pi^d_s \subseteq \Pi_{X,\omega}$, where s is the largest possible value. Let $m \leq s$. Then the submatrix $\mathbf{H}_{\mathbf{n_1}:\mathbf{n_2},\mathbf{m}}$ for $n_1 = 1 + \dim \Pi^d_{m-1}$ and $n_2 = \dim \Pi^d_m$ is a square, orthogonal matrix. Furthermore, all blocks $\mathbf{H}_{\mathbf{n},\mathbf{m}}$ for $n \leq \dim \Pi^d_{m-1}$ vanish. This means that linear combinations of the first $\dim \Pi^d_s$ rows of $\mathbf{V}$ can eliminate the first $M = \dim \Pi^d_s$ columns of $\mathbf{x}$. Let us compute these elimination coefficients and store them in $\mathbf{L}_{\mathbf{N+1},\mathbf{1}:\mathbf{M}}$.

For further elimination we require the entries $\mathbf{V}_{\mathbf{n},\mathbf{m}}$ for $n \leq N$ and $m > s$. This can be avoided if we stored nonvanishing "off-diagonal" elements of $\mathbf{H}$ in the initial basis determination for X, but here we assume that this information was discarded.

We start with degree $m = s+1$, and we denote column block m of the row vector $\mathbf{x}$ as $\mathbf{x_m}$. We updated $\mathbf{x_m}$ by performing all previous elimination steps used for previous column blocks. Once this is done, we orthogonalize $\mathbf{x_m}$ against the nonvanishing rows in column block m of $\widetilde{\mathbf{H}}$. If the remaining elements in $\mathbf{x_m}$ vanish, then we move to the next column block. Otherwise, we mark node $N+1$ as contributing to this degree and update the matrices accordingly. The precise transformations are given in Algorithm 2.

ALGORITHM 2. The algorithm for adding a new polynomial to $\Pi_{X,\omega}$ given a new node location.

Input: $N+1$ functionals λ_n on Π^d, orthogonal polynomials Φ_α
Input: block LU information for $\{\lambda_n\}_{n=1}^N$: $\mathbf{L}$, $\mathbf{U}$, $\mathbf{P}$, $\mathbf{r}$, $\widetilde{\mathbf{H}}$, approximation order s
Initialize node $N+1$ expansion coefficients: $x_\alpha = \lambda_{N+1}(\Phi_\alpha)$
$M \leftarrow \dim \Pi_s^d$
$\mathbf{L_{N+1,1:M}} \leftarrow \mathbf{x_{1:s}}\widetilde{\mathbf{H}}^T_{\mathbf{1:M,1:s}}(\mathbf{L_{1:M,1:M}U_{1:M,1:M}})^{-1}$
$\mathbf{x} \leftarrow \mathbf{x} - \mathbf{L_{N+1,1:M}V_{1:M,:}}$
$m \leftarrow s+1$
while true **do**
  rows $= (M+1) : (M + r_m)$
  $\mathbf{L_{N+1,rows}} \leftarrow \mathbf{x_m}\widetilde{\mathbf{H}}^T_{\mathbf{rows,m}}(\mathbf{L_{rows,rows}U_{rows,rows}})^{-1}$
  $\mathbf{x} \leftarrow \mathbf{x} - \mathbf{L_{N+1,rows}V_{rows,:}}$
  if $\mathbf{x_m}$ vanishes **then**
    $M \leftarrow M + r_m$, $m \leftarrow m+1$
  else
    $\mathbf{L_{N+1,N+1}} \leftarrow \|\mathbf{x_m}\|_2$, $\widetilde{\mathbf{H}}_{\mathbf{N+1,m}} \leftarrow \mathbf{x_m}/L_{N+1,N+1}$
    $\mathbf{V_{1:M,m}} \leftarrow (\mathbf{L_{1:M,1:M}})^{-1}\mathbf{V_{1:M,m}}$
    $\mathbf{U_{1:M,N+1}} \leftarrow \mathbf{V_{1:M,m}}\widetilde{\mathbf{H}}^T_{\mathbf{N+1,m}}$
    for $n = M+1, \ldots, N$ **do**
      $m \leftarrow m(n)$
      $\mathbf{U_{N+1,n}} \leftarrow \mathbf{H_{n,m}x_m}$
    end for
    exit
  end if
end while

Note that $\mathbf{L}$ and $\mathbf{U}$ are no longer triangular matrices, and $\widetilde{\mathbf{H}}$ is no longer block diagonal, but an appropriate row permutation will restore these properties. Thus, one need only update the permutation matrix $\mathbf{P}$ (see Algorithm 1) to restore the familiar structure of Figure 1, with $N+1$ rows. The complexity of adding a node is far less than the entire basis determination step—we require only one *step* of a QR process for each degree m that we search.

4.5. Proof of Theorem 3.6. With the algorithm presented, we may now prove our earlier assertion that the Hermite least orthogonal interpolant is equivalent to the least interpolant of [7]. This section presents two lemmas which we combine at the end to prove the desired result.

We must first review some properties of the original least interpolant of [7]. For any $g : \mathbb{R}^d \to \mathbb{R}$ analytic at the origin, define

$$g_\downarrow(x) = \sum_{\alpha:|\alpha|=k} (D^\alpha g)(0)\frac{x^\alpha}{\alpha!},$$

where k is chosen as the first nonnegative integer such that the above expression is nonvanishing. Given N points $x^{(n)} \in \mathbb{R}^d$ the (traditional) least interpolant is defined as the unique interpolating polynomial from the polynomial space

$$(\exp(X))_\downarrow = \operatorname{span}\{g_\downarrow : g \in \exp X\},$$

where $\exp X$ is the subspace spanned by exponentials of frequency $x^{(n)}$,

$$\exp(X) = \operatorname{span}\{\exp_{x^{(n)}} : x^{(n)} \in X\},$$

and $\exp_x$ is the map $y \mapsto \exp(y \cdot x)$ for $y \in \mathbb{R}^d$ with $a \cdot b$ the Euclidean inner product in d dimensions.

LEMMA 4.1. *Let $G_x : \mathbb{R}^d \to \mathbb{R}$ be defined by*

$$G_x(y) = \exp\left(-\frac{1}{2}\|x\|_2^2 + (cx) \cdot y\right)$$

for any scalar $c \neq 0$. Then $G(X) = \operatorname{span}\{G_{x^{(n)}} : x^{(n)} \in X\}$ is equal to $\exp(X)$.

Proof. The only technicality is the scaling factor c. This is resolved by using a property of the traditional least interpolation space: $\Pi_{cX} = \Pi_X$ for all nonvanishing scalar c [8]. ∎

In the remainder of this section, we consider only the Hermite polynomial weight function $\omega(x) = (2\pi)^{-d/2}\exp(-\frac{1}{2}\|x\|_2^2)$ over all $\mathbb{R}^d$. The spaces $\mathcal{V}_k^\omega$ are now unique, and we make use of one explicit representation for this orthogonal polynomial family: $\Phi_\alpha = \prod_{q=1}^d H_{\alpha_q}$, where H_n is the univariate (orthonormal) Hermite polynomial of degree n. Having made Φ_α well defined, let $[x]^\alpha$ be the weighted monomials: $[x]^\alpha = x^\alpha/\sqrt{\alpha!}$. Let $\mathbf{C}$ be the matrix that connects the monomials to the Hermite polynomials:

$$\Phi_\alpha = \sum_{|\beta| \leq |\alpha|} C_{\alpha,\beta}[x]^\beta.$$

We consider $\mathbf{C}$ as a well-defined two-dimensional array by imposing a strict total ordering on $\mathbb{N}_0^d$ that respects the ℓ^1 norm, and the same ordering is applied for both Φ_α and $[x]^\alpha$. $\mathbf{C}$ is invertible (and lower triangular).

Noting that the Taylor expansion of $\exp(x \cdot y)$ around the origin is given by

$$\exp(x \cdot y) = \sum_\alpha [x]^\alpha [y]^\alpha$$

allows us to use Algorithm 1 to compute the required block LU decomposition for the traditional least interpolant.

LEMMA 4.2. *Let $V_{n,\alpha} = [x^{(n)}]^\alpha$. Then the algorithm defined by Algorithm* 1*, along with* (15) *and* (16)*, results in expansion coefficients of the traditional least interpolant in the basis $[x]^\alpha$.*

We can now finish the proof of Theorem 3.6.

Proof of Theorem 3.6. We will show that computations for the traditional least interpolant can be reformulated as a transformation of the ω-least interpolant. Let $\mathbf{V}^\omega$ be the matrix with entries $V^\omega_{n,\alpha} = \Phi_\alpha(x^{(n)})$ and $\mathbf{V}$ be the matrix with entries $V_{n,\alpha} = [x^{(n)}]^\alpha$. Then we have $\mathbf{V}\mathbf{C}^T = \mathbf{V}^\omega$. Similarly, if $f \in \Pi^d$ has expansions $f = \sum_\alpha \hat{f}^\omega \Phi_\alpha = \sum_\alpha \hat{f}[\cdot]^\alpha$, then $\mathbf{C}^T\hat{\mathbf{f}}^\omega = \hat{\mathbf{f}}$.

We note that the univariate Hermite polynomial exponential generating function is an exponential:

$$\exp\left(-\frac{1}{2}x^2\right)\exp_{x\sqrt{2}} = \pi^{1/4}\sum_{n=0}^\infty \Phi_n(\cdot)[x]^n.$$

(Recall that Φ_n are orthonormal, resulting in a $1/\sqrt{n!}$ instead of a $1/n!$ in the formula.) By taking tensor products, we obtain

$$G_x(y) = \frac{1}{\pi^{d/4}} \exp\left(-\frac{1}{2}\|x\|_2^2\right) \exp_{x\sqrt{2}}(y) = \sum_\alpha \Phi_\alpha(y)[x]^\alpha.$$

Given X, Lemma 4.1 states that $(G_X)_{\downarrow,\omega} = (\exp X)_\downarrow$. That is, the "least" operator $(\cdot)_\downarrow$ is invariant at $\exp X$ under the replacement $[\cdot]^\alpha \to \Phi_\alpha$. Then we may replace the appropriate quantities in the algorithm to obtain the same result.

By Lemma 4.2, the input to the algorithm for the traditional least interpolant from the space $\exp X$ is $\mathbf{V}$ and produces coefficients $\mathbf{c}$ given some data $\mathbf{f}$. Under the replacement discussed in the above,

$$\mathbf{V}\mathbf{c} = \mathbf{f} \quad \longrightarrow \quad \left(\mathbf{V}\mathbf{C}^T\right)\left(\mathbf{C}^{-T}\mathbf{c}\right) = \mathbf{f},$$

where the equality on the right is just a reformulation of the traditional least interpolant. However, this equality is just $\mathbf{V}^\omega \mathbf{c}^\omega = \mathbf{f}$, which coincides with inputs to the algorithm for the ω-least interpolant and the resulting output of Φ_α coefficients $\mathbf{c}^\omega$. ∎

The method of proof above allows us to generalize this result to formally connect the least orthogonal operator $(\cdot)_{\downarrow,\omega}$ with the traditional least operator $(\cdot)_\downarrow$.

COROLLARY 4.3. *Let ω be any tensor-product weight function satisfying the assumptions of section* 2.3. *Define*

$$G_x^\omega(\cdot) = \sum_\alpha [x]^\alpha \Phi_\alpha(\cdot)$$

as the exponential generating function for the orthogonal polynomial family associated with ω with $\Phi_\alpha = \prod_{q=1}^d \Phi_{\alpha_q}^{[q]}$, where $\Phi_m^{[k]}$ is the univariate orthonormal polynomial family associated with dimension k. Then $\left(G_X^\omega\right)_\downarrow = \delta(X)_{\downarrow,\omega}$.

While interesting, this result is not constructive: given f analytic at the origin, the traditional least interpolant for the space G_X^ω interpolates the functionals $f \mapsto \sum_\alpha [x]^\alpha D^\alpha \Phi_\alpha(0)$, which does not necessarily correspond to any well-known functional. Of course, if ω is a standard multivariate Gaussian density, this functional is a (scaled) point evaluation. The tensor product assumption is made in Corollary 4.3 because the function G_x^ω is not invariant under arbitrary orthogonal transformations of the basis Φ_α for $\mathcal{V}_{|\alpha|}^\omega$. In other words, the correspondence $[\cdot]^\alpha \leftrightarrow \Phi_\alpha$ is not straightforward in the nontensor product case.

5. Examples. In this section we give examples of implementation aspects of the least orthogonal interpolant. We first explore some simple examples in two dimensions to show that the choice of the weight function ω that defines the polynomial space $\delta(X)_{\downarrow,\omega} = \Pi_{X,\omega}$ is important. We also show that the choice of interpolation space $\Pi_{X,\omega}$ tends to concentrate accuracy of the interpolant in areas of high probability. We follow these examples with higher-dimensional examples: we show an example of regression versus interpolation for a multivariate Gaussian function and explore the linear conditioning of computing least orthogonal interpolants.

5.1. Two-dimensional examples. We consider some two-dimensional examples to explore how the least orthogonal interpolant depends on the choice of probability measure ω.

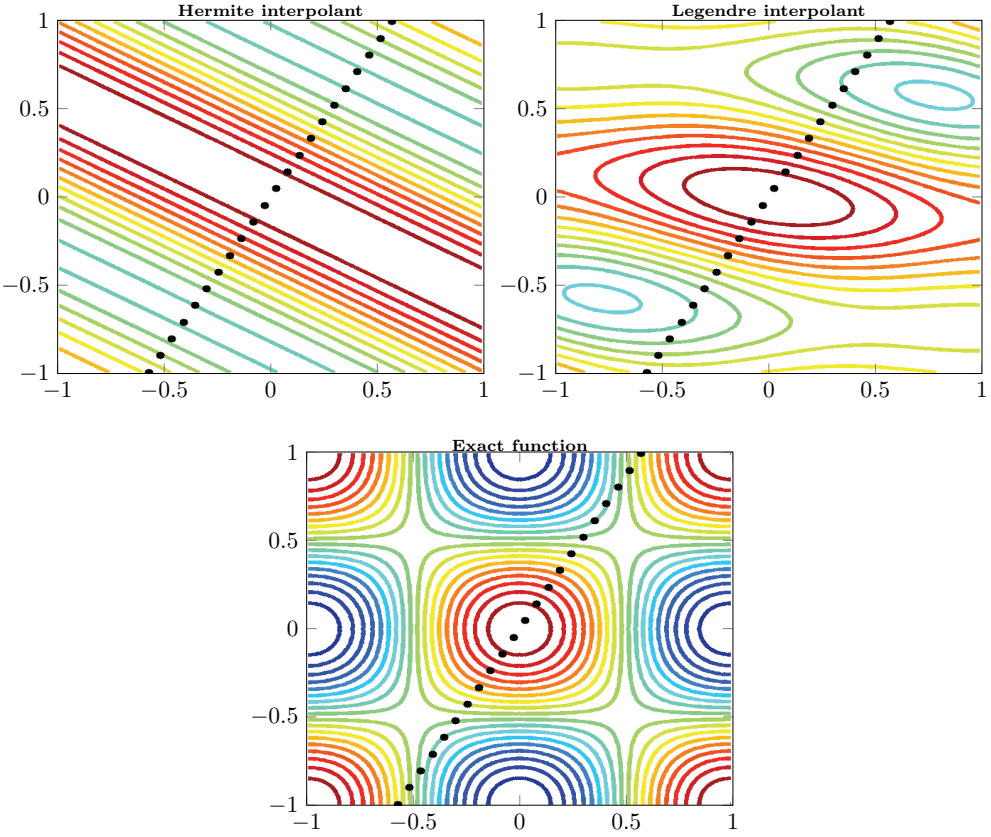

FIG. 2. *Contour plots for the least orthogonal interpolant for the Hermite (top left) and Legendre (top right) weight functions. The points where interpolation is enforced are marked on the plots by* 20 *collinear black dots. Dark lines indicate lower values.*

5.1.1. Interpolation examples. We distribute 20 collinear equispaced two-dimensional nodes $\{x^{(n)}\}$ skewed 30° clockwise from the x_2-axis (the vertical axis). Note that this choice of nodes would induce a singular Vandermonde matrix for the classical polynomial interpolation. We use the test function

$$f(x) = \cos(\pi x_1)\cos(\pi x_2) \tag{17}$$

to generate samples $\lambda_n(f) = \delta^*_{x^{(n)}}(f)$ that we interpolate using the least orthogonal interpolant. We consider two regions Ω with weight functions ω:

1. $\Omega = \mathbb{R}^2$ with $\omega \propto \exp(-\|x\|^2)$. This defines Hermite orthogonal polynomials.
2. $\Omega = [-1,1]^2$ with $\omega = \frac{1}{4}$. This defines Legendre orthogonal polynomials.

The interpolation polynomials are shown in Figure 2, in contour lines along with the data points. We observe that both interpolations accomplish what they are intended to do—they interpolate the function data precisely. In this sense, both interpolation results are "correct." Away from the data points, the interpolations are notably different. And this is acceptable because no information was provided to the interpolations away from the line. The Hermite interpolation, which coincides with the original least interpolation [7], demonstrates homogeneous behavior, while the Legendre interpolation exhibits nonhomogeneous structures because of the existence of implicit boundaries.

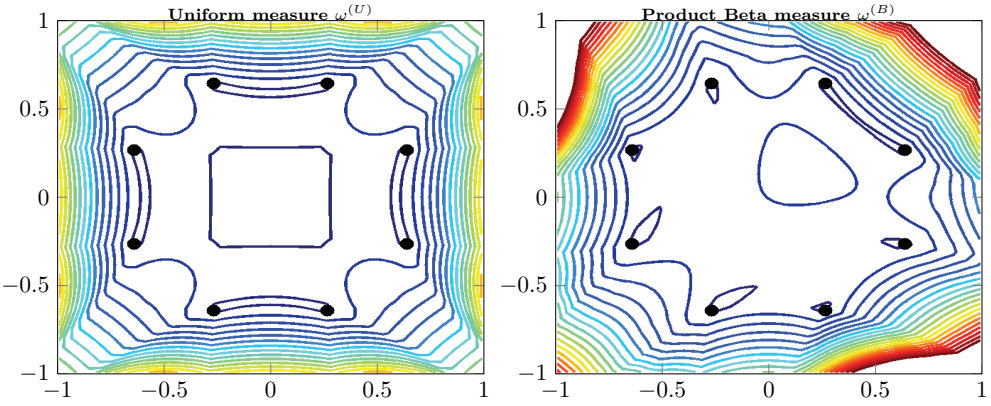

FIG. 3. *Contour plots for the Lebesgue function $\lambda(x)$ for the uniform probability measure $\omega^{(U)}$ (left) and for the product-Beta probability measure $\omega^{(B)}$ (right). The interpolation points are marked by black dots. Dark lines indicate lower values of the Lebesgue function. The high Lebesgue function values for $\omega^{(B)}$ near $x_1 = 1$ or $x_2 = 1$ indicate the poor approximation quality in regions of low probability.*

5.1.2. Interpolation accuracy. Fixing a collection of nodes $X \subset \Omega$ and the resulting space $\delta(X)_{\downarrow,\omega} \subseteq \Pi^d_{N-1}$, the Lebesgue constant is a measure of how well an interpolant from $\delta(X)_{\downarrow,\omega}$ can approximate functions continuous on Ω (cf. [5]). Here the Lebesgue constant is defined as

$$\begin{aligned} L(X,\omega) &= \max_{x\in\Omega} \lambda(x) \\ &\triangleq \max_{x\in\Omega} \sum_{n=1}^{N} |\ell_n(x)|, \end{aligned}$$

where $\ell_n(x)$ is the Lagrange interpolating polynomial at node x_n from $\delta(X)_{\downarrow,\omega}$. That is, $\ell_n(x) \in \delta(X)_{\downarrow,\omega}$ is the unique polynomial that satisfies $\ell_n(x_m) = \delta_{n,m}$, where $\delta_{n,m}$ is the Kronecker delta. $\lambda(x)$ is called the Lebesgue function. Then for all continuous f, the unique polynomial $p(f) \in \delta(X)_{\downarrow,\omega}$ that interpolates f at X satisfies

$$\|f - p(f)\|_\infty \le [1 + L(X,\omega)]\, \|f - p^*\|_\infty,$$

where p^* is the $\|\cdot\|_\infty$-best approximating polynomial from $\delta(X)_{\downarrow,\omega}$, and $\|\cdot\|_\infty$ is maximum norm, i.e., the W^0_∞ Sobolev norm. Therefore, smaller values of the Lebesgue function imply more accurate approximation.

We consider 8 points spaced equally on the circle $\|x\|_2^2 = 0.7$ on $\Omega = [-1,1]^2$ with two different weight functions: the uniform $\omega^{(U)}(x) = \frac{1}{4}$ and the product Beta distribution $\omega^{(B)} \propto (1-x_1)^{10}(1-x_2)^{10}$. The weight function $\omega^{(B)}$ has low probability in regions where $x_1 = 1$ or $x_2 = 1$. We show contour plots of the Lebesgue function $\lambda(x)$ in Figure 3. We observe that the least orthogonal interpolant exhibits a large Lebesgue-function value in areas with small probability—this indicates that the least orthogonal interpolant focuses effort on interpolation quality in areas with high ω probability.

5.2. High-dimensional examples. We now present some examples in high-dimensional spaces. Note that node selection is a particularly difficult, and understudied, topic in high dimensions. Here we employ certain "safe" and heuristic choices of nodes to demonstrate the applicability of the least orthogonal interpolation.

TABLE 1
Interpolation metrics for the least orthogonal interpolation in section 5.2.1.

Dimension d	N	CPU time (s)	k	$\dim \Pi_k^d$
2	9	0.04	4	15
3	37	0.08	6	84
4	61	0.08	5	126
5	89	0.10	5	252
6	121	0.18	5	462
7	157	0.13	4	330
8	197	0.25	4	495
9	241	0.46	4	715
10	289	1.16	4	1,001
11	341	2.52	4	1,365
12	397	6.05	4	1,820
13	457	10.09	4	2,380
14	521	16.62	4	3,060
15	589	29.84	4	3,876

5.2.1. Interpolation of a Gaussian. We consider the Gaussian test function

$$f(x) = \exp\big(-w\|x - x^{(0)}\|^2\big),$$

where $w > 0$ is a width parameter and $x^{(0)} \in \mathbb{R}^d$ is an offset coordinate. For the following example, we set $x^{(0)} = (0.1, 0.1, \ldots, 0.1) \in \mathbb{R}^d$ and $w = \frac{(d+1)}{3d}$. We interpolate this function with a variation of the Stroud-3 grid [18] on $[-1,1]^d$ using the uniform weight $\omega \equiv 1/2^d$. The grid is defined as follows: let X_0 be a standard $2d$-point Stroud-3 grid from [18]. We generate d rotations of this grid: define $T_\theta[i,j]$ as the rotation operator in d dimensions that rotates the $i-j$ coordinates by θ radians. Let $\theta_q = \pi/3$ for $1 \le q \le d-1$ and $\theta_d = \pi/4$. The qth rotation operator T_q for $1 \le q \le d$ is defined by

$$T_q = T_{\theta_q}[1,2]T_{\theta_{q+1}}[2,3]T_{\theta_{q+2}}[3,4]\cdots T_{\theta_{q+j-1}}[j,j+1]\cdots T_{\theta_{q+d-1}}[d,1],$$

where the subscript for the angles θ is evaluated mod d. Then $X_q = T_q X_0$. This results in about $2d^2$ nodes $X = \cup_{q=1}^d X_q$ (some are repeated and are removed). We generate another grid $Y = \cup_{q=1}^d Y_q$, with $Y_q = T_q Y_0$, but we set $\theta_q \leftarrow -\theta_q$ and Y_0 is a standard Stroud-3 grid with magnitude multiplied by $1/\sqrt{2}$. Thus we have about $4d^2$ nodes for interpolation. The exact tabulation is given in Table 1.

We perform least orthogonal interpolation with the uniform weight, and also cubic least-squares regression on the same data. We estimate the L^2_ω and L^∞ errors using Monte-Carlo sampling with 10^6 samples for dimensions $d = 2, 3, \ldots, 15$. The error results are plotted in Figure 4. For this example, we observe that the ability to add higher-degree information about the function via the least orthogonal interpolant allows one to obtain smaller errors than with a cubic least-squares fit. But we must remark that one cannot generalize this result freely. As stated earlier, regression and interpolation are different approaches with different properties.

For least orthogonal interpolation, we also list the total CPU time (i.e., basis determination, linear solving for $\mathbf{u}$, and transformation to gPC coefficients $\mathbf{c}$) and the largest polynomial degree used in Table 1. This is for reference purposes. The CPU time is measured on a 3.06 GHz dual processor with 4GB of RAM.

5.2.2. Linear conditioning. Univariate polynomial interpolation is notoriously ill-conditioned unless one takes special care in the choice of basis set and/or the

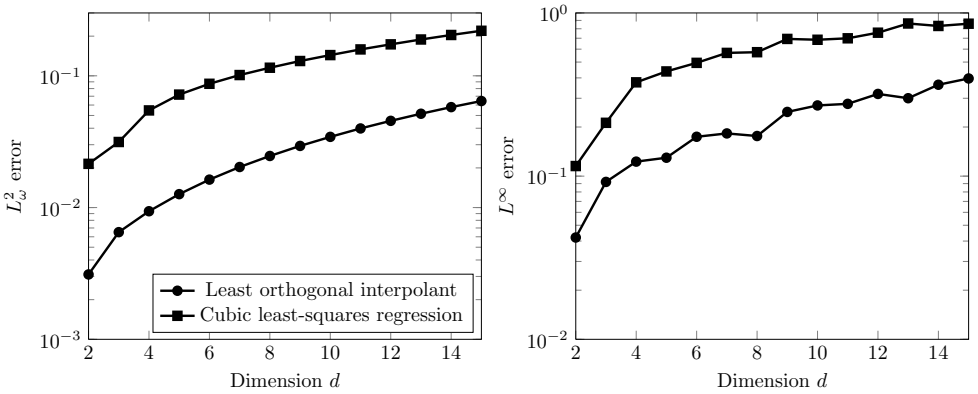

FIG. 4. *Interpolation accuracy for least orthogonal interpolation versus cubic least-squares regression. The L^2_ω (left) and L^∞ (right) errors as measured on 10^6 Monte-Carlo nodes are shown.*

choice of nodes. The concern of our discussion is when the nodal locations are outside our control. Although we could consider a rearrangement of the least orthogonal polynomial basis in an attempt to ameliorate the conditioning, since the basis $\delta(X)_{\downarrow,\omega}$ itself requires computation, we are already likely to be affected by ill-conditioning in the basis determination step itself.

Therefore, we turn our attention to the fact raised in a previous example that the least orthogonal interpolant depends on the choice of Ω and ω (or more accurately on the choice of the orthogonal basis Φ_α). Therefore we investigate the conditioning of the matrices $\mathbf{L}$ and $\mathbf{U}$ that are the output of the basis pursuit. Since the algorithm requires multiplication of the data by $\mathbf{U}^{-1}\mathbf{L}^{-1}$ to determine coefficients in the basis $(h_n)_{\downarrow,\omega}$, there is interest in exploring the stability of this operation.

In the computational sense, if we are interpolating Dirac functionals in high dimensions, one of the worst situations happens when all the points are collinear—in this case we can guarantee that a new degree will be added for each point. This forces us to use high-order polynomials to perform the interpolation, which can lead to ill-conditioning. We compare the conditioning of our method to the conditioning of the operators in the traditional least interpolant method, which by Lemma 4.2 uses weighted monomials as basis functions.

All examples here use the same collinear, equispaced nodal set as in the previous example. The orientation of the nodal set is generalized to higher dimensions $d > 2$ by use of the $d-1$ hyperspherical angular coordinates $\{\phi_q\}_{q=1}^{d-1}$. We set $\phi_q = \pi/3$ and define the unit-length coordinate $y \in \mathbb{R}^d$ such that

$$\begin{aligned} y_1 &= \cos(\phi_1), \\ y_2 &= \sin(\phi_1)\cos(\phi_2) \\ &\vdots \\ y_{d-1} &= \sin(\phi_1)\sin(\phi_2)\ldots\sin(\phi_{d-2})\cos(\phi_{d-1}), \\ y_d &= \sin(\phi_1)\sin(\phi_2)\ldots\sin(\phi_{d-2})\sin(\phi_{d-1}). \end{aligned}$$

The points are then distributed in an equispaced manner along the line passing through the origin and y; all the points lie in the domain $\Omega = [-1,1]^d$.

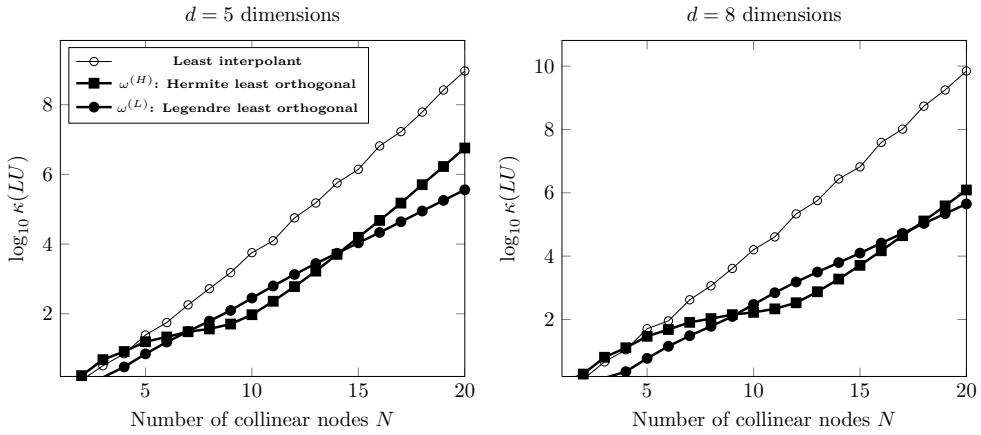

FIG. 5. $\log_{10}$ *of the linear condition numbers of the LU matrix product for different least-type basis constructions in multiple dimensions. Comparison using the monomial basis for the traditional least interpolant ("least"), the Hermite basis from the Gaussian weight* $\omega^{(H)}$*, and the Legendre basis from the uniform weight* $\omega^{(L)}$*.*

We consider three polynomial space constructions, each of which has its own LU decomposition:

- the traditional least interpolant of de Boor and Ron [7],
- the orthogonal least interpolant corresponding to a tensor-product Hermite polynomial basis with $\omega^{(H)} \propto \exp\left(-\|x\|^2\right)$,
- the orthogonal least interpolant corresponding to a tensor-product Legendre polynomial basis with uniform weight $\omega^{(L)} \propto 1$.

Though mathematically equivalent to the original least interpolation, the Hermite case of our least orthogonal interpolation allows extra numerical flexibility for relatively better conditioning. For the Hermite interpolant, we can form the one-dimensional N-point interpolation problem with corresponding Vandermonde matrix entries $V_{n,m} = \Phi_m(cx_n)$, where Φ_m is the (orthonormal) degree-m Hermite polynomial and c is a constant. For each N, there is a value of $c(N)$ that is optimal in terms of the condition number of V, and we use this $c(N)$ to scale the Hermite basis elements for all dimensions. No scaling can be performed for the traditional least interpolant, which uses monomial basis elements.

For N points spaced as described above, we compute the LU factorizations for each basis/polynomial space, and then compute the linear condition number for the LU product. The $\log_{10}$ of these numbers are plotted in Figure 5 for $d = 5$ and $d = 8$ dimensions up to $N = 20$. We see that the traditional least interpolant method has a high condition number when compared to the orthogonal alternatives, although we have not attempted to make any optimal scalings. The Hermite polynomials tend to behave better at lower degrees, but the Legendre polynomial basis performs the best for high degrees.

Therefore, we observe in the worst case that the least orthogonal interpolant (for the uniform weight) is much better conditioned than the traditional least interpolant, and we have the added benefit of not having an unspecified scaling parameter to optimize.

6. Summary. In this paper we presented the framework of *least orthogonal interpolation*, which extends the original work on least interpolation by de Boor and

Ron and allows one to construct polynomial interpolants on nodal sets of arbitrary distribution in arbitrary (finite) dimensions. The new framework makes a connection to the probability distribution and utilizes orthogonal polynomials. Hence it is suitable for conducting high-order stochastic collocation simulations and is highly flexible for practical simulations. It also incorporates the original least interpolation by de Boor and Ron as a special case of Hermite polynomials and Gaussian measure. In this work, we presented the mathematical framework of the method and a practical construction algorithm for determining the interpolation basis. While this methodology holds promise for practical computations, it also introduces many unresolved research issues, with the most prominent being the choice of nodes for "good" interpolation. As in the univariate setting, global approximation with polynomials is not appropriate when the underlying function is not smooth. In these cases one may wish to explore locally smooth basis functions such as piecewise polynomials. These issues shall be pursued in future works.

REFERENCES

[1] N. Agarwal and N. R. Aluru, *A domain adaptive stochastic collocation approach for analysis of MEMS under uncertainties*, J. Comput. Phys., 228 (2009), pp. 7662–7688.

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

[3] M. Bieri, *A Sparse Composite Collocation Finite Element Method for Elliptic sPDEs*, Technical report 2009-08, Seminar for Applied Mathematics, ETHZ, Zürich, Switzerland, 2009.

[4] M. Bieri and Ch. Schwab, *Sparse high order FEM for elliptic sPDEs*, Comput. Methods Appl. Mech. Engrg., 198 (2009), pp. 1149–1170.

[5] L. Brutman, *Lebesgue functions for polynomial interpolation—A survey*, Ann. Numer. Math., 4 (1997), pp. 111–127.

[6] P. Constantine, A. Doostan, and G. Iaccarino, *A hybrid collocation/Galerkin scheme for convective heat transfer problems with stochastic boundary conditions*, Internat. J. Numer. Methods Engrg., 80 (2009), pp. 868–880.

[7] C. de Boor and A. Ron, *On multivariate polynomial interpolation*, Constr. Approx., 6 (1990), pp. 287–302.

[8] C. de Boor and A. Ron, *Computational aspects of polynomial interpolation in several variables*, Math. Comp., 58 (1992), pp. 705–727.

[9] A. Doostan and H. Owhadi, *A non-adapted sparse approximation of PDEs with stochastic inputs*, J. Comput. Phys., 230 (2011), pp. 3015–3034.

[10] M. Eldred, *Recent advances in non-intrusive polynomial chaos and stochastic collocation methods for uncertainty analysis and design*, in Proceedings of the 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2009, AIAA-2009-2249.

[11] O. G. Ernst, A. Mugler, H. Starkloff, and E. Ullmann, *On the convergence of generalized polynomial chaos expansions*, ESAIM Math. Model. Numer. Anal., 46 (2012), pp. 317–339.

[12] J. Foo, X. Wan, and G. E. Karniadakis, *The multi-element probabilistic collocation method (ME-PCM): Error analysis and applications*, J. Comput. Phys., 227 (2008), pp. 9572–9595.

[13] B. Ganapathysubramanian and N. Zabaras, *Sparse grid collocation methods for stochastic natural convection problems*, J. Comput. Phys., 225 (2007), pp. 652–685.

[14] M. Gasca and T. Sauer, *Polynomial interpolation in several variables*, Adv. Comput. Math., 12 (2000), pp. 377–410.

[15] R. G. Ghanem and P. Spanos, *Stochastic Finite Elements: A Spectral Approach*, Springer-Verlag, New York, 1991.

[16] X. Ma and N. Zabaras, *An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations*, J. Comput. Phys., 228 (2009), pp. 3084–3113.

[17] F. Nobile, R. Tempone, and C. G. Webster, *A sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM J. Numer. Anal., 46 (2008), pp. 2309–2345.

[18] A. H. Stroud, *Numerical integration formulas of degree two*, Math. Comp., 14 (1960), pp. 21–26.

[19] D. Xiu, *Efficient collocational approach for parametric uncertainty analysis*, Commun. Comput. Phys., 2 (2007), pp. 293–309.
[20] D. Xiu, *Numerical Methods for Stochastic Computations*, Princeton University Press, Princeton, NJ, 2010.
[21] D. Xiu and J. S. Hesthaven, *High-order collocation methods for differential equations with random inputs*, SIAM J. Sci. Comput., 27 (2005), pp. 1118–1139.
[22] D. Xiu and G. E. Karniadakis, *The Wiener–Askey polynomial chaos for stochastic differential equations*, SIAM J. Sci. Comput., 24 (2002), pp. 619–644.
[23] Y. Xu, *Inzell Lectures on Orthogonal Polynomials*, Nova Science Publishers, New York, 2005.