Sparse Pseudospectral Approximation Method

Paul G. Constantine[a], Michael S. Eldred[a], Eric T. Phipps[a]

[a]*Sandia National Laboratories, Albuquerque, NM 87185*

Abstract

Multivariate global polynomial approximations – such as polynomial chaos or stochastic collocation methods – are now in widespread use for sensitivity analysis and uncertainty quantification. The pseudospectral variety of these methods uses a numerical integration rule to approximate the Fourier-type coefficients of a truncated expansion in orthogonal polynomials. For problems in more than two or three dimensions, a sparse grid numerical integration rule offers accuracy with a smaller node set compared to tensor product approximation. However, when using a sparse rule to approximately integrate these coefficients, one often finds unacceptable errors in the coefficients associated with higher degree polynomials.

By reexamining Smolyak's algorithm and exploiting the connections between interpolation and projection in tensor produce spaces, we construct a *sparse pseudospectral approximation method* that accurately reproduces the coefficients of basis functions that naturally correspond to the sparse grid integration rule. The compelling numerical results show that this is the proper way to use sparse grid integration rules for pseudospectral approximation.

Keywords: sparse grids, pseudospectral methods, polynomial chaos, stochastic collocation, non-intrusive spectral projection

1. Introduction

As the power and availability of computers has increased, the profile of simulation in scientific and engineering endeavors has risen. Computer simu-

[1]Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

lations that model complex physical phenomena now regularly aid in decision making and design processes. However, the complexity and computational cost of the codes often renders them impractical for design and uncertainty studies, where many runs at different input parameter values are necessary to compute statistics of interest. In such cases, designers use a relatively small number of high fidelity runs to build cheaper surrogate models, which are then used for the studies requiring many model evaluations.

It is now common to use a multivariate global polynomial of the input parameters as the surrogate, particularly when one desires estimates of integrated quantities such as mean and variance of simulation results. Additionally, the polynomial surrogate is typically much cheaper to evaluate as a function of the input parameters, which allows sampling and optimization studies at a fraction of the cost. In an uncertainty quantification context – where the input parameters often carry the interpretation of random variables – this polynomial approximation method appears under the labels *polynomial chaos* [1, 2] or *stochastic collocation* [3, 4], amongst others.

One of the primary disadvantages of the polynomial methods is the rapid growth in the work required to compute the approximation as the number of model input parameters increases; this generally limits the applicability of these methods to models with fewer than ten input parameters. To combat this apparent curse of dimensionality, many have proposed to use so-called sparse grid methods [5], which deliver comparable accuracy for some problems using far fewer function evaluations to build the surrogate. The sparse grid is a set of points in the input parameter space that is the union of carefully chosen tensor product grids. When the tensor grids are formed from univariate point sets with a nesting property, such as the Chebyshev points, the number of points in the union of tensor grids is greatly reduced – although this nesting feature is not necessary for the construction of the sparse grids. The points in the sparse grid can be used as a numerical integration rule [6, 7], where the weights are linear combinations of weights from the member tensor grids. Alternatively, the interpolating tensor product Lagrange polynomials constructed on the member tensor grids can be linearly combined in a similar fashion to yield a polynomial surrogate [8], since a linear combination of polynomials is itself a polynomial.

Another popular polynomial representation employs a multivariate orthogonal polynomial basis. When the coefficients of a series in this basis are computed by projecting the unknown function onto each basis, the series is a spectral projection or Fourier series [9, 10]; this is also known as the poly-

nomial chaos expansion in an uncertainty quantification context [1, 2]. The series must be truncated for computation; convergence to the true function occurs in the mean-squared sense as one adds more basis polynomials. If the integrals in the projections are approximated with a numerical integration rule, this method is known as a pseudospectral projection [11, 12]. These integral approximations only require the simulation outputs evaluated at the quadrature points of the input space.

The question that naturally arises is: *Which numerical integration rule is appropriate to approximate the Fourier coefficients?* Some early attempts used Monte Carlo integration [13], but its relative inaccuracies overwhelm the spectral accuracy of the truncated Fourier series. Other attempts used tensor product Gaussian quadrature rules, but they do not scale to high dimensional parameter spaces due to the exponential increase in the number of quadrature points with dimension. The sparse-grid quadrature rules have shown promise for retaining the spectral accuracy while alleviating the curse of dimensionality. However, in practice this approach produces unacceptable errors in the coefficients associated with the higher order basis polynomials, which forces a much stricter truncation than might be expected for the number of function evaluations [14].

This paper presents a *sparse pseudospectral approximation method* (SPAM) for computing the coefficients of the truncated Fourier series with the points of the sparse grid integration rule that eliminates the error in the coefficients associated with higher degree polynomials. This allows the number of terms in the expansion to be consistent with the number of points in the sparse-grid integration rule. The key is to separately compute the coefficients of a tensor product polynomial expansion for each tensor grid in the sparse grid. The linear combination of the tensor weights used to produce the sparse-grid integration weights is then used to linearly combine the coefficients of each tensor expansion. We show that this method produces a pointwise equivalent polynomial surrogate to the one constructed from a linear combination of tensor product Lagrange polynomials. Therefore error bounds from that context can be applied directly. Recently, in the context of spectral methods for discretized PDEs, Shen and coauthors [15, 16] proposed and analyzed a closely related sparse spectral approximation using a hierarchical basis of Chebyshev polynomials; the hierarchical structure results in increased efficiency.

The remainder of the paper is structured as follows. In Section 2, we review the relationship between Lagrange polynomial interpolation on a set of

quadrature points and the pseudospectral approximation for univariate functions; we then extend this analysis to multivariate tensor product approximation. Section 2 closes with a review of Smolyak's algorithm. In Section 3, we detail the SPAM for approximating the Fourier coefficients using the function evaluations at the sparse-grid integration points followed by some interesting analysis results. In Section 4, we present numerical experiments from (i) a collection of scalar bivariate functions and (ii) an elliptic PDE model with parameterized coefficients. In each experiment, we compare the approximate Fourier coefficients from the SPAM with ones computed directly with the sparse grid integration rule. Finally we conclude with a summary and discussion in Section 5.

2. Background and Problem Set-up

In this section, we briefly review the background necessary to understand the SPAM; in particular, we examine the relationship between the Lagrange interpolation on a set of Gaussian quadrature points and a pseudospectral approximation in a basis of orthonormal polynomials. One purpose of this review is to set up the notation, which departs slightly from the notation in the disparate references. For the orthogonal polynomials, we follow the notation of [17].

Consider a multivariate function $f : \mathcal{S} \to \mathbb{R}$, where the domain $\mathcal{S} \subset \mathbb{R}^d$ has a product structure

$$\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_d. \tag{1}$$

Define a d-dimensional point $s = (s_1, \ldots, s_d) \in \mathcal{S}$. The domain is equipped with a positive, separable weight function $w : \mathcal{S} \to \mathbb{R}_+$ where $w(s) = w_1(s_1) \cdots w_d(s_d)$ and

$$\int_{\mathcal{S}_k} s_k^a \, w_k(s_k) \, ds_k < \infty, \qquad k = 1, \ldots, d, \quad a = 1, 2, \ldots \tag{2}$$

The w_k are normalized to integrate to 1, which allows the interpretation of $w(s)$ as a probability density function. In general, we consider functions which are square-integrable on $\mathcal{S}$, i.e.

$$\int_{\mathcal{S}} f(s)^2 \, w(s) \, ds \; < \; \infty. \tag{3}$$

Such functions admit a convergent Fourier series in orthonormal basis polynomials,

$$f(s) \;=\; \sum_{i_1=1}^{\infty} \cdots \sum_{i_d=1}^{\infty} \hat{f}_{i_1,\ldots,i_d}\, \pi_{i_1}(s_1) \cdots \pi_{i_d}(s_d) \;=\; \sum_{\mathbf{i}\in\mathbb{N}^d} \hat{f}_{\mathbf{i}}\, \pi_{\mathbf{i}}(s), \tag{4}$$

where the equality is in the L_2 sense, $\mathbf{i} = (i_1, \ldots, i_d)$ is a multi-index, and

$$\hat{f}_{\mathbf{i}} \;=\; \int_{\mathcal{S}} f(s)\, \pi_{\mathbf{i}}(s)\, w(s)\, ds \tag{5}$$

is the Fourier coefficient associated with the basis polynomial $\pi_{\mathbf{i}}(s)$. The $\pi_{i_k}(s_k)$ are univariate polynomials in s_k of degree $i_k - 1$ that are orthonormal with respect to $w_k(s_k)$. In general, a pseudospectral method uses a numerical integration rule to approximate a subset of the integrals (5); the remaining terms are discarded.

While any square-integrable function admits a convergent Fourier series in theory, the polynomial approximation methods perform best on a much smaller class of smooth functions; we will restrict our attention to such function classes when citing appropriate error bounds. Before diving into the multivariate approximation, we first review the univariate case.

2.1. Gaussian Quadrature, Collocation, Pseudospectral Methods

Consider the problem set-up above with $d = 1$. Let $\boldsymbol{\pi}(s) = [\pi_1(s), \ldots, \pi_n(s)]^T$ be a vector of the first n polynomials that are orthonormal with respect to the weight function $w(s)$. The components of $\boldsymbol{\pi}(s)$ satisfy a recurrence relationship, which we can write in matrix form as

$$s\boldsymbol{\pi}(s) = \mathbf{J}\boldsymbol{\pi}(s) + \beta_{n+1}\pi_{n+1}(s)\mathbf{e}_n, \tag{6}$$

where $\mathbf{e}_n$ is an n-vector of zeros with a one in the last entry, and $\mathbf{J}$ (known as the *Jacobi matrix*) is the symmetric, tridiagonal matrix containing the recurrence coefficients,

$$\mathbf{J} = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-1} & \alpha_{n-1} & \beta_n \\ & & & \beta_n & \alpha_n \end{bmatrix}. \tag{7}$$

The zeros of $\pi_{n+1}(s)$ generate eigenvalue/eigenvector pairs of $\mathbf{J}$ by (6), which we write

$$\mathbf{J} = \mathbf{Q}\Lambda\mathbf{Q}^T, \qquad \Lambda = \mathrm{diag}\big([\lambda_1, \ldots, \lambda_n]\big), \tag{8}$$

where $\mathbf{Q}(i,j) = \pi_i(\lambda_j)/\|\boldsymbol{\pi}(\lambda_j)\|$ are the elements of the normalized eigenvectors. The zeros λ_j of $\pi_{n+1}(s)$ are the points of the n-point Gaussian quadrature rule for $w(s)$; the quadrature weights $\nu_j \in \mathbb{R}_+$ are given by

$$\nu_j = \frac{1}{\|\boldsymbol{\pi}(\lambda_j)\|^2}, \tag{9}$$

which are the squares of the first component of the jth eigenvector. A Gaussian quadrature approximation to the integral is written

$$\int_{\mathcal{S}} f(s)\, w(s)\, ds \;\approx\; \mathcal{U}_q^n(f) \;=\; \sum_{j=1}^{n} f(\lambda_j)\, \nu_j \;=\; \mathbf{f}^T\boldsymbol{\nu}. \tag{10}$$

The $\mathcal{U}_q^n$ denotes the linear operation of quadrature applied to f; the subscript q is for quadrature. This notation will be used later when discussing sparse grids. The n-vector $\mathbf{f}$ contains the evaluations of $f(s)$ at the quadrature points, and the n-vector $\boldsymbol{\nu}$ contains the weights of the quadrature rule. It will be notationally convenient to define the matrices

$$\mathbf{P}(i,j) = \pi_i(\lambda_j), \qquad \mathbf{W} = \mathrm{diag}([\sqrt{\nu_1}, \ldots, \sqrt{\nu_n}]), \tag{11}$$

and note that the orthogonal matrix of eigenvectors $\mathbf{Q}$ can be written $\mathbf{Q} = \mathbf{PW}$.

The spectral collocation approximation of $f(s)$ constructs a Lagrange interpolating polynomial through the Gaussian quadrature points. Since the points are distinct, the $n-1$ degree interpolating polynomial is unique. We write this approximation $\mathcal{U}_l^n(f)$, where the subscript l is for *Lagrange interpolation*, as

$$f(s) \;\approx\; \mathcal{U}_l^n(f) \;=\; \sum_{i=1}^{n} f(\lambda_i)\, \ell_i(s) \;=\; \mathbf{f}^T\mathbf{l}(s). \tag{12}$$

The parameterized vector $\mathbf{l}(s)$ contains the Lagrange cardinal functions

$$\ell_i(s) = \prod_{j=1,\, j\neq i}^{n} \frac{s-\lambda_j}{\lambda_i-\lambda_j}. \tag{13}$$

By construction, the collocation polynomial $\mathcal{U}_l^n(f)$ interpolates $f(s)$ at the Gaussian quadrature points.

The pseudospectral approximation of $f(s)$ is constructed by first truncating its Fourier series at n terms and approximating each Fourier coefficient with a quadrature rule. If we use the n-point Gaussian quadrature, then we can write the approximation as

$$f(s) \;\approx\; \mathcal{U}_p^n(f) \;=\; \sum_{i=1}^{n} \hat{f}_i \, \pi_i(s) \;=\; \hat{\mathbf{f}}^T \boldsymbol{\pi}(s), \tag{14}$$

where $\hat{f}_i$ is the pseudospectral coefficient,

$$\hat{f}_i = \sum_{j=1}^{n} f(\lambda_j) \, \pi_i(\lambda_j) \, \nu_j, \tag{15}$$

and the vector $\hat{\mathbf{f}}$ contains all coefficient approximations; the subscript p on $\mathcal{U}_p^n$ is for *pseudospectral*. Note that we have overloaded the notation by defining $\hat{f}_i$ as the pseudospectral coefficient (15), instead of the true Fourier coefficient in (5). We next state two lemmas about the relationship between the spectral collocation and pseudospectral approximations for future reference.

Lemma 1. *The vector of evaluations of f at the quadrature points $\mathbf{f}$ is related to the pseudospectral coefficients $\hat{\mathbf{f}}$ by*

$$\hat{\mathbf{f}} \;=\; \mathbf{QWf} \;=\; \mathbf{PW}^2\mathbf{f}. \tag{16}$$

Proof. This is easily verified by equation (15) using the matrices defined in (11). □

Lemma 2. *The pseudospectral approximation $\mathcal{U}_p^n(f)$ is equal to the spectral collocation approximation $\mathcal{U}_l^n(f)$ for all $s \in \mathcal{S}$.*

Proof. By the uniqueness of the Lagrange polynomial interpolation, we can write $\mathbf{Pl}(s) = \boldsymbol{\pi}(s)$. Since $\mathbf{P} = \mathbf{QW}^{-1}$, we have $\mathbf{l}(s) = \mathbf{WQ}^T\boldsymbol{\pi}(s)$. Then

$$\begin{aligned}
\mathcal{U}_l^n(f) &= \mathbf{f}^T\mathbf{l}(s) \\
&= \mathbf{f}^T\mathbf{WQ}^T\boldsymbol{\pi}(s) \\
&= \hat{\mathbf{f}}^T\boldsymbol{\pi}(s) \\
&= \mathcal{U}_p^n(f),
\end{aligned}$$

as required. □

Lemma 2 implies that the pseudospectral approximation $\mathcal{U}_p^n(f)$ interpolates $f(s)$ at the Gaussian quadrature points. However, the equivalence expressed in Lemma 2 breaks down in two important cases. When the number of terms in the orthogonal series is less than the number of points in the quadrature rules, the orthogonal series representation no longer produces the same polynomial as the Lagrange interpolant. Also, if a quadrature rule that is not the Gaussian quadrature rule is used to approximate the Fourier coefficients, then the discrete Fourier transform from Lemma 1 is no longer valid. The latter situation may occur if an alternative quadrature rule holds practical advantages over the Gaussian quadrature rule.

Remark 1. *We have restricted our attention to orthonormal polynomials and Gaussian quadrature rules for a given weight function. However, transformations similar to Lemma 1 apply for Chebyshev polynomials and Clenshaw-Curtis quadrature rules using a fast Fourier transform. For an insightful discussion of the comparisions between these methods of integration and approximation, see [18].*

2.2. Tensor Product Extensions

When $d > 1$, the above concepts extend directly via a tensor product construction. For a given multi-index $\mathbf{n} = (n_1, \ldots, n_d) \in \mathbb{N}^d$, it is convenient to define the set of multi-indices

$$\mathcal{I}_{\mathbf{n}} = \{\mathbf{i} \ : \ \mathbf{i} \in \mathbb{N}^d,\, 1 \leq i_k \leq n_k,\, k = 1, \ldots, d\}. \tag{17}$$

We can use this index set to reference components of the tensor product approximations.

Tensor product Gaussian quadrature rules are constructed by taking cross products of univariate Gaussian quadrature rules. For multi-index $\mathbf{n}$, let $G_{\mathbf{n}}$ be the set of d-variate Gaussian quadrature points,

$$G_{\mathbf{n}} \ = \ \{\lambda_{\mathbf{i}} = (\lambda_{i_1}, \ldots, \lambda_{i_d}) \ : \ \mathbf{i} \in \mathcal{I}_{\mathbf{n}}\}, \tag{18}$$

where the points λ_{i_k} with $i_k = 1, \ldots, n_k$ are the univariate quadrature points for $w_k(s_k)$. The associated quadrature weights $W_{\mathbf{n}}$ are given by

$$W_{\mathbf{n}} \ = \ \left\{ \nu_{\mathbf{i}} = \nu_{i_1} \cdots \nu_{i_d} \ : \ \mathbf{i} \in \mathcal{I}_{\mathbf{n}} \right\}. \tag{19}$$

In words, the tensor product quadrature weights are products of the univariate weights. To approximate the integral of $f(s)$, compute

$$\begin{aligned}\int_{\mathcal{S}} f(s)\, w(s)\, ds &\approx (\mathcal{U}_q^{n_1} \otimes \cdots \otimes \mathcal{U}_q^{n_d})(f) \\ &= \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} f(\lambda_{i_1}, \ldots, \lambda_{i_d})\, \nu_{i_1} \cdots \nu_{i_d} \\ &= \sum_{\mathbf{i} \in \mathcal{I}_{\mathbf{n}}} f(\lambda_{\mathbf{i}})\, \nu_{\mathbf{i}} \\ &= \mathbf{f}_{\mathbf{n}}^T\, \boldsymbol{\nu}_{\mathbf{n}}\end{aligned}$$

where $\mathbf{f}_{\mathbf{n}}$ is the vector of function evaluations at the tensor grid of quadrature points, and $\boldsymbol{\nu}_{\mathbf{n}}$ is a vector of the tensor product quadrature weights.

The spectral collocation approximation on the points $G_{\mathbf{n}}$ uses a basis of product-type Lagrange cardinal functions. Define the vector of these basis polynomials by

$$\mathbf{l}_{\mathbf{n}}(s) = \mathbf{l}_{n_1}(s_1) \otimes \cdots \otimes \mathbf{l}_{n_d}(s_d), \tag{20}$$

where $\mathbf{l}_{n_k}(s_k)$ is a vector of the univariate Lagrange cardinal functions constructed on the univariate quadrature rule defined by λ_{i_k}; see (13). Then the tensor product spectral collocation approximation for the multi-index $\mathbf{n}$ is given by

$$f(s) \approx (\mathcal{U}_l^{n_1} \otimes \cdots \otimes \mathcal{U}_l^{n_d})(f) \tag{21}$$

$$= \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} f(\lambda_{i_1}, \ldots, \lambda_{i_d})\, \ell_{i_1}(s_1) \cdots \ell_{i_d}(s_d) \tag{22}$$

$$= \sum_{\mathbf{i} \in \mathcal{I}_{\mathbf{n}}} f(\lambda_{\mathbf{i}})\, \ell_{\mathbf{i}}(s) \tag{23}$$

$$= \mathbf{f}_{\mathbf{n}}^T\, \mathbf{l}_{\mathbf{n}}(s). \tag{24}$$

The tensor product pseudospectral approximation uses a product type multivariate orthonormal polynomial basis, which is simply a Kronecker product of the univariate orthonormal polynomials. For a multi-index $\mathbf{n}$, let $\boldsymbol{\pi}_{n_k}(s_k)$ be the vector of univariate polynomials that are orthonormal with respect to $w_k(s_k)$ for $k = 1, \ldots, d$. Then the vector

$$\boldsymbol{\pi}_{\mathbf{n}}(s) = \boldsymbol{\pi}_{n_1}(s_1) \otimes \cdots \otimes \boldsymbol{\pi}_{n_d}(s_d) \tag{25}$$

contains polynomials that are orthonormal with respect to $w(s)$; we can uniquely reference a component of the vector $\boldsymbol{\pi}_{\mathbf{n}}(s)$ by $\pi_{\mathbf{i}}(s)$ with $\mathbf{i} \in \mathcal{I}_{\mathbf{n}}$. The tensor product pseudospectral approximation for the multi-index $\mathbf{n}$ is given by

$$
\begin{aligned}
f(s) \approx\ & (\mathcal{U}_p^{n_1} \otimes \cdots \otimes \mathcal{U}_p^{n_d})(f) && (26)\\
&= \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} \hat{f}_{i_1,\ldots,i_d}\, \pi_{i_1}(s_1) \cdots \pi_{i_d}(s_d) && (27)\\
&= \sum_{\mathbf{i}\in\mathcal{I}_{\mathbf{n}}} \hat{f}_{\mathbf{i}}\, \pi_{\mathbf{i}}(s) && (28)\\
&= \hat{\mathbf{f}}_{\mathbf{n}}^T\, \boldsymbol{\pi}_{\mathbf{n}}(s), && (29)
\end{aligned}
$$

where $\hat{\mathbf{f}}_{\mathbf{n}}$ is the vector of pseudospectral coefficients

$$
\begin{aligned}
\hat{f}_{\mathbf{i}} &= \sum_{j_1=1}^{n_1} \cdots \sum_{j_d=1}^{n_d} f(\lambda_{j_1}, \ldots, \lambda_{j_d})\, \pi_{i_1}(\lambda_{j_1}) \cdots \pi_{i_d}(\lambda_{j_d})\, \nu_{j_1} \cdots \nu_{j_d} && (30)\\
&= \sum_{\mathbf{j}\in\mathcal{I}_{\mathbf{n}}} f(\lambda_{\mathbf{j}})\, \pi_{\mathbf{i}}(\lambda_{\mathbf{j}})\, \nu_{\mathbf{j}}. && (31)
\end{aligned}
$$

The extensions of Lemmas 1 and 2 are then straightforward. For the multi-index $\mathbf{n}$, define the matrices

$$
\mathbf{Q} = \mathbf{Q}_{n_1} \otimes \cdots \otimes \mathbf{Q}_{n_d}, \qquad \mathbf{P} = \mathbf{P}_{n_1} \otimes \cdots \otimes \mathbf{P}_{n_d}, \qquad \mathbf{W} = \mathbf{W}_{n_1} \otimes \cdots \otimes \mathbf{W}_{n_d}. \tag{32}
$$

The proofs of Lemmas 1 and 2 hold with

$$
\mathbf{f} = \mathbf{f}_{\mathbf{n}}, \qquad \hat{\mathbf{f}} = \hat{\mathbf{f}}_{\mathbf{n}}, \qquad \mathbf{l}(s) = \mathbf{l}_{\mathbf{n}}(s), \qquad \boldsymbol{\pi}(s) = \boldsymbol{\pi}_{\mathbf{n}}(s). \tag{33}
$$

This is easily verified by employing the mixed product property of Kronecker products. In words, we have that the Lagrange interpolant constructed on a tensor product of Gaussian quadrature points (i.e., tensor product collocation) produces the same polynomial approximation as a truncated Fourier expansion with a tensor product basis, where the coefficients are computed with the tensor product Gaussian quadrature rule (i.e., tensor product pseudospectral). This equivalence occurs when the number of quadrature points in each variable is equal to the number of univariate basis polynomials in each variable; in other words, the number of points is the maximum degree plus one in each variable.

2.3. Smolyak's Algorithm and Sparse Grids

The inescapable challenge for tensor product approximation is the exponential increase in the work required to compute the approximation as the dimension increases. An n-point quadrature rule in each of d dimensions uses n^d function evaluations. Thus, tensor product approximation quickly becomes infeasible beyond a handful of dimensions. Smolyak's algorithm [19] attempts to alleviate this curse of dimensionality while retaining integration and interpolation accuracy for certain classes of functions.

The majority of sparse grid applications in the literature rely on Smolyak's algorithm. The most common derivation starts by defining a linear operation (e.g., integration, interpolation, or projection) on a univariate function. We can generalize the notation used in (10), (12), and (14) by writing the linear operation as $\mathcal{U}^m(f)$. However, it is common to reinterpret the parameter m in this context as a choice for how the number of points grows as m is incremented. For example, $n_m = m$ for $m > 0$ would correspond to (10), (12), and (14). Another common growth relationship is

$$n_1 = 1, \qquad n_m = 2^{m+1} - 1, \quad m > 1. \tag{34}$$

Such a relationship is useful when the quadrature/interpolation point sets are nested, i.e., the points of the n-point rule are a subset of the points in the $2n+1$ rule. This notably occurs for rules based on (i) Chebyshev points [17], (ii) Gauss-Patterson[2] quadrature formulas [20], or (iii) equidistant points. In the case of a closed region of interpolation/integration, one may include and reuse the endpoints of the interval in the sequence of approximations; see for example the popular Clenshaw-Curtis integration rules [21]. Nested point sets can greatly increase efficiency if $f(s)$ is very expensive.

Define $|\mathbf{m}| = m_1 + \cdots + m_d$. Given a univariate linear operator, Smolyak's algorithm can be written

$$\mathcal{A} = \sum_{\mathbf{m} \in \mathcal{I}} c(\mathbf{m}) \, (\mathcal{U}^{m_1} \otimes \cdots \otimes \mathcal{U}^{m_d}). \tag{35}$$

In the standard formulation [8, 6], the set of admissible multi-indices $\mathcal{I}$ is

$$\mathcal{I} \;=\; \left\{ \mathbf{m} \in \mathbb{N}^d \;:\; l + 1 \leq |\mathbf{m}| \leq l + d \right\} \tag{36}$$

[2]The Gauss-Patterson rules contain a specific pattern of nesting that is not applicable for arbitrary n. See the reference for further details.

for a given level parameter l. In this case, the coefficients $c(\mathbf{m})$ are

$$c(\mathbf{m}) \;=\; (-1)^{l+d-|\mathbf{m}|} \binom{d-1}{l+d-|\mathbf{m}|}. \tag{37}$$

However, adaptive and anisotropic versions of Smolyak's algorithm may contain different choices for $\mathcal{I}$ and $c(\mathbf{m})$; such variations are useful if a function's variability can be primarily attributed to a subset of the inputs. See [22, 23] for details on such methods.

For our purposes, it is sufficient to note that Smolyak's algorithm amounts to a linear combination of tensor product operations. The specific tensor products are chosen so that no constituent tensor grid contains too many nodes. In the case of nested univariate rules, a node may be common to many tensor products. In practice, one may structure the computation to evaluate the function once per node in the union of tensor product grids – as opposed to once per node per tensor grid. This greatly simplifies the sparse grid integration, which can be written as a set of nodes and weights. If a node is common to multiple constituent tensor grids, then its corresponding weight is computed as a linear combination of the tensor grid weights; the coefficients of the linear combination are exactly $c(\mathbf{m})$. It is worth noting that the weights of a sparse grid rule can be negative, which precludes its use as a positive definite weighted inner product.

3. Sparse Pseudospectral Approximation Method

In practice, one may wish to take advantage of the relatively small number of points in the sparse grid quadrature rule when computing a pseudospectral approximation. This is often done by first truncating the Fourier series representation of $f(s)$ (see (5)), and then approximating its spectral coefficients with a sparse grid quadrature rule. Unfortunately, choosing the parameters of the sparse grid rule that will accurately approximate the integral formulation of the Fourier coefficient is not straightforward. The is because – in constrast to tensor product approximation – the Lagrange interpolating polynomial is *not* equivalent to a truncated pseudospectal approximation with sparse grid integration, where the number of basis polynomials is equal to the number of points in the quadrature rule. The general wisdom has been to truncate conservatively for a sparse grid quadrature rule constrained by a computational budget; such heuristics become more complicated when anisotropic sparse grid rules are used.

The SPAM approaches this problem from a different perspective; it is merely the proper application of Smolyak's algorithm to the tensor product pseudospectral projection. We take advantage of the equivalence between tensor product pseudospectral and spectral collocation approximations to construct spectral approximations that naturally correspond to a given sparse grid quadrature rule. In essence, since the sparse grid quadrature rule is constructed by taking linear combinations of tensor product quadrature rules, we can take the same linear combination of tensor product pseudospectral expansions to produce an approximation in a basis of multivariate orthogonal polynomials; a linear combination of expansions can be easily computed by linearly combining the pseudospectral coefficients corresponding to the same basis polynomial. Each tensor product pseudospectral expansion is simply a transformation from the Lagrange basis using Lemma 1. In the numerical examples of Section 4, we show compelling evidence that this procedure is superior to directly applying the sparse grid quadrature rule to the integral formulation of the Fourier coefficients.

More precisely, let $\mathcal{I}$ and $c(\mathbf{m})$ be the admissible index set and coefficient function for a given sparse grid quadrature rule. Then the sparse pseudospectral approximation is given by

$$\begin{aligned} f(s) &\approx \mathcal{A}_p(f) && (38)\\ &= \sum_{\mathbf{m}\in\mathcal{I}} c(\mathbf{m})\,(\mathcal{U}_p^{m_1}\otimes\cdots\otimes\mathcal{U}_p^{m_d})(f) && (39)\\ &= \sum_{\mathbf{m}\in\mathcal{I}} c(\mathbf{m})\,\hat{\mathbf{f}}_{\mathbf{m}}^T\,\boldsymbol{\pi}_{\mathbf{m}}(s) && (40) \end{aligned}$$

where $\hat{\mathbf{f}}_{\mathbf{m}}$ and $\boldsymbol{\pi}_{\mathbf{m}}(s)$ are defined as in (29). In practice, we linearly combine the coefficients corresponding to common basis polynomials. With a slight abuse of notation, let $\{\boldsymbol{\pi}(s)\}$ be the set of basis polynomials corresponding to a vector $\boldsymbol{\pi}(s)$; the common basis set for $\mathcal{A}_p(f)$ is defined by

$$\Pi = \bigcup_{\mathbf{m}\in\mathcal{I}} \{\boldsymbol{\pi}_{\mathbf{m}}(s)\}. \tag{41}$$

Then we can write

$$\mathcal{A}_p(f) = \sum_{\pi(s)\in\Pi} \hat{f}_\pi\,\pi(s). \tag{42}$$

The coefficient corresponding to $\pi(s)$ is given by

$$\hat{f}_\pi = \sum_{\mathbf{m}\in\mathcal{I}} c(\mathbf{m})\, \hat{f}_{\mathbf{i},\mathbf{m}}, \tag{43}$$

where

$$\hat{f}_{\mathbf{i},\mathbf{m}} = \begin{cases} \hat{f}_{\mathbf{i}} & \text{if } \pi(s) = \pi_{\mathbf{i}}(s) \text{ with } \mathbf{i} \in \mathcal{I}_{\mathbf{m}}, \\ 0 & \text{otherwise.} \end{cases} \tag{44}$$

In words, (42) simply rearranges the terms in the sum so that each polynomial basis appears only once. The next theorem allows us to apply existing analysis results for sparse grid interpolation schemes to the SPAM.

Theorem 1. *Under the conditions of Lemma 2, the sparse pseudospectral approximation $\mathcal{A}_p(f)$ is point-wise equivalent to the sparse grid interpolation approximation.*

Proof. Using the tensor product version of Lemma 2, we can write

$$\begin{aligned} \mathcal{A}_p(f) &= \sum_{\mathbf{m}\in\mathcal{I}} c(\mathbf{m})\, \hat{\mathbf{f}}_{\mathbf{m}}^T\, \boldsymbol{\pi}_{\mathbf{m}}(s) \\ &= \sum_{\mathbf{m}\in\mathcal{I}} c(\mathbf{m})\, \mathbf{f}_{\mathbf{m}}^T\, \mathbf{l}_{\mathbf{m}}(s), \end{aligned}$$

where $\mathbf{f}_{\mathbf{m}}$ and $\mathbf{l}_{\mathbf{m}}(s)$ are defined in (21). This completes the proof. □

As a result of this theorem, all of the error analysis for sparse grid collocation and interpolation methods applies directly to the sparse pseudospectral approximation. We refer the interested reader to references [8, 6, 5] for such details, and we instead focus on the numerical experiments in the following section. But before we move on, we prove an interesting fact about the mean of $\mathcal{A}_p(f)$.

Corollary 1. *The mean of the sparse pseudospectral approximation $\mathcal{A}_p(f)$ is equal to the mean of $f(s)$ approximated with the associated sparse grid quadrature rule.*

Proof. By orthogonality, the mean of a polynomial expanded in an orthonormal basis is equal to the coefficient of the zero degree term, which is 1. Define $\hat{f}_{\mathbf{1}}$ to be the coefficient of the constant term in $\mathcal{A}_p(f)$. The constant term

also appears in each constituent tensor product pseudospectral approximation; denote this by $\hat{f}_{\mathbf{1},\mathbf{m}}$ for the multi-index $\mathbf{m}$. Therefore, by (44),

$$\begin{aligned}\hat{f}_{\mathbf{1}} &= \sum_{\mathbf{m}\in\mathcal{I}} c(\mathbf{m})\, \hat{f}_{\mathbf{1},\mathbf{m}} \\ &= \sum_{\mathbf{m}\in\mathcal{I}} c(\mathbf{m})\, (\mathcal{U}_q^{m_1} \otimes \cdots \otimes \mathcal{U}_q^{m_d})(f),\end{aligned}$$

which is exactly the definition of sparse grid integration. □

4. Numerical Experiments

In the following numerical experiments, we compare the coefficients computed with the SPAM to direct approximation of the Fourier coefficients with the corresponding sparse grid quadrature rule. To make the comparison fair, we apply the sparse grid rule directly to each coefficient corresponding to the basis set (41) for the sparse pseudospectral approximation. We construct each sparse grid rule using (i) univariate non-nested Gauss-Legendre quadrature points for a uniform weight function on the square $[-1,1]^2$, (ii) n_m defined as in (34), and (iii) $\mathcal{I}$ and $c(\mathbf{m})$ defined as in (36) and (37). The choice of the uniform weight function implies the $\pi_i(s)$ are the normalized Legendre polynomials for the pseudospectral approximation. For all experiments, we compute the largest feasible tensor product pseudospectral approximation and call the resulting coefficients the truth. In all cases, the apparent decay in the tensor product pseudospectral coefficients assures us that we have used a sufficiently high order approximation to bestow the title *truth*.

4.1. Five Bivariate functions

In the first experiment, we compare both methods on five bivariate functions; see Table 1. For each function, we compute a tensor product pseudospectral approximation of order 255 in each variable – 65,536 total quadrature points. We plot the log of the magnitude of the pseudospectral coefficients with a surface plot to visually observe their decay. We then plot the log of the magnitude of the sparse pseudospectral coefficients corresponding to a level $l = 7$ sparse grid compared to the same sparse grid approximation of the Fourier coefficients.

#	$f(s_1, s_2)$
1	$s_1^{10} s_2^{10}$
2	$e^{s_1+s_2}$
3	$\sin(5(s_1 - 0.5)) + \cos(3(s_2 - 1))$
4	$1/(2 + 16(s_1 - 0.1)^2 + 25(s_2 + 0.1)^2)$
5	$(\lvert s_1 - 0.2\rvert + \lvert s_2 + 0.2\rvert)^3$

Table 1: The five bivariate test functions.

With a level 7 grid, sparse pseudospectral approximation contains a maximum univariate degree of 129. For each test function, the corresponding set of figures contains

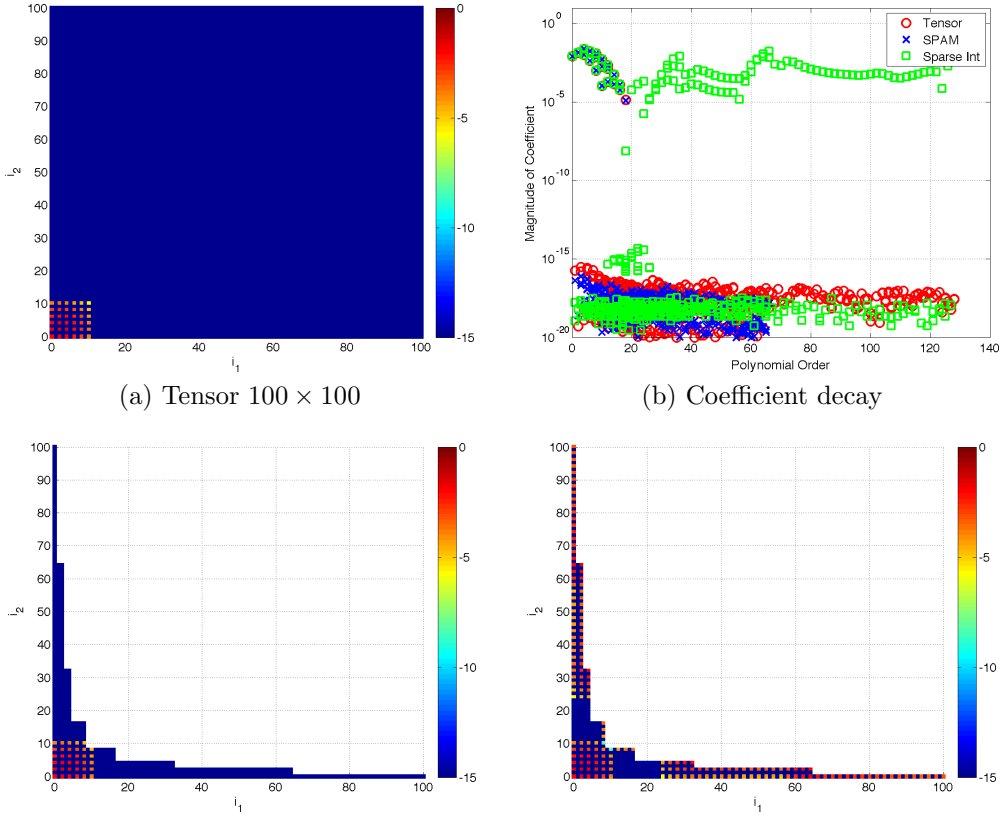
(i) the tensor product pseudospectral coefficients up to maximum univariate degree 100,
(ii) the SPAM coefficients up to maximum univariate degree 100,
(iii) the sparse grid integration approximation of the Fourier coefficients up to maximum univariate degree 100,
(iv) a summary plot with coefficients up to univariate degree 129 ordered by total order.

As a general comment, we see that the sparse grid integration produces largely incorrect values for coefficients associated with higher degree polynomials. More specific comments for the individual test functions are as follows:

1. $s_1^{10} s_2^{10}$: This function evaluates the performance of the methods on a monomial. We know that coefficients associated with polynomials of degree greater than 10 in either x or y should be zero by orthogonality. Additionally, since the monomial is an even function over the domain with a symmetric weight function, the coefficients corresponding to odd degree polynomials in either variable ought to be zero. This is verified in the tensor product pseudospectral coefficients and respected by the SPAM coefficients. However, the direct sparse integration produces non-zero values for coefficients that should be zero. See Figure 1.
2. $e^{s_1+s_2}$: This function is analytic in both variables with rapid decay of the Fourier coefficients. Observe that the direct sparse integration yields large values for coefficients corresponding to the higher order polynomials. See Figure 2.

3. $\sin(5(s_1 - 0.5)) + \cos(3(s_2 - 1))$: In the language of the ANOVA decomposition [24], this function has only main effects. Thus, the Fourier coefficients for polynomials with mixed terms corresponding to interaction effects should be zero. Again, this is apparent in the tensor product pseudospectral coefficients, and it is respected by the SPAM coefficients. The direct sparse integration, however, produces non-zero values for coefficients of the mixed polynomials; see Figure 3.
4. $1/(2 + 16(s_1 - 0.1)^2 + 25(s_2 + 0.1)^2)$: The pseudospectral coefficients of this rational function decay relatively slowly; notice it needs up to degree 40 polynomials in each variable to reach numerical precision, according to the tensor product pseudospectral coefficients. The SPAM coefficients do a much better job capturing the true decay of the Fourier coefficients than the direct integration method, which does not appear to decay at all. See Figure 4.
5. $(|s_1 - 0.2| + |s_2 + 0.2|)^3$: This function has discontinuous first derivatives, so we expect only first order algebraic convergence of its Fourier coefficients; on a log scale they decay very slowly. However, the interaction effects disappear after degree three in either variable. Again, this is visible in the tensor product pseudospectral coefficients and respected by the SPAM coefficients, but the direct sparse grid integration produces non-zero values for coefficients that ought to be zero. See Figure 5.

In general, we find that the SPAM coefficients are significantly more accurate than the direct application of the sparse grid integration rules to the Fourier coefficients. This observation is rather counterintuitive. One may expect that the sparse grid rule, by evaluating the product of the function times the basis polynomial at more locations, would yield a more accurate approximation. But this is clearly not the case for these examples. In Figure 6c, we plot the decay of the truncation error for the sparse approximations as the level increases. We approximate the truncation error by the sum of squares of the neglected coefficients from the tensor product expansion. Since both approximations use the same basis sets, this approximate truncation error is identical. In Figures 6a and 6b we plot the decay in the error of the approximated coefficients as the level increases; the error in the coefficients is squared and summed. We see that the error in the SPAM coefficients decays roughly like the truncation error, while the error in the direct sparse grid integration does not decay. Of course, this summary plot ignores what is



(a) Tensor 100×100 (b) Coefficient decay

(c) SPAM (d) Sparse grid integration

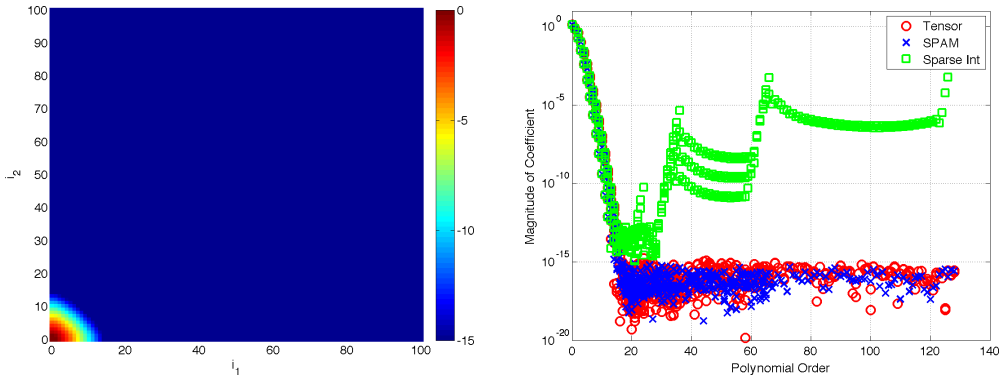
Figure 1: Fourier coefficient approximations for $s_1^{10} s_2^{10}$.

most visible in Figures 1-5, which is that some of the coefficients associated with lower order polynomials may be approximated well; it is the coefficients of the higher order terms that contain most of the error.

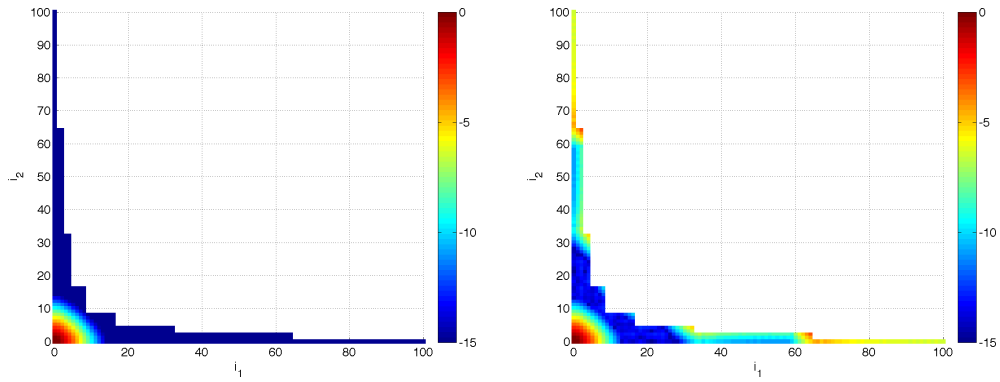
4.2. PDE with Random Input Data

The last numerical example we examine is a linear elliptic diffusion equation with a stochastic diffusion coefficient. Let $D = [0,1] \times [0,1]$ and $(\Omega, \mathcal{B}, P)$ be a complete probability space. We seek the function $u : D \times \Omega \to \mathbb{R}$ such that the following holds P-a.e.:

$$\begin{aligned} -\nabla \cdot (a(x,\omega)\nabla u(x,\omega)) &= 1, \quad x \in D, \\ u(x) &= 0, \quad x \in \partial D. \end{aligned} \tag{45}$$

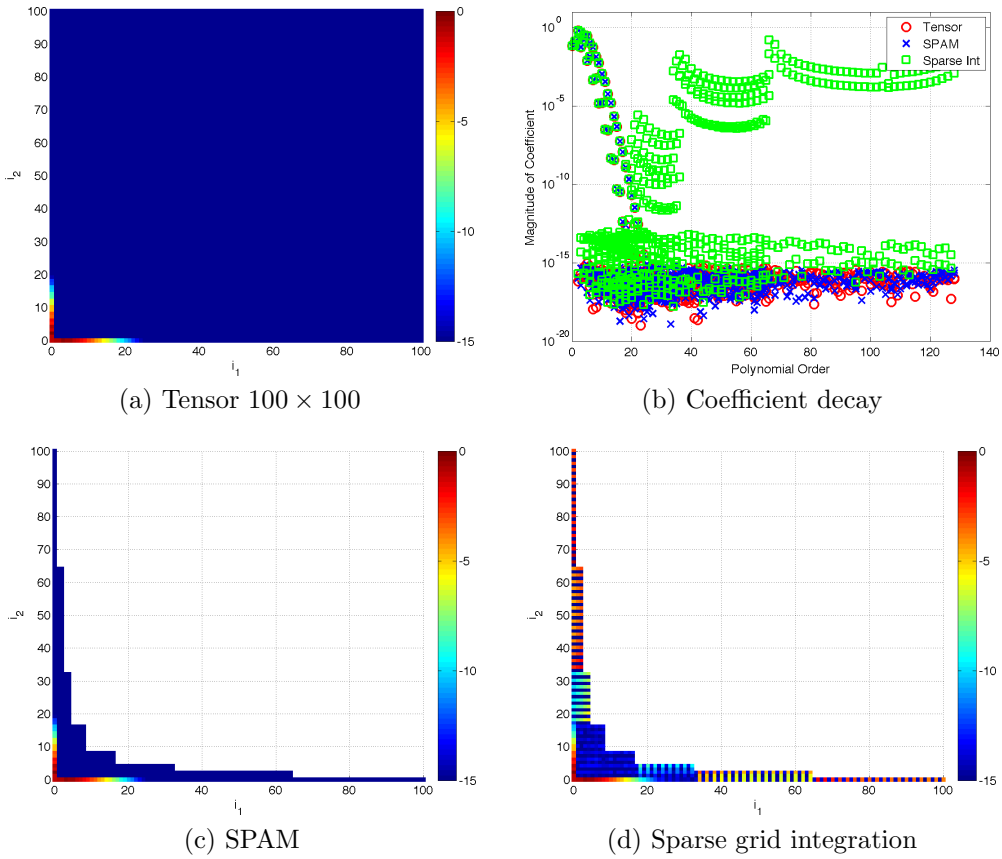


(a) Tensor 100×100 (b) Coefficient decay



(c) SPAM (d) Sparse grid integration

Figure 2: Fourier coefficient approximations for $e^{s_1+s_2}$.



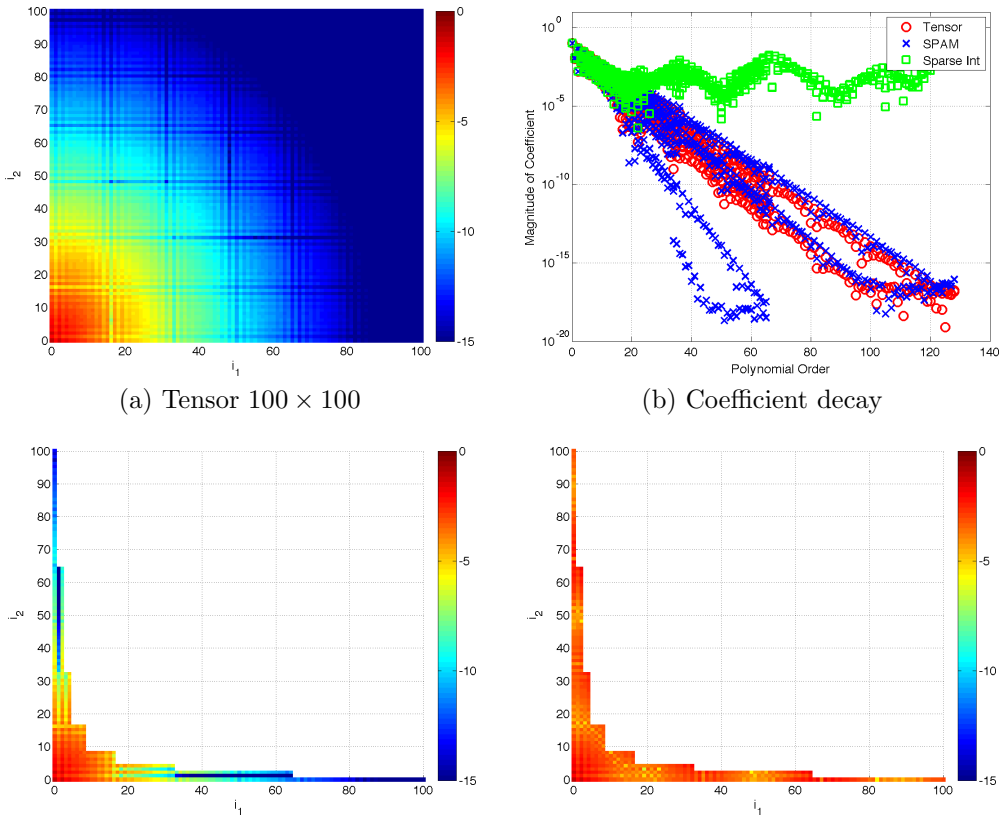
(a) Tensor 100×100

(b) Coefficient decay

(c) SPAM

(d) Sparse grid integration

Figure 3: Fourier coefficient approximations for $\sin(5(s_1 - 0.5)) + \cos(3(s_2 - 1))$.



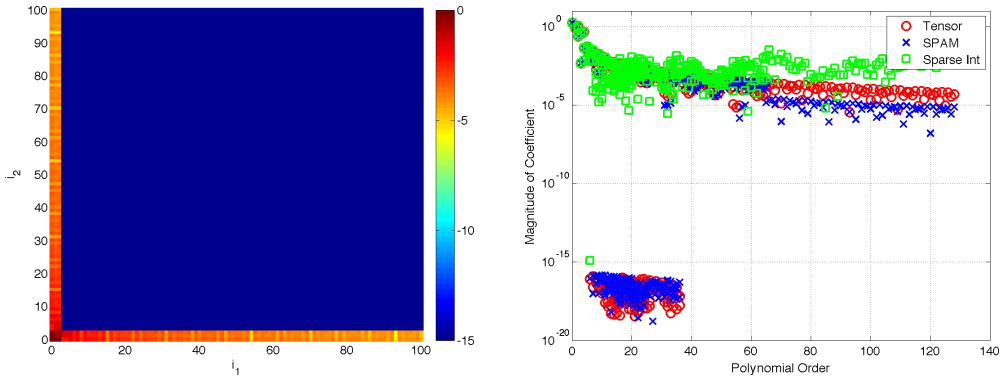
(a) Tensor 100×100

(b) Coefficient decay

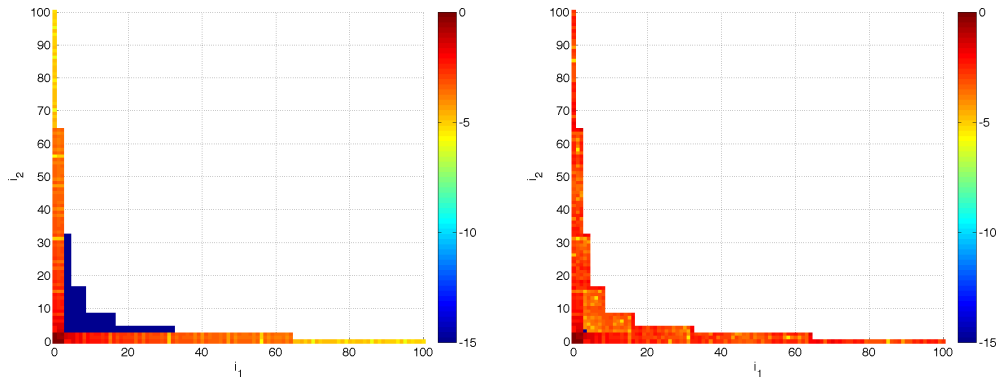
(c) SPAM

(d) Sparse grid integration

Figure 4: Fourier coefficient approximations for $1/(2+16(s_1-0.1)^2+25(s_2+0.1)^2)$.

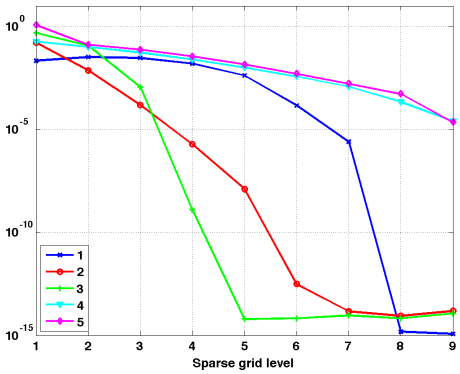


(a) Tensor 100×100 (b) Coefficient decay

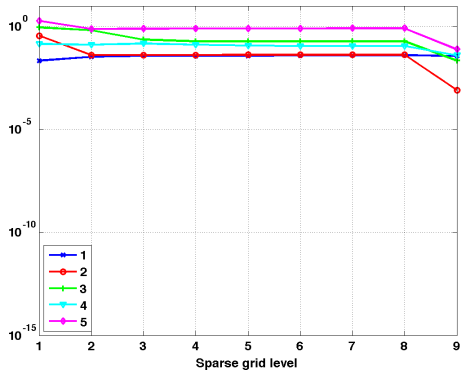


(c) SPAM (d) Sparse grid integration

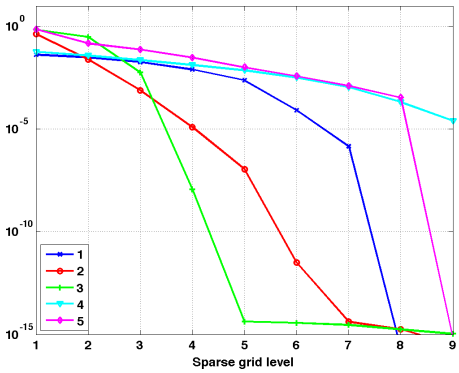
Figure 5: Fourier coefficient approximations for $(|s_1 - 0.2| + |s_2 + 0.2|)^3$.



(a) SPAM coefficient error



(b) Sparse grid integration coefficient error



(c) Truncation error

Figure 6: Comparison of truncation error to error coefficient approximation between SPAM and direct sparse grid integration for each of the five test functions, numbered according to Table 1.

Instead of the whole solution u, we are interested in computing the response function

$$g(\omega) = \int_D u(x,\omega)dx \tag{46}$$

which is the spatial mean of u over D.

The diffusion coefficient $a(x,\omega)$ is modeled as a uniform random field with exponential covariance:

$$C(x,y) \equiv E[a(x,\omega)a(y,\omega)] = \sigma^2 e^{\|x-y\|_1/L} \tag{47}$$

where $\sigma = 0.1$ is the standard deviation of the field and $L = 1$ is the correlation length. It is approximated through a truncated Karhunen-Loéve expansion [25]:

$$a(x,\omega) \;\approx\; \hat{a}_d(x, s(\omega)) \;=\; a_0(x) \;+\; \sum_{k=1}^{d} \sqrt{\lambda_k} a_k(x)\, s_k(\omega), \tag{48}$$

where $a_0(x) = \mu = 0.2$ is the mean of the random field, $(\lambda_k, a_k(x))$, $k = 1,\ldots,d$ are eigenvalue-eigenfunction pairs for the covariance operator:

$$\int_D C(x,y)a_k(x)dx = \lambda_k a_k(y), \quad y \in D, \tag{49}$$

and $s = (s_1,\ldots,s_d)$ are uncorrelated, uniform random variables on $[-1,1]$. We make the further modeling assumption that the random variables are independent. Define $\Gamma = [-1,1]^d$ to be the range of s and

$$w(s) = \begin{cases} 1/2^d & s \in [-1,1]^d \\ 0 & \text{otherwise} \end{cases} \tag{50}$$

to be the density of s. The eigenvalues and eigenfunctions are computed using a pseudo-analytic procedure described in [2]. The eigenvalues are sorted in decreasing order, and we use the first $d = 5$ eigenvalues/eigenfunctions to approximate the random field.

Let $\pi_i : [-1,1] \to \mathbb{R}$, $i = 1,2,\ldots$ be the normalized Legendre polynomial of order $i-1$. For a given multi-index $\mathbf{i} = (i_1,\ldots,i_d)$, define the tensor product polynomial

$$\pi_{\mathbf{i}}(s) = \pi_{i_1}(s_1)\ldots\pi_{i_d}(s_d). \tag{51}$$

Given a set $\mathcal{I}$ of multi-indices, we approximate $g(\omega)$ by

$$\hat{g}(\omega) = \sum_{\mathbf{i} \in \mathcal{I}} \hat{g}_{\mathbf{i}} \pi_{\mathbf{i}}(s(\omega)) \tag{52}$$

where the unknown coefficients $\hat{g}_{\mathbf{i}}$ are computed through pseudospectral projection using both SPAM and sparse grid integration. For a given s, the corresponding response g is computed by solving

$$\begin{aligned} -\nabla \cdot (\hat{a}_d(x,s)\nabla u(x) &= 1, \quad x \in D, \\ u(x) &= 0, \quad x \in \partial D, \\ g &= \int_D u(x)dx. \end{aligned} \tag{53}$$

These equations are discretized using piecewise linear finite elements over quadrilateral mesh cells of size 1/32. The resulting linear algebraic equations are solved via preconditioned GMRES using an algebraic multigrid preconditioner with tolerance of 10^{-12}. The finite element equations were implemented and solved using a variety of packages within the Trilinos solver framework [26]. The resulting SPAM and sparse grid integrations were provided by the Dakota package [27].

Note that instead of using the growth relationship in (34), we choose

$$n_m = 2m - 1, \quad m \geq 1. \tag{54}$$

This growth relationship yields tensor grids with many fewer points compared to (34). The multiplication factor 2 ensures that all tensor grids will share the mid-point of the domain, which reduces the total number of function evaluations. The corresponding coefficients of the stochastic response function g are plotted in Figure 7 by the degree of the corresponding multivariate polynomial. One can see as the order of the polynomials increases, the coefficients generated by SPAM decay as they should, whereas those generated through direct sparse integration begin to diverge for the higher order polynomials. Note, however, that the difference is not as pronounced compared to the bivariate test cases. We attribute this to the use of the growth relationship (54), as opposed to (34) used with the bivariate functions.

5. Conclusions

Sparse grid integration rules are constructed as linear combinations of tensor product quadrature rules. By taking advantage of the equivalence be-

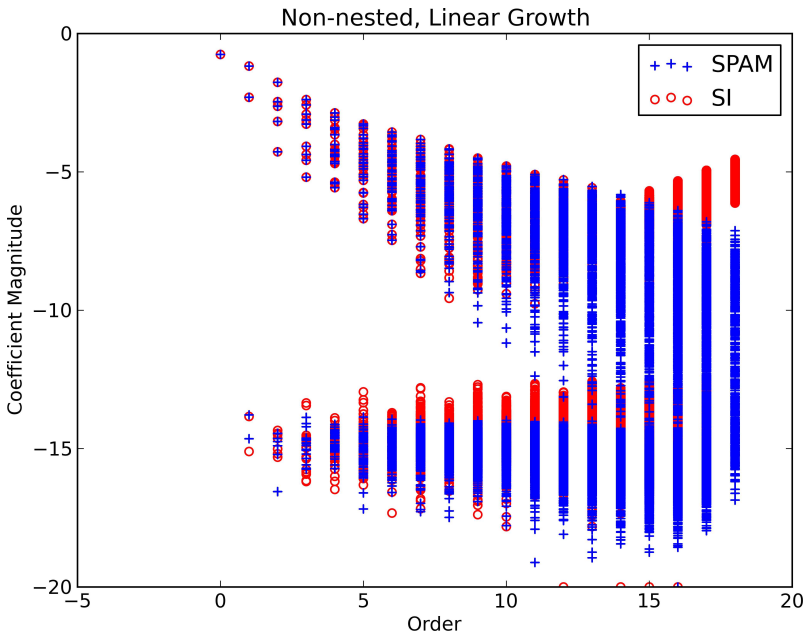


Figure 7: Comparison of approximate Fourier coefficients of the stochastic response (52) of the linear diffusion problem (45) using SPAM and sparse integration. The figure plots the coefficients according to the degree of associated polynomial.

tween the tensor product Lagrange interpolant and a pseudospectral approximation with a tensor product orthogonal polynomial basis, we can linearly combine the tensor product polynomial expansions associated with each tensor grid quadrature rule to produce a sparse pseudospectral approximation. We have numerically compared this approach to direct sparse grid integration of the Fourier coefficients. The experiments show convincingly that the direct integration approach produces inaccurate approximations of the Fourier coefficients associated with the higher order polynomial basis functions, while the SPAM coefficients are much more accurate.

The difference between SPAM and the sparse grid integration of the Fourier coefficients is present in all Smolyak type approximations – including anisotropic and adaptive variants. While not presented explicitly in this paper due to space limitations, the authors have conducted similar studies on such variants with similar results. The conclusions are clear. Given a function evaluated at the nodes of a sparse grid integration rule, the proper way to approximate the Fourier coefficients of an orthogonal expansion is the SPAM.

References

[1] D. Xiu, G. E. Karniadakis, The Wiener-Askey polynomial chaos for stochastic differential equations, SIAM Journal of Scientific Computing 24 (2002) 619 – 644.

[2] R. Ghanem, P. D. Spanos, Stochastic Finite Elements: A Spectral Approach, Springer-Verlag, New York, 1991.

[3] D. Xiu, J. S. Hesthaven, High order collocation methods for differential equations with random inputs, SIAM Journal of Scientific Computing 27 (2005) 1118 – 1139.

[4] I. Babuška, F. Nobile, R. Tempone, A stochastic collocation method for elliptic partial differential equations with random input data, SIAM Journal of Numerical Analysis 45 (2007) 1005 – 1034.

[5] H.-J. Bungartz, M. Griebel, Sparse grids, Acta Numerica (2004) 147–269.

[6] E. Novak, K. Ritter, High dimensional integration of smooth functions over cubes, Numerische Mathematik 75 (1) (1996) 79–97.

[7] F. Nobile, R. Tempone, C. G. Webster, A sparse grid stochastic collocation method for partial differential equations with random input data, SIAM Journal of Numerical Analysis 46 (2008) 2309 – 2345.

[8] V. Barthelmann, E. Novak, K. Ritter, High dimensional polynomial interpolation on sparse grids, Advances in Computer Mathematics 12 (2000) 273–288.

[9] G. Szegö, Orthogonal Polynomials, American Mathematical Society, Providence, RI, 1939.

[10] C. F. Dunkl, Y. Xu, Orthogonal Polynomials in Several Variables, Cambridge University Press, 2001.

[11] D. Xiu, Efficient collocational approach for parametric uncertainty analysis, COMMUNICATIONS IN COMPUTATIONAL PHYSICS 2 (2) (2007) 293–309.

[12] O. P. L. Maitre, M. T. Reagan, H. N. Najm, R. G. Ghanem, O. M. Knio, A stochastic projection method for fluid flow: Ii. random process, Journal of Computational Physics 181 (1) (2002) 9 – 44. doi:DOI: 10.1006/jcph.2002.7104.
URL http://www.sciencedirect.com/science/article/pii/S0021999102971044

[13] M. T. Reagan, H. N. Najm, R. G. Ghanem, O. M. Knio, Uncertainty quantification in reacting-flow simulations through non-intrusive spectral projection, Combustion and Flame 132 (3) (2003) 545 – 555. doi:DOI: 10.1016/S0010-2180(02)00503-5.
URL http://www.sciencedirect.com/science/article/pii/S0010218002005035

[14] M. S. Eldred, C. G. Webster, P. G. Constantine, Evaluation of non-intrusive approaches for Wiener-Askey generalized polynomial chaos, AIAA Paper (2008-1892).

[15] J. Shen, L.-L. Wang, Sparse spectral approximations of high-dimensional problems based on hyperbolic cross, SIAM Journal of Numerical Analysis 48 (2010) 1087–1109.

[16] J. Shen, H. Yu, Efficient spectral sparse grid methods and applications to high-dimensional elliptic problems, SIAM J. Sci. Comput. 32 (2010) 3228–3250.

[17] W. Gautschi, Orthogonal Polynomials: Computation and Approximation, Clarendon Press, Oxford, 2004.

[18] L. N. Trefethen, Is Gauss quadrature better than Clenshaw-Curtis?, SIAM Review 50 (2008) 67–87.

[19] S. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of function, Soviet Math. Dokl. 4 (1963) 240–143.

[20] T. N. L. Patterson, The optimum addition of points to quadrature formulae, Mathematics of Computation 22 (1968) 847–856.

[21] C. W. Clenshaw, A. R. Curtis, A method for numerical integration on an automatic computer, Numerische Mathematik 2 (1960) 197–205.

[22] F. Nobile, R. Tempone, C. G. Webster, An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data, SIAM Journal of Numerical Analysis 46 (2008) 2411–2442.

[23] T. Gerstner, M. Griebel, Numerical integration using sparse grids, Numerical Algorithms 18 (1998) 209–232.

[24] R. Liu, A. B. Owen, Estimating mean dimensionality of analysis of variance decompositions, Journal of the American Statistical Association 101 (2006) 712–721.

[25] M. Loève, Probability Theory II, Springer-Verlag, 1978.

[26] M. Heroux, R. Bartlett, V. Howle, R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, A. Salinger, H. Thornquist, R. Tuminaro, J. Willenbring, A. Williams, K. Stanley, An overview of the Trilinos package, ACM Trans. Math. Softw. 31 (3), `http://trilinos.sandia.gov/`.

[27] M. S. Eldred, B. M. Adams, D. M. Gay, L. P. Swiler, K. Haskell, W. J. Bohnhoff, J. P. Eddy, W. E. Hart, J.-P. Watson, P. D. Hough, T. G. Kolda, DAKOTA, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis, Tech. Rep. SAND2006-6337, Sandia National Laboratories (2006).