

A METHOD FOR DIMENSIONALLY ADAPTIVE SPARSE  
TRIGONOMETRIC INTERPOLATION OF PERIODIC FUNCTIONS\*ZACHARY MORROW<sup>†</sup> AND MIROSLAV STOYANOV<sup>‡</sup>

**Abstract.** We present a method for dimensionally adaptive sparse trigonometric interpolation of multidimensional periodic functions belonging to a smoothness class of finite order. This method targets applications where periodicity must be preserved and the precise anisotropy is not known a priori. To the authors' knowledge, this is the first instance of a dimensionally *adaptive* sparse interpolation algorithm that uses a *trigonometric interpolation* basis. The motivating application behind this work is the adaptive approximation of a multi-input model for a molecular potential energy surface (PES) where each input represents an angle of rotation. Our method is based on an anisotropic quasi-optimal estimate for the decay rate of the Fourier coefficients of the model; a least-squares fit to the coefficients of the interpolant is used to estimate the anisotropy. Thus, our adaptive approximation strategy begins with a coarse isotropic interpolant, which is gradually refined using the estimated anisotropic rates. The procedure takes several iterations where ever-more accurate interpolants are used to generate ever-improving anisotropy rates. We present several numerical examples of our algorithm where the adaptive procedure successfully recovers the theoretical “best” convergence rate, including an application to a periodic PES approximation. An open-source implementation of our algorithm resides in the Tasmanian UQ library developed at Oak Ridge National Laboratory.

**Key words.** sparse interpolation, trigonometric interpolation, adaptive refinement, periodicity-preserving approximation

**AMS subject classifications.** 65D05, 65D15, 65T40, 92E10

**DOI.** 10.1137/19M1283483

**1. Introduction.** Consider the approximation of a periodic multidimensional function (i.e., model)  $f : \mathbb{T}^d \rightarrow \mathbb{R}$ , where  $\mathbb{T} = [0, 1]$  represents the unit interval and  $f$  is both differentiable and periodic up to a given finite order. Since the domain in our motivating application is a torus, we chose the letter  $\mathbb{T}$  to help reinforce the notion of periodicity. The order of differentiability and periodicity can vary with each dimension. The main challenge of approximating a computationally expensive model in a multidimensional (multi-input) setting is the rapid growth of required model simulations per dimension  $d$ , a phenomenon known as the *curse of dimensionality* [3].

---

\*Submitted to the journal's Methods and Algorithms for Scientific Computing section August 26, 2019; accepted for publication (in revised form) May 4, 2020; published electronically August 24, 2020.

<https://doi.org/10.1137/19M1283483>

**Funding:** The work of the first author was supported by an NSF Graduate Research Fellowship under grant DGE-1746939 and by an appointment to the Oak Ridge National Laboratory Advanced Short-Term Research Opportunity (ASTRO) Program, sponsored by the U.S. Department of Energy and administered by the Oak Ridge Institute for Science and Education. The work of the second author was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration; by the U.S. Defense Advanced Research Projects Agency, Defense Sciences Office under contract and award HR0011619523 and 1868-A017-15; and by the Laboratory Directed Research and Development program at the Oak Ridge National Laboratory, which is operated by UT-Battelle, LLC, for the U.S. Department of Energy under 674 contract DE-AC05-00OR22725. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

<sup>†</sup>Department of Mathematics, North Carolina State University, Raleigh, NC 27695 (zbmorrow@ncsu.edu).

<sup>‡</sup>Computational and Applied Mathematics Group, Oak Ridge National Laboratory, Oak Ridge, TN 37830 (stoyanovmk@ornl.gov).

Various techniques exist to mitigate, or in rare cases eliminate, the curse of dimensionality. Global and derivative-based sensitivity analysis enables the identification of noninfluential parameters, thereby reducing the effective dimensionality of the problem to include only inputs and directions that contribute toward the model output variability [18, 56]. Other methods seek to reduce the complexity of the target function  $f$  by approximating it with functions that are in some sense “simpler,” e.g., [4, 10, 11, 22, 24, 39, 45, 57]. This could be done, for instance, by projection onto or interpolation within a polynomial or trigonometric function space, both of which use samples, i.e., the values of the target function for a set of independent inputs. Sampling methods are attractive because those can be wrapped around existing third-party or black-box models in a nonintrusive way, i.e., without the need to modify the original solver.

Let  $\{\phi_\nu\}_{\nu \in \mathbb{N}^d}$  be an orthonormal basis for the Hilbert space where  $f$  resides, e.g.,  $L^2(\mathbb{T}^d)$ , and let  $\Lambda \subset \mathbb{N}^d$  be finite.<sup>1</sup> It is well known that orthogonal projection of  $f$  onto  $S = \text{span}\{\phi_\nu\}_{\nu \in \Lambda}$  yields the optimal  $L^2$  error [29, p. 352], that is,

$$(1.1) \quad c_\nu = \langle f, \phi_\nu \rangle_{L^2(\mathbb{T}^d)} \quad \Rightarrow \quad \left\| f - \sum_{\nu \in \Lambda} c_\nu \phi_\nu \right\|_{L^2(\mathbb{T}^d)} = \min_{g \in S} \|f - g\|_{L^2(\mathbb{T}^d)}.$$

Here,  $\{c_\nu\}_{\nu \in \Lambda}$  are the optimal expansion coefficients, and  $\langle \cdot, \cdot \rangle$  and  $\|\cdot\|$  denote the  $L^2(\mathbb{T}^d)$  inner product and norm, respectively. In general, the integral coefficients in (1.1) must be evaluated numerically, e.g., with a multidimensional numerical quadrature. Thus, projection methods often come at a high computational cost due to the large number of function samples necessary to approximate  $c_\nu$  to a sufficient accuracy which could far exceed the number of basis functions [2, 57, 58]. In contrast, interpolation methods require a single sample per basis function, although the resulting approximation is not Hilbert-optimal. The interpolation error is bounded by the best projection error multiplied by a penalty term called the *Lebesgue constant*, but the degradation in accuracy is usually offset by the reduction in computational cost. In particular, sparse interpolation methods [50], which are a focus of this paper, often have a better overall convergence rate with respect to the number of samples [57].

Aiming to further improve the convergence rate of sparse-grid interpolation, many methods gauge the approximation error in order to determine the most important directions and spatial locations in which to sample next. Such procedures are known as *adaptive refinement*, and the overall goal is to select the samples that would result in the fastest convergence rate. Bungartz and Griebel formulated this procedure as a knapsack problem in which they maximize the added accuracy subject to cost constraints at each refinement iteration until the interpolation error reaches a desired accuracy [4]. The greedy construction of an approximate knapsack set has demonstrated good performance in many applications; however, the exact solution to the knapsack problem remain intractable and the greedy approach is susceptible to stagnation and premature termination due to nonmonotonic behavior of  $c_\nu$  when the grid is still fairly coarse (also called the *preasymptotic regime*).

Recent developments in quasi-optimal approximation utilize theoretical upper bounds on the decay rates of the expansion coefficients  $c_\nu$  (rather than the exact  $c_\nu$  themselves) in order to refine the approximation iteratively. For a fixed number of terms  $M$  in the expansion, such approaches often result in a tighter error bound. Much of the previous work on quasi-optimal approximation is done in the context of

---

<sup>1</sup>This paper adopts the convention  $\mathbb{N} = \{0, 1, 2, \dots\}$ .

Legendre expansions of holomorphic functions. Authors have deployed quasi-optimal approximation in the context of projection [59] and sparse-grid interpolation [37, 57].

In contrast to previous work, this paper considers periodic functions, i.e.,  $f \in L^2(\mathbb{T}^d)$ , such that, for each  $1 \leq k \leq d$ ,

$$\left\| \frac{\partial^m f}{\partial x_k^m} \right\|_{L^2(\mathbb{T}^d)} < \infty, \quad \frac{\partial^m f(0)}{\partial x_k^m} = \frac{\partial^m f(1)}{\partial x_k^m} \quad \forall m \in \{0, 1, \dots, n_k\},$$

where  $n_k$  is the order of periodicity in dimension  $k$ . The periodic boundary conditions lead to the natural choice of basis of trigonometric polynomials [21]. In this paper, we present a multidimensional sparse-grid trigonometric-basis adaptive interpolation technique for periodic functions with different degrees of smoothness in each direction. The smoothness affects the convergence rate, so we will use anisotropic grids with rates of anisotropy estimated using a least-squares fitting of the Fourier coefficients of the interpolant, which is similar to previous work in the context of total-degree, polynomial-based interpolation for holomorphic functions [57]. To the authors' knowledge, this is the first instance of a method for adaptive sparse-grid interpolation with a trigonometric basis.

The rest of the paper is organized as follows. In section 2, we derive the trigonometric quasi-optimal approximation space using theoretical bounds on the decay rates of the Fourier coefficients. In section 3, we describe anisotropic sparse trigonometric interpolation and present our adaptive refinement algorithm. In section 4, we provide several numerical examples using both simple polynomials with known degrees of periodicity and the potential energy surface (PES) model. Finally, section 5 offers a brief summary of our results.

**2. Quasi-optimal function space.** First, we consider the space of multidimensional periodic functions. Using upper bounds on the Fourier coefficients, we derive the quasi-optimal approximation space in the context of projection. From projection, we proceed to interpolation and derive the quasi-optimal interpolation space. We conclude by discussing how to estimate the anisotropic coefficients of the target function *on-the-fly*.

Let  $H^n(\mathbb{T}) \subset C^n(\mathbb{T})$ , with  $n \geq 0$ , denote the space of  $n$ -times continuously differentiable functions  $f : \mathbb{T} \rightarrow \mathbb{R}$  such that  $f$  has  $n$  periodic derivatives and  $f^{(n+1)}$  is piecewise continuous with only finitely many jump discontinuities. Functions arising from science and engineering applications often satisfy this mild requirement of piecewise differentiability (see, e.g., [32, 35] and section 4). In the  $d$ -dimensional case, for  $\mathbf{n} = (n_1, n_2, \dots, n_d)$ , we define

$$H^\mathbf{n}(\mathbb{T}^d) = H^{n_1}(\mathbb{T}) \otimes \cdots \otimes H^{n_d}(\mathbb{T})$$

so that for any  $f \in H^\mathbf{n}(\mathbb{T}^d)$ ,  $1 \leq k \leq d$ , and  $(x_1, x_2, \dots, x_d) \in \mathbb{T}^d$

$$f(x_1, \dots, x_{k-1}, x, x_{k+1}, \dots, x_d) \in H^{n_k},$$

i.e., restricting  $f$  to a single dimension yields a function in  $H^{n_k}(\mathbb{T})$ . Here, without loss of generality, we take the canonical  $\mathbb{T}^d = [0, 1]^d$  since any arbitrary hypercube  $\Gamma = \bigotimes_{k=1}^d [a_k, b_k]$  can be translated to  $\mathbb{T}^d$  with a simple affine transformation.

The coefficients of the  $L^2$ -Fourier expansion of  $f \in H^\mathbf{n}$  are defined as

$$(2.1) \quad c_{\mathbf{j}}(f) = \int_{\mathbb{T}^d} \exp(-2\pi i \mathbf{j} \cdot \mathbf{x}) f(\mathbf{x}) d\mathbf{x}, \quad \mathbf{j} \in \mathbb{Z}^d,$$

with  $i^2 = -1$  and  $\mathbf{j} \cdot \mathbf{x} = \sum_{k=1}^d j_k x_k$ . In a single-dimensional context, using Theorems 1.6, 4.4, and 4.5 from [25, pp. 4, 25] and trivial reindexing, we obtain

$$(2.2) \quad |c_j(f)| \leq \frac{C(f)}{(1 + |j|)^{n+2}}, \quad j \in \mathbb{Z}, \quad f \in H^n(\mathbb{T}),$$

for some constant  $C(f) > 0$  that depends on  $f$ . Furthermore, since  $f^{(n+1)}$  has jump discontinuities, the bound in (2.2) is asymptotically sharp [14, p. 200]. In a multidimensional context, using the tensor-product structure of the space, we have

$$(2.3) \quad |c_{\mathbf{j}}(f)| \leq \frac{C(f)}{\prod_{k=1}^d (1 + |j_k|)^{n_k+2}}, \quad \mathbf{j} \in \mathbb{Z}^d, \quad f \in H^{\mathbf{n}}(\mathbb{T}^d).$$

Function spaces like this have appeared in the literature as weighted Sobolev and Korobov spaces. In an early work on sparse trigonometric interpolation, Halatschek [17] considered the Korobov space

$$(2.4) \quad E_a^d = \left\{ f \in L^2([0, 1]^d) : \exists C > 0 \text{ s.t. } \forall \mathbf{j} \in \mathbb{Z}^d, |c_{\mathbf{j}}(f)| \leq C \prod_{k=1}^d (1 + |j_k|)^{-a} \right\},$$

where  $a > 1$  is a smoothness parameter. In general,  $a$  may take on any real value greater than one, but integer values have an interpretation in terms of the order of differentiability [43]. One may directly connect  $a \in \{2, 3, \dots\}$  back to  $H^{\mathbf{n}}(\mathbb{T}^d)$  and (2.3) by taking  $\mathbf{n} = (a-2, \dots, a-2)$ . There is also a precedent in the literature for our consideration of anisotropic, rather than isotropic, approximations for functions obeying (2.3). Authors have recently studied tractability questions in anisotropic Korobov spaces [31, 42, 43] in addition to approximation in anisotropic Sobolev and Besov spaces [15, 49]. There is also a long tradition of dimensionally and spatially adaptive refinement within the context of sparse-grid interpolation with a piecewise or Lagrange polynomial basis, e.g., [4, 16, 22, 23, 24, 26, 27, 34, 36, 37, 38, 44, 45, 53, 55, 57]. Our method differs from previous work by using trigonometric basis functions rather than polynomials and by estimating the anisotropy using a fit to a sharp bound on the decay rate of Fourier coefficients (2.3); see subsection 2.3. Furthermore, to the authors' knowledge, this method is the first to use adaptive refinement with trigonometric interpolation on sparse grids.

**2.1. Quasi-optimal projection space.** Consider the projection of  $f \in H^{\mathbf{n}}(\mathbb{T}^d)$  onto a space of real trigonometric polynomials defined by the finite lower complete<sup>2</sup> multi-index set  $\Lambda \subset \mathbb{N}^d$ ,

$$(2.5) \quad \mathbb{P}_{\Lambda} = \text{span} \bigcup_{\nu \in \Lambda} \mathbb{P}_{\nu} = \text{span} \bigcup_{\nu \in \Lambda} \bigotimes_{k=1}^d \mathbb{P}_{\nu_k},$$

where

$$(2.6) \quad \mathbb{P}_n = \text{span}\{\exp(2\pi i j x)\}_{j=-n}^n, \quad i^2 = -1,$$

which is the set of all univariate trigonometric polynomials of degree at most  $n$ . In the context of the sparse grid construction presented in section 3, it is more convenient

---

<sup>2</sup>A set  $\Lambda$  is called *lower* if  $\nu \in \Lambda$  implies  $\{\mathbf{i} \in \mathbb{N}^d : \mathbf{i} \leq \nu\} \subset \Lambda$ , where  $\mathbf{i} \leq \nu$  if and only if  $i_k \leq \nu_k$  for each  $1 \leq k \leq d$ .

to represent the spaces in terms of nonnegative multi-index sets. As a result, we introduce the reindexing

$$(2.7) \quad \phi_\nu(x) = \exp(2\pi i \sigma(\nu) x), \quad \nu \in \mathbb{N},$$

where

$$(2.8) \quad \sigma(\nu) = \begin{cases} -\nu/2, & \nu \text{ even}, \\ (\nu+1)/2, & \nu \text{ odd}, \end{cases}$$

so that

$$\mathbb{P}_n = \text{span}\{\phi_\nu\}_{\nu=0}^{2n}.$$

Note that when using the scalar subscript on  $\mathbb{P}_n$  we are in fact referring to the one-dimensional space defined by the lower-complete set  $\{\nu \in \mathbb{N} : \nu \leq n\}$ .

Let  $f_\Lambda$  be the best approximation to  $f$  in  $\mathbb{P}_\Lambda$  in the  $L^2(\mathbb{T}^d)$  sense, i.e.,  $f_\Lambda$  is defined from the orthogonal decomposition of  $f$  in terms of trigonometric polynomials  $\phi_\nu(\mathbf{x}) \in \mathbb{P}_\Lambda$ ,

$$(2.9) \quad f_\Lambda(\mathbf{x}) = \sum_{\nu \in \Lambda} c_\nu \phi_\nu(\mathbf{x}), \quad \phi_\nu(\mathbf{x}) = \prod_{k=1}^d \phi_{\nu_k}(x_k),$$

which is the familiar Fourier series. It follows that the best  $M$ -term approximation space for projection is associated with the  $M$  largest Fourier coefficients of  $f$ . Therefore, taking the upper bound in (2.3) with  $\alpha_k = n_k + 2$ , we obtain the quasi-optimal space

$$(2.10) \quad \Lambda^\alpha(L) = \{\mathbf{i} \in \mathbb{N}^d : (\mathbf{i} + \mathbf{1})^\alpha \leq L\}, \quad \text{where } \nu^\alpha = \prod_{k=1}^d \nu_k^{\alpha_k},$$

and the parameter  $L \in \mathbb{N}$  discretizes the multi-index space into levels. Note that the structure of the multi-indexes corresponds to a hyperbolic space, which is in contrast to the total-degree space, commonly used for sparse grids [1, 17, 40, 41]:

$$(2.11) \quad \Lambda_{TD}^\alpha(L) = \{\mathbf{i} \in \mathbb{N}^d : \alpha \cdot \mathbf{i} \leq L\},$$

where  $\alpha \cdot \mathbf{i} = \sum_{k=1}^d \alpha_k i_k$ .

**2.2. Quasi-optimal interpolation.** Projection yields the optimal  $L^2$  error, but computing  $c_\nu$  to a sufficient accuracy involves a number of function samples typically much larger than the size of the basis  $\mathbb{P}_{\Lambda(L)}$ . In contrast, interpolation requires exactly the same number of samples as basis functions at the cost of slight reduction of accuracy. Thus, in many practical situations, interpolation results in a better overall convergence rate.

In this subsection, we consider  $f_{\Lambda(L)}$  as an interpolatory (rather than projective) approximation to  $f$ . Specifically,  $f_{\Lambda(L)}$  is obtained from applying the interpolation operator  $I_{\Lambda(L)}$  to  $f$ , where the operator is exact for all functions in  $\mathbb{P}_{\Lambda(L)}$ . Following classical results in interpolation (e.g., [13]), for all  $\phi \in \mathbb{P}_{\Lambda(L)}$  we have

$$\begin{aligned} \|f - f_{\Lambda(L)}\|_\infty &= \|f - \phi + \phi - f_{\Lambda(L)}\|_\infty \\ &= \|f - \phi + I_{\Lambda(L)}[\phi - f]\|_\infty \\ &\leq \|f - \phi\|_\infty + \|I_{\Lambda(L)}\| \|f - \phi\|_\infty, \end{aligned}$$

where

$$(2.12) \quad \|I_{\Lambda(L)}\| = \sup_{\|u\|_\infty=1} \|I_{\Lambda(L)}[u]\|_\infty$$

and  $\|\cdot\|_\infty$  is the  $L^\infty$  norm on  $\mathbb{T}^d$ . Taking the infimum over all  $\phi \in \mathbb{P}_{\Lambda(L)}$ , we obtain

$$(2.13) \quad \|f - f_{\Lambda(L)}\|_\infty \leq (1 + \mathbb{L}_{\Lambda(L)}) \inf_{\phi \in \mathbb{P}_{\Lambda(L)}} \|f - \phi\|_\infty,$$

where  $\mathbb{L}_{\Lambda(L)}$  is norm of the interpolation operator, commonly called the *Lebesgue constant*. Note that in our context the word *constant* is a misnomer since it strongly depends on the approximation set  $\Lambda(L)$  and the specific choice of samples; a detailed discussion is included in section 3. Using (2.9), we observe that

$$(2.14) \quad \inf_{\phi \in \mathbb{P}_{\Lambda(L)}} \|f - \phi\|_\infty \leq \left\| f - \sum_{\nu \in \Lambda(L)} c_\nu \phi_\nu \right\|_\infty,$$

where the coefficients  $c_\nu$  come from the best  $L^2$  approximation. We can chain together (2.13)–(2.14) and note that the only difference from the optimal  $L^2$  approximation comes from using the  $L^\infty$  rather than  $L^2$  norm. We end up deriving the same quasi-optimal space as in (2.10) by heuristically approximating interpolation error as a combination of  $L^2$  projection error and the Lebesgue constant. Other papers have included more parameters in the quasi-optimal interpolation space in order to incorporate the effects of the Lebesgue constant [57], but this is not necessary in our context; see subsection 3.3.

**2.3. Estimating anisotropy.** The specific values of the entries in the anisotropy vector  $\alpha$ , while critical for constructing a quasi-optimal approximation, are seldom known a priori. In this section, we describe a method for estimating the anisotropy from an already constructed approximation  $f_{\Lambda(L)}$  for some lower set  $\Lambda(L)$ . By definition, since  $f_{\Lambda(L)} \in \mathbb{P}_{\Lambda(L)}$

$$f_{\Lambda(L)}(\mathbf{x}) = \sum_{\nu \in \Lambda(L)} \hat{c}_\nu \phi_\nu(\mathbf{x}),$$

where  $\hat{c}_\nu$  are either the projection coefficients from subsection 2.1 or a corresponding set of interpolation coefficients. Oftentimes, the  $\hat{c}_\nu$  are explicitly computed as part of the respective projection or interpolation procedure and hence available at no additional cost. If the estimate in (2.3) bounds the decay of  $\hat{c}_\nu$  sharply, then

$$(2.15) \quad |\hat{c}_\nu| \approx C(f) \prod_{k=1}^d (1 + |\sigma(\nu_k)|)^{-\alpha_k} \quad \forall \nu \in \Lambda(L),$$

where we make use of  $\sigma(\nu)$  in (2.8) to reindex  $j \in \mathbb{Z}^d$  in the estimate (2.3) to  $\nu \in \Lambda(L)$ .

The hyperbolic space defined in (2.10) does not depend on the constant  $C(f)$ ; thus, we focus our attention on estimating  $\alpha$ , and we include the constant only as a regularizing term. If  $\Lambda(L)$  is defined by (2.10) and  $L \geq 2$ , we can replace the approximate sign in (2.15) by an equal sign and solve the system of equations; however, in practice, the estimate is only an asymptotic upper bound and the individual coefficients  $\hat{c}_\nu$  can vary in the preasymptotic regime, which gives an effect similar to noise.

Thus, we take more samples in each direction and solve for the effective rates of decay from an overdetermined set of equations. Taking the log of both sides and changing signs, we obtain

$$(2.16) \quad -\log(|\hat{c}_\nu|) \approx -\log(C(f)) + \boldsymbol{\alpha} \cdot \log(\tilde{\boldsymbol{\sigma}}(\nu) + \mathbf{1}) \quad \forall \nu \in \Lambda(L),$$

where

$$(2.17) \quad \log(\mathbf{i}) = (\log(i_1), \dots, \log(i_d)), \quad \tilde{\boldsymbol{\sigma}}(\nu) = (|\sigma(\nu_1)|, \dots, |\sigma(\nu_d)|).$$

We average out the fluctuations in the coefficient values by taking the least-squares solution, i.e., the solution that minimizes the  $\ell^2$  norm

$$(2.18) \quad \min_{\boldsymbol{\alpha} \in \mathbb{R}^d, \bar{C} \in \mathbb{R}} \frac{1}{2} \sum_{\nu \in \Lambda(L)} (\bar{C} + \boldsymbol{\alpha} \cdot \log(\tilde{\boldsymbol{\sigma}}(\nu) + \mathbf{1}) + \log(|\hat{c}_\nu|))^2,$$

which can be written in a matrix form

$$(2.19) \quad \min_{\mathbf{v} \in \mathbb{R}^{d+1}} \frac{1}{2} \|\mathbf{Av} - \mathbf{b}\|_2^2,$$

where the rows of  $\mathbf{A}$  have the form

$$(1, \log(|\sigma(\nu_1)| + 1), \log(|\sigma(\nu_2)| + 1), \dots, \log(|\sigma(\nu_d)| + 1)),$$

the vector  $\mathbf{b}$  holds the corresponding entries of  $-\log(|\hat{c}_\nu|)$ , and the solution vector is  $\mathbf{v} = (\bar{C}, \alpha_1, \dots, \alpha_d)^T$ . Note that  $\bar{C}$  is not the same as  $\log(C(f))$  from (2.3), since  $C(f)$  defines an upper bound while the curve defined by  $\bar{C}$  and  $\boldsymbol{\alpha}$  has coefficients  $-\log(\hat{c}_\nu)$  both above and below.

The anisotropic weights used in (2.10) are defined as  $\alpha_k = n_k + 2$  where  $n_k$  are integer orders of differentiability, thus it seems natural to restrict  $\alpha_k$  to integer numbers; however, such restriction is unnecessary and we can use any positive real numbers. We see this by observing that the  $L$  parameter is a dummy discretization variable in (2.10) and more obviously in (2.11). If we multiply both  $\boldsymbol{\alpha}$  and  $L$  by the same positive constant, the total degree space (2.11) remains the same; similarly, in the hyperbolic case (2.10), we can multiply  $\boldsymbol{\alpha}$  and raise  $L$  to the same positive power. For the isotropic total degree space with  $\boldsymbol{\alpha} = \mathbf{1}$ ,  $L$  indicates the total degree polynomial order; however, the relation is lost for an isotropic space with  $\boldsymbol{\alpha} \neq \mathbf{1}$  and even more so for an anisotropic space. In the hyperbolic cross section construction, the notion of  $L$  does not relate to polynomial order even for  $\boldsymbol{\alpha} = \mathbf{1}$ . In both cases, the factor that determines the structure of the quasi-optimal space is the ratio between the pairs of coefficients in  $\boldsymbol{\alpha}$ . Since most least-squares solvers operate on real valued numbers, it is convenient to remove any integer restrictions in (2.18)–(2.19) and seek  $\mathbf{v} \in \mathbb{R}^{d+1}$ .

Equation (2.19) admits a unique solution as long as  $\mathbf{A}$  has full column rank, i.e., as long as we have at least two coefficients in each direction to estimate the corresponding decay rate. However, since the coefficients may not decay monotonically and since the accuracy of the solution heavily depends on the condition number of the matrix  $\mathbf{A}$ , the approximation with only two coefficients will not suffice. The estimated  $\boldsymbol{\alpha}$  may be too inaccurate or even yield negative decay rates, which according to (2.10) results in  $\Lambda(L)$  with infinitely many multi-indexes. Nevertheless, we employ the estimate in an adaptive refinement strategy presented in subsection 3.4, and in Remark 3.5 we propose an ad hoc strategy, specific to the refinement procedure, that would allow us to move forward with the adaptive steps even if some of the computed  $\alpha_k$  are negative.

**3. Sparse trigonometric interpolation.** In this section, we describe a sparse-grid approach for constructing an interpolant  $f_{\Lambda(L)} \in \mathbb{P}_{\Lambda(L)}$  based on the values of  $f$  at a set of nodes  $\mathbf{x}_1, \dots, \mathbf{x}_M \in \mathbb{T}^d$ . Here, we make no optimality assumptions about  $\Lambda(L)$ ; we only assume that  $\Lambda(L)$  is a lower set. After discussing the sparse interpolation algorithm, we examine the sparse-grid Lebesgue constant. We conclude by coupling the anisotropy-estimation procedure of subsection 2.3 with a general sparse trigonometric interpolation algorithm in order to produce a dimensionally adaptive interpolant.

**3.1. One-dimensional and fully tensorized rule.** First, we define the one-dimensional  $m$ -point trigonometric interpolation rule. The nodes and (global) basis functions are

$$(3.1) \quad x_j = \frac{j}{m}, \quad \phi_j(x) = \exp(2\pi i \sigma(j) x), \quad j = 0, 1, \dots, m-1.$$

Because approximations using cosines and sines up to mode  $n$  require complex exponentials with powers ranging between  $-n \leq j \leq n$  [52], we use the reindexing  $\sigma(j)$  first introduced in (2.8). To resolve all sine and cosine modes up to  $n$ , it is necessary and sufficient to take  $m = 2n + 1$  points [52]. The interpolation operator is

$$(3.2) \quad \mathcal{U}^m : C^0(\mathbb{T}) \rightarrow \mathbb{P}_{(m-1)/2}, \quad \mathcal{U}^m[f](x) = \sum_{j=0}^{m-1} \hat{c}_j \phi_j(x),$$

where  $\hat{c}_j$  are the interpolation coefficients,  $\mathbb{P}_n$  is defined by (2.6), and  $C^0(\mathbb{T})$  is the set of all real-valued continuous functions on  $\mathbb{T}$ .

*Remark 3.1.* One-dimensional (nonadaptive) trigonometric interpolation with an even number of points has precedents in the literature [9, 52]. Indeed, much work with Fourier transforms on sparse grids uses  $2^l$  points at each level since powers of 2 are highly amenable to fast Fourier transforms [15, 17]. However, for the purposes of interpolation, an even number of points  $m = 2n$  exactly reproduces all modes up to  $n - 1$  due to the missing conjugate exponent for mode  $n$ . Put differently, there is an additional basis function for  $m = 2n$  vs.  $m = 2n - 1$  without a general increase in exactness in terms of both sines and cosines. Therefore, we use an odd number of interpolation nodes, which means  $\mathcal{U}^m$  is exact up to mode  $(m - 1)/2$ .

The interpolation conditions at the nodes  $x_j$

$$\mathcal{U}^m[f](x_j) = f(x_j), \quad j = 0, 1, \dots, m-1,$$

can be imposed by selecting the coefficients as [15, 17, 52]

$$(3.3) \quad \hat{c}_j = \frac{1}{m} \sum_{p=0}^{m-1} f(x_p) [\phi_j(x_p)]^*,$$

where  $[\cdot]^*$  denotes complex conjugation. With some algebra work, it becomes apparent that (3.3) is a normalized and reindexed discrete Fourier transform. Because the target function  $f \in C^0(\mathbb{T})$  is real-valued, symmetry in  $\hat{c}_j$  and  $\phi_j$  makes the interpolant real-valued in exact arithmetic; in implementation when computing (3.2) numerically, we only compute the real part.

We extend the one-dimensional construction to a multidimensional context using tensor products of the points and basis functions expressed in multi-index notation.

Let  $\mathbf{m} = (m_1, m_2, \dots, m_d)$  represent the vector with (potentially) a different number of points in each dimension, and then

$$\mathbf{x}_j = (x_{j_1}, x_{j_2}, \dots, x_{j_d}),$$

$$\phi_j(\mathbf{x}) = \prod_{k=1}^d \phi_{j_k}(x_k) = \exp\left(2\pi i \sum_{k=1}^d \sigma(j_k) \cdot x_k\right).$$

The anisotropic fully tensorized operator becomes

$$\mathcal{U}^{\mathbf{m}} : C^0(\mathbb{T}^d) \rightarrow \bigotimes_{k=1}^d \mathbb{P}_{(m_k-1)/2},$$

$$\mathcal{U}^{\mathbf{m}}[f](\mathbf{x}) = \sum_{\mathbf{j} \leq \mathbf{m}-1} \hat{c}_j \phi_j(\mathbf{x}).$$

As one may expect, the fully tensorized interpolation coefficients are analogous to the one-dimensional case,

$$(3.4) \quad \hat{c}_j = \frac{1}{m_1, \dots, m_d} \sum_{\mathbf{p} \leq \mathbf{m}-1} f(\mathbf{x}_p) [\phi_j(\mathbf{x}_p)]^*,$$

which is a normalized and reindexed  $d$ -dimensional discrete Fourier transform.

The fully tensorized construction is easy to implement, using only one-dimensional nodes and basis functions and employing a suitable algorithm for fast-Fourier transform. However, the resulting approximation belongs to a fully tensorized space which is very far from optimal. First, we observe that the extra point incurred by the even rules in 1 dimension combines with all the “good” points in other dimensions and results in a much higher penalty, e.g., in 6 dimensions the 4-point rule has 4096 points and it covers the same basis as the 3-point rule with only 729 points, thus wasting the majority of the computational effort. Restricting our attention to rules with only odd number of points, we then look at our estimate for the quasi-optimal basis. At  $L = 2$ , in 6 dimensions, the space has 7 exponential powers, which results in 13 basis functions (all powers except 0 require 2 basis functions), and the smallest fully tensorized space including  $L = 2$  hyperbolic space has the aforementioned 729 points. Fully tensorized interpolation is not a feasible approach in a multidimensional context.

**3.2. Sparse-grid interpolation.** Sparse-grid interpolation aims at exploiting the implementational simplicity of fully tensorized rules while alleviating (and sometimes completely avoiding) the restrictions on the basis space. To this end, sparse-grid algorithms employ a family of one-dimensional interpolation rules with a different number of points and basis functions and combine (superimpose) a set of anisotropic full tensors interpolants into a single grid. The set of tensors is chosen so that the combined approximation space includes a desired (quasi-) optimal space with as little extra basis functions as possible.

Starting with the one-dimensional nodes and basis (3.1), we select the family of rules via the node growth  $m(l)$ , i.e.,  $m(l)$  is a strictly increasing function indicating the number of points on level  $l \geq 0$ . See Remark 3.2 for the specific choice used in our examples. Following the approach used in [57], we define the surplus operators

$$\Delta^{m(l)} = \mathcal{U}^{m(l)} - \mathcal{U}^{m(l-1)}, \quad \Delta^{\mathbf{m}(i)} = \bigotimes_{k=1}^d \Delta^{m(i_k)}$$

with the convention  $\Delta^{m(0)} = \mathcal{U}^{m(0)}$ . For any lower set  $\Theta(L)$ , we define the generalized interpolation operator

$$(3.5) \quad I_{\Theta(L)} = \sum_{\mathbf{i} \in \Theta(L)} \Delta^{\mathbf{m}(\mathbf{i})}.$$

Our objective is to relate  $\Theta(L)$  to the quasi-optimal  $\Lambda(L)$ , but first we observe that

$$(3.6) \quad \mathcal{U}^{\mathbf{m}(\mathbf{i})} = \sum_{\mathbf{j} \leq \mathbf{i}} \Delta^{\mathbf{m}(\mathbf{j})}$$

since the sum is telescoping [4, 39, 57]. We wish to express

$$(3.7) \quad I_{\Theta(L)} = \sum_{\mathbf{i} \in \Theta(L)} t_{\mathbf{i}} \mathcal{U}^{\mathbf{m}(\mathbf{i})}$$

for some coefficients  $\{t_{\mathbf{i}}\}_{\mathbf{i} \in \Theta(L)}$ . By substituting (3.6) into (3.7) and equating the coefficients of the  $\Delta^{\mathbf{m}(\mathbf{i})}$  operators, we derive the system

$$(3.8) \quad \sum_{\substack{\mathbf{i} \in \Theta(L) \\ \mathbf{j} \leq \mathbf{i}}} t_{\mathbf{i}} = 1 \quad \forall \mathbf{j} \in \Theta(L).$$

The system (3.8) can be expressed as an upper triangular matrix of zeroes and ones, with a diagonal of all ones, so a unique integer solution  $\{t_{\mathbf{i}}\}_{\mathbf{i} \in \Theta(L)}$  does indeed exist.

The interpolation nodes associated with  $I_{\Theta(L)}$  are the union of the nodes of all tensors  $\mathcal{U}^{\mathbf{m}(\mathbf{i})}$ . If we want to minimize the number of nodes (and the associated expensive simulation of the target model), it is best to reuse the nodes as much as possible, i.e., we want the nodes associated with  $\mathcal{U}^{m(l)}$  to be a subset of the nodes of  $\mathcal{U}^{m(l+1)}$ . It is well known that such nested rules are advantageous for sparse-grid methods, and thus we restrict our attention to only those  $m(l)$  that satisfy the nested property; see Remark 3.2.

Let  $\Theta_m(L)$  denote the multi-indexes of the interpolation nodes for the interpolant defined by  $\Theta(L)$ . Then exploiting the nested structure of the rule gives

$$(3.9) \quad \Theta_m(L) = \bigcup_{\mathbf{i} \in \Theta(L)} \{ \mathbf{j} \in \mathbb{N}^d : \mathbf{j} \leq \mathbf{m}(\mathbf{i}) - \mathbf{1} \},$$

which comes from

$$\{x_{\mathbf{j}}\}_{\mathbf{j} \in \Theta_m(L)} = \bigcup_{\mathbf{i} \in \Theta(L)} \{x_{\mathbf{j}}\}_{\mathbf{j} \leq \mathbf{m}(\mathbf{i}) - \mathbf{1}}.$$

With (3.7)–(3.9), we can explicitly write the sparse trigonometric interpolant as

$$(3.10) \quad I_{\Theta(L)}[f](\mathbf{x}) = \sum_{\mathbf{j} \in \Theta_m(L)} \sum_{\substack{\mathbf{i} \in \Theta(L) \\ \mathbf{j} \leq \mathbf{m}(\mathbf{i}) - \mathbf{1}}} t_{\mathbf{i}} \hat{c}_{\mathbf{j}}^{\mathbf{i}} \phi_{\mathbf{j}}(\mathbf{x}) = \sum_{\mathbf{j} \in \Theta_m(L)} w_{\mathbf{j}} \phi_{\mathbf{j}}(\mathbf{x}).$$

Here,  $\hat{c}_{\mathbf{j}}^{\mathbf{i}}$  and  $\phi_{\mathbf{j}}$  are defined in subsection 3.1, and

$$(3.11) \quad w_{\mathbf{j}} = \sum_{\substack{\mathbf{i} \in \Theta(L) \\ \mathbf{j} \leq \mathbf{m}(\mathbf{i}) - \mathbf{1}}} t_{\mathbf{i}} \hat{c}_{\mathbf{j}}^{\mathbf{i}}, \quad \mathbf{j} \in \Theta_m(L).$$

Since each of the tensor operators exactly reproduces the basis functions, constructing  $I_{\Theta(L)}[\phi_j](\mathbf{x})$  means that the only nonzero coefficient will be the corresponding  $\hat{c}_j^i$  and from (3.8) it follows that the interpolant is exact for all basis functions, i.e., the union of the space of all tensors. Therefore, Theorem 1 in [57] applies; since the exactness of level  $l \in \mathbb{N}$  is  $(m(l) - 1)/2$ , then for an arbitrary lower trigonometric polynomial space  $\mathbb{P}_{\Lambda(L)}$ , we define the optimal sparse grid tensor set by

$$(3.12) \quad \Theta^{opt}(L) = \{\mathbf{i} \in \mathbb{N}^d : (\mathbf{m}(\mathbf{i} - \mathbf{1}) + \mathbf{1})/2 \in \Lambda(L)\}.$$

That is,  $\Theta^{opt}(L)$  is the smallest set of tensors that results in an interpolant which is exact for  $\mathbb{P}_{\Lambda(L)}$ . Depending on the choice of  $m(l)$ , the actual interpolation space may be larger.

*Remark 3.2.* We choose  $m(l) = 3^l$ , which gives us nodes that are both nested and odd, i.e., we avoid the cost of extra basis functions noted in Remark 3.1. The Radix-3 FFT algorithms are slightly less efficient than the Radix-2 and Radix-4 variants; however, from (3.11), we observe that the weight can be precomputed and reused every time we need to compute the value of the interpolant. Thus, the FFT procedure is a one-time effort resulting in a small increase in computational cost, which is far offset by the reduction in interpolation nodes and model simulations. Furthermore, when targeting hyperbolic cross-section interpolation space, e.g., (2.10), exponentially growing  $m(l)$  have a natural advantage. Suppose that  $\Theta(L)$  is chosen as a total degree multi-index space:

$$(3.13) \quad \Theta(L) = \left\{ \mathbf{i} \in \mathbb{N}^d : \sum_{k=1}^d \alpha_k i_k \leq L \right\} \Rightarrow \prod_{k=1}^d (3^{i_k})^{\alpha_k} \leq 3^L.$$

Then from the definition of  $\Theta_m(L)$  in (3.9) and the one-to-one correspondence between nodes and basis functions, we have that the resulting space will include all  $\phi_j$  for  $j$  such that

$$\mathbf{j} + \mathbf{1} \leq \mathbf{m}(\mathbf{i}), \quad \text{i.e.,} \quad j_k + 1 \leq 3^{i_k} \quad \forall 1 \leq k \leq d.$$

Combining the above with (3.13), we get the anisotropic cross-section space:

$$\prod_{k=1}^d (j_k + 1)^{\alpha_k} \leq 3^L.$$

Therefore, using an exponential  $m(l)$  and total degree multi-index selection produces an interpolant in a hyperbolic cross-section space, modulo a constant in the exponent and some rounding in the actual implementation due to noninteger anisotropic rates.

**3.3. Lebesgue constant.** Consider the Lebesgue constant  $\mathbb{L}_n$  for  $(2n+1)$ -point trigonometric interpolation. Well-known estimates for  $\mathbb{L}_n$  exist [9], including quite sharp ones [8, 48]. In terms of the Lebesgue constant, trigonometric interpolation is closely related to polynomial interpolation with the Clenshaw–Curtis and Chebyshev nodes [6, 9]. In [48], Rivlin showed the equality

$$(3.14) \quad \mathbb{L}_n = \frac{2}{\pi} \ln(n) + \beta_n, \quad n \geq 1,$$

where  $\beta_n$  decreases monotonically from  $5/3$  to

$$\frac{2}{\pi} \left( \ln \left( \frac{16}{\pi} \right) + \gamma \right) \approx 1.404,$$

where  $\gamma$  is the Euler–Mascheroni constant. For our family of one-dimensional rules defined by  $m(l) = 3^l$ , the Lebesgue constant grows as a logarithm in the number of points and linearly in level.

In general, no sharp estimates exist for the Lebesgue constant of sparse interpolation with space of exactness  $\mathbb{P}_{\Lambda(L)}$ . However, Lemma 3.1 in [5] yields an upper bound, i.e., when the Lebesgue constant of the one-dimensional family of rules exhibits a polynomial growth, the sparse grids constant grows no faster than the polynomial with one additional power. Let  $\Theta(L)$  be the optimal multi-index set corresponding to  $\Lambda(L)$ . Because the one-dimensional rules obey (3.14), we can bound the Lebesgue constant of the sparse interpolation operator  $I_{\Theta(L)}$  in (3.10) as

$$(3.15) \quad \|I_{\Theta(L)}\| \leq C^d (\#\Theta(L))^2,$$

where  $\#\Theta(L)$  is the number of multi-indexes in  $\Theta(L)$ . As in [5], the operator norm is defined by (2.12).

In [57], Stoyanov and Webster considered many different one-dimensional rules, some of which have Lebesgue constants that grow slowly at first but increase quite rapidly after only a few levels. As a result, the authors included an additional parameter to account for the different “effective” Lebesgue constant in each dimension since some dimensions may have far fewer points than others. However, we do not consider such a correction in our context for the following reasons. First, the rapid increase in Lebesgue constants in [57] was not observed for the Clenshaw–Curtis nodes, which have the same linear-in-level and logarithmic-in-nodes growth as our rule. Second, the variability of  $\beta_n$  in (3.14) is small. Third, by using (3.14) in (2.13) and following the derivation in subsection 2.3, the Lebesgue penalty term in (2.18) is the logarithm of a logarithm, which is negligible in practical situations.

**3.4. Adaptive refinement.** In this subsection, we will link the anisotropy least-squares problem from subsection 2.3 to the sparse trigonometric interpolation algorithm in subsection 3.2. We use (2.18), but with the sparse interpolation coefficients  $w_j$  given by (3.11), which are linear combinations of  $\hat{c}_j^i$  over the constituent tensors of the sparse grid. Also, by (3.1) and (3.3), the sparse discrete Fourier coefficient  $w_j$  corresponds to mode  $\sigma(j)$ . In general, multi-index space is not easily discretizable in the sense of (2.10) after adaptive refinement takes place, so we omit the dependence on  $L$  here. Thus, for a general lower set  $\Lambda$  and the corresponding  $\Theta_m$ , the relevant least-squares problem becomes

$$(3.16) \quad \min_{\alpha \in \mathbb{R}^d, \bar{C} \in \mathbb{R}} \frac{1}{2} \sum_{j \in \Theta_m} (\bar{C} + \alpha \cdot \log(\tilde{\sigma}(j) + 1) + \log(|w_j|))^2,$$

where  $\Theta_m$  is defined by (3.9) and  $\tilde{\sigma}(j)$  by (2.17).

The motivation for the least-squares fitting (2.18) can be demonstrated in the following one-dimensional example:

$$(3.17) \quad f(x) = x \sin(\pi x) + x \sin(5\pi x), \quad x \in [-1, 1],$$

where  $f \in H^0(\mathbb{T})$ . In Figure 3.1, we show the computed continuous and discrete Fourier coefficients, the theoretical decay rate according to (2.3), and the decay rate estimate coming from (2.18). We observe that on each level, for the largest indexes in  $\Lambda$ , the discrete Fourier coefficients differ systematically from the continuous Fourier coefficients. This can be explained by observing that the discrete Fourier transform (3.3)

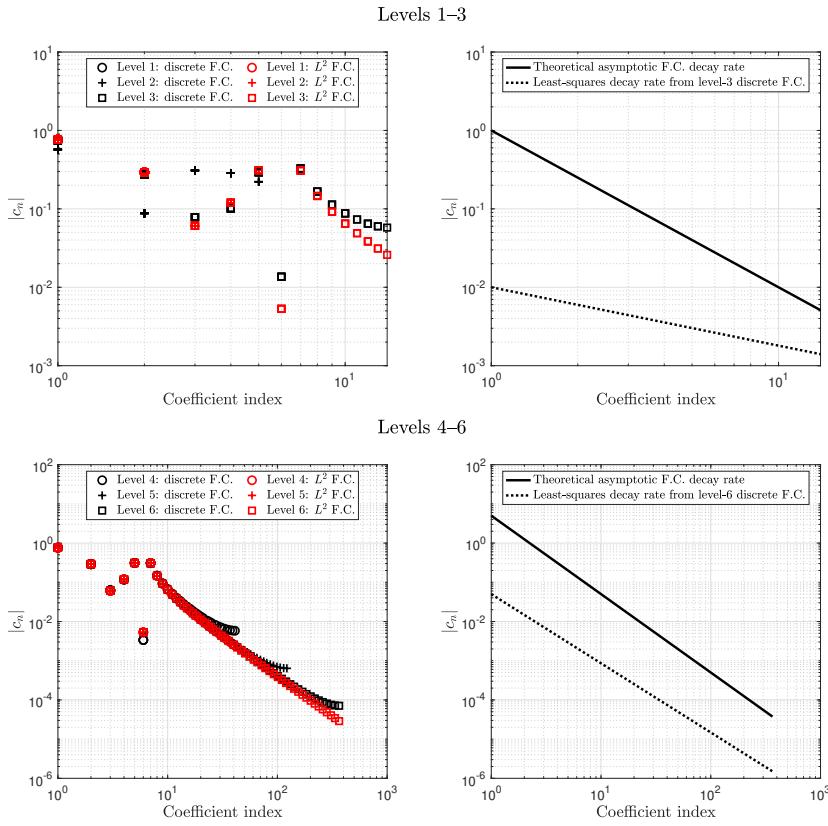


FIG. 3.1. Left column: discrete and  $L^2$  Fourier coefficients (F.C.) of (3.17) on one-dimensional grids of various sizes. Right column: decay rates from (2.2) and (3.16).

is a left-hand Riemann sum discretization of (2.1); therefore, the largest indexes in  $\Lambda$  correspond to the highest frequencies, and the discretization is not able to resolve those to the same degree of accuracy. A refinement criterion could be based on the surplus or the correction introduced by the high frequencies, e.g., similar to the greedy knapsack problem [4], but such refinement would be guided by the least accurate coefficients. This phenomenon is not present in the methods using hierarchical Lagrange approximation where adding more indexes to  $\Lambda$  would not alter the current set of polynomial coefficients. Furthermore, there are unpredictable fluctuations in the preasymptotic low-frequencies. Since the breaking point between the two regimes is unknown, we use the least-squares approach defined in (3.16) to incorporate *all* coefficients and balance out the “noise-like” effects.

We show pseudocode for our algorithm in Algorithm 3.1. Importantly, since the solution of the least-squares problem (3.16) is heavily dependent on  $\Theta_m$ , then one should choose  $\Lambda_0$  so that it contains enough points to compute an initial anisotropy estimate that is reliable for each direction. Thus, we select isotropic  $\Lambda_0$  and in practical application, the choice is usually guided by considerations regarding the minimum number of samples needed to saturate the computational resources, e.g., the number of computing nodes (see Remarks 3.3 and 3.5). After we compute approximate anisotropic coefficients, we want the set  $\Lambda^\alpha(L)$  in (2.10) to be invariant when  $\alpha$  is multiplied by a constant factor; for instance,  $\Lambda^{(1,1)}(L)$  should be equal to  $\Lambda^{(2,2)}(L)$  and  $\Lambda^{(1,2)}(L)$

equal to  $\Lambda^{(2,4)}(L)$ . Additionally, we want  $\Lambda(L)$  to grow conservatively, not suddenly or dramatically, when incrementing  $L$ . Thus, we normalize  $\alpha$  by dividing by the smallest element before constructing  $\Lambda^\alpha(L)$ .

---

**Algorithm 3.1** Adaptive refinement algorithm.

---

**Input:**  $L_0 \geq 2$ ,  $f \in H^{\mathbf{n}}(\mathbb{T}^d)$ ,  $m(l)$  defined in Remark 3.1

**Output:** quasi-optimal  $\Lambda$

- 1:  $n \leftarrow 0$
  - 2: Start with isotropic  $\Lambda = \Lambda^{\mathbf{1}}(L_0)$  from (2.10); define  $\Theta$  according to (3.12)
  - 3: Compute the samples of  $f$  and load the values into the grid
  - 4: **while** num\_samples < budget **do**
  - 5:     Solve (3.16) for  $\hat{\alpha}$ ; set  $\hat{\alpha} \leftarrow \hat{\alpha}/(\min_k \hat{\alpha}_k)$  and apply Remark 3.5
  - 6:     Find  $L_{n+1}$  such that  $\Lambda^{\hat{\alpha}}(L_{n+1}) \not\subseteq \Lambda$ ; define  $\Theta^{\hat{\alpha}}(L_{n+1})$  by (3.12)
  - 7:      $\Lambda \leftarrow \Lambda \cup \Lambda^{\hat{\alpha}}(L_{n+1})$ ;  $\Theta \leftarrow \Theta \cup \Theta^{\hat{\alpha}}(L_{n+1})$ ;  $n \leftarrow n + 1$
  - 8:     Compute the samples of  $f$  at the new points
  - 9: **end while**
- 

*Remark 3.3.* One may modify line 6 of Algorithm 3.1 so that the number of new points is large enough to exploit parallel computations of the sample of  $f$ . When the model is sufficiently complex, parallelism must be exploited to make the problem feasible. A possible drawback is that, if the initial grid is too coarse, Algorithm 3.1 may add a large number of unnecessary nodes due to an unreliable initial anisotropy estimate.

*Remark 3.4.* The quasi-optimal approach aims to construct the best  $M$ -term approximation for some  $M$  given beforehand. For this reason, the termination criterion in line 4 relies on reaching some predetermined computational budget. Alternatively, if computing resources are sufficiently abundant, one could terminate upon reaching a desired error tolerance.

*Remark 3.5.* Given a black-box model, it is not feasible to determine a priori the appropriate size of  $\Lambda(L_0)$  that would yield a stable initial estimate of the anisotropic coefficients. However, the weights are only used to guide the refinement process. Thus, if we encounter a negative weight  $\alpha_k \leq 0$  for some direction  $k$ , we can simply replace that weight with the smallest positive one, which will force the refinement to put additional points in direction  $k$ , which in turn will improve the estimate in the following iterations. If all weights are negative, then we continue the refinement using isotropic weights  $\alpha = \mathbf{1}$ . The correction strategy will allow us to work past negative weights, but it is still possible for a coarse grid to yield positive yet incorrect weight that would deteriorate the convergence. However, the theoretical estimates are only asymptotic and in our numerical examples we observe the opposite behavior, namely, that the preasymptotic weights improve the initial error compared to the optimal analytic weights, e.g., in Figure 4.2. Therefore, in our examples we use isotropic initial  $\Lambda(L_0)$  with  $L_0 = 3$ , which is one more than the absolute minimum.

**4. Numerical results.** We include several examples in this section to illustrate the performance of Algorithm 3.1. We will apply our algorithm to purpose-built periodic polynomials of known anisotropy and then to the chemistry problem that motivated this work. These simulations use the open-source Tasmanian package developed at Oak Ridge National Laboratory [54], which implements Algorithm 3.1 for sparse trigonometric interpolation.

First, to obtain a theoretical convergence rate for our interpolation algorithm, let  $f \in H^n(\mathbb{T})$ . Using a theorem of [46, Jackson's theorem], we can bound the infimum term in (2.13) by

$$\inf_{T \in \mathbb{P}_\Lambda} \|f - T\|_\infty \leq \frac{C(f)}{N^{M+1}},$$

where  $C > 0$  is a constant depending on  $f$ ,  $M = \min_k n_k$ , and  $N = \#\Theta_m^{\text{opt}}$  is the number of nodes. Then, using (3.15) and heuristically approximating  $\#\Theta_m^{\text{opt}}$  as  $\log(N)$  in light of Remark 3.2,

$$(4.1) \quad \|f - I_\Theta[f]\|_\infty \leq O(\log^2(N)/N^{M+1})$$

for  $N$  sufficiently large.

*Remark 4.1* (alternative function space). As noted in subsection 2.2, much early work on sparse grids sought to approximate function spaces of some total degree (2.11). In Fourier interpolation, the total-degree space is suitable for target functions  $f$  having a holomorphic extension in component  $k$  within a polyellipse of radius  $\alpha_k$  around the real axis, for each  $1 \leq k \leq d$ . To see why, suppose  $f$  is a function satisfying the previous analyticity assumptions. From, e.g., [25, p. 27], the Fourier coefficients obey the asymptotically sharp estimate

$$(4.2) \quad |c_\nu(f)| \leq C(f) \exp(-\alpha \cdot \tilde{\sigma}(\nu)), \quad \nu \in \mathbb{N}^d,$$

where  $\tilde{\sigma}(\nu)$  is defined by (2.17). By taking the negative logarithm of the right-hand side of (4.2) and ignoring the constant, we obtain the total-degree space (2.11). For functions of this type, the least-squares problem (3.16) in subsection 3.4 becomes

$$(4.3) \quad \min_{\alpha \in \mathbb{R}^d, \bar{C} \in \mathbb{R}} \frac{1}{2} \sum_{j \in \Theta_m} (\bar{C} + \alpha \cdot \tilde{\sigma}(j) + \log(|w_j|))^2.$$

Our numerical examples will include a modification of Algorithm 3.1 that uses the total-degree space (2.11) and the least-squares problem (4.3).

**4.1. Periodic polynomials.** We manufacture some multidimensional target functions that are engineered to have a certain order of differentiability and periodicity. We define the univariate functions  $g_i : [-1, 1] \rightarrow \mathbb{R}$  as

$$\begin{aligned} g_1(x) &= x^3 - x, \\ g_2(x) &= \frac{x^4}{4} - \frac{x^2}{2}, \\ g_3(x) &= \frac{x^5}{20} - \frac{x^3}{6} + \frac{7x}{60}, \\ g_4(x) &= \frac{x^6}{120} - \frac{x^4}{24} + \frac{7x^2}{120}, \\ g_5(x) &= \frac{x^7}{840} - \frac{x^5}{120} + \frac{7x^3}{360} - \frac{31x}{2520}, \end{aligned}$$

which we have derived by starting with  $g_1(x)$  and integrating repeatedly and choosing the constant to preserve periodicity. By construction,  $g_k \in H^k([-1, 1])$ , where we translate  $[-1, 1]$  to  $[0, 1]$  using a linear transformation and note that the  $k + 1$ th derivative is discontinuous across the periodic boundary. For  $1 \leq i \leq 5$ , we normalize

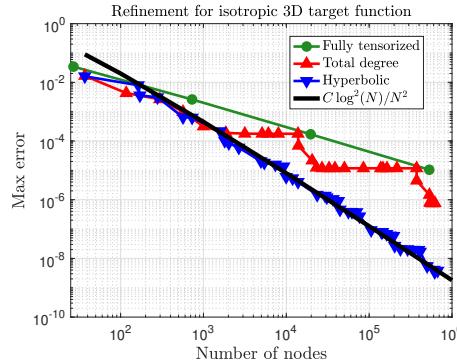


FIG. 4.1. Isotropic refinement for trigonometric interpolation of  $f_{(1,1,1)}(\mathbf{x}) = \prod_{k=1}^3 h_1(x_k)$  with different choices of  $\Lambda^\alpha(L)$ . Here,  $\alpha = \mathbf{1}$  and refinement occurs solely by incrementing  $L$ . As expected, the hyperbolic cross-section refinement converges at the expected rate and outperforms the total-degree and fully tensorized methods.

in the sup-norm by taking  $h_i = g_i/\|g_i\|_{L^\infty([-1,1])}$ . Thus, the multivariate target functions are

$$(4.4) \quad f_i(\mathbf{x}) = \prod_{k=1}^d h_{i_k}(x_k), \quad 1 \leq i \leq 5.$$

The domain of interpolation for (4.4) is  $\Gamma = [-1, 1]^d$ . The Fourier coefficients obey the estimate (2.3), so a hyperbolic function space like (2.10) is appropriate, as Figure 4.1 demonstrates. We calculate the error by drawing 2000 validation points  $\mathbf{x}_j \sim \mathcal{U}(\Gamma)$ , where  $\mathcal{U}(\Gamma)$  is the uniform distribution on  $\Gamma$ , with

$$\text{error} = \max_{1 \leq j \leq 2000} |f(\mathbf{x}_j) - I_\Theta[f](\mathbf{x}_j)|.$$

In the isotropic example, the initial grids have approximately the same number of nodes, and we refine up to a maximum of 700000 nodes.

We now consider target functions with various numbers of inputs and anisotropy. The initial grid for each refinement strategy has approximately the same number of nodes, and we refine up to a maximum of 200000 nodes.

Next we consider an anisotropic example. In Figure 4.2, we compare different anisotropic grids, and we use the six-dimensional target function

$$(4.5) \quad \tilde{f}(\mathbf{x}) = h_1(x_1)h_5(x_4) + h_2(x_2)h_5(x_5) + h_3(x_3)h_5(x_6).$$

Since  $h_k \in H^k(\Gamma)$ , then by (2.3) and (2.10), we know the anisotropy of  $\tilde{f}$  beforehand:

$$\alpha = (1, 2, 3, 5, 5, 5) + \mathbf{2} = (3, 4, 5, 7, 7, 7).$$

The line in Figure 4.2 labeled ‘‘Analytical hyperbolic’’ uses the known anisotropy  $\alpha$ , while the adaptive strategies solve the relevant least-squares problem for  $\hat{\alpha}$  at each refinement iteration. In terms of convergence behavior, all strategies with a hyperbolic cross-section space outperform the total-degree space of Remark 4.1. Additionally, the adaptive algorithms based on solving the least-squares problem (3.16) converge at a similar rate as using the known target space  $\Lambda^\alpha(L)$  directly. Both the adaptive and

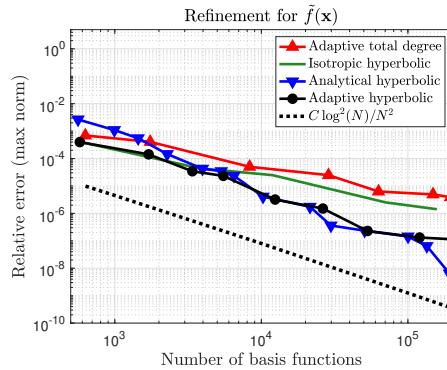


FIG. 4.2. Convergence results for (4.5). The adaptive hyperbolic cross section methods match the convergence rate of the analytic anisotropy, but without using any prior knowledge.

TABLE 4.1

Anisotropy ratios for two-dimensional product functions at the end of refinement. Column 2 uses Algorithm 3.1 and column 3 uses Remark 4.1.

$f_i$	Final $\hat{\alpha}_1/\hat{\alpha}_2$ (hyperbolic)	Final $\hat{\alpha}_1/\hat{\alpha}_2$ (TD)	True $\alpha_1/\alpha_2$
(1, 2)	0.72	0.73	0.75
(1, 3)	0.61	0.61	0.60
(1, 4)	0.49	0.49	0.50
(1, 5)	0.45	0.45	0.43
(2, 3)	0.84	0.84	0.80
(2, 4)	0.68	0.68	0.67
(2, 5)	0.62	0.62	0.57
(3, 4)	0.81	0.81	0.83
(3, 5)	0.74	0.74	0.71
(4, 5)	0.91	0.91	0.86

analytical anisotropic strategies converge at approximately the rate given in (4.1). This shows that Algorithm 3.1 is well suited to handle periodic models where the anisotropy is not known *a priori*.

At the end of refinement, we obtain the following anisotropy estimates (normalized so that  $\hat{\alpha}_1 = \alpha_1 = 3$ ):

$$\begin{aligned}\hat{\alpha}_{hyp} &= (3.00, 3.53, 4.35, 5.58, 5.70, 5.73), \\ \hat{\alpha}_{TD} &= (3.00, 3.63, 4.51, 6.11, 5.73, 5.40).\end{aligned}$$

In Table 4.1, we show the anisotropy ratios at the end of adaptive refinement for two-dimensional product polynomials of the form (4.4). We compute the true anisotropy ratio for  $f_i$  by recalling  $\alpha_k = i_k + 2$ . Both Algorithm 3.1 and the modifications in Remark 4.1 are reasonably able to detect the relative anisotropy of the target function.

**4.2. Particle in a two-dimensional box.** We construct a two-dimensional particle in a box (PIB) system. This is a staple example in textbooks on quantum mechanics, e.g., [32]. Here, the anisotropy arises from different perturbations in the  $x$  and  $y$  directions. As discussed in [32], the Hamiltonian for the unperturbed one-dimensional PIB on the interval  $[0, 1]$  is

$$(4.6) \quad \hat{\mathcal{H}} = -\frac{1}{2} \frac{d^2}{dx^2} + V(x), \quad V(x) = \begin{cases} 0, & x \in [0, 1], \\ \infty & \text{else,} \end{cases}$$

where we have used atomic units and set the particle mass equal to the electron rest mass,  $m_e = 1$ . For  $n = 1, 2, \dots$ , the normalized wavefunctions satisfy

$$\hat{\mathcal{H}}\psi_n = E_n \psi_n \quad \Rightarrow \quad \psi_n(x) = \sqrt{2} \sin(n\pi x), \quad E_n = \frac{1}{2}n^2\pi^2.$$

Inspired by exercises in quantum mechanics textbooks [32, p. 261], we use the potentials

$$f_1(x) = \begin{cases} 15, & x \in [0, 1/4] \cup [3/4, 1], \\ 0 & \text{else,} \end{cases} \quad f_2(y) = 60 \left( y - \frac{1}{2} \right)^2$$

in our two-dimensional perturbed PIB system and treat them as perturbations. Note that the maximum value of each perturbation is less than  $E_2^{(0)}$ , the energy of the unperturbed  $n = 2$  energy level. The two-dimensional Hamiltonian is

$$(4.7) \quad \hat{\mathcal{H}} = -\frac{1}{2}\nabla^2 + V(x) + V(y) + f_1(x) + f_2(y),$$

where  $V$  is given in (4.6). The full two-dimensional wavefunction has the form

$$\Psi_{\mathbf{n}}(x, y) = \psi_{n_1}(x) \psi_{n_2}(y),$$

where  $n_k$  is the quantum number in dimension  $k$ .

We will demonstrate the performance of various refinement strategies on the wavefunction of (4.7) corresponding to  $n_x = n_y = 2$ . To evaluate the target wavefunction, we first decompose (4.7) into the  $x$  and  $y$  parts and apply first-order nondegenerate perturbation theory (see, e.g., [32, p. 233]). The first-order correction to the wavefunction for the  $x$  component is

$$(4.8) \quad \psi_{2,x}^{(1)}(x) = \sum_{n \neq 2} \frac{\int_0^1 \psi_n^{(0)}(u) f_1(u) \psi_2^{(0)}(u) du}{E_2^{(0)} - E_n^{(0)}} \psi_n^{(0)}(x),$$

where  $\psi_n^{(0)}$  and  $E_n^{(0)}$  are the unperturbed wavefunctions and energies corresponding to (4.7). Similarly, for the  $y$  component, we get

$$(4.9) \quad \psi_{2,y}^{(1)}(y) = \sum_{n \neq 2} \frac{\int_0^1 \psi_n^{(0)}(u) f_2(u) \psi_2^{(0)}(u) du}{E_2^{(0)} - E_n^{(0)}} \psi_n^{(0)}(y).$$

Thus, we take the two-dimensional target wavefunction as

$$(4.10) \quad \Psi_{2,2}(x, y) = \left( \psi_2^{(0)}(x) + \psi_{2,x}^{(1)}(x) \right) \left( \psi_2^{(0)}(y) + \psi_{2,y}^{(1)}(y) \right).$$

We evaluate the integral coefficients in (4.8)–(4.9) with Maple and find that the only nonzero coefficients correspond to functions of the form  $\sin(2\pi kx)$ , yielding an a priori anisotropy estimate. The  $L^2$ -Fourier coefficients of  $\Psi_{2,2}(x, y)$  decay like  $O(1/k^3)$  in the  $x$  component and  $O(1/k^5)$  in  $y$ , where  $k$  is the coefficient index. This both justifies the use of approximation space  $\Lambda_{hyp}^{\alpha}$  and gives the prior anisotropy  $\alpha = (3, 5)$ . Computationally, we truncate the series in (4.8)–(4.9) at  $N = 10^4$  terms and use (4.10) as our target function. In practice, though, one would not use Fourier interpolation on a known truncated Fourier series; instead, a more accurate solution technique

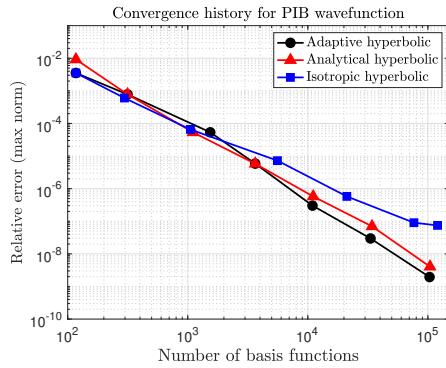
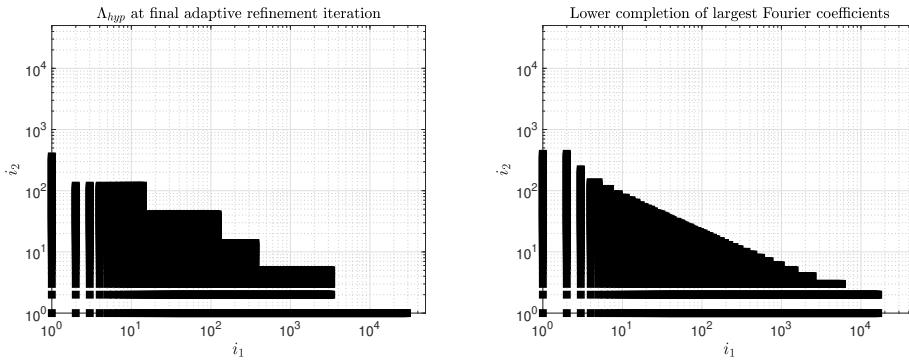


FIG. 4.3. Convergence history of approximating (4.10) with various techniques.

FIG. 4.4.  $\Lambda_{hyp}$  at final iteration of adaptive refinement (left); lower completion of indices for the largest Fourier coefficients (right). Note the logarithmic scaling. Equal vertical and horizontal axes are chosen to display the anisotropy.

would provide the target wavefunction, e.g., [7]. Perturbation theory, however, is straightforward enough to use for the end goal of demonstrating the convergence behavior of our adaptive refinement method.

For sparse interpolation, we use the hyperbolic index set  $\Lambda_{hyp}$  and refine according to three strategies: adaptive (Algorithm 3.1), analytical anisotropy, and isotropic. We show the convergence behavior in Figure 4.3. Similarly to subsection 4.1, adaptive refinement performs as well as analytical anisotropic refinement, but without any prior knowledge. Asymptotically, the errors of the analytical anisotropic and adaptive strategies in Figure 4.3 decay at roughly the same rate and are an order of magnitude better than isotropic refinement. Furthermore, since we know the Fourier coefficients explicitly as a result of (4.8)–(4.9), we may construct the optimal lower approximation space directly from the explicit coefficients. In Figure 4.4 we show  $\Lambda_{hyp}$  at the final iteration of adaptive refinement along with the smallest lower set of size  $\#(\Lambda_{hyp})$  containing the  $N \leq \#(\Lambda_{hyp})$  largest Fourier coefficients. Figure 4.4 shows that adaptive refinement closely resembles the lower set containing the  $N \leq \#(\Lambda_{hyp})$  largest Fourier coefficients, except for some rectangular gaps introduced by the growth rule  $m(l) = 3^l$ . We chose equal vertical and horizontal axes to make the anisotropy clear.

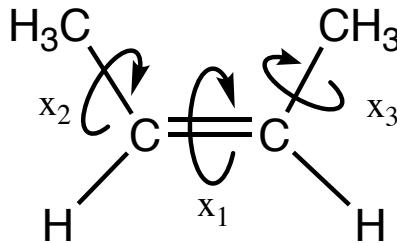


FIG. 4.5. Molecular structure of 2-butene, labeled with rotations of interest.

**4.3. The 2-butene potential energy surface.** Now we consider the motivating application of this paper: the adaptive approximation of a molecule's PES, where the anisotropy  $\boldsymbol{\alpha}$  is not known beforehand. The molecule of interest is 2-butene, whose molecular structure is shown in Figure 4.5.

At the quantum-mechanical level, the energy  $\mathcal{E}_n$  of a molecule with an arrangement of nuclei described by  $\mathbf{q}$  satisfies the Schrödinger equation [32]:

$$(4.11) \quad \hat{\mathcal{H}}(\mathbf{q}) \Psi_n(\mathbf{y}; \mathbf{q}) = \mathcal{E}_n(\mathbf{q}) \Psi_n(\mathbf{y}; \mathbf{q}).$$

Above,  $\hat{\mathcal{H}}$  is the molecular Hamiltonian operator,  $\mathcal{E}_n$  is the energy of electronic state  $n \geq 0$ , and  $\Psi_n$  is the (possibly complex-valued) wavefunction of state  $n$  as a function of electron position  $\mathbf{y}$ . All terms depend parametrically on the nuclear geometry  $\mathbf{q}$ . Physically,  $|\Psi_n(\mathbf{y}; \mathbf{q})|^2$  is the probability distribution function of observing an electron of energy state  $n$  at position  $\mathbf{y}$  in a molecule of geometry  $\mathbf{q}$ . As a function of geometry,  $\mathcal{E}_n(\mathbf{q})$  is the PES corresponding to energy state  $n$ .

For an  $N$ -atom molecule, one may express  $\mathbf{q}$  in Cartesian coordinates as a vector with  $3N$  components or in internal coordinates (bond lengths, bond angles, and torsion angles) as a vector of  $3N - 6$  components. We opt for the latter, which has fewer components and directly enables the varying of geometric features. Only a few of the geometry components, denoted by  $\mathbf{x}$  (the *design variables*), may be needed in a particular study; we optimize over the rest,  $\boldsymbol{\xi}$  (the *remainder variables*):

$$(4.12) \quad E_n(\mathbf{x}) = \min_{\boldsymbol{\xi}} \mathcal{E}_n(\mathbf{x}, \boldsymbol{\xi}).$$

$E_n(\mathbf{x})$  is called the *relaxed PES* for state  $n$ . Chemical intuition and knowledge of the system guides the selection of design variables. Furthermore, for rotational design variables  $\boldsymbol{\theta}$ , a polynomial interpolant does not guarantee periodicity of  $\nabla E_n$  with respect to  $\boldsymbol{\theta}$ , which leads to nonphysical phenomena (e.g., nonconservation of energy). Therefore, a trigonometric interpolation basis is appropriate when  $\mathbf{x}$  contains only bond angles and torsion angles.<sup>3</sup>

As hinted earlier, solving the optimization (4.12) subject to the generalized eigenvalue problem (4.11) is a prohibitively expensive calculation. To trim down computational cost, it is common practice in quantum chemistry to use approximate Hamiltonians and wavefunctions [32]. In our case, we use density functional theory (with the B3LYP hybrid functional) to simplify the Hamiltonian [20, 28, 51], and we approximate the wavefunctions with the 6-311G\* Pople basis set [30]. We use the Gaussian 16 software package [12] to handle the approximation of Hamiltonians and

<sup>3</sup>Bond lengths, in general, are not periodic over an interpolation domain, so approximation by trigonometric polynomials would lead to inaccuracies at the domain boundary [19].

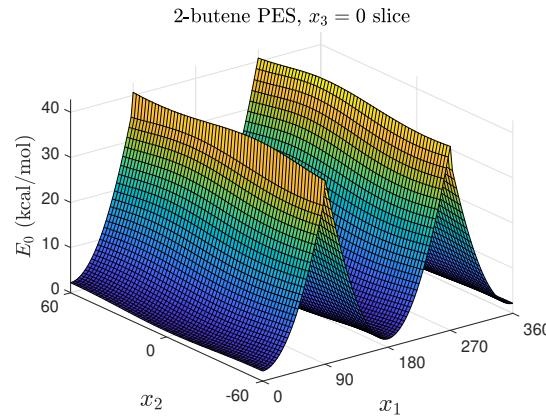


FIG. 4.6. Slice of 2-butene PES for  $x_3 = 0$ .

wavefunctions. By default, Gaussian 16 performs the optimization in (4.12) using a variant of the EDIIS algorithm tuned for molecular geometry optimizations [33].

Previous work constructed a sparse polynomial interpolant of  $E_0(\mathbf{x})$  and  $E_1(\mathbf{x})$  for 2-butene to study the transition from the *cis*- to *trans*- conformation via the first singlet excited state [35]. We use the same design variables from that study, shown in Figure 4.5. The design variable  $x_1$  is more influential on the PES than  $x_2$  and  $x_3$ , but the exact anisotropy is not known in advance.

The domain for our 2-butene ground-state ( $n = 0$ ) PES is  $\Gamma = [0, 360] \times [-60, 60] \times [-60, 60]$ . The coordinates  $x_2$  and  $x_3$  correspond to dihedral rotations of  $\text{CH}_3$ , which have period  $120^\circ$ . We show a slice of the 2-butene PES in Figure 4.6. The ridges at  $x_1 = 90$  and  $x_1 = 270$  indicate a discontinuous first derivative, so we hypothesize that the hyperbolic function space (2.10) and Algorithm 3.1 are appropriate for this problem.

There are numerous sources of noise going into the evaluation of  $E_0(\mathbf{x})$ : density functional theory approximates the Hamiltonian, the 6-311G\* basis set approximates the wavefunction  $\Psi$ , and the optimization (4.12) has internal stopping criteria. Therefore, we do not report the max error of the interpolant  $I_\Theta[E_0](\mathbf{x})$ , which could be heavily skewed by noninterpolatory error. Instead, we give the root-mean-square error (RMSE) over 2000 validation points drawn uniformly over  $\Gamma$ :

$$(4.13) \quad \text{RMSE} = \sqrt{\frac{\sum_{j=1}^{2000} (E_0(\mathbf{x}_j) - I_\Theta[E_0](\mathbf{x}_j))^2}{2000}}, \quad \mathbf{x}_j \sim \mathcal{U}(\Gamma).$$

In our sparse grid constructions, we use  $\Theta_{opt}$  based on the hyperbolic function space  $\Lambda$  in (2.10) as well as the total-degree space (2.11). For each variety of  $\Theta$ , we refine both adaptively (according to Algorithm 3.1 or Remark 4.1) and isotropically (taking  $\boldsymbol{\alpha} = \mathbf{1}$  and incrementing  $L$ ). In all cases, we initialize each grid with 37 nodes. Each function sample takes approximately 30 seconds to evaluate, and occasionally the optimization (4.12) may fail to converge to the correct (or any) local minimum. Due to limitations on available computing time, we refine up to a maximum of only 4000 nodes. If a refinement strategy terminates prior to 4000 nodes, that is because the next increment of  $L$  would result in the number of nodes exceeding 4000. We

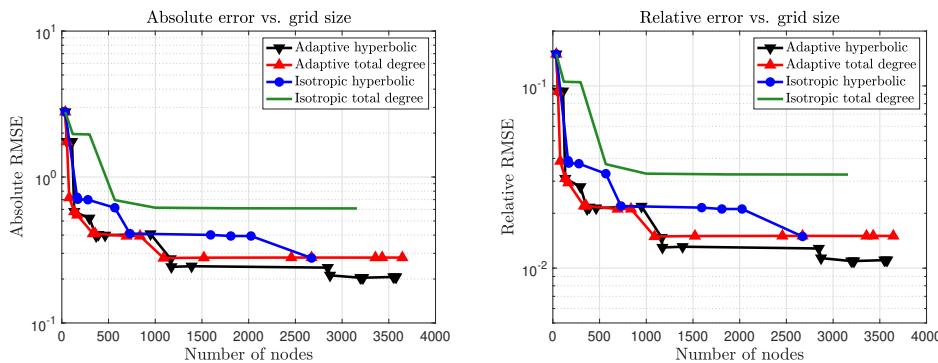


FIG. 4.7. Absolute (left) and relative (right) error results for sparse interpolation of 2-butene ground-state PES.

show the results in Figure 4.7. Following Pople in his 1998 Nobel lecture, we adopt 1 kcal/mol as the threshold of acceptable chemical accuracy for energies [47].

First, we note that the asymptotic absolute RMS errors in Figure 4.7 are consistent with Pople's definition of chemical accuracy for energies (i.e., less than 1 kcal/mol). Furthermore, the limiting relative error for adaptive hyperbolic refinement is approximately 1%. Second, even though the one-dimensional interpolation rule (3.1) grows exponentially, we can still add smaller batches of nodes at each iteration by using (3.12) and Algorithm 3.1, which mitigates the exponential growth of (3.1). Third, in both the adaptive and isotropic cases, the asymptotic error is lower for a hyperbolic cross-section than for a total-degree space.

**5. Conclusion and future work.** In this work, we have presented a quasi-optimal dimensionally adaptive method for sparse interpolation with a trigonometric basis. Our approach targets applications where the surrogate models must be periodicity-preserving and where the anisotropy is not known beforehand. For target functions of known finite smoothness, our algorithm matches the theoretical convergence rate, outperforms the total-degree space asymptotically, and produces a good approximation to the anisotropy. The open-source and freely available Tasmanian package contains a user-friendly implementation.

In the future, we will apply adaptive refinement to more complicated chemical systems. In particular, we aim to approximate potential energy surfaces where the geometry domain includes bond lengths, bond angles, and dihedral angles—a mix of periodic and nonperiodic inputs. To do this, we will apply trigonometric and polynomial interpolation to the periodic and nonperiodic components, respectively.

## REFERENCES

- [1] V. BARTHELMANN, E. NOVAK, AND K. RITTER, *High dimensional polynomial interpolation on sparse grids*, Adv. Comput. Math., 12 (2000), pp. 273–288, <https://doi.org/10.1023/A:1018977404843>.
- [2] J. BECK, F. NOBILE, L. TAMELLINI, AND R. TEMPONE, *Convergence of quasi-optimal stochastic Galerkin methods for a class of PDEs with random coefficients*, Comput. Math. Appl., 67 (2014), pp. 732–751, <https://doi.org/10.1016/j.camwa.2013.03.004>.
- [3] R. BELLMAN, *Adaptive Control Processes: A Guided Tour*, Princeton University Press, Princeton, NJ, 1961.
- [4] H.-J. BUNGARTZ AND M. GRIEBEL, *Sparse grids*, Acta Numer., 13 (2004), pp. 147–269, <https://doi.org/10.1017/S0962492904000182>.

- [5] A. CHKIFA, A. COHEN, AND C. SCHWAB, *High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs*, Found. Comput. Math., 14 (2014), pp. 601–633, <https://doi.org/10.1007/s10208-013-9154-z>.
- [6] C. W. CLENSHAW AND A. R. CURTIS, *A method for numerical integration on an automatic computer*, Numer. Math., 2 (1960), pp. 197–205, <https://doi.org/10.1007/BF01386223>.
- [7] A. CONSORTINI AND B. FRIEDEN, *Quantum-mechanical solution for the simple harmonic oscillator in a box*, Nuov Cim B, 35 (1976), pp. 153–164, <https://doi.org/10.1007/BF02724052>.
- [8] V. K. DZYADYK, S. Y. DZYADYK, AND A. S. PRYPIK, *Asymptotic behavior of Lebesgue constants in trigonometric interpolation*, Ukrainian Math. J., 33 (1981), pp. 553–559, <https://doi.org/10.1007/BF01085428>.
- [9] H. EHЛИCH AND K. ZELLER, *Auswertung der Normen von Interpolationsoperatoren*, Math. Ann., 164 (1966), pp. 105–112, <https://doi.org/10.1007/BF01429047>.
- [10] M. ELDRED AND J. BURKARDT, *Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification*, in Proceedings of AIAA, 2009, pp. 1–20, <https://doi.org/10.2514/6.2009-976>.
- [11] M. ELDRED, C. WEBSTER, AND P. CONSTANTINE, *Evaluation of non-intrusive approaches for Wiener-Askey generalized polynomial chaos*, in Proceedings of AIAA, 2008, ch. 2008-1892, pp. 1–22, <https://doi.org/10.2514/6.2008-1892>.
- [12] M. J. FRISCH, G. W. TRUCKS, H. B. SCHLEGEL, G. E. SCUSERIA, M. A. ROBB, J. R. CHEESEMAN, G. SCALMANI, V. BARONE, G. A. PETERSSON, H. NAKATSUJI, X. LI, M. CARICATO, A. V. MARENICH, J. BLOINO, B. G. JANESKO, R. GOMPERTS, B. MENNUCCI, H. P. HRATCHIAN, J. V. ORTIZ, A. F. IZMAYLOV, J. L. SONNENBERG, D. WILLIAMS-YOUNG, F. DING, F. LIPPARINI, F. EGIDI, J. GOINGS, B. PENG, A. PETRONE, T. HENDERSON, D. RANASINGHE, V. G. ZAKRZEWSKI, J. GAO, N. REGA, G. ZHENG, W. LIANG, M. HADA, M. EHARA, K. TOYOTA, R. FUKUDA, J. HASEGAWA, M. ISHIDA, T. NAKAJIMA, Y. HONDA, O. KITAO, H. NAKAI, T. VREVEN, K. THROSSELL, J. A. MONTGOMERY, JR., J. E. PERALTA, F. OGLIARO, M. J. BEARPARK, J. J. HEYD, E. N. BROTHERS, K. N. KUDIN, V. N. STAROVEROV, T. A. KEITH, R. KOBAYASHI, J. NORMAND, K. RAGHAVACHARI, A. P. RENDELL, J. C. BURANT, S. S. IYENGAR, J. TOMASI, M. COSSI, J. M. MILLAM, M. KLENE, C. ADAMO, R. CAMMI, J. W. OCHTERSKI, R. L. MARTIN, K. MOROKUMA, O. FARKAS, J. B. FORESMAN, AND D. J. FOX, *Gaussian 16 Revision A.03*, Gaussian, Wallingford, CT, 2016.
- [13] W. GAUTSCHI, *Numerical Analysis*, Birkhäuser, Basel, 2012.
- [14] L. GRAFAKOS, *Classical Fourier Analysis*, Springer, New York, 2014.
- [15] M. GRIEBEL AND J. HAMAEKERS, *Fast discrete Fourier transform on generalized sparse grids*, in Proceedings of Sparse Grids and Applications—Munich 2012, J. Garcke and D. Pflüger, eds., Lect. Notes Comput. Sci. Eng. 97, Springer, New York, 2014, pp. 75–107, [https://doi.org/10.1007/978-3-319-04537-5\\_4](https://doi.org/10.1007/978-3-319-04537-5_4).
- [16] M. D. GUNZBURGER, C. G. WEBSTER, AND G. ZHANG, *Stochastic finite element methods for partial differential equations with random input data*, Acta Numer., 23 (2014), pp. 521–650, <https://doi.org/10.1017/S0962492914000075>.
- [17] K. HALLATSCHEK, *Fouriertransformation auf dünnen Gittern mit hierarchischen Basen*, Numer. Math., 63 (1992), pp. 83–97, <https://doi.org/10.1007/BF01385849>.
- [18] J. HART, A. ALEXANDERIAN, AND P. GREMAUD, *Efficient computation of Sobol' indices for stochastic models*, SIAM J. Sci. Comput., 39 (2017), pp. A1514–A1530, <https://doi.org/10.1137/16M106193X>.
- [19] G. HELMBERG, *The Gibbs phenomenon for Fourier interpolation*, J. Approx. Theory, 78 (1994), pp. 41–63, <https://doi.org/10.1006/jath.1994.1059>.
- [20] P. HOHENBERG AND W. KOHN, *Inhomogeneous electron gas*, Phys. Rev., 136 (1964), pp. B864–B871, <https://doi.org/10.1103/PhysRev.136.B864>.
- [21] D. JACKSON, *The Theory of Approximation*, American Mathematical Society, Providence, RI, 1930.
- [22] J. D. JAKEMAN, R. ARCHIBALD, AND D. XIU, *Characterization of discontinuities in high-dimensional stochastic problems on adaptive sparse grids*, J. Comput. Phys., 230 (2011), pp. 3977–3997, <https://doi.org/10.1016/j.jcp.2011.02.022>.
- [23] J. D. JAKEMAN, A. NARAYAN, AND D. XIU, *Minimal multi-element stochastic collocation for uncertainty quantification of discontinuous functions*, J. Comput. Phys., 242 (2013), pp. 790–808, <https://doi.org/10.1016/j.jcp.2013.02.035>.
- [24] J. D. JAKEMAN AND S. G. ROBERTS, *Local and dimension adaptive stochastic collocation for uncertainty quantification*, in Sparse Grids and Applications, Springer, New York, 2012, pp. 181–203, [https://doi.org/10.1007/978-3-642-31703-3\\_9](https://doi.org/10.1007/978-3-642-31703-3_9).
- [25] Y. KATZNELSON, *An Introduction to Harmonic Analysis*, 3rd ed., Cambridge University Press, Cambridge, 2004.

- [26] V. KHAKHUTSKYY AND M. HEGLAND, *Spatially-dimension-adaptive sparse grids for online learning*, in Proceedings of Sparse Grids and Applications—Stuttgart 2014, Springer, New York, 2016, pp. 133–162, [https://doi.org/10.1007/978-3-319-28262-6\\_6](https://doi.org/10.1007/978-3-319-28262-6_6).
- [27] A. KLIMKE AND B. WOHLMUTH, *Algorithm 847: Spinterp: Piecewise multilinear hierarchical sparse grid interpolation in MATLAB*, ACM Trans. Math. Software, 31 (2005), pp. 561–579, <https://doi.org/10.1145/1114268.1114275>.
- [28] W. KOHN AND L. J. SHAM, *Self-consistent equations including exchange and correlation effects*, Phys. Rev., 140 (1965), pp. A1133–A1138, <https://doi.org/10.1103/PhysRev.140.A1133>.
- [29] E. KREYSZIG, *Introductory Functional Analysis with Applications*, Wiley, New York, 1978.
- [30] R. KRISHNAN, J. S. BINKLEY, R. SEEGER, AND J. A. POPLE, *Self-consistent molecular orbital methods. XX. A basis set for correlated wave functions*, J. Chem. Phys., 72 (1980), pp. 650–654, <https://doi.org/10.1063/1.438955>.
- [31] P. KRITZER, F. PILLICHSHAMMER, AND H. WOZNIAKOWSKI, *Multivariate integration of infinitely many times differentiable functions in weighted Korobov spaces*, Math. Comp., 83 (2014), pp. 1189–1206, <https://doi.org/10.1090/S0025-5718-2013-02739-1>.
- [32] I. N. LEVINE, *Quantum Chemistry*, 7th ed., Pearson, London, 2014.
- [33] X. LI AND M. J. FRISCH, *Energy-represented direct inversion in the iterative subspace within a hybrid geometry optimization method*, J. Chem. Theory Comput., 2 (2006), pp. 835–839, <https://doi.org/10.1021/ct050275a>.
- [34] 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, <https://doi.org/10.1016/j.jcp.2009.01.006>.
- [35] J. NANCE, E. JAKUBIKOVA, AND C. T. KELLEY, *Reaction path following with sparse interpolation*, J. Chem. Theory Comput., 10 (2014), pp. 2942–2949, <https://doi.org/10.1021/ct5004669>.
- [36] A. NARAYAN AND J. D. JAKEMAN, *Adaptive Leja sparse grid constructions for stochastic collocation and high-dimensional approximation*, SIAM J. Sci. Comput., 36 (2014), pp. A2952–A2983, <https://doi.org/10.1137/140966368>.
- [37] F. NOBILE, L. TAMELLINI, AND R. TEMPONE, *Convergence of quasi-optimal sparse-grid approximation of Hilbert-space-valued functions: Application to random elliptic PDEs*, Numer. Math., 134 (2016), pp. 343–388, <https://doi.org/10.1007/s00211-015-0773-y>.
- [38] F. NOBILE, R. TEMPONE, AND C. G. WEBSTER, *An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM J. Numer. Anal., 46 (2008), pp. 2411–2442, <https://doi.org/10.1137/070680540>.
- [39] 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, <https://doi.org/10.1137/060663660>.
- [40] E. NOVAK AND K. RITTER, *High dimensional integration of smooth functions over cubes*, Numer. Math., 75 (1996), pp. 79–97, <https://doi.org/10.1007/s002110050231>.
- [41] E. NOVAK AND K. RITTER, *Simple cubature formulas with high polynomial exactness*, Constr. Approx., 15 (1999), pp. 499–522, <https://doi.org/10.1007/s003659900119>.
- [42] E. NOVAK AND H. WOZNIAKOWSKI, *Tractability of Multivariate Problems*, European Mathematical Society, Zürich, Switzerland, 2008.
- [43] A. PAPAGEORGIOU AND H. WOZNIAKOWSKI, *Tractability through increasing smoothness*, J. Complexity, 26 (2010), pp. 409–421, <https://doi.org/10.1016/j.jco.2009.12.004>.
- [44] D. PFLÜGER, *Spatially adaptive refinement*, in Sparse Grids and Applications, Springer, New York, 2012, pp. 243–262, [https://doi.org/10.1007/978-3-642-31703-3\\_12](https://doi.org/10.1007/978-3-642-31703-3_12).
- [45] D. PFLÜGER, B. PEHERSTORFER, AND H.-J. BUNGARTZ, *Spatially adaptive sparse grids for high-dimensional data-driven problems*, J. Complexity, 26 (2010), pp. 508–522, <https://doi.org/10.1016/j.jco.2010.04.001>.
- [46] A. PINKUS, *Negative theorems in approximation theory*, Amer. Math. Monthly, 110 (2003), pp. 900–911, <https://doi.org/10.1080/00029890.2003.11920030>.
- [47] J. A. POPLE, *Nobel lecture: Quantum chemical models*, Rev. Modern Phys., 71 (1999), pp. 1267–1274, <https://doi.org/10.1103/RevModPhys.71.1267>.
- [48] T. J. RIVLIN, *The Chebyshev Polynomials*, 1st ed., Wiley, New York, 1974.
- [49] W. SICKEL AND T. ULLRICH, *Tensor products of Sobolev–Besov spaces and applications to approximation from the hyperbolic cross*, J. Approx. Theory, 161 (2009), pp. 748–786, <https://doi.org/10.1016/j.jat.2009.01.001>.
- [50] S. A. SMOLYAK, *Quadrature and interpolation formulas for tensor products of certain classes of functions*, Dokl. Akad. Nauk SSSR, 148 (1963), pp. 1042–1045.
- [51] P. J. STEPHENS, F. J. DEVLIN, C. F. CHABALOWSKI, AND M. J. FRISCH, *Ab initio calculation of vibrational absorption and circular dichroism spectra using density functional force fields*, J. Phys. Chem., 98 (1994), pp. 11623–11627, <https://doi.org/10.1021/j100096a001>.

- [52] J. STOER AND R. BULIRSCH, *Introduction to Numerical Analysis*, 2nd ed., Springer, New York, 1993.
- [53] M. STOYANOV, *Adaptive sparse grid construction in a context of local anisotropy and multiple hierarchical parents*, in Proceedings of Sparse Grids and Applications—Miami 2016, Springer, New York, 2018, pp. 175–199, [https://doi.org/10.1007/978-3-319-75426-0\\_8](https://doi.org/10.1007/978-3-319-75426-0_8).
- [54] M. STOYANOV, *TASMANIAN Sparse Grids (version 6.0)*, Technical report ORNL/TM-2015/596, Oak Ridge National Laboratory, 2018; also available online from <https://tasmanian.ornl.gov/>.
- [55] M. STOYANOV, P. SELESON, AND C. WEBSTER, *A surrogate modeling approach for crack pattern prediction in peridynamics*, in Proceedings of the 19th AIAA Non-Deterministic Approaches Conference, 2017, <https://doi.org/10.2514/6.2017-1326>.
- [56] M. STOYANOV AND C. G. WEBSTER, *A gradient-based sampling approach for dimension reduction of partial differential equations with stochastic coefficients*, Int. J. Uncertain. Quantif., 5 (2015), pp. 49–72, <https://doi.org/10.1615/Int.J.UncertaintyQuantification.2014010945>.
- [57] M. K. STOYANOV AND C. G. WEBSTER, *A dynamically adaptive sparse grids method for quasi-optimal interpolation of multidimensional functions*, Comput. Math. Appl., 71 (2016), pp. 2449–2465, <https://doi.org/10.1016/j.camwa.2015.12.045>.
- [58] R. A. TODOR AND C. SCHWAB, *Convergence rates for sparse chaos approximations of elliptic problems with stochastic coefficients*, IMA J. Numer. Anal., 27 (2007), pp. 232–261, <https://doi.org/10.1093/imanum/drl025>.
- [59] H. TRAN, C. G. WEBSTER, AND G. ZHANG, *Analysis of quasi-optimal polynomial approximations for parameterized PDEs with deterministic and stochastic coefficients*, Numer. Math., 137 (2017), pp. 451–493, <https://doi.org/10.1007/s00211-017-0878-6>.