

ORNL REPORT

ORNL/TM-2015/341

Unlimited Release

Printed July 2015

A Dynamically Adaptive Sparse Grid Method for Quasi-Optimal Interpolation of Multidimensional Analytic Functions

M. K. Stoyanov, C. G. Webster

Prepared by
Department of Computational and Applied Mathematics
Computer Science and Mathematics Division
Oak Ridge National Laboratory
One Bethel Valley Road, Oak Ridge, Tennessee 37831

The Oak Ridge National Laboratory is operated by UT-Battelle, LLC,
for the United States Department of Energy under Contract DE-AC05-00OR22725.
Approved for public release; further dissemination unlimited.

DOCUMENT AVAILABILITY

Reports produced after January 1, 1996, are generally available free via the U.S. Department of Energy (DOE) Information Bridge.

Web site <http://www.osti.gov/bridge>

Reports produced before January 1, 1996, may be purchased by members of the public from the following source.

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
Oak Ridge, TN 37831

Telephone 703-605-6000 (1-800-553-6847)

TDD 703-487-4639

Fax 703-605-6900

E-mail info@ntis.gov

Web site <http://www.ntis.gov/support/ordernowabout.htm>

Reports are available to DOE employees, DOE contractors, Energy Technology Data Exchange (ETDE) representatives, and International Nuclear Information System (INIS) representatives from the following source.

Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone 865-576-8401

Fax 865-576-5728

E-mail reports@osti.gov

Web site <http://www.osti.gov/contact.html>

NOTICE

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.



A DYNAMICALLY ADAPTIVE SPARSE GRID METHOD FOR QUASI-OPTIMAL INTERPOLATION OF MULTIDIMENSIONAL ANALYTIC FUNCTIONS

Miroslav K. Stoyanov ^{*} Clayton G. Webster [†]

Abstract. In this work we develop a dynamically adaptive sparse grids (SG) method for quasi-optimal interpolation of multidimensional analytic functions defined over a product of one dimensional bounded domains. The goal of such approach is to construct an interpolant in space that corresponds to the “best M -terms” based on sharp a priori estimate of polynomial coefficients. In the past, SG methods have been successful in achieving this, with a traditional construction that relies on the solution to a Knapsack problem: only the most profitable hierarchical surpluses are added to the SG. However, this approach requires additional sharp estimates related to the size of the analytic region and the norm of the interpolation operator, i.e., the Lebesgue constant. Instead, we present an iterative SG procedure that adaptively refines an estimate of the region and accounts for the effects of the Lebesgue constant. Our approach does not require any a priori knowledge of the analyticity or operator norm, is easily generalized to both affine and non-affine analytic functions, and can be applied to sparse grids build from one dimensional rules with arbitrary growth of the number of nodes. In several numerical examples, we utilize our dynamically adaptive SG to interpolate quantities of interest related to the solutions of parametrized elliptic and hyperbolic PDEs, and compare the performance of our quasi-optimal interpolant to several alternative SG schemes.

1. Introduction. This paper considers constructing an approximations to multidimensional analytic functions defined over a product of one dimensional bounded domains. The main challenge facing all methods in this context is the *curse of dimensionality*, i.e., the computational complexity of approximation techniques increases exponentially with the number of dimensions. To alleviate the curse, methods have been proposed that reduce the dimensionality of the problem [19, 26], reduce the complexity the target function [2, 11], or approximate the function in an optimal polynomial subspace [9, 23, 24]. We take the latter approach and we build upon the recent results in best M -terms approximation [5, 9, 29], where the function is projected onto the polynomial space associated with the dominant coefficients of either a Taylor or Legendre expansion. In implementation, finding the optimal space is intractable and instead sharp a priori estimates of the expansion coefficients are used to select a quasi-optimal space. Such approach can achieve sub-exponential convergence rate in the context of both projection, e.g., [1, 12, 28] and interpolation, e.g., [5, 22], however, the quasi-optimal methods rely heavily on a priori estimates of the size of the region of analyticity of the function and sharp estimates are available only in few special cases.

Given a suitable polynomial space, orthogonal projection results in the best L^2 approximation, however, the projection approach often times comes at a heavy computational cost [1, 12, 28]. In contrast, sampling based techniques require only the values of the function at a set of nodes, e.g., Monte Carlo random sampling for computing the statistical moments of a function [17, 21], and Sparse Grids (SG) method for high order polynomial approximation [18, 23, 24], which is the focus on this work. SG sampling does not result in best approximation in the associated polynomial space and the error is magnified by

^{*}Department of Computational and Applied Mathematics, Oak Ridge National Laboratory, One Bethel Valley Road, P.O. Box 2008, MS-6367, Oak Ridge, TN 37831-6164 (stoyanovmk@ornl.gov)

[†]Department of Computational and Applied Mathematics, Oak Ridge National Laboratory, One Bethel Valley Road, P.O. Box 2008, MS-6164, Oak Ridge, TN 37831-6164 (webstercg@ornl.gov)

the norm of the SG operator, a.k.a., the Lebesgue constant. However, sampling tends to be computationally cheaper than projection as well as more susceptible to parallelization which usually offsets the moderate increase of the error. In addition, sampling procedures can wrap around simulation software that computes single realization of the function, which simplifies the implementation and allows the use of legacy and third party code.

Sparse grids algorithms construct multidimensional function approximation from a linear combination of tensors of one dimensional interpolation rules. Quasi-optimal SG are traditionally constructed as the solution to a Knapsack problem [3, 22], where the selected set of tensors is associated with the largest profit index that is derived from an a priori estimate of the hierarchical surplus, the Lebesgue constant, and the number of samples in a tensor. In the case when the one dimensional rules grow by one node at a time, a near optimal greedy procedure using the Taylor coefficients of the function can construct a suitable approximation [4, 5], however, without a priori assumptions, selecting the optimal set of coefficients comes at a very high computational cost.

In this work, we present an iterative procedure for constructing a sequence of SG interpolants with increasing number of nodes and accuracy, that does not require a priori estimates of the region of analyticity. We focus our attention to the nested SG case, where all nodes associated with one grid are also utilized by the next grid in the sequence, thus reusing all available samples. We review popular one dimensional nested rules such as Clenshaw-Curtis [7] and Leja [6, 10] and we present several new rules based on greedy minimization of operator norms. In addition, for any chosen rule and any arbitrary lower (i.e., admissible [3]) polynomial space, we present a strategy for selecting the minimal set of tensors that yields an interpolant in that space. Every interpolant in the sequence is constructed using this strategy, which circumvents the Knapsack problem and allows us to restrict our attention to the selection of the optimal polynomial spaces.

The quasi-optimal polynomial space associated with Legendre coefficients is a total degree space with a small logarithmic correction [1, 29]. However, while the Legendre space is optimal with respect to projection, in the context of interpolation, the quasi-optimal estimate does not account for the effect of the Lebesgue constant. Using estimates of the operator norm of the one dimensional rules, we add a strong correction to the total degree space to arrive at an estimate for the quasi-optimal interpolation space. Our estimate is parametrized by two vectors associated with the size of the analytic region of the function and the growth of the Lebesgue constant of the interpolation rules.

In order to keep our approach free from a priori assumptions, we present a procedure for dynamically estimating the two vector parameters. For each interpolant in the sequence, we consider the orthogonal decomposition of the interpolant into a linear combination of multivariate Legendre polynomials. Then, we seek the vectors that give the best fit of our quasi-optimal estimate to the decay rate of the Legendre coefficients, i.e., using least-squares approach. The polynomial space used for the construction of the next interpolant in the sequence is optimal with respect to the parameters inferred from the previous interpolant. The number of additional nodes in each interpolant can be chosen arbitrarily, however, few nodes result in more frequent update of the parameter vectors which leads to better accuracy, while larger number of nodes allows for greater parallelization.

The procedure for estimating the quasi-optimal polynomial space can be coupled with any approximation strategy that satisfies a mild assumption regarding the growth of the

Lebesgue constant. One potential alternative is to use interpolation based on Fekete points, however, even in moderate dimensions, finding those points involves an ill-conditioned and prohibitively expensive problem. Other popular alternatives are the optimization based methods that construct the an approximation based on minimization of ℓ^2 (e.g., least-squares [14]) or ℓ^1 (e.g., compressed sensing [15]) norms. Those methods can be applied to sets of random samples, however, the number of samples needed to construct the approximation always exceeds the cardinality of the optimal polynomial space. We assume that we can choose the abscissas for each samples and we want to exploit the fact that the range of an interpolation operator has exactly the same degrees of freedom as the number of interpolation nodes. Thus, the sparse grids interpolants are best suited for our context.

The rest of the paper is organized as follows, in §2 we derive an estimate of the quasi-optimal interpolation space and we present an iterative procedure for generating a sequence of quasi-optimal polynomial spaces. In §3, we present a strategy for constructing sparse grids operators with minimal number of nodes and we present several one dimensional interpolation rules. In §4 we present several numerical examples.

2. Quasi-optimal polynomial space. We consider the problem of approximating a multivariate function $f(\mathbf{y}) : \Gamma \rightarrow \mathbb{R}$, where $\Gamma \subset \mathbb{R}^d$ is a d -dimensional hypercube, i.e., $\Gamma = \bigotimes_{k=1}^d \Gamma_k$ and without loss of generality we let $\Gamma_k = [-1, 1]$. We assume that $f(\mathbf{y})$ admits holomorphic extension to a poly-ellipse in complex plane, i.e.,

Assumption 1 (Holomorphic extension). *For a vector $\boldsymbol{\rho} \in \mathbb{R}^d$ with $\rho_k > 1$, the map $z \rightarrow f(z)$ is holomorphic in an open neighborhood of the poly-ellipse*

$$\mathcal{E}_{\boldsymbol{\rho}} = \bigcup_{\theta \in [0, 2\pi]} \bigotimes_{1 \leq k \leq d} \left\{ z_k \in \mathbb{C} : |\Re(z_k)| \leq \frac{\rho_k + \rho_k^{-1}}{2} \cos(\theta), |\Im(z_k)| \leq \frac{\rho_k - \rho_k^{-1}}{2} \sin(\theta) \right\} \quad (2.1)$$

where $\Re(z_k)$ and $\Im(z_k)$ indicate the real and complex part of z_k .

Due to this assumption, we aim at approximating $f(\mathbf{y})$ with globally defined polynomials over Γ . To achieve this goal, we introduce a multivariate polynomial space

$$\mathcal{P}_{\Lambda(p)}(\Gamma) = \text{span} \{ \mathbf{y}^{\boldsymbol{\nu}} : \boldsymbol{\nu} \in \Lambda(p) \},$$

in which it will be convenient to use multi-index notation¹, where $\Lambda(p)$ is a sequence of lower multi-index set². A global polynomial approximation of $f(\mathbf{y})$ in $\mathcal{P}_{\Lambda(p)}(\Gamma)$ has the form

$$f_{\Lambda(p)} = \sum_{\boldsymbol{\nu} \in \Lambda(p)} c_{\boldsymbol{\nu}} \phi_{\boldsymbol{\nu}}(\mathbf{y}),$$

where $\text{span} \{ \phi_{\boldsymbol{\nu}}(\mathbf{y}) : \boldsymbol{\nu} \in \Lambda(p) \} = \mathcal{P}_{\Lambda(p)}(\Gamma)$, and the choice of $\phi_{\boldsymbol{\nu}}(\mathbf{y})$ and $c_{\boldsymbol{\nu}}$ is method specific. To alleviate the curse of dimensionality, the polynomial space $\mathcal{P}_{\Lambda(p)}(\Gamma)$ should be chosen with as few degrees of freedom necessary to approximate $f(\mathbf{y})$ with sufficient

¹For the remainder of the paper we let \mathbb{N} be the set of natural numbers including zero, and $\Lambda, \Theta \subset \mathbb{N}^d$ will denote set of multi-indexes. For any two vectors, we define $\mathbf{y}^{\boldsymbol{\nu}} = \prod_{k=1}^d y_k^{\nu_k}$ with the usual convention $0^0 = 1$.

²A set Λ is called lower or admissible if $\boldsymbol{\nu} \in \Lambda$ implies $\{ \mathbf{i} \in \mathbb{N}^d : \mathbf{i} \leq \boldsymbol{\nu} \} \subset \Lambda$, where $\mathbf{i} \leq \boldsymbol{\nu}$ if and only if $i_k \leq \nu_k$ for all $1 \leq k \leq d$.

accuracy. Using Assumption 1, let $\alpha_k = \log(\rho_k)$ for $1 \leq k \leq d$ dictate the anisotropy in each direction, then the most common choice of the polynomial space is the tensor product space $\mathcal{P}_{\Lambda^{TP}(p)}(\Gamma)$, where

$$\Lambda^{TP}(p) = \{\boldsymbol{\nu} \in \mathbb{N}^d : \max_{1 \leq k \leq d} \alpha_k \nu_k \leq p\},$$

and the cardinality depends exponentially on the dimension. Several alternative polynomial spaces have been proposed, namely total degree space with $\Lambda^{TD}(p) = \{\boldsymbol{\nu} \in \mathbb{N}^d : \boldsymbol{\alpha} \cdot \boldsymbol{\nu} \leq p\}$, hyperbolic cross section $\Lambda^{HC}(p) = \{\boldsymbol{\nu} \in \mathbb{N}^d : (\boldsymbol{\nu} + \mathbf{1})^\alpha \leq p\}$ and Smolyak $\Lambda^{Sm}(p) = \{\boldsymbol{\nu} \in \mathbb{N}^d : \boldsymbol{\alpha} \cdot \log_2(\boldsymbol{\nu} + \mathbf{1}) \leq p\}$, where \cdot indicates vector dot product and $\log(\boldsymbol{\nu} + \mathbf{1}) = \bigotimes_{k=1}^d \log(\nu_k + 1)$ (see [18] and references therein).

Our approach is motivated by recent work on best M -term quasi-optimal Galerkin approximation [1, 8, 28]. Consider the orthogonal decomposition of $f(\mathbf{y})$

$$f(\mathbf{y}) = \sum_{\boldsymbol{\nu} \in \mathbb{N}^d} c_\nu L_\nu(\mathbf{y}),$$

where $L_\nu(\mathbf{y})$ are the multivariate Legendre polynomials. For any lower set $\Lambda(p)$ the projection of $f(\mathbf{y})$ onto $\mathcal{P}_{\Lambda(p)}(\Gamma)$ is given by $f_{\Lambda(p)}(\mathbf{y}) = \sum_{\boldsymbol{\nu} \in \Lambda(p)} c_\nu L_\nu(\mathbf{y})$ and the approximation error is

$$\|f - f_{\Lambda(p)}\|_{L^2}^2 = \sum_{\boldsymbol{\nu} \notin \Lambda(p)} |c_\nu|^2$$

Therefore, the best M -term space for projection is the space associated with the M largest coefficients. The coefficients are not known a priori, however, when Assumption 1 is satisfied, a sharp upper bound to $|c_\nu|$ is given by

$$|c_\nu| \leq C \cdot \exp(-\boldsymbol{\alpha} \cdot \boldsymbol{\nu}) \prod_{k=1}^d \sqrt{2\nu_k + 1}, \quad (2.2)$$

for some constant C [1, 29]. The quasi-optimal projection space is then associated with the multi-indexes that yield the largest upper estimate, i.e.,

$$\Lambda^\alpha(p) = \left\{ \boldsymbol{\nu} \in \mathbb{N}^d : \boldsymbol{\alpha} \cdot \boldsymbol{\nu} - \frac{1}{2} \sum_{k=1}^d \log(\nu_k + 0.5) \leq p \right\}$$

where we derive the condition by taking the log of the right hand side of (2.2), changing the sign and ignoring the constant. In this context, $p \in \mathbb{N}$ is an arbitrary variable used to discretize the index space into levels.

2.1. Quasi-optimal interpolation. We are interested in constructing approximation using a computationally cheap sampling scheme. Projection results in optimal $L^2(\Gamma)$ error, however, interpolatory approximations are not optimal. Let $\Lambda(L)$ be a lower set and let $f_{\Lambda(L)}(\mathbf{y})$ be in interpolatory approximation of $f(\mathbf{y})$ in $\mathcal{P}_{\Lambda(L)}(\Gamma)$, then

$$\|f - f_{\Lambda(L)}\|_{L^\infty} \leq (1 + \mathcal{C}_{\Lambda(L)}) \inf_{p \in \mathcal{P}_{\Lambda(L)}(\Gamma)} \|f - p\|_{L^\infty}, \quad (2.3)$$

where $\mathcal{C}_{\Lambda(L)}$ is the Lebesgue constant, i.e., norm of the interpolation operator, that manifests as an additional penalty term. Observe that

$$\inf_{p \in \mathcal{P}_{\Lambda(L)}(\Gamma)} \|f - p\|_{L^\infty} \leq \left\| f - \sum_{\nu \in \Lambda(L)} c_\nu L_\nu \right\|_{L^\infty}, \quad (2.4)$$

therefore, the dominant polynomial space associated with the infimum term in (2.3) can be (heuristically) approximated by estimate (2.2), i.e., the difference comes only from using $L^2(\Gamma)$ as opposed to the $L^\infty(\Gamma)$ norm. However, to define a proper quasi-optimal interpolation space, (2.2) has to be combined with an estimate of the Lebesgue constant. Here we make the following assumption³:

Assumption 2 (Lebesgue constant). *Let \mathcal{C}_ν indicate the Lebesgue constant associated with the smallest lower set that contains ν , i.e., $\mathcal{C}_\nu = \mathcal{C}_{\{j \in \mathbb{N}^d: j \leq \nu\}}$, then we assume that*

$$\mathcal{C}_\nu \leq C_\gamma \prod_{k=1}^d (\nu_k + 1)^{\gamma_k}, \quad (2.5)$$

for some constants C_γ and $(\gamma_k)_{1 \leq k \leq d}$.

Multiplying (2.2) by (2.5), and using that $\nu_k + 0.5 < \nu_k + 1$ we arrive at the estimate

$$\mathcal{C}_\nu \cdot C \cdot \exp(-\alpha \cdot \nu) \prod_{k=1}^d \sqrt{2\nu_k + 1} \leq C \cdot C_\gamma \cdot \sqrt{2} \cdot \exp(-\alpha \cdot \nu) \prod_{k=1}^d (\nu_k + 1)^{\gamma_k + \frac{1}{2}} \quad (2.6)$$

As before, taking the log of (2.6), reversing the sign and ignoring the constants, we define the quasi-optimal interpolation space

$$\Lambda^{\alpha, \beta}(L) = \left\{ \nu \in \mathbb{N}^d : \alpha \cdot \nu + \beta \cdot \log(\nu + \mathbf{1}) \leq L \right\}, \quad (2.7)$$

where $\beta_k = -\gamma_k - \frac{1}{2}$. The vectors α and β give rise to a discretization of the multi-index space, however, the notion of levels does not transcend the specific α and β , since (2.7) depends on the scaling of the vector components. Given parameter vectors and a sequence of levels $\{L_n\}_{n=0}^\infty \in \mathbb{N}^d$, we can construct the corresponding quasi-optimal approximations $f_{\Lambda^{\alpha, \beta}(L_n)}(\mathbf{y})$, however, finding suitable α and β is of consideration.

Remark 2.1 (Preserving the property lower). *The entries of β in (2.7) could be negative, hence, $\Lambda^{\alpha, \beta}(L)$ is not necessarily a lower set. Lower sets have the advantage that the polynomial space associated with the multi-indices is independent from the hierarchical basis used, e.g., Legendre basis, monomials, Newton polynomials. We are interested in constructing a sequence of lower sets, thus, if $\Lambda^{\alpha, \beta}(L)$ is not a lower set, we replace (2.7) with*

$$\Lambda^{\alpha, \beta}(L) = \bigcup_{\nu \in \mathbb{N}^d, \alpha \cdot \nu + \beta \cdot \log(\nu + \mathbf{1}) \leq L} \{j \in \mathbb{N}^d : j \leq \nu\}$$

³In §3 we derive an estimate for the Lebesgue constant associated with our sparse grids construction and we demonstrate that Assumption 2 is indeed satisfied.

2.2. Estimating the parameters. Sharp a priori bounds of the region of analytic extension of $f(\mathbf{y})$ and the corresponding α are seldom available, likewise estimates of the Lebesgue constant are either overly conservative or the constant fluctuates in a wide range and predicting the effective γ_k is very difficult. Thus, we need a procedure to dynamically estimate the *effective* parameters α and β that give the best fit of (2.6) to the behavior of $f(\mathbf{y})$.

Assume that we have constructed an interpolant $f_{\Lambda(L)}(\mathbf{y})$ for some lower set $\Lambda(L)$. Here $\Lambda(L)$ could be chosen according to (2.7) for some α and β , or according to total degree, hyperbolic cross section or Smolyak formulas [18]. Since $f_{\Lambda(L)}(\mathbf{y}) \in \mathcal{P}_{\Lambda(L)}$ there are coefficients \hat{c}_{ν} for $\nu \in \Lambda(L)$ such that

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

where $L_{\nu}(\mathbf{y})$ are the multidimensional Legendre polynomials. By orthogonality, each of the coefficients is

$$\hat{c}_{\nu} = \int_{\Gamma} f_{\Lambda(L)}(\mathbf{y}) L_{\nu}(\mathbf{y}) d\mathbf{y}, \quad (2.8)$$

where the integral can be computed with a multidimensional quadrature rule and note this can be done without additional evaluations of $f(\mathbf{y})$. Since $L_{\nu}(\mathbf{y})$ and $f_{\Lambda(L)}(\mathbf{y})$ are polynomials, it is sufficient to use a quadrature that can integrate exactly all polynomials in $\mathcal{P}_{2\Lambda(L)}$.

We assume that $|\hat{c}_{\nu}|$ decay at a rate guided by (2.6) for some α and β , i.e.,

$$|\hat{c}_{\nu}| \propto \exp(-\alpha \cdot \nu)(\nu + \mathbf{1})^{-\beta} \implies \log(|\hat{c}_{\nu}|) \approx -\hat{C} - \alpha \cdot \nu - \beta \cdot \log(\nu + \mathbf{1}), \quad (2.9)$$

for some constant \hat{C} . In (2.9), all \hat{c}_{ν} are known and we can solve for α , β and \hat{C} , however, the decay rate of the coefficients is not monotone and hence we look for the parameters that give the “best fit”. Here best is used in ℓ^2 sense, i.e., we infer approximate $\hat{\alpha}$ and $\hat{\beta}$ from the solution to the convex minimization problem

$$\min_{\alpha, \beta, \hat{C}} \frac{1}{2} \sum_{\nu \in \Lambda(L)} \left(\hat{C} + \alpha \cdot \nu + \beta \cdot \log(\nu + \mathbf{1}) + \log(|\hat{c}_{\nu}|) \right)^2 \quad (2.10)$$

For sufficiently large $\Lambda(L)$, (2.10) admits a unique solution. However, the accuracy of the estimated $\hat{\alpha}$ and $\hat{\beta}$ strongly depends on the size of $\Lambda(L)$, in fact, estimates obtained using this least squares approach are valid only for sets close to $\Lambda(L)$. Furthermore, the constant \hat{C} is not used in (2.7) and since $\hat{C} \exp(-\alpha \cdot \nu)(\nu + \mathbf{1})^{-\beta}$ does not give an upper bound on the coefficients, \hat{C} cannot be used to estimate the true approximation error; in this context, \hat{C} plays the role of a *dummy* variable.

Remark 2.2 (Ad hoc stability constraint). *In general, $|\hat{c}_{\nu}|$ do not decay monotonically and if $\Lambda(L)$ is small, some of the estimated parameters in $\hat{\alpha}$ could be negative. The entries of $\hat{\alpha}$ are associated with the convergence rate in different direction and negative entries indicate that for this specific direction and specific $\Lambda(L)$ we are observing divergence. In case of a negative $\hat{\alpha}_k$, we use an ad hoc correction, where we replace the negative entries with the smallest positive one, i.e., we assume that in the limit the diverging direction will in fact*

converge, albeit slowly.

The least-squares approach allows us to construct a sequence of polynomial spaces indexed by sets $\{\Lambda_n\}_{n=0}^\infty$. We start with either $\Lambda_0 = \Lambda^{\alpha_0, \beta_0}(L_0)$ for some initial guess of α_0, β_0 and L_0 , or Λ_0 can be chosen according to total degree, hyperbolic cross section or Smolyak formulas. Then from Λ_n and the interpolant $f_{\Lambda_n}(\mathbf{y})$, we infer the best fit parameters $\hat{\alpha}_{n+1}$ and $\hat{\beta}_{n+1}$ and construct the next index set as

$$\Lambda_{n+1} = \Lambda_n \cup \Lambda^{\hat{\alpha}_{n+1}, \hat{\beta}_{n+1}}(L_{n+1}), \quad (2.11)$$

where we take the union to ensure that the sequence is nested and L_{n+1} controls the number of additional degrees of freedom in the polynomial space. Using small L_{n+1} , i.e., taking the smallest L_{n+1} such that $\Lambda^{\hat{\alpha}_{n+1}, \hat{\beta}_{n+1}}(L_{n+1}) \not\subset \Lambda_n$, leads to a more frequent update of the parameters $\hat{\alpha}$ and $\hat{\beta}$ and hence better accuracy, however, larger L_{n+1} leads to more samples needed for constructing $f_{\Lambda_{n+1}}(\mathbf{y})$ and hence more opportunity for parallelization.

Next, we present a specific sparse grids strategy for constructing $f_{\Lambda_n}(\mathbf{y})$.

3. Sparse grids interpolation. In this section we present a general sparse grids interpolation approach, which consists of evaluating $f(\mathbf{y})$ at a set of nodes $\mathbf{y}_1, \dots, \mathbf{y}_m \in \Gamma$ and constructing an interpolant $f_{\Lambda(L)}(\mathbf{y}) \in \mathcal{P}_{\Lambda(L)}$, where $\Lambda(L)$ is an arbitrary lower index set, i.e., not necessarily constructed according to (2.7). The properties of the SG interpolant are determined by the Lebesgue constant and growth of nodes in the one dimensional family of interpolants that induce the grid. Minimizing the number of nodes is desirable and ideally we want the number of samples to not exceed the cardinality of $\Lambda(L)$, however, interpolants with more nodes often times result in smaller operator norm and potentially more accurate approximation. For a given $\Lambda(L)$ and one dimensional rule, we present a strategy for constructing the SG with smallest number of nodes that produces an interpolant in $\mathcal{P}_{\Lambda(L)}$. We also present several novel one dimensional rules.

3.1. Constructing optimal interpolant. A nested one dimensional family of interpolation rules is defined by a distinct sequence of nodes $\{y_j\}_{j=1}^\infty \in [-1, 1]$ and a strictly increasing growth function $m : \mathbb{N} \rightarrow \mathbb{N}$, i.e., $m(0) > 0$ and $m(l+1) > m(l)$. For $l = 0, 1, 2, \dots$ we define the l -th level interpolation operator

$$\mathcal{U}^{m(l)} : \mathbb{C}^0(\Gamma) \rightarrow \mathcal{P}_{m(l)-1}([-1, 1]), \quad \text{by} \quad \mathcal{U}^{m_l}[g](y) = \sum_{j=1}^{m(l)} g(y_j) \psi_j^l(y),$$

where $g(y) \in \mathbb{C}^0([-1, 1])$, the Lagrange basis functions are $\psi_j^l(y) = \prod_{i=1, i \neq j}^{m_l} \frac{y-y_i}{y_j-y_i}$ and $\mathcal{P}_{m(l)-1}([-1, 1]) = \text{span}\{y^\nu : 0 \leq \nu \leq m(l)-1\}$. In addition, for $l > 0$, we define the surplus operators $\Delta^{m(l)} = \mathcal{U}^{m(l)} - \mathcal{U}^{m(l-1)}$ and for notational convenience let $\Delta^0 = \mathcal{U}^{m(0)}$. Several specific examples of y_j and m_l are listed in Table 1 in §3.3. Note, we are explicitly assuming the interpolants have nested nodes and strictly increasing $m(l)$, see Remarks 3.1 and 3.3.

Taking the d -dimensional tensors, we have

$$\mathbf{m}(\mathbf{i}) = \bigotimes_{k=1}^d m(i_k) : \mathbb{N}^d \rightarrow \mathbb{N}^d, \quad \mathbf{y}_j = \bigotimes_{k=1}^d y_{j_k} \in \Gamma, \quad \Psi_j^{\mathbf{i}}(\mathbf{y}) = \prod_{k=1}^d \psi_{j_k}^{i_k},$$

where $\mathbf{i}, \mathbf{j} \in \mathbb{N}^d$, $\mathbf{y} \in \Gamma$ and $\psi_{j_k}^{i_k}$ is evaluated at the corresponding k -th component of \mathbf{y} . The tensor operators are given by

$$\Delta^{\mathbf{m}(\mathbf{i})} = \bigotimes_{k=1}^d \Delta^{m(i_k)}, \quad \mathcal{U}^{\mathbf{m}(\mathbf{i})} = \bigotimes_{k=1}^d \mathcal{U}^{m(i_k)}, \quad \mathcal{U}^{\mathbf{m}(\mathbf{i})}[f](\mathbf{y}) = \sum_{\mathbf{1} \leq \mathbf{j} \leq \mathbf{m}(\mathbf{i})} f(\mathbf{y}_{\mathbf{j}}) \Psi_{\mathbf{j}}^{\mathbf{i}}(\mathbf{y})$$

Note that by the telescoping properties of $\Delta^{\mathbf{m}(\mathbf{i})}$ we have that $\mathcal{U}^{\mathbf{m}(\mathbf{i})} = \sum_{\mathbf{j} \leq \mathbf{i}} \Delta^{\mathbf{m}(\mathbf{j})}$, i.e., each full tensor operator $\mathcal{U}^{\mathbf{m}(\mathbf{i})}$ can be decomposed into a sum of surplus operators.

Every multidimensional polynomial space can be included in the range of a full tensor interpolation operator, however, full tensors have a rigid structure and often times require an excessive number of samples. Sparse grids offer a flexible alternative, where the interpolant is constructed from a sparse set of the surplus operators $\Theta(L)$, i.e.,

$$I_{\Theta(L)}^{\mathbf{m}} = \sum_{\mathbf{i} \in \Theta(L)} \Delta^{\mathbf{m}(\mathbf{i})} \quad (3.1)$$

For any lower index set $\Theta(L)$, (3.1) is an interpolation operator with nodes $\{\mathbf{y}_{\mathbf{j}}\}_{\mathbf{j} \in \Theta_m(L)}$, where

$$\Theta_m(L) = \bigcup_{\mathbf{i} \in \Theta(L)} \{\mathbf{j} \in \mathbb{N}^d : \mathbf{1} \leq \mathbf{j} \leq \mathbf{m}(\mathbf{i})\} = \bigcup_{\mathbf{i} \in \Theta(L)} \{\mathbf{j} \in \mathbb{N}^d : \mathbf{m}_{i-1} + \mathbf{1} \leq \mathbf{j} \leq \mathbf{m}(\mathbf{i})\} \quad (3.2)$$

and $I_{\Theta(L)}^{\mathbf{m}}[f] \in \mathcal{P}_{\Theta_m(L)-1}$ with $\Theta_m(L) - \mathbf{1} = \{\mathbf{j} \in \mathbb{N}^d : \mathbf{j} + \mathbf{1} \in \Theta_m(L)\}$ [22]. By definition of $\Delta^{\mathbf{m}(\mathbf{i})}$, there exists a set of integer weights $\{t_{\mathbf{j}}\}_{\mathbf{j} \in \Theta(L)}$, satisfying the system of equations $\sum_{\mathbf{j} \leq \mathbf{i}, \mathbf{j} \in \Theta(L)} t_{\mathbf{j}} = 1$ for every $\mathbf{i} \in \Theta(L)$, and the interpolant can be written explicitly as

$$I_{\Theta(L)}^{\mathbf{m}}[f](\mathbf{y}) = \sum_{\mathbf{j} \in \Theta_m(L)} f(\mathbf{y}_{\mathbf{j}}) \sum_{\mathbf{i} \in \Theta(L), \mathbf{m}(\mathbf{i}) \leq \mathbf{j}} t_{\mathbf{i}} \Psi_{\mathbf{j}}^{\mathbf{i}}(\mathbf{y}) \quad (3.3)$$

Interpolant (3.3) is constructed from the samples $f(\mathbf{y}_{\mathbf{j}})$ for $\mathbf{j} \in \Theta_m(L)$, and since the sets of the second union of (3.2) are disjoint, we have a direct relationship between $\Theta(L)$, $m(l)$ and the number of nodes.

Remark 3.1. *If the one dimensional family of rules is not nested then generally (3.1) is not an interpolant. Even in the non-nested case, operator of the form (3.1) can produce an accurate approximation to $f(\mathbf{y})$ [22, 24], however, the approximation belongs to a space of polynomials with cardinality much less than the required number of function evaluations. Excess sampling unnecessarily increases the computational cost and thus we restrict our attention to nested rules.*

Next, we present the main result of this section, where we consider the optimal choice of tensor set that will guarantee the range of the interpolation operator includes a given polynomial space.

Theorem 1 (Minimal polynomial interpolant). *Let $\Lambda(L)$ be a lower set of polynomial multi-indexes and define*

$$\Theta^{opt}(L) = \left\{ \mathbf{i} \in \mathbb{N}^d : \mathbf{m}(\mathbf{i} - \mathbf{1}) \in \Lambda(L) \right\}, \quad (3.4)$$

where for notational convenience we let $m(-1) = 0$. Then, $\mathcal{P}_{\Lambda(L)} \subset \mathcal{P}_{\Theta_m^{opt}(L)-1}$ and with respect to the particular $m(l)$, $\Theta^{opt}(L)$ is minimal, i.e., if $\Theta(L)$ is a lower set such that $\mathcal{P}_{\Lambda(L)} \subset \mathcal{P}_{\Theta_m(L)-1}$, then $\Theta(L)$ is a superset of $\Theta^{opt}(L)$.

Proof: For arbitrary $\mathbf{j} \in \Lambda(L)$, by monotonicity of $m(l)$, there is \mathbf{i} such that $m(i_k - 1) \leq j_k \leq m(i_k) - 1$ for $k = 1, 2, \dots, d$. Then, according to (3.4)

$$\mathbf{m}(\mathbf{i} - \mathbf{1}) \leq \mathbf{j} \implies \mathbf{i} \in \Theta^{opt}(L), \quad \text{and} \quad \mathbf{j} \leq \mathbf{m}(\mathbf{i}) - \mathbf{1} \implies \mathbf{j} \in \Theta_m^{opt}(L) - \mathbf{1}.$$

To show that $\Theta^{opt}(L)$ is minimal, suppose $\Theta(L)$ is a lower set and $\Lambda(L) \subset \Theta_m(L) - \mathbf{1}$. For arbitrary $\mathbf{i} \in \Theta^{opt}(L)$, let $\mathbf{j} = \mathbf{m}(\mathbf{i} - \mathbf{1})$, then

$$\mathbf{j} = \mathbf{m}(\mathbf{i} - \mathbf{1}) \implies \mathbf{j} \in \Lambda(L) \implies \mathbf{j} \in \Theta_m(L) - \mathbf{1} \implies \text{exists } \mathbf{i}' \in \Theta(L) \text{ s.t. } \mathbf{j} \leq \mathbf{m}(\mathbf{i}') - \mathbf{1}.$$

Thus, $\mathbf{m}(\mathbf{i} - \mathbf{1}) \leq \mathbf{m}(\mathbf{i}') - \mathbf{1}$ and by monotonicity of $m(l)$ we have $\mathbf{i} \leq \mathbf{i}'$. Since $\mathbf{i}' \in \Theta(L)$ and $\Theta(L)$ is a lower set, $\mathbf{i} \in \Theta(L)$. \square

Corollary 3.2. For $\Lambda^{\alpha, \beta}(L)$ defined in (2.7), the optimal tensor set is

$$\Theta^{\alpha, \beta}(L) = \left\{ \mathbf{i} \in \mathbb{N}^d : \alpha \cdot \mathbf{m}(\mathbf{i} - \mathbf{1}) + \beta \cdot \log(\mathbf{m}(\mathbf{i} - \mathbf{1}) + \mathbf{1}) \leq L \right\} \quad (3.5)$$

Armed with this result, we define the sparse grids interpolants associated with the sequence of polynomial spaces defined in (2.11). Let Θ_0 be the optimal tensor set associated with Λ_0 , then $f_{\Lambda_0}(\mathbf{y}) = I_{\Theta_0}^m[f](\mathbf{y})$ and for $n \in \mathbb{N}$

$$\Theta_{n+1} = \Theta_n \bigcup \Theta^{\hat{\alpha}_{n+1}, \hat{\beta}_{n+1}}(L_{n+1}), \quad \text{and} \quad f_{\Lambda_{n+1}}(\mathbf{y}) = I_{\Theta_{n+1}}[f](\mathbf{y})$$

where $\hat{\alpha}_{n+1}$ and $\hat{\beta}_{n+1}$ are estimated from (2.10). Note that unless $m(l) = l + 1$ the interpolants constructed in this fashion will be associated with polynomial spaces larger than \mathcal{P}_{Λ_n} . The traditional Knapsack approach uses $m(l)$ explicitly into the definition of the profit index [3, 22], however, in our context, rules with $m(l) > l + 1$ are associated with smaller Lebesgue constant and thus $m(l)$ affects Λ_n implicitly through the β term.

Remark 3.3. *Isotropic total degree space defined by $\sum_{k=1}^d j_k \leq L$ is an example of a particular polynomial space of interest, in fact, large amount of the initial work in sparse grids was aimed at constructing total degree interpolants. The work [25] give an optimal construction using one dimensional rules with occasionally repeating number of nodes (as opposed to strictly increasing). However, Corollary 3.2 with $\alpha = \mathbf{1}$ and $\beta = \mathbf{0}$ gives us the same result as the slow-growth method and hence the assumption $m(l) < m(l + 1)$ is not restrictive.*

3.2. Lebesgue constant. In the one dimensional context, the norm of $\mathcal{U}^{m(l)}$ can be estimated numerically and, in some cases, sharp theoretical estimates are also available. Thus, we assume there exists $\{\lambda_l\}_{l=0}^{\infty} \in \mathbb{R}$ so that $\|\mathcal{U}^{m(l)}\|_{L^\infty} \leq \lambda_l$, and for specific examples see §3.3. However, even if λ_l is sharp, there is no known *sharp* analytic estimate of $\|I_{\Theta(L)}^m\|_{L^\infty}$ for a general $\Theta(L)$ and numerical estimates are computationally impractical. In the case when λ_l exhibits polynomial growth, i.e., $\lambda_l \leq C_\gamma(l + 1)^\gamma$ for some $C_\gamma, \gamma \in \mathbb{R}$, Lemma 3.1 in [5] shows that

$$\|I_{\Theta(L)}^m\|_{C^0} \leq C_\gamma^d (\#\Theta(L))^{\gamma+1} \quad (3.6)$$

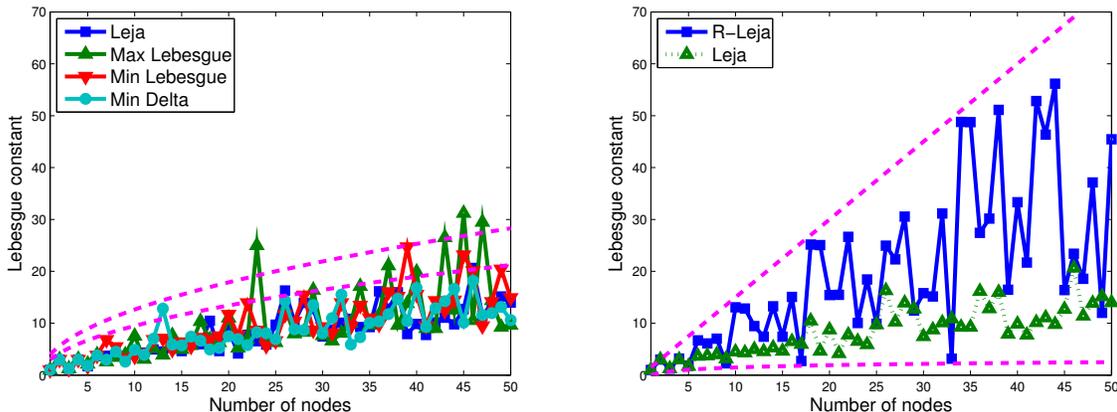


Figure 1: Interpolation operator norm for different sequence of points. Left: the two dashed lines correspond to $3\sqrt{l+1}$ and $4\sqrt{l+1}$. Right: the reference lines are $\frac{3}{2}(l+1)$ and $\frac{2}{\pi} \log(2^l+1)$.

where $\#\Theta(L)$ indicates the number of elements in $\Theta(L)$. This result is not sharp and the effective operator norm is usually much smaller than what (3.6) suggests, however, estimate (3.6) indicates that the two main factors contributing to the operator norm are the growth of λ_l (i.e., C_γ and γ) and the cardinality of $\Theta(L)$.

The polynomial growth of λ_l is sufficient to satisfy Assumption 2. The norm of a full tensor operator is $\|\mathcal{U}^{\mathbf{m}(\mathbf{i})}\|_{L^\infty} \leq \prod_{k=1}^d \lambda_{i_k}$, and the associated polynomial space is indexed by $\{\boldsymbol{\nu} \in \mathbb{N}^d : \boldsymbol{\nu} \leq \mathbf{m}(\mathbf{i}) - \mathbf{1}\}$. By monotonicity of $m(l)$, if \mathbf{i} is the index of the smallest full tensor containing $\boldsymbol{\nu}$, then $\mathbf{i} \leq \boldsymbol{\nu}$ and

$$C_\nu \leq \prod_{k=1}^d \lambda_{i_k} \leq C_\gamma^d \prod_{k=1}^d (i_k + 1)^\gamma \leq C_\gamma^d \prod_{k=1}^d (\nu_k + 1)^\gamma \quad (3.7)$$

Note that (3.7) implies that γ_k is independent from k , however, a one dimensional family of rules can exhibit very slow increase of the Lebesgue constant for the first several nodes, followed by much sharper increase. Thus, when $f(\mathbf{y})$ exhibits strong anisotropic behavior (i.e., the components of $\boldsymbol{\alpha}$ differ significantly), there is a large discrepancy in the number of one dimensional nodes associated with various directions, and different γ_k yield a much sharper estimate.

3.3. One dimensional rules. In this section, we present multiple one dimensional interpolation rules with different nodes y_j , growth sequence m_l and operator norm λ_l . Ideally, we want a rule with slowly growing $m(l)$ and λ_l , since small m_l results in fewer interpolation nodes and smaller λ_l leads to better accuracy of the approximation, however, those are competing goals. Rules with rapidly increasing $m(l)$ lead to Θ_n with small cardinality and interpolants with small Lebesgue constant. However, rapid growth of $m(l)$ also leads to interpolants with more nodes than degrees of freedom of Λ_n . It is hard to predict a priori the optimal relation between $m(l)$ and λ_l , and in what follows, we present a list of several candidate interpolation rules.

The roots and extrema of Chebyshev polynomials are a popular choice for interpolation nodes and the Clenshaw-Curtis [7] rule is one of the most widely used. The Clenshaw-Curtis nodes y_j are defined by

$$y_1 = 0, \quad y_2 = 1, \quad y_3 = -1, \quad \text{for } j > 3, \quad y_j = \cos\left(2^{-\lceil \log_2(j-1) \rceil} (2j-3)\pi\right), \quad (3.8)$$

where $\lceil x \rceil = \min\{i \in \mathbb{Z} : i \geq x\}$. The growth sequence starts with $m(0) = 1$ and for $l > 0$ we have $m(l) = 2^l + 1$. The operator norm increases logarithmically in number of nodes

$$\lambda_l = \frac{2}{\pi} \log(m(l) - 1) + 1 = \frac{2}{\pi} \log(2^l) + 1. \quad (3.9)$$

Another similar example is the Fejer type-2 interpolation based on the interior roots of Chebyshev polynomials [16]. The nodes are given by

$$y_j = \cos\left(2^{-\lceil \log_2(j+1) \rceil} (2j+1)\pi\right), \quad (3.10)$$

with growth sequence $m(l) = 2^{l+1} - 1$. Similar to Clenshaw-Curtis, the operator norm exhibits logarithmic dependence on $m(l)$.

Remark 3.4. For interpolation rules that exhibit logarithmic increase of λ_l , replacing $(\nu_k + 1)^\gamma$ in Assumption 2 with $\log(\nu_k + 1)$ may yield a sharper estimate. However, the effects of the logarithmic term are negligible even for relatively small α . See Figure 2 in §4.

A nested rule based on Chebyshev nodes and a slower growing m_l is the \mathcal{R} -Leja sequence presented in [6]. Let $\{\theta_k\}_{k=1}^\infty$ be the sequence defined recursively by

$$\theta_1 = 0, \quad \theta_2 = \pi, \quad \theta_3 = \frac{\pi}{2}, \quad \text{for } j > 3, \quad \theta_j = \begin{cases} \theta_{j-1} + \pi, & j \text{ is odd} \\ \frac{1}{2}\theta_{\frac{j}{2}+1}, & j \text{ is even} \end{cases} \quad (3.11)$$

then the \mathcal{R} -Leja nodes are $y_j = \cos(\theta_j)$ with $m(l) = l + 1$, and $\lambda_l \leq 2(l+1)^2 \log(l+1)$. The single point growth gives great flexibility, since the number of nodes needed to construct $f_{\Lambda_n}(\mathbf{y})$ always equals the number of indexes in Λ_n , however, the operator norm of \mathcal{R} -Leja based interpolants is usually larger than Clenshaw-Curtis and Fejer rules.

The quadratic estimate of the Lebesgue constant for the \mathcal{R} -Leja sequence is sharp for the worst case, however, the actual penalty fluctuates between quadratic and logarithmic (in the number of nodes), which gives rise to a family of rules with different $m(l)$ [6]. First, we define the centered \mathcal{R} -Leja sequence as

$$y_1 = 0, \quad y_2 = 1, \quad y_3 = -1, \quad y_j = \cos(\theta_j), \quad (3.12)$$

where θ_j is defined as in (3.11). With the centered rule, if we select $m(l) = 2^l + 1$ (and $m(0) = 1$), then the resulting one dimensional interpolants are identical to those defined by the Clenshaw-Curtis rule. In general, \mathcal{R} -Leja sub-sequences with exponentially growing $m(l)$ exhibit linear (in l) increase of λ_l . Two specific examples include the \mathcal{R} -Leja double-2

rule defined by

$$m(0) = 1, \quad m(1) = 3, \quad \text{for } l > 1, \quad m(l) = 2^{\lfloor \frac{l}{2} \rfloor + 1} \left(1 + \frac{l}{2} - \left\lfloor \frac{l}{2} \right\rfloor \right) + 1, \quad (3.13)$$

and the \mathcal{R} -Leja double-4 rule defined by

$$m(0) = 1, \quad m(1) = 3, \quad \text{for } l > 1, \quad m(l) = 2^{2 + \lfloor \frac{l-2}{4} \rfloor} \left(1 + \frac{l-2}{4} - \left\lfloor \frac{l-2}{4} \right\rfloor \right) + 1, \quad (3.14)$$

where $\lfloor x \rfloor = \max\{i \in \mathbb{Z} : i \leq x\}$. The word *double* in the name refers to the fact that l appears in the exponents of 2 (i.e., we are doubling the number of nodes) and the numbers 2 and 4 refer to the delay of the doubling (i.e., $m(l+2) - 1 = 2(m(l) - 1)$ and $m(l+4) - 1 = 2(m(l) - 1)$ respectively). Yet another option is to use the odd sub-sequence, namely $m(l) = 2l + 1$. The \mathcal{R} -Leja odd rules result in interpolants with symmetric distribution of nodes and slightly lower operator norm. See Table 1 for a summary of the \mathcal{R} -Leja rules.

The \mathcal{R} -Leja sequence is constructed as the solution to a greedy optimization problem defined on the unit disk in the complex plane, the resulting complex nodes are projected onto the real line. The optimization problem can also be defined on $[-1, 1]$ as $y_1 = 0$ and

$$y_{j+1} = \operatorname{argmax}_{y \in [-1, 1]} \prod_{i=1}^j |y - y_i|, \quad (3.15)$$

where, if the optimization admits more than one solution, we take the right-most one. This construction leads to Leja interpolation [10] and the number of points in Leja interpolants can grow one at a time, i.e., $m(l) = l + 1$, or we can use only the odd rules, i.e., $m(l) = 2l + 1$. Unlike the \mathcal{R} -Leja sequence, there is no known sharp estimate of the operator norm of Leja interpolants, however, numerical tests shows that for $l \leq 50$ we can take $\lambda_l \approx 3(l + 1)^{1/2}$.

Alternative greedy sequences can be constructed by replacing (3.15) with maximization of the Lebesgue function

$$y_{j+1} = \operatorname{argmax}_{y \in [-1, 1]} \sum_{j'=1}^j \prod_{i=1, i \neq j'}^j \left| \frac{y - y_i}{y_{j'} - y_i} \right|, \quad (3.16)$$

minimization of $\|\mathcal{U}^{m(l)}\|_{L^\infty}$

$$y_{j+1} = \operatorname{argmin}_{y \in [-1, 1]} \max_{y' \in [-1, 1]} \prod_{i=1}^j \left| \frac{y' - y_i}{y - y_i} \right| + \sum_{j'=1}^j \left| \frac{y' - y}{y_{j'} - y} \right| \prod_{i=1, i \neq j'}^j \left| \frac{y' - y_i}{y_{j'} - y_i} \right|, \quad (3.17)$$

or minimization of $\|\Delta^{m(l)}\|_{L^\infty}$

$$y_{j+1} = \operatorname{argmin}_{y \in [-1, 1]} \max_{y' \in [-1, 1]} \left(1 + \sum_{i=1}^j \prod_{j'=1, j' \neq i}^j \left| \frac{y - y_{j'}}{y_i - y_{j'}} \right| \right) \prod_{j'=1}^j \left| \frac{y' - y_{j'}}{y - y_{j'}} \right|. \quad (3.18)$$

In (3.16)-(3.18) the growth can be prescribed either as a one or two points (i.e., $m(l) = l + 1$ or $m(l) = 2l + 1$). Numerical estimates of λ_l for each sequence are given in Figure 1.

Remark 3.5 (Alternative interpolation rules). *Each of the optimization problems (3.15) - (3.18) can be redefined over a multidimensional domain and a greedy procedure can be devised that constructs an interpolant for an arbitrary $\Lambda(L)$. However, The optimization*

Name	m_l	λ_l	y_k
Clenshaw-Curtis	$1, 2^l + 1$	$\frac{2}{\pi} \log(2^l + 1)$	see (3.8)
Fejer type 2	$2^{l+1} - 1$	$\frac{2}{\pi} \log(2^{l+1} - 1)$	see (3.10)
\mathcal{R} -Leja	$l + 1$	$1.5(l + 1)$	$\cos(\theta_k)$ (see 3.11)
\mathcal{R} -Leja double-2	see (3.13)	$1.5(l + 1)$	see (3.12)
\mathcal{R} -Leja double-4	see (3.14)	$1.5(l + 1)$	see (3.12)
\mathcal{R} -Leja odd	$2l + 1$	$3(l + 1)$	see (3.12)
Leja	$l + 1$	$3\sqrt{l + 1}$	see (3.15)
Leja odd	$2l + 1$	$6\sqrt{l + 1}$	see (3.15)
max-Lebesgue	$l + 1$	$4\sqrt{l + 1}$	see (3.16)
max-Lebesgue odd	$2l + 1$	$8\sqrt{l + 1}$	see (3.16)
min-Lebesgue	$l + 1$	$4\sqrt{l + 1}$	see (3.17)
min-Lebesgue odd	$2l + 1$	$8\sqrt{l + 1}$	see (3.17)
min-delta	$l + 1$	$3\sqrt{l + 1}$	see (3.18)
min-delta odd	$2l + 1$	$6\sqrt{l + 1}$	see (3.18)

Table 1: Summary of one dimensional rules. Note that the increase of the Lebesgue penalty λ_l is measured empirically from the first 50 nodes of the corresponding sequence.

problem is ill-conditioned and feasible only for few dimensions and moderate cardinality of $\Lambda(L)$, and therefore not practical. A notable exception is the Magic Points procedure [20], which is an extension of (3.15) to a multidimensional domain of arbitrary geometry, and the method can be used for non-polynomial interpolation. However, in the case of a hypercube Γ and a lower polynomial space, the functional associated with the Magic Points greedy problem is a product of one dimensional functionals of type (3.15), and the maximum is a tensor of one dimensional Leje nodes. Thus, one possible realization of the Magic Points algorithm is a sparse grid induced from Leja nodes.

3.4. Summary of our method. Our dynamically adaptive sparse grids interpolaiton strategy is summarized in the following algorithm:

Algorithm 1 (Anisotropic Dynamically Adaptive Multidimensional Approximation).

Given target function $f(\mathbf{y}) \in \mathbb{C}^0(\Gamma)$, where Γ is a hypercube in \mathbb{R}^d

Select one dimensional nodes $\{y_j\}_{j=1}^{\infty}$ and growth rule $m(l)$ (e.g., form Table 1)

Select initial Λ_0 , e.g., (2.7), total degree, hyperbolic cross section or Smolyak formulas.

Let $\Theta_0 = \Theta_0^{opt}$ according to (3.4)

for $n = 0, 1, 2, \dots$ **to** ∞ **do**

 Construct $f_{\Lambda_n}(\mathbf{y}) = I_{\Theta_n}[f](\mathbf{y})$ according to (3.3), i.e., compute the necessary $f(\mathbf{y}_j)$

 Compute the coefficients $\hat{c}_i^{(l)}$ according to (2.8)

 Solve the minimization problem (2.10) for $\hat{\alpha}_{n+1}$ and $\hat{\beta}_{n+1}$

 If necessary, apply the ah hoc correction of Remark 2.2

 Let $\Theta_{n+1} = \Theta_n \cup \Theta^{\hat{\alpha}_{n+1}, \hat{\beta}_{n+1}}(L_{n+1})$ according to (3.5)

 If necessary, find L_{n+1} large enough to exploit parallelism

end for

Evaluating $f_{\Lambda_n}(\mathbf{y})$ is computationally cheap and the coefficients \hat{c}_ν are computed using only $f_{\Lambda_n}(\mathbf{y})$, however, for a high dimensional problem this process can still take 10-20 minutes on 6-core CPU. A potentially cheaper alternative is to represent the interpolant using Newton hierarchical polynomials. We define the hierarchical basis

$$h_1(y) = 1, \quad \text{and for } j > 1, \quad h_j(y) = \prod_{i=1}^{j-1} \frac{y - y_i}{y_j - y_i}, \quad H_j(\mathbf{y}) = \prod_{k=1}^d h_{j_k} \quad (3.19)$$

Then, for any lower tensor set $\Theta(L)$ and $\Theta_m(L)$ as defined in (3.2), there are surplus coefficients $\{s_j\}_{j \in \Theta_m(L)}$ satisfying the system of equations $\sum_{\mathbf{1} \leq j \leq i} s_j H_j(\mathbf{y}_i) = f(\mathbf{y}_i)$ for every $\mathbf{i} \in \Theta_m(L)$ and the interpolant can be written as

$$I_{\Theta(L)}^m[f](\mathbf{y}) = \sum_{j \in \Theta_m(L)} s_j H_j(\mathbf{y}) \quad (3.20)$$

The surplus coefficients are much cheaper to compute than the Legendre coefficients, and s_j can be used as an alternative to \hat{c}_ν . When the one dimensional growth functions is $m(l) = l + 1$, then we can infer $\hat{\alpha}$ and $\hat{\beta}$ from

$$\min_{\alpha, \beta, \hat{C}} \frac{1}{2} \sum_{j \in \Theta_m(L)} \left(\hat{C} + \alpha \cdot j + \beta \cdot \log(j + 1) + \log(|s_j|) \right)^2 \quad (3.21)$$

i.e., using s_j in place of \hat{c}_ν in (2.10). However, this approach is not suited for the case when $m(l) > l + 1$. Any interpolant can be written in surplus form, and the sub-ordering of $\{y_j\}_{j=m(l-1)+1}^{m(l)}$ does not affect the nodes, polynomial space, or Lebesgue constant. However, the surpluses do depend on the sub-ordering, and therefore, s_j are influenced by λ_l associated with growth $m(l) = l + 1$ rather than $m(l) > l + 1$.

Remark 3.6. When $\Theta(L)$ is a lower index set the three constructions (3.1), (3.3) and (3.20) result in identical interpolants. In addition, the associated polynomial space is determined solely by $\Theta(L)$ and the growth sequence $m(l)$, i.e., the choice of nodes does not affect the range of the operator (only the norm). The three formulas can be generalized to not lower tensor sets $\Theta(L)$, however, if $\Theta(L)$ is not a lower set, then the three approximations are not equal and only (3.20) gives an interpolant. Furthermore, when $\Theta(L)$ is not lower, the associated polynomial spaces depend on the specific choice of y_j . In our experiments, interpolants constructed from not lower tensor sets are less accurate and we restrict our attention to lower sets $\Theta(L)$.

4. Numerical results. In this section we present several numerical examples using functions $f(\mathbf{y})$ that depend on the solutions to discretized linear and nonlinear parametric PDEs. We compare the convergence rates of SG interpolants for different selections of the tensor sets, including total degree space and the Θ_n constructed using Algorithm 1. We also test the performance of the one dimensional rules from Table 1.

4.1. Parametrized elliptic equation. The first three examples use similar setup involving the parametrized elliptic equation defined by

$$\begin{cases} -\nabla_{\mathbf{x}} \cdot (a(\mathbf{x}, \mathbf{y}) \nabla_{\mathbf{x}} u(\mathbf{x}, \mathbf{y})) = b(\mathbf{x}), & \mathbf{x} \in D \\ u(\mathbf{x}, \mathbf{y}) = 0, & \mathbf{x} \in \partial D, \end{cases} \quad (4.1)$$

where $\mathbf{y} \in \Gamma$ and $D = [0, 1] \otimes [0, 1]$. The parametrized coefficient $a(\mathbf{x}, \mathbf{y})$ is such that

$$0 < \min_{\mathbf{x}, \mathbf{y}} a(\mathbf{x}, \mathbf{y}) \leq \max_{\mathbf{x}, \mathbf{y}} a(\mathbf{x}, \mathbf{y}) < \infty,$$

in which case for every $\mathbf{y} \in \Gamma$ exists a unique $u(\mathbf{x}, \mathbf{y}) \in \mathbb{H}_0^1(D)$ that satisfies (4.1), e.g., [24]. For a bounded functional $Q : \mathbb{H}_0^1(D) \rightarrow \mathbb{R}$ we define the quantity of interest (QoI)

$$f(\mathbf{y}) = Q(u(\mathbf{x}, \mathbf{y})) : \Gamma \rightarrow \mathbb{R}.$$

The first three numerical examples differ only in the choice of $a(\mathbf{x}, \mathbf{y})$, $b(\mathbf{x})$ and $Q(u(\mathbf{x}, \mathbf{y}))$. The above setup has been thoroughly studied in literature, e.g., [1, 5, 8, 9, 23, 24], and makes for a good test bed of novel techniques.

4.1.1. Karhunen-Loéve expansion. Let $(\Omega, \mathcal{F}, \mathbb{P})$ denote a complete probability space, with sample space Ω , σ -algebra $\mathcal{F} = 2^\Omega$ and probability measure $\mathbb{P} : \mathcal{F} \rightarrow [0, 1]$. For $x \in [0, 1]$ and $\eta \in \Omega$. Let $a_1(x, \eta)$ define a random field with covariance function

$$\text{Cov} \left[\log(a_1 - 0.5) \right] (x, x') = \exp \left(-\frac{(x - x')^2}{4} \right)$$

Using Karhunen-Loéve expansion, we parametrize the random field using the dominant seven eigenvalues and eigenfunctions

$$a_1(x, \mathbf{y}) \approx 0.5 + \exp \left(1 + \frac{\sqrt[4]{9\pi}}{2} y_1 + \frac{\sqrt[4]{9\pi}}{\sqrt{2}} \sum_{k=1}^3 e^{-\frac{(k\pi)^2}{32}} (y_{2k} \sin(k\pi x_1) + y_{2k+1} \cos(k\pi x_1)) \right),$$

where \mathbf{y} is uniformly distributed over Γ [23, 24]. Assuming that the diffusion coefficient in (4.1) depends only on the first component of \mathbf{x} , we define $a_1(\mathbf{x}, \mathbf{y}) = a_1(x_1, \mathbf{y})$. The source term is $b_1(\mathbf{x}) = \cos(x_1) \sin(x_2)$ and the quantity of interest is the $L^2(D)$ norm of the solution $u(\mathbf{x}, \mathbf{y})$

$$f_1(\mathbf{y}) = \left(\int_D u^2(\mathbf{x}, \mathbf{y}) d\mathbf{x} \right)^{1/2}, \quad \mathbf{y} \in \Gamma \subset \mathbb{R}^7 \quad (4.2)$$

Analytic expression for $f_1(\mathbf{y})$ is not available, however, for a given $\mathbf{y}_j \in \Gamma$, we can approximate (4.1) with a finite element method (e.g., [27]) and compute a sufficiently accurate approximation to $f_1(\mathbf{y}_j)$. The error associated with this additional approximation is beyond the scope of this paper and we focus our attention to the discretized $f_1(\mathbf{y})$.

Figure 2 shows a comparison of eight different types of interpolation methods applied to $f_1(\mathbf{y})$. The left plot shows the results using the Clenshaw-Curtis rule, while the right plot uses the Leja nodes. Each plot shows the results from four different selections of the tensor

set and the error is estimated from 1000 uniformly distributed random samples $\{\tilde{\mathbf{y}}_i\}_{i=1}^{1000}$

$$error = \max_{1 \leq i \leq 1000} |I_{\Theta_n}[f_1](\tilde{\mathbf{y}}_i) - f_1(\tilde{\mathbf{y}}_i)|. \quad (4.3)$$

The isotropic case corresponds to the construction of isotropic total degree polynomial space, i.e., as defined in Remark 3.3. However, the e^{-k^2} term in (??) decays fast with increasing k and thus y_k variables corresponding to larger k will have lesser effect on the overall variation of $f_1(\mathbf{y})$. The isotropic approach fails to capture the different behavior of y_k and hence it is the worst performing scheme for this example. Anisotropic SG method has been proposed in [23] and using $a_1(x, \mathbf{y})$ anisotropic weights are derived

$$\boldsymbol{\alpha} = (0.85, 0.80, 0.80, 1.0, 1.0, 1.6, 1.6),$$

and the analytic anisotropic polynomial space is defined by

$$\Lambda^\alpha(L) = \{\boldsymbol{\nu} \in \mathbb{N}^d : \boldsymbol{\alpha} \cdot \boldsymbol{\nu} \leq L\}.$$

Using $\Lambda^\alpha(L)$, we construct the corresponding interpolant defined in Theorem 1, the results are plotted as the dashed lines in Figure 2.

For comparison purposes, Figure 2 also plots a *Dynamic Total Degree* example that uses a modified version of Algorithm 1. In the total degree case, we remove the β term from the weight function (2.9) and the least-squares estimate (2.10), and we compute only $\hat{\boldsymbol{\alpha}}$. For both Clenshaw-Curtis and Leja nodes, the total degree approach captures the anisotropic behavior of $f_1(\mathbf{y})$ and the method exhibits faster convergence. The estimated normalized parameters $\hat{\boldsymbol{\alpha}}$ are given on Table 2. Since the dominant polynomial space is independent from the scaling of $\hat{\boldsymbol{\alpha}}$, we divide all entries of $\hat{\boldsymbol{\alpha}}$ by the average $\frac{1}{7}|\hat{\boldsymbol{\alpha}}|_1$. Both Clenshaw-Curtis and Leja nodes result in nearly identical $\hat{\boldsymbol{\alpha}}$ even though Clenshaw-Curtis parameters are based on the projection of the interpolant (2.8) while the Leja parameters are computed using the hierarchical surpluses (3.21). The Clenshaw-Curtis nodes have a lower Lebesgue constant, however, the Clenshaw-Curtis nodes result in grids with more nodes due to the exponential growth of $m(l)$. For this particular problem using dynamic total degree nodes, the lower operator norm of Clenshaw-Curtis nodes leads to faster convergence.

Figure 2 also shows the results from applying Algorithm 1 to $f_1(\mathbf{y})$, and the curve is labeled *Dynamic Curved*. When using Clenshaw-Curtis nodes, there is no significant difference between the total degree method and the quasi-optimal interpolation (2.7). The estimated (normalized) decay parameters $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ are listed in Table 3 and we see that $\hat{\boldsymbol{\beta}}$ is relatively small. In contrast, when using the Leja nodes, the quasi-optimal estimate (2.7) leads to a significant improvement in convergence and from Table 3 we see that the $\hat{\boldsymbol{\beta}}$ estimated parameters are smaller compared to the Clenshaw-Curtis ones, which is due to the larger Lebesgue constant.

Figure 3 shows the results of applying Algorithm 1 to $f_1(\mathbf{y})$ and using different rules from Table 1. The curves are not smooth since the results are affected by error in the least-squares approximation, fluctuations of the operator norm (see Figure 1), the random samples used to compute the error (4.3), and, when $m(l) > l + 1$ the range of the resulting interpolant is a super-set of the estimated optimal polynomial space. However, all interpolants have similar convergence rate within the “noise” in the method. Table 4 gives the relative variation of $\hat{\boldsymbol{\alpha}}$

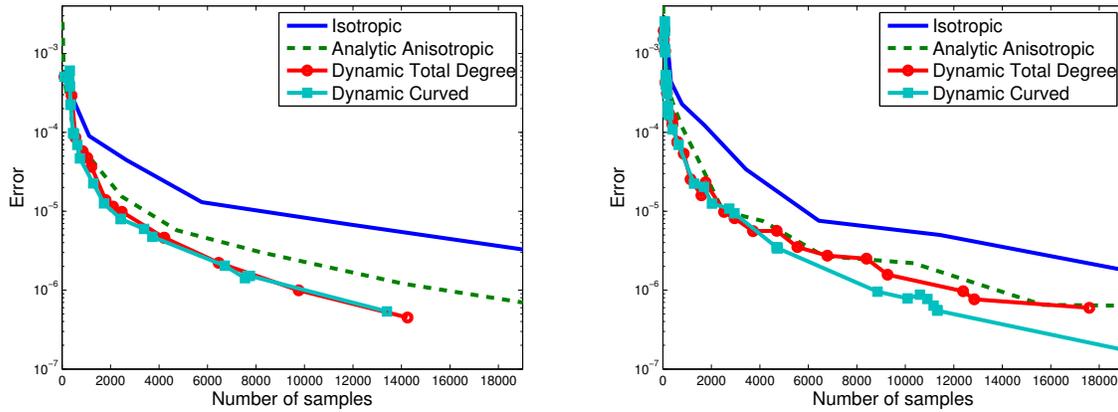


Figure 2: Applying four different construction of interpolants for $f_1(\mathbf{y})$. Left: using Clenshaw-Curtis nodes. Right: using Leja nodes.

Dimension	$\hat{\alpha}_1$	$\hat{\alpha}_2$	$\hat{\alpha}_3$	$\hat{\alpha}_4$	$\hat{\alpha}_5$	$\hat{\alpha}_6$	$\hat{\alpha}_7$
Clenshaw-Curtis	0.58	0.70	0.57	0.99	0.93	1.57	1.66
Leja	0.61	0.70	0.58	0.96	0.91	1.55	1.68
Analytic	0.78	0.73	0.73	0.92	0.92	1.46	1.46

Table 2: Estimated $\hat{\alpha}$ parameters for $f_1(\mathbf{y})$ using total degree estimate.

Dimension	$\hat{\alpha}_1$	$\hat{\alpha}_2$	$\hat{\alpha}_3$	$\hat{\alpha}_4$	$\hat{\alpha}_5$	$\hat{\alpha}_6$	$\hat{\alpha}_7$
Clenshaw-Curtis	0.64	0.77	0.80	0.97	0.99	1.48	1.34
Leja	0.67	0.80	0.85	1.08	1.05	1.43	1.12
Dimension	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	$\hat{\beta}_5$	$\hat{\beta}_6$	$\hat{\beta}_7$
Clenshaw-Curtis	-0.46	-0.51	-0.94	-0.31	-0.47	-0.29	0.09
Leja	-1.00	-1.14	-1.68	-1.24	-1.27	-0.96	-0.21

Table 3: Estimated $\hat{\alpha}$ and $\hat{\beta}$ parameters for $f_1(\mathbf{y})$.

Dimension	$\delta\hat{\alpha}_1$	$\delta\hat{\alpha}_2$	$\delta\hat{\alpha}_3$	$\delta\hat{\alpha}_4$	$\delta\hat{\alpha}_5$	$\delta\hat{\beta}_1$	$\delta\hat{\beta}_2$	$\delta\hat{\beta}_3$	$\delta\hat{\beta}_4$	$\delta\hat{\beta}_5$
Clenshaw-Curtis	0.80	0.45	0.82	0.50	0.67	1.27	0.88	0.93	1.70	0.89

Table 4: Relative variation of $\hat{\alpha}$ and $\hat{\beta}$ parameters for $f_1(\mathbf{y})$ over all rules from Table 1 .

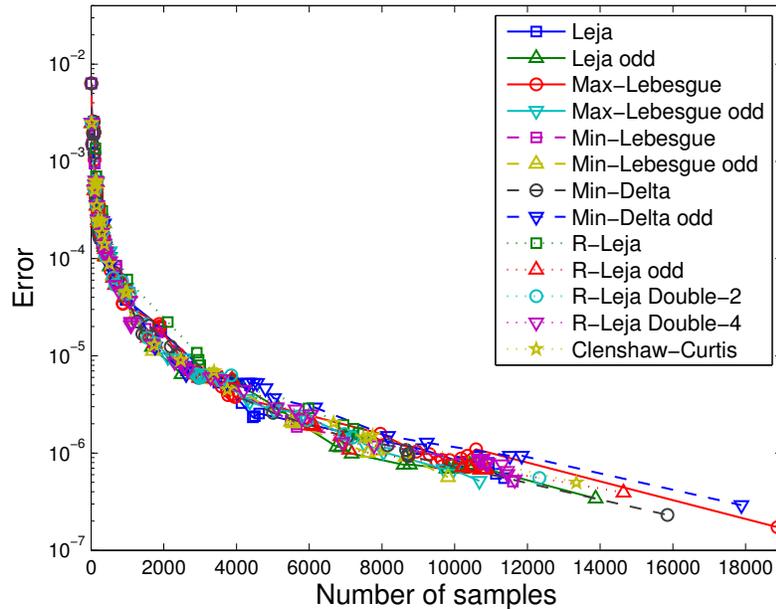


Figure 3: Algorithm 1 applied to $f_1(\mathbf{y})$ using all rules from Table 1.

and $\hat{\beta}$ components for all rules in Table 1, where $\delta\alpha_k$ is computed as the difference between the largest and smallest estimate for α_k divided by the average (same for β_k). From the first 5 components, we see that the components of $\hat{\beta}$ vary over a much wider range than those of $\hat{\alpha}$, since the $\hat{\alpha}$ is affected by only by the “noise” while $\hat{\beta}$ is also affected by the different Lebesgue constants for the different rules. Note that the table lists only the first 5 components, since y_6 and y_7 are associated with the smallest eigenvalue of the Karhunen-Loève expansion, $f_1(\mathbf{y})$ is least sensitive to y_6 and y_7 , and the constructed interpolants have few nodes in those two directions, which leads to unreliable estimates.

4.1.2. Inclusion problem. In this section we apply our approach to two inclusion problems, presented in [1], where we consider the domain given on Figure 4.

Isotropic case. The (almost) isotropic inclusion problem associates each of the eight disks with a component of $\mathbf{y} \in \bigotimes_{k=1}^8 [0.01, 1]$ and we define

$$a_2(\mathbf{x}, \mathbf{y}) = \begin{cases} y_k, & \mathbf{x} \in \text{Disk } k \\ 1, & \text{otherwise.} \end{cases} \quad b_2(\mathbf{x}) = \begin{cases} 100, & \mathbf{x} \in \text{Box} \\ 0, & \text{otherwise.} \end{cases} \quad f_2(\mathbf{y}) = \int_{\text{Box}} u(\mathbf{x}, \mathbf{y}) d\mathbf{x}$$

Figure 5 shows a plot of the convergence rate of interpolants based on Clenshaw-Curtis and Leja nodes. Figure 5 compares three possible selections of the polynomial space, isotropic total degree (see Remark 3.3), dynamic curved (i.e., Algorithm 1), and dynamic total degree (i.e., Algorithm 1 ignoring β). The disks are not equidistant from the box and thus the components of \mathbf{y} do not have equal influence on $f_2(\mathbf{y})$, however, the anisotropy is weak and the performance of the the anisotropic total selection is indistinguishable from the isotropic

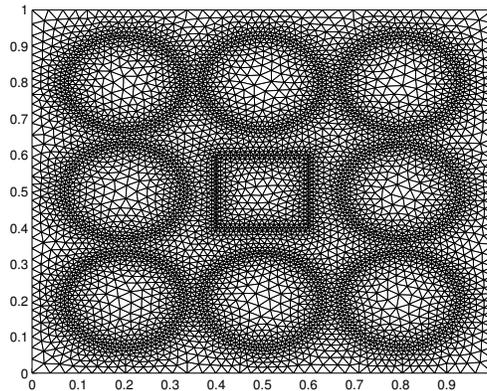


Figure 4: Mesh for the inclusion problem. Each disk has radius $\frac{2}{15}$ and the centers are at 0.2, 0.5 and 0.8 along the horizontal and vertical axis. The square box is located at the center and has side of 0.2.

Dimension	$\hat{\alpha}_1$	$\hat{\alpha}_2$	$\hat{\alpha}_3$	$\hat{\alpha}_4$	$\hat{\alpha}_5$	$\hat{\alpha}_6$	$\hat{\alpha}_7$	$\hat{\alpha}_8$
Clenshaw-Curtis	0.86	1.32	0.67	1.32	1.25	0.67	1.26	0.66
Leja	0.94	1.32	0.66	1.32	1.22	0.66	1.22	0.66
Dimension	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	$\hat{\beta}_5$	$\hat{\beta}_6$	$\hat{\beta}_7$	$\hat{\beta}_8$
Clenshaw-Curtis	0.66	-0.49	1.41	-0.50	-0.20	1.40	-0.20	1.47
Leja	0.17	-1.24	0.75	-1.24	-0.86	0.74	-0.86	0.79

Table 5: Estimated $\hat{\alpha}$ and $\hat{\beta}$ parameters for $f_2(\mathbf{y})$.

interpolant. On the other hand, our quasi-optimal approach has a significantly better convergence. Table 5 gives the final values of the estimated $\hat{\alpha}$ and $\hat{\beta}$ parameters, similar to the Karhunen-Loéve example, the $\hat{\alpha}$ parameters are almost identical for both Clenshaw-Curtis and Leja nodes, however, the $\hat{\beta}$ parameters for the Leja nodes are smaller due to the larger operator norm. Despite the seemingly small values of $\hat{\beta}$, the added flexibility of the log term in (2.7) results in accurate interpolants with thousands of fewer nodes as compared to the total degree construction.

It is interesting to note that several of the $\hat{\beta}_k$ estimates in Table 5 are in fact positive, as opposed to the negative values suggested in the theoretical derivation of (2.7) and Assumption 2. This indicates that the coefficients \hat{c}_i exhibit a combination of exponential and algebraic decay and the added flexibility of β (i.e., imposing no restrictions on the individual $\hat{\beta}_k$), results in a more accurate estimate of the optimal polynomial space. Thus, our approach is applicable to a wider class of functions, namely those obeying estimate (2.7) for any β , albeit rigorous characterization of this class of functions is not currently available.

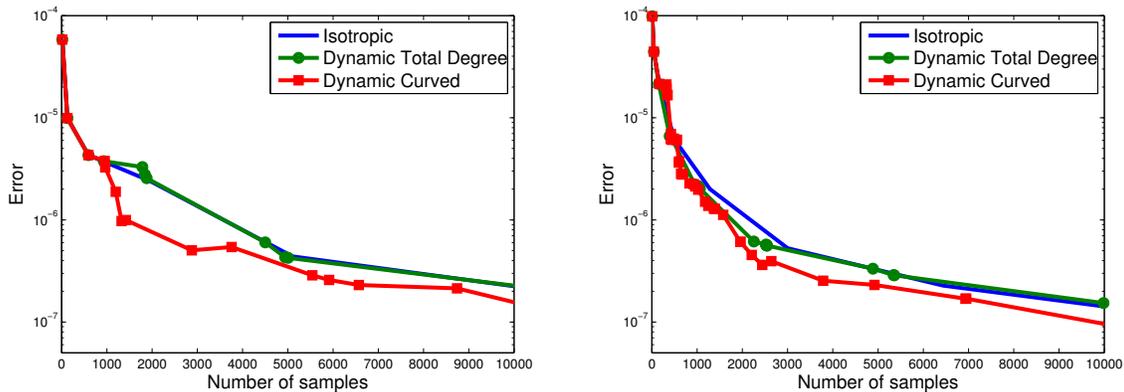


Figure 5: Applying three different construction of interpolants for $f_2(\mathbf{y})$. Left: using Clenshaw-Curtis nodes. Right: using Leja nodes.

Anisotropic case. The second variation of the inclusion problem uses only the four corner disks

$$\begin{aligned} y_{\text{top-left}} &\in [0.109, 1], & y_{\text{top-right}} &\in [0.2575, 1], \\ y_{\text{bottom-left}} &\in [0.010, 1], & y_{\text{bottom-right}} &\in [0.4060, 1]. \end{aligned}$$

The diffusion coefficient is constant 1 outside the four boxes. The four parameter forcing term and quantity of interest are now defined over the entire domain, i.e., $b_3(\mathbf{x}) = 100$ and

$$f_3(\mathbf{y}) = \int_D u(\mathbf{x}, \mathbf{y}) d\mathbf{x}. \quad (4.4)$$

Despite the smaller number of dimensions, the four parameter problem is more difficult, due to the smaller a_{\min} and smaller region of analyticity. For this problem, anisotropic total degree weights have been derived

$$\boldsymbol{\alpha} = (40, 317, 137, 227), \quad (4.5)$$

and numerical results show that (4.5) are reasonably accurate when used in the context of Galerkin projection [1]. Figure 6 shows a comparison in the convergence rate between interpolants constructed via the total degree weights given in (4.5) and the application of Algorithm 1, Table 6 lists the estimated $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ parameters. The convergence rate of the interpolants is closer to algebraic, leading to a large $\boldsymbol{\beta}$, which dominates for indexes with small entries. Thus, in the context of interpolation, the total degree space estimated by the weights (4.5) is far from optimal. Initially, the least squares method struggles to estimate the quasi-optimal parameters, however, once a sufficient number of samples are computed, the convergence rate of Algorithm 1 dramatically increases leading to enormous savings compared to the total degree selection. Furthermore, due to the small region of analyticity of $f_3(\mathbf{y})$, the difference in Lebesgue constant between Clenshaw-Curtis and Leja has a significant effect with the former method outperforming the latter.

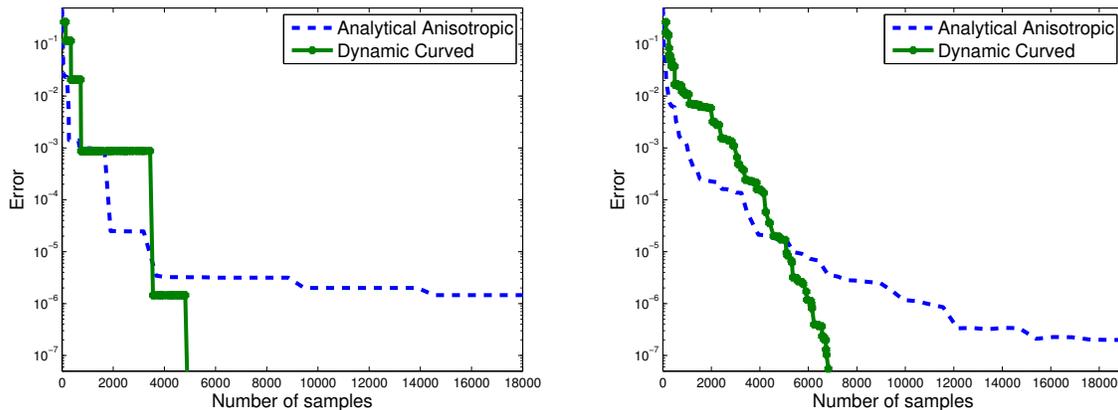


Figure 6: Applying three different construction of interpolants $f_3(\mathbf{y})$. Left: using Clenshaw-Curtis nodes. Right: using Leja nodes.

Dimension	$\hat{\alpha}_1$	$\hat{\alpha}_2$	$\hat{\alpha}_3$	$\hat{\alpha}_4$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$
Clenshaw-Curtis	0.18	2.38	0.18	1.27	12.70	10.64	13.62	12.00
Leja	0.24	2.22	0.24	1.30	6.95	5.69	8.75	6.94

Table 6: Estimated $\hat{\alpha}$ and $\hat{\beta}$ parameters for $f_3(\mathbf{y})$.

4.2. Parametrized steady state Burgers equation. Consider the steady state Burgers equation defined over the domain $D = [0, 1] \otimes [0, 0.5] \setminus [0.15, 0.25] \otimes [0.15, 0.35]$ (see Figure 7). The right most wall of the domain is associated with homogenous Neumann boundary conditions, i.e., $\partial D_n = \{1\} \otimes [0, 0.5]$, the remainder of the boundary, i.e., ∂D_d , is associated with the Dirichlet conditions defined by

$$u_b(\mathbf{x}) = \begin{cases} 16x_2(0.5 - x_2), & x_1 = 0 \\ 0, & \text{otherwise.} \end{cases} \quad (4.6)$$

A time dependent control problem using the above domain is described in [13]. In our context, we are interested in parametrizing the steady state Burgers equation given by

$$\begin{cases} -\nabla_{\mathbf{x}} \cdot (a(\mathbf{y})\nabla_{\mathbf{x}}u(\mathbf{x}, \mathbf{y})) + (\mathbf{v}(\mathbf{y}) \cdot \nabla_{\mathbf{x}}u(\mathbf{x}, \mathbf{y}))u(\mathbf{x}, \mathbf{y}) = 0, & \mathbf{x} \in D, \\ u(\mathbf{x}, \mathbf{y}) = u_b(\mathbf{x}), & \mathbf{x} \in \partial D_d, \\ \frac{\partial}{\partial x_1}u(\mathbf{x}, \mathbf{y}) = 0, & \mathbf{x} \in \partial D_n, \end{cases} \quad (4.7)$$

where $\mathbf{y} \in \otimes_{k=1}^3[-1, 1]$ and

$$a(\mathbf{y}) = \frac{1}{200 + 100y_3}, \quad \mathbf{v}(\mathbf{y}) = \begin{pmatrix} 1 + 0.2y_1 \\ 0.1y_2 \end{pmatrix}, \quad f_4(\mathbf{y}) = \int_{0.05}^{0.45} \int_{0.6}^{0.8} u(\mathbf{x}, \mathbf{y}) dx_1 dx_2$$

Figure 7 plots the nominal solution corresponding to $\mathbf{y} = \mathbf{0}$. There is no available result regarding the analyticity of $f_4(\mathbf{y})$ or the corresponding optimal polynomial space. However, the components of the convection term $\mathbf{v}(\mathbf{y})$ affect different derivatives of the non-uniform solution, y_1 and y_2 are multiplied by different coefficients, and y_3 affects the diffusion term

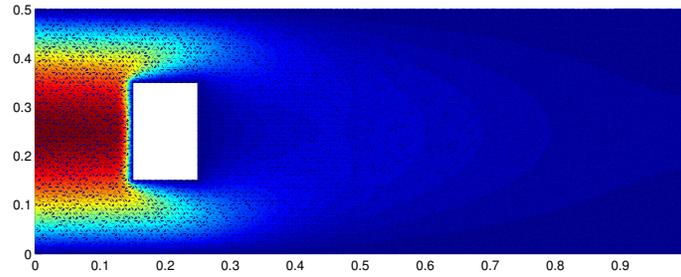


Figure 7: The domain associated with the Burgers equation (4.7), the color map corresponds to the nominal solution $\mathbf{y} = \mathbf{0}$.

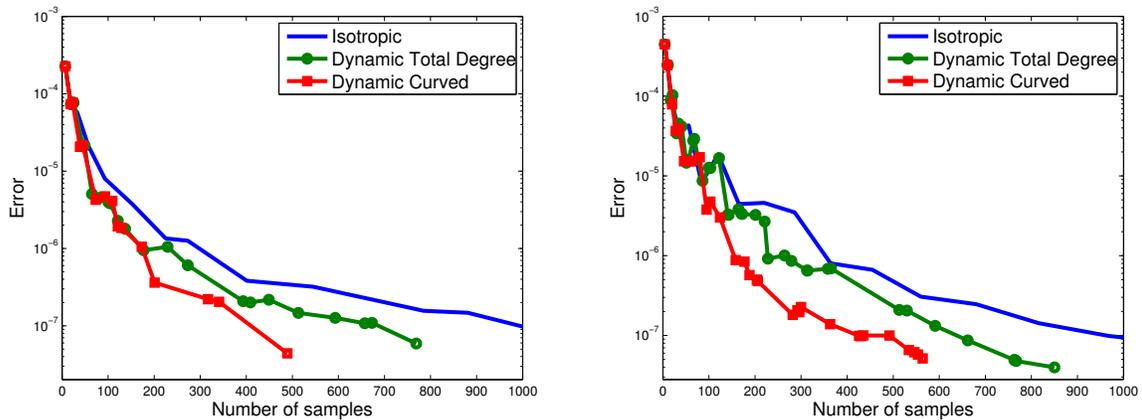


Figure 8: Applying three different construction of interpolants for $f_4(\mathbf{y})$. Left: using Clenshaw-Curtis nodes. Right: using Leja nodes.

which has very different effect on $u(\mathbf{x}, \mathbf{y})$. Thus, we expect anisotropic decay behavior of $f_4(\mathbf{y})$.

Figure 8 plots the result of interpolating $f_4(\mathbf{y})$ with three different polynomial selection schemes, the isotropic total degree (see Remark 3.3), and Algorithm 1 ignoring and including β . We use Leja and Clenshaw-Curtis nodes and the estimated parameters are given in Table 7. We observe results very similar to the previous examples, which shows that our approach extends to problems associated with nonlinear PDEs.

5. Conclusion. In this work, we have presented an adaptive algorithm for constructing a sequence of interpolants of a function defined over a product of one dimensional intervals and admitting analytic extension to poly-ellipse in complex space. Following recent results in “best M -term” approximation, we derive a heuristic estimate for the optimal interpolation space, which we parametrize by two vectors, one depending on the region of analytic extension of the function and one depending on the Lebesgue constant of the interpolation scheme. Traditional methods for (quasi-)optimal approximation rely on a priori estimates, instead, we present a procedure for constructing a sequence of polynomial spaces, where

Dimension	$\hat{\alpha}_1$	$\hat{\alpha}_2$	$\hat{\alpha}_3$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$
Clenshaw-Curtis	1.22	1.53	0.25	-1.57	-1.71	0.25
Leja	1.18	1.46	0.35	-2.37	-2.40	-0.76

Table 7: Estimated $\hat{\alpha}$ and $\hat{\beta}$ parameters for $f_4(\mathbf{y})$.

each space is derived from an estimate inferred from an interpolant in the previous space. Each interpolant is constructed using a sparse grids approach, and we present a strategy for selecting the tensor rules so that the resulting grid is optimal (i.e., fewest nodes) with respect to the polynomial space. We also present several novel interpolation rules derived from greedy optimization of operator norms. Our numerical experiments demonstrate that the method can be applied to a wide range of problems without the need for a priori estimates of the region of analyticity of the function, so long as the function is analytic and the one dimensional family of rules inducing the sparse grid exhibits polynomial growth of the Lebesgue constant.

Acknowledgements. This material is based upon work supported in part by the U.S. Air Force of Scientific Research under grant number 1854-V521-12 and by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under contract and award numbers ERKJ259; 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 Contract DE-AC05-00OR22725.

REFERENCES

- [1] J. BECK, F. NOBILE, L. TAMELLINI, AND R. TEMPONE, *Convergence of quasi-optimal stochastic Galerkin methods for a class of PDEs with random coefficients*, Computers & Mathematics with Applications, 67 (2014), pp. 732 – 751. High-order Finite Element Approximation for Partial Differential Equations.
- [2] P. BINEV, A. COHEN, W. DAHMEN, R. DEVORE, G. PETROVA, AND P. WOJTASZCZYK, *Convergence rates for greedy algorithms in reduced basis methods*, SIAM Journal on Mathematical Analysis, 43 (2011), pp. 1457–1472.
- [3] H.-J. BUNGARTZ AND M. GRIEBEL, *Sparse grids*, Acta numerica, 13 (2004), pp. 147–269.
- [4] A. CHKIFA, A. COHEN, R. DEVORE, AND C. SCHWAB, *Sparse adaptive Taylor approximation algorithms for parametric and stochastic elliptic PDEs*, ESAIM: Mathematical Modelling and Numerical Analysis, 47 (2013), pp. 253–280.
- [5] A. CHKIFA, A. COHEN, AND C. SCHWAB, *High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs*, Foundations of Computational Mathematics, 14 (2014), pp. 601–633.
- [6] M. A. CHKIFA, *On the Lebesgue constant of Leja sequences for the complex unit disk and of their real projection*, Journal of Approximation Theory, 166 (2013), pp. 176–200.
- [7] C. W. CLENSHAW AND A. R. CURTIS, *A method for numerical integration on an automatic computer*, Numerische Mathematik, 2 (1960), pp. 197–205.
- [8] A. COHEN, R. DEVORE, AND C. SCHWAB, *Convergence rates of best N -term Galerkin approximations for a class of elliptic sPDEs*, Foundations of Computational Mathematics, 10 (2010), pp. 615–646.

- [9] A. COHEN, R. DEVORE, AND C. SCHWAB, *Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDEs*, Analysis and Applications, 9 (2011), pp. 11–47.
- [10] S. DE MARCHI, *On Leja sequences: some results and applications*, Applied Mathematics and Computation, 152 (2004), pp. 621–647.
- [11] R. DEVORE, G. PETROVA, AND P. WOJTASZCZYK, *Greedy algorithms for reduced bases in Banach spaces*, Constructive Approximation, 37 (2013), pp. 455–466.
- [12] N. DEXTER, C. G. WEBSTER, AND G. ZHANG, *Explicit cost bounds of stochastic Galerkin approximations for parametrized PDEs with random coefficients*, Tech. Rep. ORNL/TM-2015/328, Oak Ridge National Laboratory, One Bethel Valley Road, Oak Ridge, TN, 2015.
- [13] S. M. DJOUADI, R. C. CAMPHOUSE, AND J. H. MYATT, *Optimal order reduction for the two-dimensional Burgers' equation*, in Decision and Control, 2007 46th IEEE Conference on, IEEE, 2007, pp. 3507–3512.
- [14] A. DOOSTAN AND G. IACCARINO, *A least-squares approximation of partial differential equations with high-dimensional random inputs*, Journal of Computational Physics, 228 (2009), pp. 4332–4345.
- [15] A. DOOSTAN AND H. OWHADI, *A non-adapted sparse approximation of PDEs with stochastic inputs*, Journal of Computational Physics, 230 (2011), pp. 3015–3034.
- [16] L. FÉJER, *On the infinite sequences arising in the theories of harmonic analysis, of interpolation, and of mechanical quadratures*, Bulletin of the American Mathematical Society, 39 (1933), pp. 521–534.
- [17] G. FISHMAN, *Monte Carlo*, Springer Series in Operations Research, Springer-Verlag, New York, 1996. Concepts, algorithms, and applications.
- [18] M. D. GUNZBURGER, C. G. WEBSTER, AND G. ZHANG, *Stochastic finite element methods for partial differential equations with random input data*, Acta Numerica, 23 (2014), pp. 521–650.
- [19] M. LOÈVE, *Probability theory I*, 4th ed., Grad. Texts in Math. 45, Springer, 1987.
- [20] Y. MADAY, N. C. NGUYEN, A. T. PATERA, AND G. S. PAU, *A general, multipurpose interpolation procedure: the magic points*, Communications on Pure and Applied Analysis, 8 (2009).
- [21] H. NIEDERREITER, *Quasi-Monte Carlo Methods*, Wiley Online Library, 1992.
- [22] F. NOBILE, L. TAMELLINI, AND R. TEMPONE, *Convergence of quasi-optimal sparse grid approximation of Hilbert-valued functions: application to random elliptic PDEs*, Tech. Rep. Nr. 12.2014, Mathematics Institute of Computational Science and Engineering, MATHICSE Station 8 - CH-1015 - Lausanne - Switzerland, 2014.
- [23] F. NOBILE, R. TEMPONE, AND C. G. WEBSTER, *An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM Journal on Numerical Analysis, 46 (2008), pp. 2411–2442.
- [24] ———, *A sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM Journal on Numerical Analysis, 46 (2008), pp. 2309–2345.
- [25] E. NOVAK AND K. RITTER, *Simple cubature formulas with high polynomial exactness*, Constructive approximation, 15 (1999), pp. 499–522.
- [26] M. STOYANOV AND C. G. WEBSTER, *A gradient-based sampling approach for dimension reduction of partial differential equations with stochastic coefficients*, International Journal for Uncertainty Quantification, 5 (2015).
- [27] G. STRANG AND G. J. FIX, *An analysis of the finite element method*, vol. 212, Prentice-Hall Englewood Cliffs, NJ, 1973.
- [28] R. A. TODOR AND C. SCHWAB, *Convergence rates for sparse chaos approximations of elliptic problems with stochastic coefficients*, IMA Journal of Numerical Analysis, 27 (2007), p. 232.

- [29] H. TRAN, C. G. WEBSTER, AND G. ZHANG, *Analysis of quasi-optimal polynomial approximations for parametrized PDEs with deterministic and stochastic coefficients*, Tech. Rep. ORNL/TM-2014/468, Oak Ridge National Laboratory, One Bethel Valley Road, Oak Ridge, TN, 2014.

