INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2011; **85**:1365–1389
Published online 2 September 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme.3019

Weighted Smolyak algorithm for solution of stochastic differential equations on non-uniform probability measures

Nitin Agarwal and N. R. Aluru*, †

Department of Mechanical Science and Engineering, Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, 405 N. Mathews Avenue, Urbana, IL 61801, U.S.A.

SUMMARY

This paper deals with numerical solution of differential equations with random inputs, defined on bounded random domain with non-uniform probability measures. Recently, there has been a growing interest in the stochastic collocation approach, which seeks to approximate the unknown stochastic solution using polynomial interpolation in the multi-dimensional random domain. Existing approaches employ sparse grid interpolation based on the Smolyak algorithm, which leads to orders of magnitude reduction in the number of support nodes as compared with usual tensor product. However, such sparse grid interpolation approaches based on piecewise linear interpolation employ uniformly sampled nodes from the random domain and do not take into account the probability measures during the construction of the sparse grids. Such a construction based on uniform sparse grids may not be ideal, especially for highly skewed or localized probability measures. To this end, this work proposes a weighted Smolyak algorithm based on piecewise linear basis functions, which incorporates information regarding non-uniform probability measures, during the construction of sparse grids. The basic idea is to construct piecewise linear univariate interpolation formulas, where the support nodes are specially chosen based on the marginal probability distribution. These weighted univariate interpolation formulas are then used to construct weighted sparse grid interpolants, using the standard Smolyak algorithm. This algorithm results in sparse grids with higher number of support nodes in regions of the random domain with higher probability density. Several numerical examples are presented to demonstrate that the proposed approach results in a more efficient algorithm, for the purpose of computation of moments of the stochastic solution, while maintaining the accuracy of the approximation of the solution.

Received 3 September 2009; Revised 14 July 2010; Accepted 18 July 2010

KEY WORDS: Smolyak algorithm; sparse grid; non-uniform distributions; stochastic collocation method; stochastic differential equations; uncertainty quantification

1. INTRODUCTION

In recent years, there has been a growing interest in analyzing and quantifying the effect of random input parameters during the simulation of physical phenomenon governed by ordinary or partial differential equations (ODE or PDE). The source of random inputs includes uncertainty in system parameters, boundary or initial conditions, etc. These uncertain input parameters can be modeled as random processes, using which the original governing equations can be reformulated as stochastic ordinary or partial differential equations (SODEs/SPDEs). Over the years, much of the research effort has focussed on developing numerical techniques to efficiently solve these

*Correspondence to: N. R. Aluru, Department of Mechanical Science and Engineering, Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, 405 N. Mathews Avenue, Urbana, IL 61801, U.S.A.
†E-mail: aluru@illinois.edu

differential equations with random inputs (for an overview, see [1] and references therein). Traditionally, sampling-based methods such as Monte Carlo (MC) method have been used for systems with random inputs. These methods involve generating various realizations of the input parameters according to the underlying probability distribution, and repeatedly employing the deterministic solver for each realization. Although the MC method is straightforward to implement and readily generates the required statistics, the simulations may become expensive as it offers slow convergence rate. Notably, the convergence rate for the MC method does not depend on the number of random dimensions or the smoothness of the stochastic solution in the random domain. The convergence of the MC method can be improved by using techniques such as the Latin Hypercube Sampling [2], the quasi-Monte Carlo method [3], the Markov Chain MC method [4], etc.

Among the non-sampling approaches, the most widely used method includes the stochastic Galerkin approach, which seeks to directly construct an approximation for the unknown stochastic solution based on complete orthogonal polynomials. The stochastic Galerkin method was initially developed by Ghanem and Spanos [5] using the Wiener–Hermite polynomial chaos expansion [6]. The method was further generalized to improve performance for a wider class of problems, such as using hypergeometric polynomials from the Askey scheme to obtain exponential convergence rate for non-Gaussian random processes [7], wavelet-based Weiner–Haar basis to deal with sharp or even discontinuous variation in the random domain [8, 9] and piecewise polynomial expansions [10, 11]. The stochastic Galerkin method takes advantage of an assumption of smoothness of the stochastic solution in the random domain and provides high accuracy and faster convergence rate. However, as the number of stochastic dimensions of the problem increases, the number of basis functions needed to obtain accurate results increases rapidly, which reduces the efficiency. In addition, the coupled nature of the deterministic equations that need to be solved to determine the modes of the solution makes the implementation non-trivial. It may be further complicated in situations when the governing equations take a complicated form, such as non-linear terms and coupled multiphysics.

Recently, there has been a growing interest in another class of methods known as stochastic collocation method [12–14]. The basic idea of the stochastic collocation approach is to approximate the unknown stochastic solution by a polynomial interpolation function in the multi-dimensional random domain. The interpolation is constructed by repeatedly solving (sampling) the deterministic problem at a *pre-determined* set of nodes. This approach offers high resolution as the stochastic Galerkin method as well as easy implementation as the sampling-based methods. The computational effort required for the collocation approach depends on the number of support nodes used for constructing the interpolation. As a result, the key issue for this approach is the *selection* of the support nodes, such that using the minimal number of nodes one achieves a good approximation. One such possible choice for constructing the multi-dimensional interpolation is to use tensor product of appropriate one-dimensional interpolation functions. For example, in [13] a methodology was proposed which employs a collocation scheme based on tensor product of one-dimensional interpolation functions using Gauss quadrature nodes. Although, the tensor product approach easily extends the one-dimensional interpolation formula to multi-dimensional case, it rapidly gets expensive, as the number of realizations required to construct the interpolation increases exponentially with the number of random dimensions.

A more attractive choice, as proposed in [12], is based on sparse grids generated using the Smolyak algorithm [15]. Smolyak's construction, also referred as *sparse grid*, *hyperbolic cross* or *Boolean blending*, provides a general tool for constructing efficient algorithms for solving multivariate problems. Since its introduction [15], it has been analyzed by many authors (e.g. [16–18]) and applied to many problems including multi-dimensional integration [17], approximation of multivariate functions [16, 19] and solution of differential and integral equations [20] (see [20] for an in-depth review of sparse grids). In the context of interpolation, it has been shown that the Smolyak construction leads to algorithms that are *almost optimal*, in the sense that it preserves the quality of the univariate interpolation formula (up to logarithmic terms) in the multi-dimensional case, for different classes of functions with bounded mixed derivatives [16]. For sufficiently smooth functions, the Smolyak algorithm may be used to construct multi-dimensional interpolants with

DOI: 10.1002/nme

orders of magnitude reduction in the number of support nodes to give the same level of approximation as the usual tensor product. Sparse grid interpolants can be constructed based on Lagrange interpolation [12, 21, 22] or piecewise linear basis functions [14, 23].

In the context of solution of differential equations with random inputs, in addition to approximating the unknown stochastic solution, we are also interested in computing its moments, e.g. mean and variance. These moments are defined as weighted integrals of the stochastic solution with respect to the probability measures in the multi-dimensional random domain. For best performance, it is required to include the effect of underlying probability measures while constructing the approximation. To this end, the stochastic Galerkin approach employs polynomials that are orthogonal with respect to the underlying probability measure, such as the Askey scheme of polynomials [7] for certain standard distributions or numerically generated polynomials for arbitrary distributions. For the stochastic collocation approaches based on high-order polynomial interpolation, one may employ Lagrange interpolation based on the roots of the orthogonal polynomials, chosen from the Askey scheme [24] or generated numerically [25]. However, the stochastic collocation approach based on piecewise linear interpolation is restricted to constructing highly accurate approximation for the unknown stochastic solution, and then computing the moments based on the constructed approximation. Such a procedure employs the information regarding the probability measure only for the computation of moments, and not while constructing the approximation or during the *node selection* step. In this work, we are interested in the collocation approach based on piecewise linear interpolation owing to its applicability to a wider class of problems.

The collocation approach using piecewise linear interpolation employs sparse grids which uniformly sample the unknown function in the random domain. Thus, in an effort to achieve high accuracy for the moments, such a construction would attempt to accurately approximate the stochastic solution throughout the random domain, which may unnecessarily increase the computational cost. For example, while computing moments based on highly skewed or localized probability distributions, it may only be required to approximate the solution with high accuracy in regions of high probability density, while maintaining reasonable accuracy in other parts of the random domain. Recently, several adaptive stochastic collocation approaches [26–28] have been proposed, for effectively approximating the stochastic solution in the presence of discontinuities or sharp variations in the random domain. Based on appropriate error criterion, these approaches construct the approximation in an adaptive manner, such that higher number of support nodes are placed close to the discontinuities or sharp variations. Such adaptive approaches may also be used in the context of non-uniform probability measures, for constructing approximations with varying levels of accuracy in different regions of the random domain. However, as one needs to evaluate the adaptivity criterion, the implementation of such approaches may not be as straightforward as the standard sparse grid interpolation algorithm. We should also mention that one could employ the Rosenblatt transformation [29] to directly map the set of underlying random variables with non-uniform distribution to one with a uniform distribution. However, for the class of methods that seek to construct an approximation for the stochastic solution using interpolation, such a transformation cannot guarantee good performance for arbitrary probability measures (as discussed further in Appendix A).

We recognize that in order to develop numerical algorithms for the solution of stochastic differential equations, which are also efficient for computation of moments of the stochastic solution, we need to incorporate the information regarding the non-uniform probability measures during the construction of the approximation or the node selection step. To this end, we need to consider algorithms for constructing multi-dimensional approximation, which are optimal with respect to the approximation error measured in a weighted sense. One such algorithm has been proposed in [30], which constructs a piecewise constant interpolant for weighted L_1 approximation of multivariate functions. The approximation error is measured in the weighted L_1 semi-norm, under which the approximation problem is directly related to the weighted integration problem. As mentioned in [30], any algorithm developed for the L_1 approximation of functions, yields an integration algorithm with the error at least as small as the approximation error. Another algorithm for weighted L_1 approximation has been proposed in [31], which seeks to construct

DOI: 10.1002/nme

piecewise polynomial interpolants for the multivariate objective functions, based on Smolyak's construction. This algorithm employs one-dimensional piecewise polynomial interpolants, where the interpolation points are specially chosen depending on the regularity of the objective function and the prescribed integration weight. The framework developed here is based on such a choice of univariate interpolation nodes, which has been further modified in order to ensure nestedness property for the resulting sparse grids.

In this work, we propose a *weighted Smolyak algorithm* based on piecewise linear basis functions, which incorporates information regarding the probability measures, during the construction of sparse grids. The proposed algorithm assumes that the probability measure admits a tensor product structure and is thus applicable for the solution of differential equations with random inputs which can be represented in terms of independent random variables. The basic idea is to construct univariate piecewise linear interpolation formulas, that employ higher number of support nodes in regions of higher probability density based on the marginal probability distributions. Using these *weighted* univariate interpolation formulas, we then construct the *weighted* sparse grid interpolants, based on the standard Smolyak algorithm. This algorithm results in sparse grids with higher number of support nodes in regions of the multi-dimensional random domain with higher probability density. We consider several numerical examples to demonstrate that the proposed approach results in a more efficient algorithm, for the purpose of computation of moments of the stochastic solution with respect to non-uniform probability measures while maintaining the accuracy of the approximation of the solution.

The remainder of the paper is organized as follows: In Section 2 we present the problem formulation and briefly discuss the stochastic collocation approach for the solution of stochastic differential equations. In Section 3 we introduce the standard Smolyak construction for multi-dimensional interpolation and integration. Using this, we describe the proposed weighted Smolyak algorithm in Section 4. In Section 5 we consider several numerical examples to demonstrate the efficiency of the proposed approach. We finally conclude the discussion in Section 6.

2. PROBLEM FORMULATION

2.1. Stochastic differential equations

Let $(\Theta, \mathscr{F}, \mathscr{P})$ denote a probability space, where Θ is the set of elementary events, $\mathscr{F}$ is the σ-algebra of events and $\mathscr{P}$ is the probability measure. The symbol θ specifies an elementary event in Θ and in the following presentation any quantity with θ-dependence denotes a random quantity. Let $\Omega \subset \mathbb{R}^d$ $(d = 1, 2, 3)$ denote a d-dimensional physical domain with boundary $\partial\Omega$. We consider the following problem: find a stochastic function $u(\mathbf{x}, \theta) : \bar{\Omega} \times \Theta \to \mathbb{R}$, such that $\mathscr{P}$-almost everywhere $\theta \in \Theta$, the following holds:

$$\mathscr{L}(u; \mathbf{x}, \theta) = f(\mathbf{x}, \theta), \quad (\mathbf{x}, \theta) \in \Omega \times \Theta, \tag{1}$$

with boundary condition

$$\mathscr{B}(u; \mathbf{x}, \theta) = g(\mathbf{x}, \theta), \quad (\mathbf{x}, \theta) \in \partial\Omega \times \Theta, \tag{2}$$

where $\mathscr{L}$ denotes a linear or non-linear differential operator, $\mathscr{B}$ denotes a boundary operator (which takes different forms on different parts of the boundary $\partial\Omega$), and $\mathbf{x} \in \mathbb{R}^d$ is a point in the physical space Ω. We assume that the boundary $\partial\Omega$ is sufficiently regular and the forcing functions f and g are properly posed, such that the stochastic differential equations (Equations (1)–(2)) are well posed $\mathscr{P}$-almost everywhere $\theta \in \Theta$.

2.2. Finite-dimensional noise assumption

The first step toward numerical solution of Equations (1)–(2) is to reduce the infinite-dimensional probability space into a finite-dimensional space. This can be accomplished by characterizing the probability space in terms of a finite number of random variables. The choice of methodology

DOI: 10.1002/nme

and the ability to correctly characterize infinite-dimensional random processes largely depend on the extent of available information regarding these processes. One such possible choice is the Karhunen–Loève (KL) expansion [32], which employs a spectral decomposition of the covariance function of the input random processes, to represent them in terms of a set of *uncorrelated* random variables [33, 34]. Polynomial chaos expansion can also be used to represent random fields [35–37], where the coefficients of the PC modes can be determined based on the principles of maximum likelihood [36] or maximum entropy [37]. Further, we assume that the set of finite number of random variables, denoted as $\boldsymbol{\xi}=\{\xi_i\}_{i=1}^n$ are mutually independent. However, we must note that representing the input random parameters in terms of independent random variables may not be a trivial exercise. For example, when KL-expansion is used for representing *non-Gaussian* random processes, these random variables would not necessarily be independent. In this work, we assume that the random input parameters are already characterized in terms of a finite number of mutually independent random variables with sufficient accuracy.

Let $\boldsymbol{\xi}=\{\xi_i\}_{i=1}^n$ represent mutually independent random variables with images $\Gamma_i \equiv \xi_i(\Theta)$ and probability density functions $\rho_i : \Gamma_i \to \mathbb{R}^+$, for $i=1,\ldots,n$. Then, the joint probability density function $\rho(\boldsymbol{\xi})$ is given as

$$\rho(\boldsymbol{\xi})=\prod_{i=1}^{n}\rho_i(\xi_i) \quad \forall \boldsymbol{\xi}\in\Gamma, \tag{3}$$

where $\Gamma=\prod_{i=1}^n \Gamma_i$ represents the support of the set of random variables. Using this, we can rewrite Equations (1)–(2) as: we seek stochastic function $u(\mathbf{x},\boldsymbol{\xi}):\bar{\Omega}\times\Gamma\to\mathbb{R}$ such that

$$\mathscr{L}(u;\mathbf{x},\boldsymbol{\xi})=f(\mathbf{x},\boldsymbol{\xi}), \quad (\mathbf{x},\boldsymbol{\xi})\in\Omega\times\Gamma, \tag{4}$$

with boundary condition

$$\mathscr{B}(u;\mathbf{x},\boldsymbol{\xi})=g(\mathbf{x},\boldsymbol{\xi}), \quad (\mathbf{x},\boldsymbol{\xi})\in\partial\Omega\times\Gamma, \tag{5}$$

which represents a $(d+n)$-dimensional system, where d and n refer to the dimensionality of the physical space Ω and the random space Γ, respectively.

2.3. *Numerical solution*: *Stochastic collocation method*

Recently, the stochastic collocation approach has emerged as an attractive alternative to the widely popular stochastic Galerkin method, for the numerical solution of stochastic systems (as given by Equations (4)–(5)). The collocation approach offers high resolution and fast convergence rate as the Galerkin method, as well as ease of implementation as the sampling-based methods, such as MC method. The basic idea of the stochastic collocation approach is to approximate the unknown stochastic solution by a polynomial interpolation function in the multi-dimensional random space. Given a set of support nodes $\Theta_N=\{\boldsymbol{\xi}^i\}_{i=1}^N\subset\Gamma$, an interpolation for the unknown stochastic solution $u:\bar{\Omega}\times\Gamma\to\mathbb{R}$ can be constructed as follows:

$$\hat{u}(\mathbf{x},\boldsymbol{\xi})\equiv\mathscr{I}u(\mathbf{x},\boldsymbol{\xi})=\sum_{i=1}^{N}u(\mathbf{x},\boldsymbol{\xi}^i)L_i(\boldsymbol{\xi}), \tag{6}$$

where $\{u(\mathbf{x},\boldsymbol{\xi}^i)\}_{i=1}^N$ are the values sampled at the support nodes Θ_N, and $\{L_i(\boldsymbol{\xi})\}_{i=1}^N$ represent the interpolation basis functions, such that $L_i(\boldsymbol{\xi}^j)=\delta_{ij}$, $1\leqslant i,j\leqslant N$. Using this approximation in the governing equations (Equations (4)–(5)), with the property of the interpolation polynomials $L_i(\boldsymbol{\xi}^j)=\delta_{ij}$, the collocation procedure immediately leads to: for $k=1,\ldots,N$,

$$\mathscr{L}(u(\mathbf{x},\boldsymbol{\xi}^k);\mathbf{x},\boldsymbol{\xi}^k)=f(\mathbf{x},\boldsymbol{\xi}^k), \quad \mathbf{x}\in\Omega, \tag{7}$$

subject to boundary condition

$$\mathscr{B}(u(\mathbf{x},\boldsymbol{\xi}^k);\mathbf{x},\boldsymbol{\xi}^k)=g(\mathbf{x},\boldsymbol{\xi}^k), \quad \mathbf{x}\in\partial\Omega. \tag{8}$$

DOI: 10.1002/nme

Thus, the stochastic collocation procedure reduces to solving a set of N decoupled deterministic systems, at each nodal point $\boldsymbol{\xi}^k, k=1,\ldots,N$, in the given set of support nodes Θ_N.

Using the sampled values for the unknown stochastic solution, we can compute the required moments in a straightforward manner. For example, the mth moment $\mu_m(u)(x)=\mathbb{E}[u^m(\mathbf{x},\boldsymbol{\xi})]$, where $\mathbb{E}[\cdot]$ denotes the expectation operator, can be computed as

$$\mu_m(u)(\mathbf{x}) = \mathbb{E}[u^m(\mathbf{x},\boldsymbol{\xi})] = \int_\Gamma u^m(\mathbf{x},\boldsymbol{\xi})\rho(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi} s \tag{9}$$

$$= \sum_{i=1}^{N} u^m(\mathbf{x},\boldsymbol{\xi}^i)w_i, \quad w_i = \int_\Gamma L_i(\boldsymbol{\xi})\rho(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}, \tag{10}$$

where $\{w_i\}_{i=1}^N$ are the weights that can be pre-computed, using the prior knowledge of the basis functions in analytical form.

The construction of the multivariate interpolation as given by Equation (6) is central to the stochastic collocation approach. Although there exists a well-developed and extensive classical theory of univariate interpolation, such a construction in the multivariate case is not trivial. The computational effort required for the collocation approach is typically N times the effort required for the deterministic problem, where N represents the total number of nodes in the set Θ_N. Thus, the key issue for the stochastic collocation procedure is the *selection* of this set of nodes Θ_N, such that using the minimal number of nodes one achieves a good approximation. One such possible choice, as proposed in [12] and further explored by other authors (e.g. [14, 28]), is based on sparse grids generated using the Smolyak algorithm [15]. These approaches involve constructing highly accurate sparse grid interpolation for the unknown stochastic solution, and then computing the moments for the stochastic solution using Equations (9)–(10). It is worth noting that these existing approaches only employ the probability measure $\rho(\boldsymbol{\xi})$ for the computations of moments, and not during *node selection*. In this work, we propose a weighted Smolyak algorithm, which allows one to incorporate arbitrary non-uniform probability measures during the node selection step, such that one places higher number of nodes in regions of high probability. The resulting weighted sparse grids lead to a significant improvement in the accuracy of computed moments as compared to the standard algorithm.

3. SMOLYAK ALGORITHM

Smolyak's construction provides a general tool for constructing efficient algorithms for solving multivariate problems. Since its introduction [15], it has been applied to many problems including multi-dimensional integration [17], approximation of multivariate functions [16, 19] and solution of differential and integral equations [20]. In the context of multivariate interpolation, the Smolyak algorithm provides a way to extend the univariate interpolation formula to higher dimensions using the minimal number of support nodes. It constructs the multi-dimensional interpolants as an algebraic sum of low-order tensor products based on appropriate one-dimensional interpolation functions. Such a construction leads to orders of magnitude reduction in the number of support nodes, while maintaining the interpolation quality of the univariate formula for higher dimensions up to a logarithmic factor [16]. Without loss of generality, in the following presentation, we assume that the bounded support of the random variables $\{\xi_i\}_{i=1}^n$ is $\Gamma_i=[0,1]$, and thus the bounded random domain $\Gamma=[0,1]^n$ is an n-hypercube.

3.1. Univariate interpolation

Since the Smolyak algorithm is constructed based on one-dimensional algorithms, we first consider the univariate case. Let $f:[0,1]\to\mathbb{R}$ be a function in one-dimensional, which is approximated using a sequence of interpolation formulas given as,

$$\mathscr{I}^k(f)(\xi) = \sum_{j=1}^{m_k} f(\xi_j^k)l_j^k(\xi) \quad \text{for each } k\geqslant 1, \tag{11}$$

DOI: 10.1002/nme

with the set of support nodes $\mathscr{X}^k=\{\xi_j^k|\,\xi_j^k\in[0,1],\,j=1,\ldots,m_k\}$, and interpolation basis functions $\ell^k=\{l_j^k|l_j^k\in C[0,1],\ \ j=1,\ldots,m_k\}$, such that $l_j^k(\xi_i^k)=0,\forall i\neq j$. Here k and m_k refer to the *depth* of interpolation and the total number of support nodes at depth k, respectively. One possible choice would be to use the piecewise linear basis functions with equidistant nodes [14, 19, 28]. For this choice, the set of equidistant nodes $\mathscr{X}^k$ can be described using,

$$m_k=\begin{cases}1 & \text{if } k=1,\\ 2^{k-1}+1 & \text{if } k>1,\end{cases} \tag{12}$$

$$\xi_j^k=\begin{cases}0.5 & \text{for } j=1 \text{ if } m_k=1,\\ \dfrac{j-1}{m_k-1} & \text{for } j=1,\ldots,m_k \text{ if } m_k>1,\end{cases} \tag{13}$$

which results in a nested set of nodes, such that $\mathscr{X}^k\subset\mathscr{X}^{k+1},\forall k\geqslant 1$. Using this set of support nodes, the piecewise linear basis functions can be defined as the standard hat functions as follows:

$$l_j^k(\xi)=1 \quad \text{for } k=1 \text{ and,} \tag{14}$$

$$l_j^k(\xi)=\begin{cases}1-(m_k-1)|\xi-\xi_j^k| & \text{if } |\xi-\xi_j^k|<\dfrac{1}{m_k-1}\\ 0 & \text{otherwise,}\end{cases} \tag{15}$$

for $k>1$ and $j=1,\ldots,m_k$.

Among other choices for constructing univariate interpolation formulas, one may also employ Lagrange interpolation based on non-equidistant Gauss quadrature points or the extrema of the Chebyshev polynomials (also known as Chebyshev–Gauss–Lobatto nodes). For sufficiently smooth objective functions, Lagrange interpolation offers faster error decay with increasing number of nodes as compared with piecewise linear basis functions. However, for functions that exhibit limited regularity, its performance may degrade significantly. In this work, we employ piecewise linear basis functions as it results in algorithms which are applicable to a larger class of functions.

3.2. Univariate to multivariate interpolation

Starting with the univariate interpolation formula (Equation (11)), the sparse interpolant $A_{q,n}(f)$, for a multi-dimensional function $f:[0,1]^n\to\mathbb{R}$ is given by the Smolyak algorithm (see e.g. [16–18]) as,

$$\mathscr{I}(f)\equiv A_{q,n}(f)=\sum_{q+1\leqslant|\mathbf{k}|\leqslant q+n}(-1)^{q+n-|\mathbf{k}|}\cdot\binom{n-1}{q+n-|\mathbf{k}|}\cdot(\mathscr{I}^{k_1}\otimes\ldots\otimes\mathscr{I}^{k_n})(f), \tag{16}$$

where q is the depth of interpolation ($q\geqslant 0, q\in\mathbb{N}_0$), n is the number of stochastic dimensions and $|\mathbf{k}|=k_1+k_2\cdots+k_n$, where $k_i, i=1,\ldots,n$, can be thought of as the level of interpolation along the $i^{th}-$dimension. In order to construct the sparse interpolant $A_{q,n}(f)$ from scratch, one needs to evaluate the function at sparse grid points $\mathscr{H}_{q,n}$ given by,

$$\mathscr{H}_{q,n}=\bigcup_{q+1\leqslant|\mathbf{k}|\leqslant q+n}(\mathscr{X}^{k_1}\times\cdots\times\mathscr{X}^{k_n}). \tag{17}$$

Remark 1

We must note that since the univariate node distribution defined by Equations (12)–(13) results in nested grids $\mathscr{X}^k\subset\mathscr{X}^{k+1},\forall k\geqslant 1$, the resulting sparse grids are also nested, such that $\mathscr{H}_{q,n}\subset\mathscr{H}_{q+1,n},\forall q\geqslant 0$. The nestedness of sparse grids is an important property, as it facilitates reuse of functional evaluations as one seeks to increase the accuracy of approximation by increasing the depth of interpolation q.

DOI: 10.1002/nme

3.3. Multivariate integration

Similar to interpolation, one can also employ the Smolyak algorithm to extend the given one-dimensional quadrature rule for integrating a one-dimensional function to higher dimensions. Suppose we are interested in approximating the multi-dimensional integral given as:

$$\mathbb{I}_n(f) = \int_\Gamma f(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}. \tag{18}$$

Based on the one-dimensional interpolation formulas $\mathscr{I}^k,\ k \geqslant 1$ (Equation (11)), we can define a sequence of one-dimensional quadrature rules as follows:

$$\mathbb{I}_1(f) = \int_0^1 f(\xi)\,\mathrm{d}\xi \approx \mathscr{Q}^k(f) = \sum_{j=1}^{m_k} f(\xi_j^k) w_j^k \quad \text{for each } k \geqslant 1 \quad \text{such that } w_j^k = \int_0^1 l_j^k(\xi)\,\mathrm{d}\xi, \tag{19}$$

where $\{w_j^k\}$ are the quadrature weights. Following this, the one-dimensional integration rule can be extended to integrate multi-dimensional functions as

$$\mathbb{I}_n(f) \approx Q_{q,n}(f) = \sum_{q+1 \leqslant |\mathbf{k}| \leqslant q+n} (-1)^{q+n-|\mathbf{k}|} \cdot \binom{n-1}{q+n-|\mathbf{k}|} \cdot (\mathscr{Q}^{k_1} \otimes \cdots \otimes \mathscr{Q}^{k_n})(f), \tag{20}$$

where $q \geqslant 0$ refers to the *depth* parameter, such that increasing q increases the accuracy of integration.

We must note that in the context of solving stochastic differential equations, in addition to approximating the stochastic solutions, we are also interested in computing their moments, which are defined as weighted integrals in the multi-dimensional random domain (as given by Equation (9)). As mentioned in [17, Section 4, Remark 4], the procedure given by Equation (20) can be easily modified to approximate weighted integrals defined as,

$$\mathbb{I}_n^w(f) = \int_\Gamma f(\boldsymbol{\xi})\rho(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}, \tag{21}$$

for situations where $\rho(\boldsymbol{\xi}) = \prod_{i=1}^n \rho_i(\xi_i)$ admits a tensor structure. One simply needs to redefine the one-dimensional quadrature rules for each dimension i as:

$$\mathbb{I}_1^w(f) \approx \mathscr{Q}_i^k(f) = \sum_{j=1}^{m_k} f(\xi_j^k) w_{i,j}^k \quad \text{for each } k \geqslant 1 \quad \text{such that } w_{i,j}^k = \int_0^1 l_j^k(\xi)\rho_i(\xi)\,\mathrm{d}\xi. \tag{22}$$

Using the redefined one-dimensional quadrature rules in Equation (20), we can easily obtain the quadrature rule $Q_{q,n}(f)$ such that $\mathbb{I}_n^w(f) \approx Q_{q,n}(f)$. It should be noted that only the quadrature weights for the integration rule $Q_{q,n}$ are modified, however, the sparse grid nodes $\mathscr{H}_{q,n}$ remain the same.

3.4. Error estimates

Using the univariate error bounds, *a priori* error estimators for integration of multivariate functions using the Smolyak construction are derived in [17]. For an n-variate objective function $f : [0, 1]^n \to \mathbb{R}$, we first define,

$$F_n^r = \{f \mid D^\beta f \text{ continuous if } \beta_i \leqslant r\ \forall i\}, \quad D^\beta f = \frac{\partial^{|\beta|} f}{\partial \xi_1^{\beta_1} \cdots \xi_n^{\beta_n}}$$

with $\beta \in \mathbb{N}_0^n$ and $|\beta| = \sum_{i=1}^n \beta_i$. Then for a quadrature formula Q for any function $f \in F_n^r$, we have the error bound,

$$|\mathbb{I}_n(f) - Q(f)| \leqslant \|\mathbb{I}_n - Q\| \cdot \|f\|, \tag{23}$$

in terms of the norm on F_n^r, i.e. $\|\mathbb{I}_n - Q\|$ denotes the respective operator norm of $\mathbb{I}_n - Q$.

DOI: 10.1002/nme

For $f \in F_n^2$, the univariate error bound for integration based on piecewise linear basis functions is given as

$$\| \mathbb{I}_1 - \mathcal{Q}^k \| \leqslant c \cdot (m_k)^{-2} \quad \text{for each } k \geqslant 1, \tag{24}$$

for a constant c, where k and m_k refer to the depth of interpolation and the total number of support nodes used at that depth. Using the univariate error bound, the corresponding estimate for multi-dimensional integration can be given as [16, 17],

$$\| \mathbb{I}_n - Q_{q,n} \| \leqslant c_n \cdot (N^{-2} \cdot |\log_2 N|^{3(n-1)}), \tag{25}$$

where $c_n > 0$ is a constant which only depends on n, $Q_{q,n}$ denotes the sparse quadrature rule (given by Equation (20)) and $N = \dim(\mathcal{H}_{q,n})$ is the total number of support nodes. From the error estimate in Equation (25) we also notice that the accuracy of this method weakly depends on the number of stochastic dimensions in the logarithmic term. This implies that, unlike the MC method, the performance of the sparse grid method would suffer with increase in the number of dimensions.

Remark 2
We note that the error estimate given by Equation (25) is also applicable for weighted integrals $\mathbb{I}_n^w(f)$, for weights $\rho(\boldsymbol{\xi})$ that satisfy $g(\boldsymbol{\xi}) = f(\boldsymbol{\xi}) \cdot \rho(\boldsymbol{\xi}) \in F_n^2$. This suggests that the asymptotic convergence rate for the approximation of integrals with non-uniform weights would also be the same as that for integrals with uniform weights.

Having presented the classical Smolyak construction for multi-dimensional interpolation and integration, we now consider the weighted Smolyak algorithm in the next section.

4. WEIGHTED SMOLYAK ALGORITHM

As mentioned briefly in last section, while employing sparse grid interpolation procedure for the solution of stochastic differential equations, in addition to approximating the unknown stochastic solution, we are also interested in accurately computing its moments. The problem of computation of moments of the stochastic solution reduces to approximating integrals with respect to non-uniform weights in the multi-dimensional random domain. As mentioned, the sparse quadrature procedure for computing uniform integrals (Equation (20)) can be easily modified for computing weighted integrals. The modified quadrature formula is based on the same set of support nodes as before, which sample the stochastic solution uniformly in the random domain, and offers the same asymptotic convergence rate as that for the integrals with uniform weights. However, evaluating weighted integrals based on uniform sparse grids may not be ideal and may even require higher number of functional evaluations (as compared with the uniform integral case) to attain the asymptotic convergence rate, especially for highly skewed or localized probability measures. In order to develop numerical algorithms for solution of stochastic differential equations, which are also efficient for computation of moments of the stochastic solution, we need to incorporate the information regarding the non-uniform probability measures during the construction of the approximation or the node selection step.

To this end, we need to consider algorithms for constructing multi-dimensional approximation, which are optimal with respect to the approximation error measured in a weighted sense. A piecewise polynomial algorithm has been proposed in [31], which seeks to construct a weighted L_1 approximation for n-variate functions defined on $\mathbb{R}^n$. The approximation error is measured in the weighted L_1 semi-norm as

$$\| f - A(f) \|_{1,\rho} = \int_\Gamma |f(\boldsymbol{\xi}) - A(f)(\boldsymbol{\xi})| \rho(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi}, \tag{26}$$

where $A(f)$ represents the constructed approximation for the function f. We are interested in such algorithms that seek to optimally approximate the stochastic solution under the weighted L_1 semi-norm, as for this case, the approximation problem is directly related to the weighted

DOI: 10.1002/nme

integration problem. As mentioned in [30], any algorithm developed for the L_1 approximation of functions yields an integration algorithm with the error at least as small as the approximation error. The approximation procedure proposed in [31] employs one-dimensional piecewise polynomial interpolants, where the interpolation points are specially chosen depending on the regularity of the objective function and the prescribed integration weight. Based on such a choice for the univariate node distribution, we develop a weighted Smolyak algorithm using piecewise linear basis functions, which incorporates the integration weights $\rho(\boldsymbol{\xi})$ during the construction of sparse grids. This algorithm results in nested sparse grids with higher number of support nodes in regions of the multi-dimensional random domain with higher probability density. In the following, we present the construction of the proposed weighted Smolyak algorithm.

4.1. Selection of univariate nodes

We recall that the joint probability measure $\rho(\boldsymbol{\xi})$ can be written as $\rho(\boldsymbol{\xi})=\prod_{i=1}^{n}\rho_i(\xi_i)$ (Equation (3)), where $\rho_i:[0,1]\to\mathbb{R}^+$, for $i=1,\ldots,n$, represents the marginal probability measure in the ith-dimension. We define a sequence of interpolation formulas for each dimension i as,

$$\mathscr{I}_i^k(f)(\xi)=\sum_{j=1}^{m_k^i} f(\xi_{i,j}^k)l_{i,j}^k(\xi) \quad \text{for each } k\geqslant 1, \tag{27}$$

with the set of support nodes $\mathscr{X}_i^k=\{\xi_{i,j}^k|\xi_{i,j}^k\in[0,1],\ j=1,\ldots,m_k^i\}$, where k refers to the depth of interpolation and m_k^i refers to the number of support nodes at depth k for the ith dimension. Given the marginal distribution ρ_i, for each dimension $i=1,\ldots,n$, the set of support nodes $\mathscr{X}_i^k$ is defined using Algorithm 1. The basic idea behind Algorithm 1 is that for uniform measures ($\rho_i=1$), the set of equidistant nodes $\mathscr{X}_i^k(k>1)$ can be thought to be generated as a result of successively partitioning each element (starting with two elements $[0,0.5]$ and $[0.5,1]$ for $k=2$) into two equal elements, and choosing the endpoints of these elements as the support nodes. For the case of non-uniform measures, the set of support nodes $\mathscr{X}_i^k$ are generated by again successively partitioning the elements, but instead of just selecting the endpoints of each element, we place a number of equidistant nodes within each element. The number of equidistant nodes which are

Algorithm 1 Selection of univariate nodes

Given the dimension number i and marginal PDF $\rho_i:[0,1]\to\mathbb{R}$.
For $k=1$, $m_1^i=1$ and $\mathscr{X}_i^1=\{0.5\}$. Define $n_1^1=2$.
For $k\geqslant 2$, subdivide $[0,1]$ in $ne_k=2^{k-1}$ elements, with $h_k=1/ne_k$ and $y_j^k=jh_k,\ j=0,\ldots,ne_k$.
Set $\mathscr{X}_i^k=\emptyset$.

a. **FOR** each element $[y_{j-1}^k, y_j^k]$, $j=1,\ldots,ne_k$:
b. Define $\omega_j^k=\sup\{\rho_i(t): y_{j-1}^k\leqslant t\leqslant y_j^k\}$ and $a_j^k=(h_k)^2\omega_j^k$.
c. Define $\eta_j^k=\left\lceil \frac{ne_k(a_j^k)^{1/2}}{A_k}\right\rceil$, where $A_k=\sum_{j=1}^{ne_k}(a_j^k)^{1/2}$.
d. **IF** $\eta_j^k>0$, set $\gamma_j^k=\lceil\log_2(\eta_j^k)\rceil$.
ELSE $\gamma_j^k=0$.
END IF.
e. Select $n_j^k=\max\left[2^{\gamma_j^k}+1, \frac{n_{\lceil j/2\rceil}^{k-1}-1}{2}+1\right]$ equidistant nodes in the interval $[y_{j-1}^k, y_j^k]$, including y_{j-1}^k and y_j^k, denoted by $\mathscr{Y}_j$.
f. Add the set of nodes $\mathscr{Y}_j$ corresponding to each element $[y_{j-1}^k, y_j^k]$ to the set of nodes $\mathscr{X}_i^k$, as $\mathscr{X}_i^k=\mathscr{X}_i^k\bigcup\mathscr{Y}_j$.
g. **END FOR**.

Set $m_k^i=\text{card}(\mathscr{X}_i^k)$, where $\text{card}(\mathscr{X}_i^k)$ represents *cardinality* or number of nodes in $\mathscr{X}_i^k$.

DOI: 10.1002/nme

placed within each element is decided based on the relative weight for that element and also to ensure nestedness.

This choice of support nodes is based on the procedure outlined in [31]. However, it must be noted that we have modified the number of equidistant nodes inserted in each element (see Algorithm 1, Step 3(d)–(e)) in order to ensure that the resulting one-dimensional node distribution is nested $\mathscr{X}_i^k \subset \mathscr{X}_i^{k+1}$, $k \geqslant 1$, for $i = 1, \ldots, n$. The algorithm given in [31], having computed η_j^k (as given by Step 3(c)), prescribes $(\eta_j^k + 1)$ equidistant nodes in the interval $[y_{j-1}^k, y_j^k]$ (including the endpoints), which would result in non-nested grids for arbitrary probability measures.

The set of interpolation basis functions $\ell_i^k = \{l_{i,j}^k | l_{i,j}^k \in C[0,1], j = 1, \ldots, m_k^i\}$, such that $l_{i,s}^k(\xi_{i,t}^k) = 0, \forall s \neq t$, can be defined as the standard hat functions based on the set of support nodes $\mathscr{X}_i^k$. In Figure 1 we demonstrate the one-dimensional node selection algorithm for three different probability measures with parameters $a, b > 0$, defined as:

$$\text{Unif}(a,b) = \frac{1}{(b-a)}, \tag{28}$$

$$\text{Beta}(a,b) = \frac{1}{B(a,b)} y^{a-1}(1-y)^{b-1}, \tag{29}$$

$$\text{Gaussian}(a,b) = \frac{1}{G(a,b)} e^{-a(y-b)^2}, \tag{30}$$

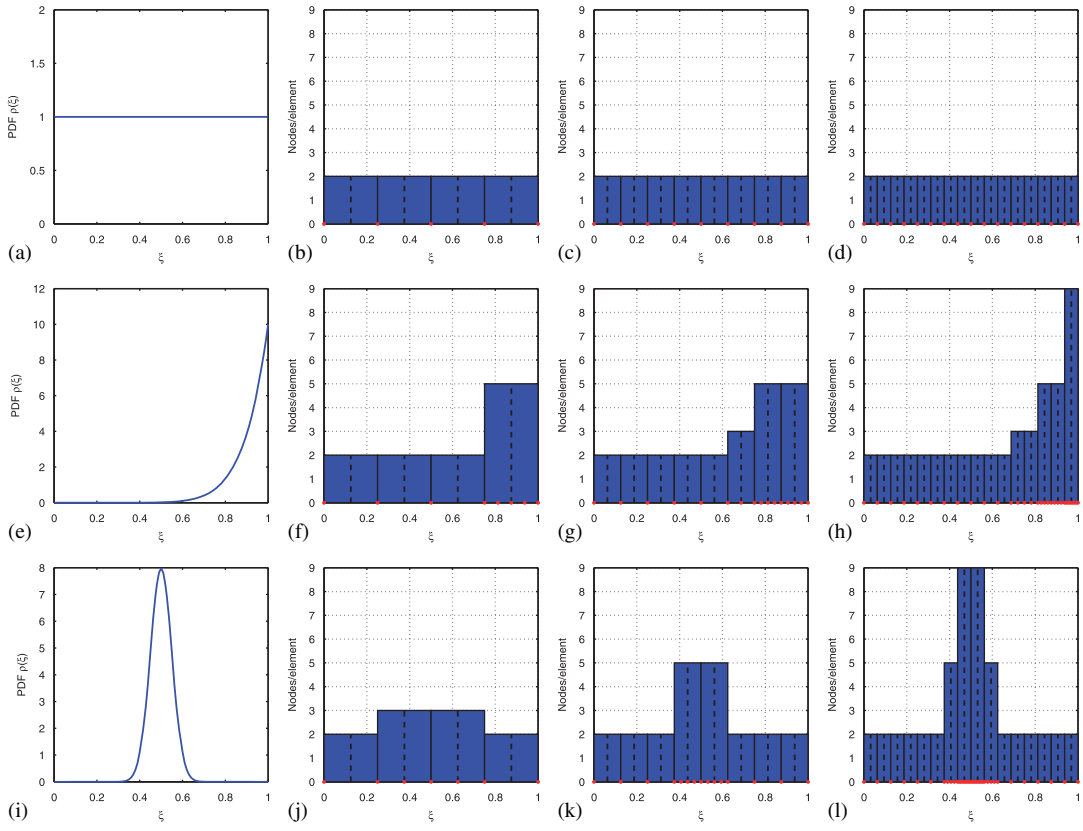

Figure 1. Illustration of univariate node selection algorithm for various probability measures: Unif(0, 1) (top row), Beta(10, 1) (middle row) and Gaussian(200, 0.5) (bottom row). For each depth k, the dashed lines denote elements or sub-intervals to be considered during Step 3 (Algorithm 1) at next depth $k+1$; (a) PDF Unif(0, 1); (b) Depth $k=3, m_k=5$; (c) Depth $k=4, m_k=9$; (d) Depth $k=5, m_k=17$; (e) PDF Beta(10, 1); (f) Depth $k=3, m_k=8$; (g) Depth $k=5, m_k=16$; (h) Depth $k=5, m_k=32$; (i) PDF Gaussian(200, 0.5); (j) Depth $k=3, m_k=7$; (k) Depth $k=4, m_k=15$; and (l) Depth $k=5, m_k=37$.

DOI: 10.1002/nme

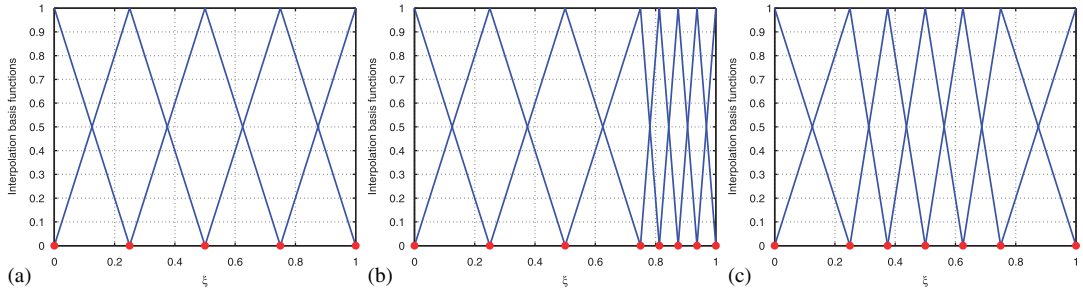

Figure 2. Univariate support nodes and basis functions for various distributions for depth $k=3$: (a) PDF Unif(0, 1); (b) PDF Beta(10, 1); and (c) PDF Gaussian(200, 0.5).

for $y \in [0,1]$, where B and G are defined as normalization constants to ensure that the total probability integrates to unity. The resulting univariate support nodes for depth $k=3$ and corresponding piecewise linear basis functions defined on those support nodes are shown in Figure 2.

4.2. *Weighted multivariate interpolation*

Having defined the sequence of weighted univariate interpolation formulas $\mathscr{I}_i^k (k \geqslant 1)$ for each dimension $i=1,\ldots,n$ (Equation (27)), the weighted sparse interpolant $A_{q,n}^w(f)$, for a multi-dimensional function $f:[0,1]^n \to \mathbb{R}$ can simply be constructed using Smolyak algorithm as,

$$\mathscr{I}(f) \equiv A_{q,n}^w(f) = \sum_{q+1 \leqslant |\mathbf{k}| \leqslant q+n} (-1)^{q+n-|\mathbf{k}|} \cdot \binom{n-1}{q+n-|\mathbf{k}|} \cdot (\mathscr{I}_1^{k_1} \otimes \ldots \otimes \mathscr{I}_n^{k_n})(f), \tag{31}$$

where, as before, $q \geqslant 0$ is the depth of interpolation, n is the number of stochastic dimensions, and $|\mathbf{k}| = k_1 + k_2, \ldots, k_n$, where k_i, $i=1,\ldots,n$, represents the level of interpolation along the ith dimension. In order to construct the weighted sparse interpolant $A_{q,n}^w(f)$, one needs functional evaluations at the weighted sparse grid nodes $\mathscr{H}_{q,n}^w$ given by:

$$\mathscr{H}_{q,n}^w = \bigcup_{q+1 \leqslant |\mathbf{k}| \leqslant q+n} (\mathscr{X}_1^{k_1} \times \cdots \times \mathscr{X}_n^{k_n}). \tag{32}$$

In Figure 3 we demonstrate the sparse grids generated using the proposed weighted Smolyak algorithm for a two-dimensional example ($n=2$), where ξ_1 and ξ_2 are assumed to be independent and identically distributed according to the probability distributions given by Equations (28)–(30). The number of nodes resulting from the uniform and weighted algorithms for $n=2, 4$ and 8 is shown in Table I.

Remark 3

We note that, for each dimension $i=1,\ldots,n$, the weighted univariate node distributions defined by Algorithm 1 results in nested grids $\mathscr{X}_i^k \subset \mathscr{X}_i^{k+1}, \forall k \geqslant 1$. As a consequence, the resulting weighted sparse grids are also nested, such that $\mathscr{H}_{q,n}^w \subset \mathscr{H}_{q+1,n}^w, \forall q \geqslant 0$.

Remark 4

We also note that for each level $q \geqslant 0$, the uniform sparse grids are subset of the corresponding weighted grids, such that $\mathscr{H}_{q,n} \subset \mathscr{H}_{q,n}^w$, in such a manner that in regions of higher probability density, the weighted algorithm automatically places more support nodes in addition to the nodes placed by the uniform algorithm. This feature of the proposed algorithm results in more efficient computation of moments of the stochastic solution, while at the same time maintaining the accuracy of the approximation of the solution as well. This point would be demonstrated through numerical examples in Section 5.

DOI: 10.1002/nme

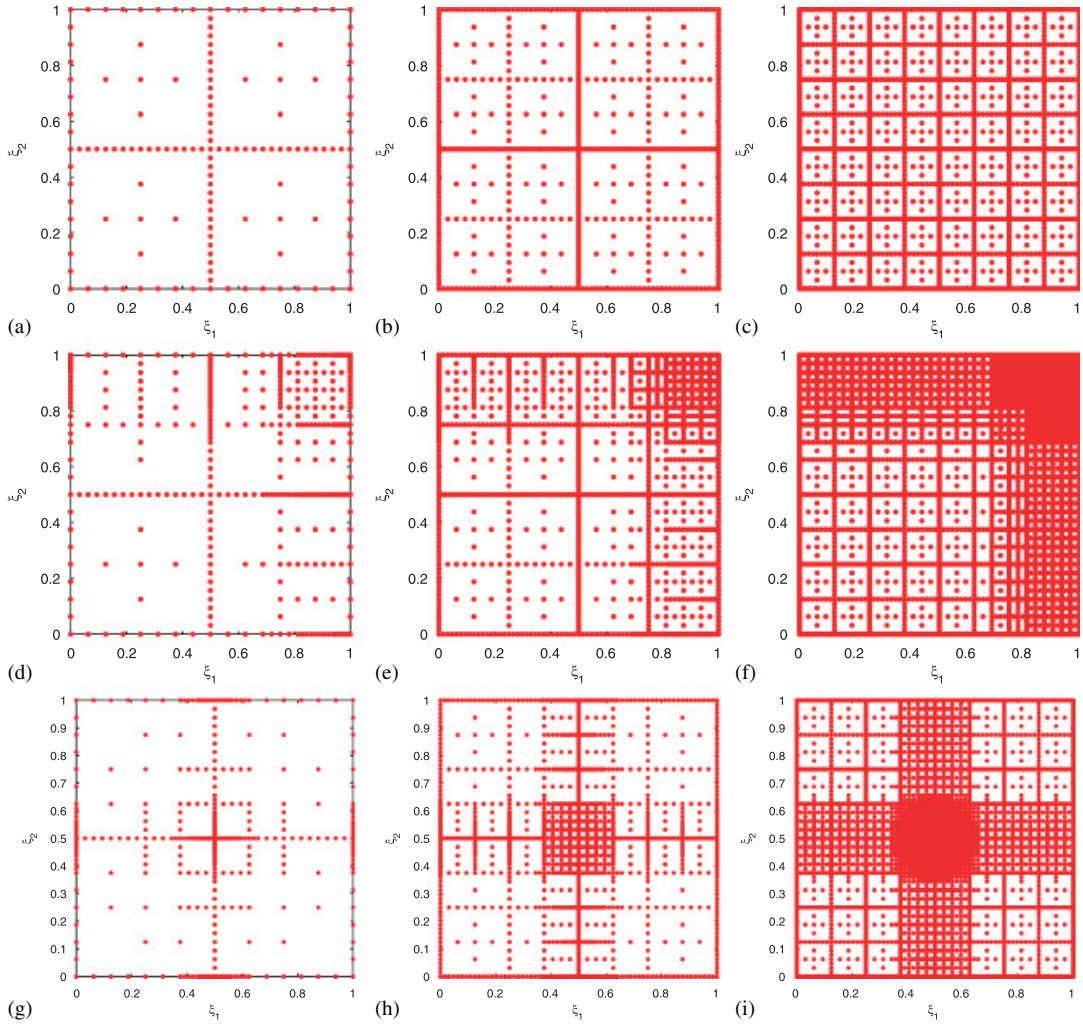

Figure 3. Sparse grids generated using weighted Smolyak algorithm ($n=2$) for the case of i.i.d. random variables $\{\xi_1, \xi_2\}$ distributed according to various probability measures: Unif(0, 1) (top row), Beta(10, 1) (middle row) and Gaussian(200, 0.5) (bottom row). (a) $q=5, N=145$; (b) $q=7, N=705$; (c) $q=9, N=3329$; (d) $q=5, N=382$; (e) $q=7, N=2036$; (f) $q=9, N=10\,182$; (g) $q=5, N=361$; (h) $q=7, N=2065$; and (i) $q=9, N=10\,665$.

4.3. Weighted multivariate integration

Based on the weighted one-dimensional interpolation rules $\mathscr{I}_i^k,\ k\geqslant 1,\ i=1,\ldots,n$ (Equation (27)), we can define a sequence of one-dimensional quadrature rules for each dimension $i=1,\ldots,n$ as follows:

$$\mathscr{Q}_i^k(f)=\sum_{j=1}^{m_k^i} f(\xi_{i,j}^k)w_{i,j}^k \quad \text{for each } k\geqslant 1 \quad \text{such that } w_{i,j}^k=\int_0^1 l_{i,j}^k(\xi)\rho_i(\xi)\mathrm{d}\xi, \tag{33}$$

where $\{w_{i,j}^k\}$ are the quadrature weights. Using the one-dimensional integration rules in Equation (20) we can obtain the multi-dimensional quadrature rule $Q_{q,n}^w(f)$, such that $\mathbb{I}_n^w(f)\approx Q_{q,n}^w(f)$.

For the choice of univariate support nodes given in [31], it has been shown that for the case of piecewise linear basis functions, the *worst case* weighted L_1 error (defined based on Equation (26)) for the sparse interpolant $A_{q,n}^w(f)$ asymptotically behaves as $\mathscr{O}(N^{-2}(\log(N))^{3(n-1)})$, where N

DOI: 10.1002/nme

Table I. Number of nodes generated using uniform and weighted Smolyak algorithm for i.i.d. random variables $\xi_i \sim \text{Gaussian}(200, 0.5),\ i = 1, \ldots, n$.

	Dimension $n=2$		Dimension $n=4$		Dimension $n=8$	
Depth q	Uniform	Weighted	Uniform	Weighted	Uniform	Weighted
0	1	1	1	1	1	1
1	5	5	9	9	17	17
2	13	17	41	49	145	161
3	29	49	137	209	849	1121
4	65	141	401	793	3937	6449
5	145	361	1105	2737	15713	32481
6	321	881	2929	8769	56737	147969
7	705	2065	7537	26465	190881	622273

represents the total number of support nodes in the weighted sparse grid $\mathscr{H}^w_{q,n}$. Further, using a well-known relation between weighted approximation and weighted integration, it has been shown that the worst-case error for the multi-dimensional integration rule $Q^w_{q,n}$ is also bounded by the corresponding error estimate derived for the approximation error. This suggests that the weighted Smolyak algorithm based on univariate node distribution proposed in [31] also offers the same convergence rate as that offered by the uniform Smolyak algorithm based on equidistant nodes (see Equation (25)). We must remark that since we have slightly modified the node distribution from that proposed in [31], in order to ensure nestedness of the resulting sparse grids, the convergence rate offered by our algorithm could be slightly different than that given in [31].

5. NUMERICAL RESULTS

In this section, we present several numerical examples to demonstrate the efficiency of the proposed weighted Smolyak algorithm. First, we consider the problem of approximation of weighted integrals in multiple dimensions that arise while computing the moments (mean and variance) of given test functions. Following that, we demonstrate the proposed approach for the solution of differential equations with random inputs.

5.1. Weighted integration problem

We consider the problem of computing the moments (mean and variance) for a family of functions, chosen from the Genz testing package [38], which has been used previously for evaluating the performance of multi-dimensional interpolation [16, 19] and integration [17] algorithms. We consider four family of functions $f_1, \ldots, f_4$, defined on $[0, 1]^n$, with a given name or attribute as shown in Table II, and approximate their moments with respect to non-uniform probability measures. Different test integrals can be obtained by varying the parameters $\mathbf{c} = (c_1, \ldots, c_n)$ and $\mathbf{d} = (d_1, \ldots, d_n)$, where the difficulty of the integrand increases monotonically with parameters $c_i > 0$ and parameters $\mathbf{d}$ act as shifting parameters. The parameters $\mathbf{c}$ and $\mathbf{d}$ are chosen randomly from $[0, 1]$, and similar to [17], we normalize c_i such that $\sum_{i=1}^{n} c_i = b_j,\ j = 1, \ldots, 4$, where b_j depends on the family of functions f_j as given in Table II.

5.1.1. Low-dimensional random inputs. We first consider a two-dimensional example ($n = 2$), where we assume y_1 and y_2 to be independent and identically distributed according to Beta$(10, 1)$ distribution, as given by Equation (29). For this case, we employ the uniform and weighted quadrature rules $Q_{q,n}(f)$ and $Q^w_{q,n}(f)$, to approximate the mean and variance for each of the test functions $f_j(\mathbf{y}),\ j = 1, \ldots, 4$. We denote the mean and variance computed using the uniform Smolyak algorithm based on level q as $\mu_{q,n}(f)$ and $v_{q,n}(f)$, respectively, and that computed using the weighted algorithm as $\mu^w_{q,n}(f)$ and $v^w_{q,n}(f)$, respectively. We report the results for levels $q = 0$ to $q = q_{\max}$, where $q_{\max} = 11$ and 12 for weighted and uniform algorithms, respectively. For the

DOI: 10.1002/nme

Table II. Genz test functions [17, 38].

j	Name	Function	Normalization constant b_j
1	Oscillatory	$f_1(\mathbf{y}) = \cos\left(2\pi d_1 + \sum_{i=1}^{n} c_i y_i\right)$	9.0
2	Product-peak	$f_2(\mathbf{y}) = \prod_{i=1}^{n} (c_i^{-2} + (y_i - d_i)^2)^{-1}$	7.25
3	Corner-peak	$f_3(\mathbf{y}) = \left(1 + \sum_{i=1}^{n} c_i y_i\right)^{-(n+1)}$	1.85
4	Gaussian	$f_4(\mathbf{y}) = \exp\left(-\sum_{i=1}^{n} c_i^2 \cdot (y_i - d_i)^2\right)$	7.03

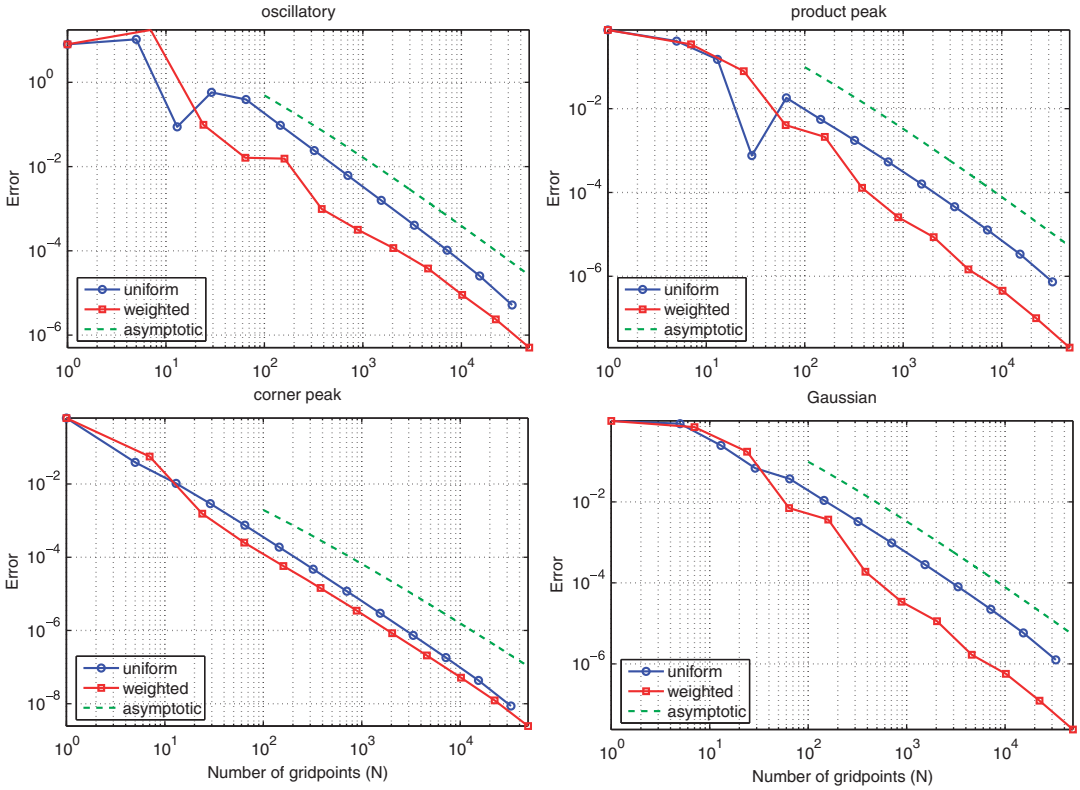

Figure 4. Normalized error in mean vs number of gridpoints N for Genz test functions [$n = 2$, $y_i \sim \text{Beta}(10, 1)$, $i = 1, 2$], using uniform and weighted Smolyak algorithms.

purpose of computing error in mean and variance, we treat the values obtained using the weighted algorithm based on level $q = 12$ (using $N = 105\,829$ support nodes) as *exact*. Finally, the error in mean and variance is normalized using these *exact* values. In Figures 4 and 5 we plot the normalized error in mean and variance, respectively, for all the test functions. In addition, we also show the asymptotic convergence rate as expected from the error estimate given by Equation (25) (dashed line). As can be seen, except for the corner-peak case, we obtain roughly an order of magnitude reduction in the error in mean and variance for all test functions, using the weighted algorithm as opposed to the uniform algorithm. In addition, in order to achieve the same level of accuracy in moments, the weighted construction requires much lesser number of functional evaluations as compared with the uniform algorithm. For example, for the oscillatory function, in order to achieve an accuracy in the range $\sim 10^{-4}$ in variance, the uniform algorithm requires $\sim 10^4$ functional evaluations, as compared with $\sim 2 \times 10^3$ evaluations for the weighted case. It is also worth noticing that, as expected, the two algorithms roughly provide the same asymptotic convergence rate as predicted by the error estimate given by Equation (25). We must mention that the lesser improvement for the case of corner-peak function can be attributed to the fact that

DOI: 10.1002/nme

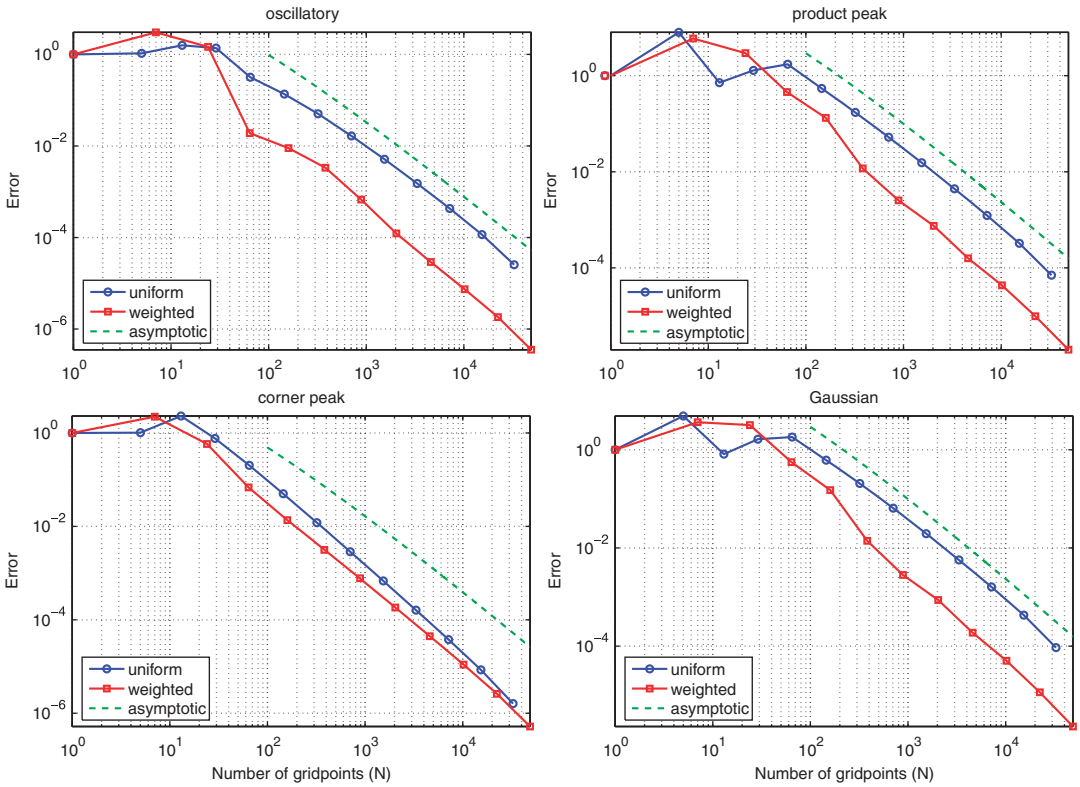

Figure 5. Normalized error in variance vs number of gridpoints N for Genz test functions [$n=2$, $y_i \sim \text{Beta}(10,1)$, $i=1,2$], using uniform and weighted Smolyak algorithms.

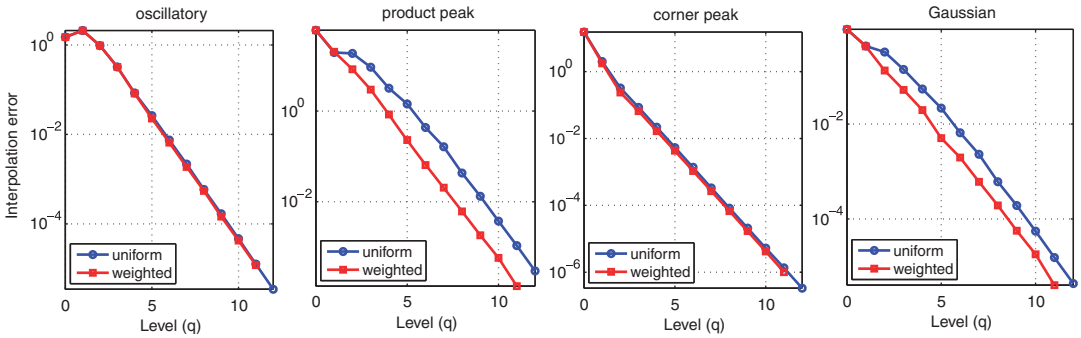

Figure 6. Interpolation error for various sparse grid levels q for Genz test functions [$n=2$, $y_i \sim \text{Beta}(10,1)$, $i=1,2$], using uniform and weighted Smolyak algorithms.

the chosen parameters **c** and **d** result in a function which peaks close to $y_1=y_2=0$. It should be noted that even for such a *bad* case, the performance of the weighted algorithm is no worse than the uniform algorithm. This example also suggests that the best performance would be obtained for adaptive refinement approaches, which incorporate information regarding the observed output form, in addition to including the a priori knowledge of the input probability measures.

In Figure 6 we plot the interpolation error for each of the test functions, obtained using uniform and weighted algorithms based on various sparse grid levels q. The interpolation error $e_{q,n}$ is computed as

$$e_{q,n} = \max_{i=1,\ldots,1000} |f(\mathbf{y}_i) - \mathcal{I}(f)(\mathbf{y}_i)| \tag{34}$$

DOI: 10.1002/nme

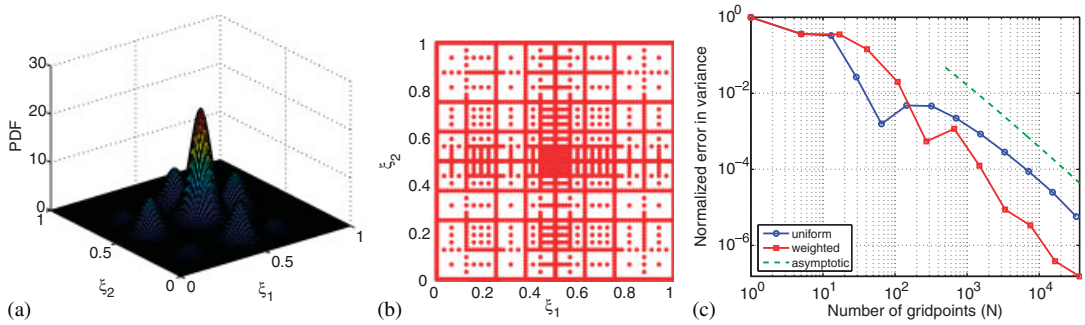

Figure 7. Computation of moments for the oscillatory function with parameters identically distributed according to non-standard mixture density $[n=2,\ y_i \sim \mathscr{M}(0.2, 0.5, 0.7),\ i=1,2]$. (a) Joint distribution of $\mathbf{y}$; (b) weighted sparse grid for level $q=8$, with gridpoints $N=3375$; and (c) normalized error in variance using uniform and weighted algorithms.

where the interpolant $\mathscr{I}(f)$ is replaced by $A_{q,n}(f)$ or $A^w_{q,n}(f)$ for uniform or weighted algorithm, respectively. The evaluation nodes $\{\mathbf{y}_i\}_{i=1}^{1000} \in [0,1]^n$ are generated randomly. As can be verified from Figure 6, for each sparse grid level q, the weighted algorithm leads to slightly lower or almost the same interpolation error as compared with the uniform algorithm. This observation is a consequence of the fact that for each level q, the sparse grids generated using the uniform algorithm are subsets of the corresponding weighted sparse grids, as mentioned in Section 4.2 (Remark 4). However, we must also note that for each level q, the weighted sparse grid $\mathscr{H}^w_{q,n}$ contains higher number of support nodes as compared with the uniform sparse grid $\mathscr{H}_{q,n}$. In this light, this observation simply signifies that while the weighted algorithm augments the set of support nodes prescribed by the uniform algorithm in order to provide significant reduction in the error in moments, such an augmented set still leads to either the same or slightly lesser interpolation error. This fact would set our algorithm apart from some other possible constructions, which may aggressively attempt to reduce only the error in moments, while not considering the approximation error. Thus, the proposed approach results in a more efficient algorithm for the computation of moments of the stochastic solution with respect to non-uniform probability measures, while maintaining the accuracy of the approximation of the solution as well. We discuss one such possible construction based on the Rosenblatt transformation [29] in Appendix A, and compare its performance with the proposed algorithm.

We also wish to highlight that the proposed weighted algorithm is easily applicable to non-standard distributions as well. To this end, we consider the mixture density given as,

$$\mathscr{M}(a,b,c) = \frac{1}{M}(\text{Gaussian}(200,a) + 3\text{Gaussian}(200,b) + \text{Gaussian}(200,c)), \tag{35}$$

where the normalization constant M is defined to ensure that the mixture density integrates to unity. Such mixture densities may be associated with various physical parameters in numerous scientific and engineering applications (e.g. distribution of Young's modulus of polysilicon induced as a result of randomness in manufacturing process parameters [39]). We consider a two-dimensional example, where y_1 and y_2 are assumed to be identically distributed according to $\mathscr{M}(0.2, 0.5, 0.7)$, and compute the moments for the oscillatory function as given in Table II. In Figures 7(a)–(c) we plot the joint distribution, weighted sparse grid obtained for $q=8$ and the normalized error in variance using uniform and weighted algorithms. As before, the weighted algorithm offers significant reduction in the number of functional evaluations required to achieve a given level of accuracy. It should be noted that the proposed algorithm only requires that the joint PDF admits a tensor structure $(\rho(\boldsymbol{\xi}) = \prod_{i=1}^n \rho_i(\xi_i))$, where the marginal distributions $\rho_i(\xi_i)$ can be different from each other. This feature is demonstrated by applying the weighted algorithm for the computation of moments for a two-dimensional oscillatory function, where we assume that y_1 and y_2 are distributed as Unif(0, 1) and $\mathscr{M}(0.2, 0.5, 0.7)$, respectively. The corresponding joint PDF, weighted

DOI: 10.1002/nme

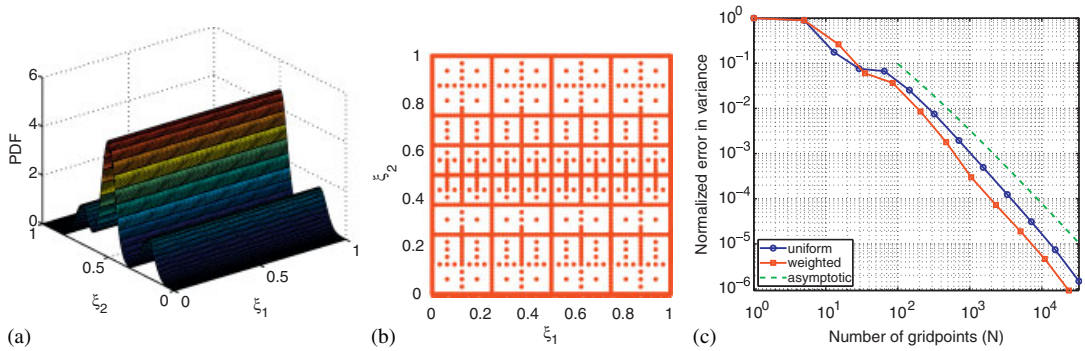

Figure 8. Computation of moments for the oscillatory function with independent but non-identically distributed parameters $[n=2,\ y_1 \sim \text{Unif}(0,1), y_2 \sim \mathcal{M}(0.2, 0.5, 0.7)]$. (a) Joint distribution of $\mathbf{y}$; (b) weighted sparse grid for level $q=8$, with gridpoints $N=2310$; (c) normalized error in variance using uniform and weighted algorithms.

sparse grid for level $q=8$, and the normalized error in variance are shown in Figures 8(a)–(c), respectively.

5.1.2. Moderate-dimensional random inputs. We also demonstrate the efficiency of the proposed algorithm for moderately high number of random dimensions ($n=4, 6$), for the case of i.i.d distributed random variables y_i, distributed according to Gaussian$(200, 0.5)$ distribution. In Figures 9 and 10 we plot the normalized error in variance for all test functions for dimensions $n=4$ and 6, respectively, using uniform and weighted algorithms. As can be seen, even for moderately high number of random dimensions, the weighted algorithm provides significant reduction in the number of functional evaluations required to obtain certain level of accuracy, for all the families of test functions considered. As compared with the low-dimensional problem considered earlier, we do not observe any change in the relative improvement offered by the weighted algorithm over the uniform approach for higher-dimensional problems.

5.2. *Kraichnan–Orszag (K-O) problem*

In order to demonstrate the efficiency of the proposed weighted algorithm for the solution of differential equations with random inputs, we now consider the three-mode Kraichnan–Orszag (K-O) problem, which has also been studied in [28, 40]. The transformed K-O problem is represented by a coupled non-linear ODE system given as [40]:

$$\frac{dy_1}{dt} = y_1 y_3, \tag{36}$$

$$\frac{dy_2}{dt} = -y_2 y_3, \tag{37}$$

$$\frac{dy_3}{dt} = -y_1^2 + y_2^2, \tag{38}$$

subjected to random initial conditions,

$$y_1(0)=\xi_1, \quad y_2(0)=\xi_2, \quad y_3(0)=\xi_3, \tag{39}$$

where $\xi_i, i=1,2,3$ are random variables in $[0, 1]$. The time integration of Equations (36)–(37) is performed using a fourth-order Runge–Kutta scheme. We are interested in computing the moments (mean and variance) of the K-O modes (y_1, y_2, y_3) for the case where $\{\xi_i\}$ are identically distributed on $[0, 1]$ according to some non-uniform distribution. As shown in Figures 11(a)–(b) the deterministic solution of the K-O problem is periodic, and the period goes to infinity if the initial conditions are located on the lines $y_1(0)=0$ or $y_2(0)=0$. From the phase plot (Figure 11(b)) we note that

DOI: 10.1002/nme

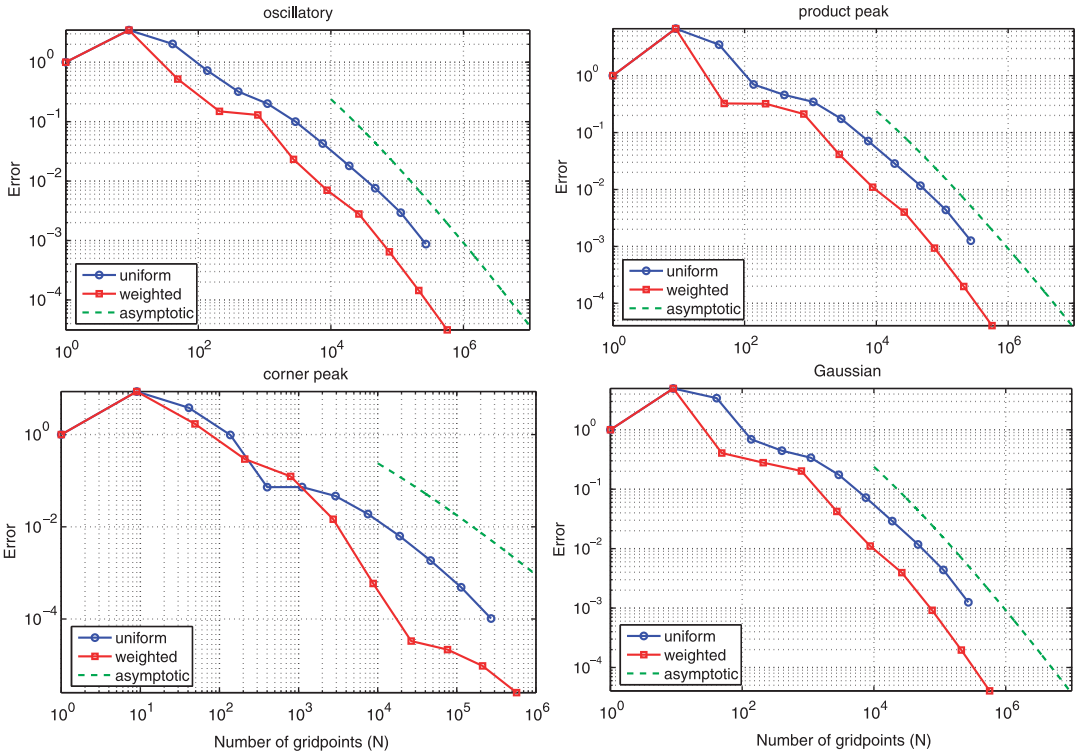

Figure 9. Normalized error in variance vs number of gridpoints N for Genz test functions $[n=4, y_i \sim \text{Gaussian}(200, 0.5),\ i=1,\ldots,4]$, using uniform and weighted Smolyak algorithms.

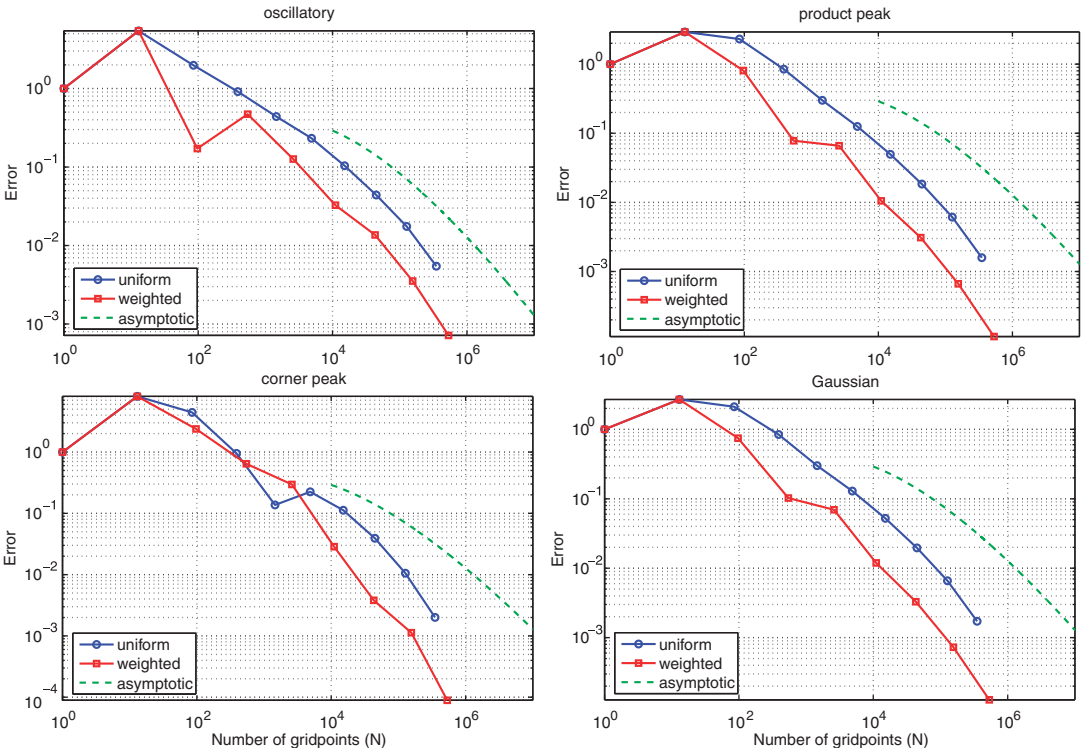

Figure 10. Normalized error in variance vs number of gridpoints N for Genz test functions $[n=6, y_i \sim \text{Gaussian}(200, 0.5),\ i=1,\ldots,6]$, using uniform and weighted Smolyak algorithms.

DOI: 10.1002/nme

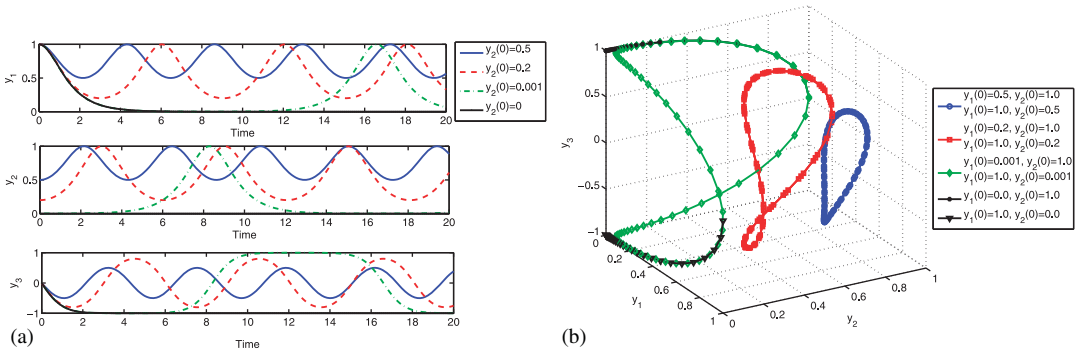

Figure 11. Solution of the Kraichnan–Orszag (K-O) problem with initial conditions $y_3(0)=0.0$ and various values of $y_1(0)$ and $y_2(0)$. (a) Evolution of modes (y_1, y_2, y_3) with time for fixed $y_1(0)=1.0$ and various values of $y_2(0)$ and (b) $y_1-y_2-y_3$ phase plot.

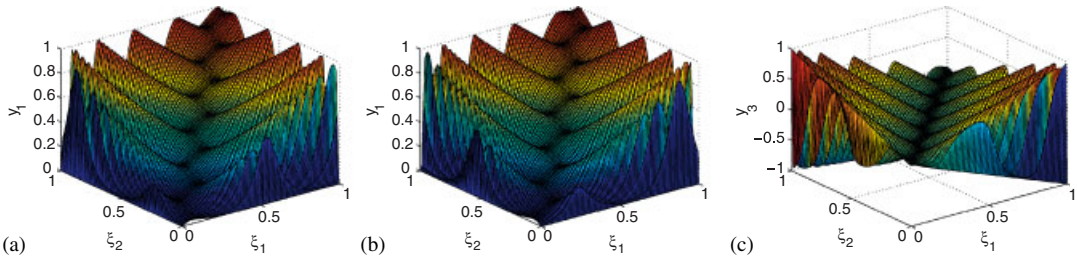

Figure 12. K-O modes (y_1, y_2, y_3) as functions of random parameters (ξ_1, ξ_2) at time $t=20$ for the 2D KO problem. (a) Mode y_1; (b) mode y_2; and (c) mode y_3.

the solution evolves into a limit cycle for initial conditions $y_{1,2}(0)>0$, and is a fixed point when $y_1(0)=0$ or $y_2(0)=0$.

We first consider a two-dimensional case (2D KO problem), where we fix $y_3(0)=0.0$ and assume ξ_1 and ξ_2 to be identically distributed according to the Gaussian$(100, 0.5)$ distribution. In Figure 12 we plot the K-O modes as a function of the random parameters (ξ_1, ξ_2) at time $t=20$, which reveals the oscillatory nature of the solution in the random domain. The error in mean and variance for the K-O modes computed using uniform and weighted algorithms is shown in Figure 13. In Table III we tabulate the maximum error in mean and variance (over all modes) for various levels of the uniform and weighted sparse grids. It must be noted that since for the purpose of comparison, we are interested in the total number of support nodes, the grid levels for the two algorithms are selected such that we obtain roughly the same number of nodes across each row. As can be seen from Figure 13 and Table III, the weighted algorithm leads up to two orders of magnitude improvement in the accuracy of moments, as compared with the uniform algorithm. Moreover, in order to achieve an accuracy in the range $\sim 10^{-5}$ for variance, the uniform algorithm requires 15 361 nodes as compared with only 3937 nodes required for the weighted algorithm.

Finally, we consider the three-dimensional KO problem, where we assume the initial conditions for all the modes to be uncertain, such that $\{\xi_i\}_{i=1}^3$ are identically distributed according to Gaussian$(100, 0.5)$ distribution. For time $t=10$, we plot the maximum error in mean and variance (over all modes) in Figure 14, and the error values for various sparse grid levels are shown in Table IV. For this case, we obtain up to an order of magnitude reduction in the error using the weighted algorithm as compared with the uniform approach.

6. CONCLUSIONS

This work presented a stochastic collocation approach for the solution of differential equations with random inputs, defined on bounded random domains with non-uniform probability measures.

DOI: 10.1002/nme

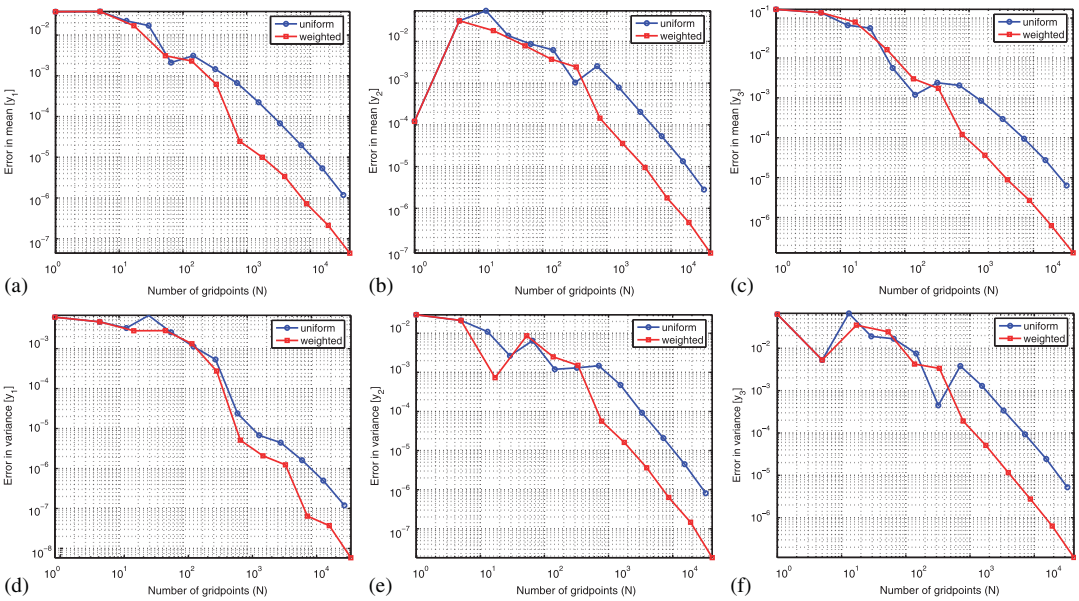

Figure 13. Error in mean (top) and variance (bottom) for the K-O modes at time $t=20$ for the 2D KO problem using uniform and weighted Smolyak algorithm. (a) Error in mean for y_1; (b) error in mean for y_2; (c) error in mean for y_3; (d) error in variance for y_1; (e) error in variance for y_2; and (f) error in variance for y_3.

Table III. Error in mean and variance (maximum over all modes) at time $t=20$ for the 2D KO problem using uniform and weighted Smolyak algorithms.

Uniform algorithm			Weighted algorithm		
Gridpoints (N)	Mean	Variance	Gridpoints (N)	Mean	Variance
29 ($q=3$)	5.59×10^{-2}	1.91×10^{-2}	53 ($q=3$)	1.61×10^{-2}	2.44×10^{-2}
705 ($q=7$)	2.55×10^{-3}	3.82×10^{-3}	781 ($q=6$)	1.44×10^{-4}	1.91×10^{-4}
3329 ($q=9$)	2.94×10^{-4}	3.39×10^{-4}	3937 ($q=8$)	9.51×10^{-6}	1.16×10^{-5}
15 361 ($q=11$)	2.74×10^{-5}	2.43×10^{-5}	19 021 ($q=10$)	6.26×10^{-7}	6.35×10^{-7}

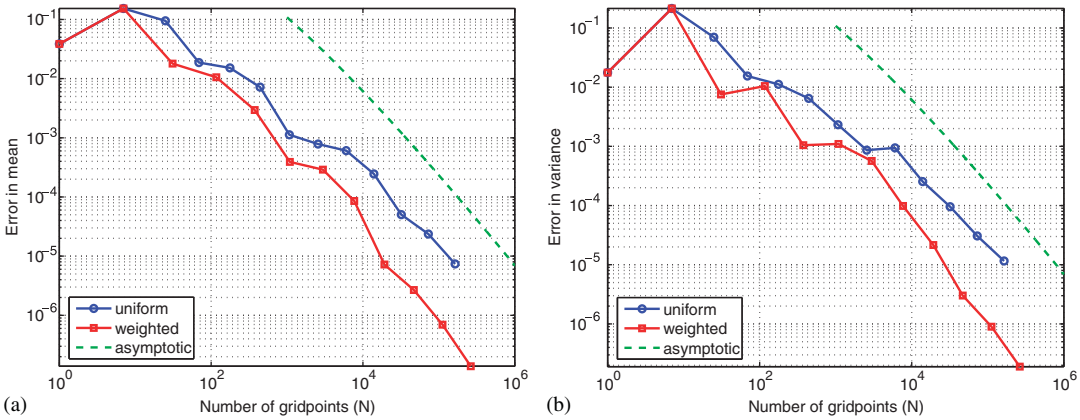

Figure 14. Error in mean and variance (maximum over all modes) at time $t=10$ for the three-dimensional KO problem using uniform and weighted Smolyak algorithm: (a) error in mean and (b) error in variance.

DOI: 10.1002/nme

Table IV. Error in mean and variance (maximum over all modes) at time $t = 10$ for the three-dimensional KO problem using uniform and weighted Smolyak algorithms.

Uniform algorithm			Weighted algorithm		
Gridpoints (N)	Mean	Variance	Gridpoints (N)	Mean	Variance
177 ($q=4$)	1.52×10^{-2}	1.11×10^{-2}	117 ($q=3$)	1.05×10^{-2}	1.04×10^{-2}
1073 ($q=6$)	1.11×10^{-3}	2.31×10^{-3}	1089 ($q=5$)	3.90×10^{-4}	1.10×10^{-3}
13 953 ($q=9$)	2.44×10^{-4}	2.54×10^{-4}	19 207 ($q=8$)	7.23×10^{-6}	2.14×10^{-5}
72 705 ($q=11$)	2.37×10^{-5}	3.07×10^{-5}	46 811 ($q=9$)	2.68×10^{-6}	3.01×10^{-6}

The collocation approach is based on a weighted Smolyak algorithm, which constructs a piecewise linear interpolation to approximate the unknown stochastic solution in the multi-dimensional random domain. The proposed algorithm incorporates information regarding the non-uniform probability measures during the node selection step, which leads to weighted sparse grids, with higher number of support nodes in regions of the random domain with higher probability density. The resulting sparse grids are nested, which facilitates reuse of functional evaluations as one seeks to increase the accuracy of approximation by increasing the depth of interpolation. Such a construction results in an efficient algorithm, for the purpose of computation of moments of the stochastic solution, while maintaining the accuracy of the approximation of the solution as well.

We presented several numerical examples to demonstrate the efficiency of the proposed approach. Specifically, we considered the problem of approximating weighted integrals, while computing moments of certain families of test functions, as well as the solution of differential equations with random inputs. It was demonstrated that for the case of highly skewed or localized probability measures, the proposed approach leads to significant improvement in the accuracy of moments (mean and variance), as compared with the uniform algorithm. Alternatively, the weighted algorithm offers significant reduction in the number of functional evaluations required to achieve a given level of accuracy in moments. We also showed that both the uniform and weighted algorithms offer the same asymptotic convergence rate, as predicted by the available theoretical error estimates.

APPENDIX A: COMPARISON WITH ROSENBLATT TRANSFORMATION

The Rosenblatt transformation [29] has been traditionally used to map a k-variate random vector with a continuous distribution to one with a uniform distribution on the k-dimensional hypercube. For the purpose of solving stochastic PDEs based on non-uniform probability measures, such a transformation can be used as a pre-processing step, where the uniformly sampled points can first be transformed using the inverse cumulative distribution function, following which the Smolyak interpolation algorithm can proceed in the usual manner. We demonstrate that for the class of methods which seek to construct an approximation for the stochastic solution using interpolation (as described in this work), such a transformation would not lead to good performance for arbitrary probability measures.

The Smolyak algorithm preserves the quality of the univariate approximation (both for interpolation or integration) while constructing the corresponding multivariate approximation. As mentioned in Section 4.3 for the algorithm described in this work, one can still expect to obtain the same convergence rate as that offered by the uniform Smolyak algorithm based on equidistant nodes. This implies that although the proposed algorithm modifies the node distribution to take the non-uniform probability measures into account, it still maintains the *almost optimal* convergence rate as offered by the uniform algorithm. However, for the approximation constructed based on the Rosenblatt transformation, one cannot guarantee the convergence rate for arbitrary distributions. Intuitively, for highly skewed distributions, the set of support nodes obtained by using the Rosenblatt transformation would be highly skewed as well, and would certainly result in higher interpolation error as

DOI: 10.1002/nme

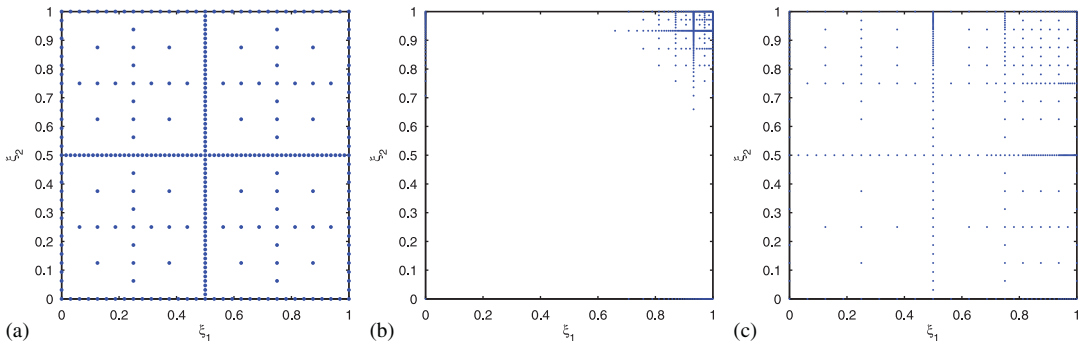

Figure A1. Sparse grids obtained using Smolyak algorithm based on uniform, Rosenblatt and weighted univariate formulas for $n=2$. (a) Uniform $q=6$, $N=321$; (b) Rosenblatt $q=6$, $N=321$; and (c) weighted $q=5$, $N=382$.

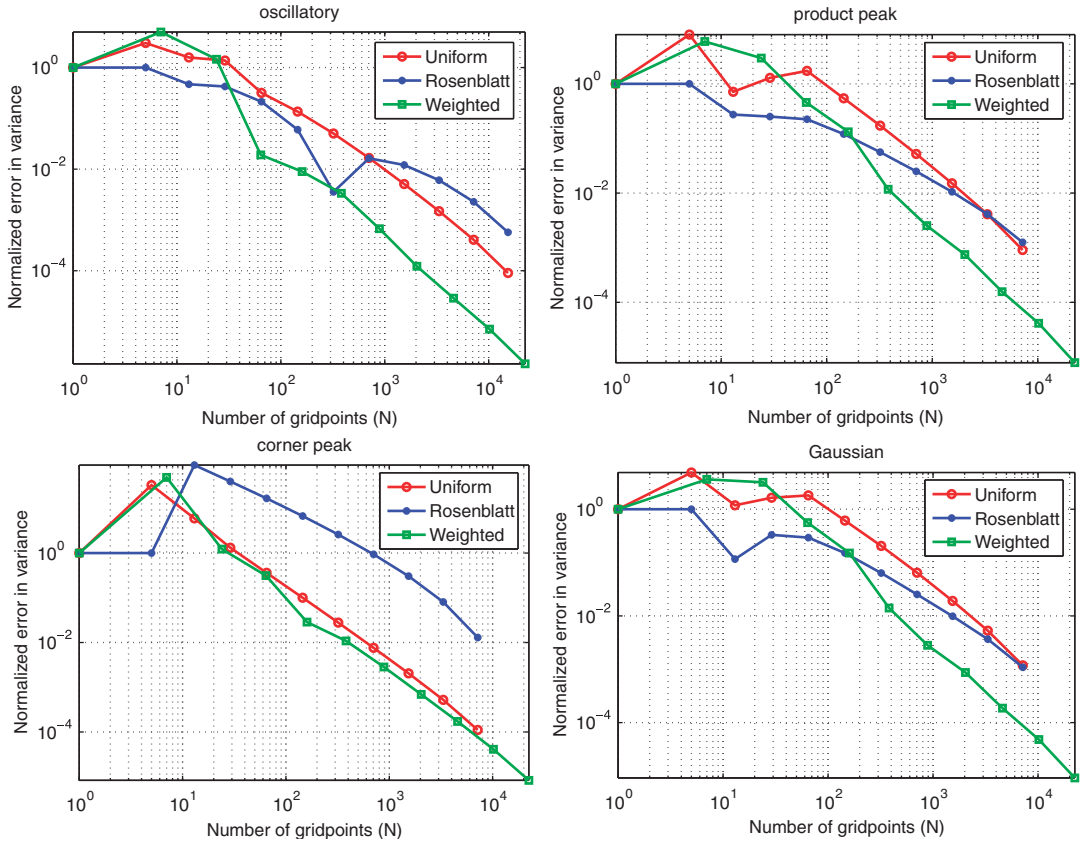

Figure A2. Normalized error in variance vs number of gridpoints N for Genz test functions [$n=2$, $y_i \sim \text{Beta}(10,1)$, $i=1,2$], using uniform, Rosenblatt and weighted Smolyak algorithms.

compared with uniformly sampled nodes. The higher interpolation error may then lead to higher error in the computation of moments as well.

We compare the performance of approximation algorithm based on the Rosenblatt transformation with the uniform and proposed weighted Smolyak algorithms, for the integration problem considered in Section 5.1. In Figure A1 we show the sparse grids obtained using the three approaches for $n=2$ and distribution Beta(10, 1). For the purpose of comparison, we have chosen the level of interpolation q such that each sparse grid roughly has the same number of nodes. As can be

DOI: 10.1002/nme

clearly seen, the sparse nodes generated following the Rosenblatt transformation are highly skewed and would result in a poor approximation. On the other hand, the sparse grid generated using the proposed algorithm (Figure A1(c)) contains more nodes in the regions of high density, while retaining the support nodes generated using the uniform algorithm.

In Figure A2 we plot the normalized error in variance using all three approaches, for the family of test functions discussed in Section 5.1. As can be clearly seen, for all of the test functions, both the uniform and the proposed weighted algorithm lead to better performance as compared with the Rosenblatt transformation.

ACKNOWLEDGEMENTS

This work is supported by the National Science Foundation under grant number 0941497 and by DOE.

REFERENCES

1. Xiu D. Fast numerical methods for stochastic computations: a review. *Communications in Computational Physics* 2009; **5**(2–4):242–272.
2. McKay MD, Beckman RJ, Conover WJ. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 1979; **2**:239–245.
3. Niederreiter H. *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM: Philadelphia, 1992.
4. Gilks W, Richardson S, Spiergelhalter D. *Markov Chain Monte Carlo in Practice*. Chapman & Hall: London, 1995.
5. Ghanem RG, Spanos P. *Stochastic Finite Elements*: *A Spectral Approach*. Springer: Berlin, 1991.
6. Wiener N. The homogeneous chaos. *American Journal of Mathematics* 1938; **60**:897–936.
7. Xiu D, Karniadakis GE. The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing* 2002; **24**(2):619–644.
8. LeMâitre OP, Ghanem RG, Knio OM, Najm HN. Uncertainty propagation using Wiener–Haar expansions. *Journal of Computational Physics* 2004; **197**:28–57.
9. LeMâitre OP, Najm HN, Ghanem RG, Knio OM. Multi-resolution analysis of Wiener-type uncertainty propagation schemes. *Journal of Computational Physics* 2004; **197**:502–531.
10. Deb M, Babuska I, Oden J. Solution of stochastic partial differential equations using Galerkin finite element techniques. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:6359–6372.
11. Babuska I, Tempone R, Zourais G. Galerkin finite element approximations of stochastic elliptic differential equations. *SIAM Journal on Numerical Analysis* 2004; **42**:800–825.
12. Xiu D, Hesthaven JS. High-order collocation methods for differential equations with random inputs. *SIAM Journal on Scientific Computing* 2005; **27**(3):1118–1139.
13. Babuska I, Nobile F, Tempone R. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis* 2007; **45**(3):1005–1034.
14. Ganapathysubramanian B, Zabaras N. Sparse grid collocation schemes for stochastic natural convection problems. *Journal of Computational Physics* 2007; **225**:652–685.
15. Smolyak S. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Soviet Mathematics—Doklady* 1963; **4**:240–243.
16. Barthelmann V, Novak E, Ritter K. High dimensional polynomial interpolation on sparse grid. *Advances in Computational Mathematics* 1999; **12**:273–288.
17. Novak E, Ritter K. High dimensional integration of smooth functions over cubes. *Numerische Mathematik* 1996; **75**:79–97.
18. Novak E, Ritter K. Simple cubature formulas with high polynomial exactness. *Constructive Approximation* 1999; **15**:499–522.
19. Klimke A, Wholmuth B. Algorithm 847: spinterp: Piecewise multilinear hierarchical sparse grid interpolation in MATLAB. *ACM Transactions on Mathematical Software* 2005; **31**(4):561–579.
20. Bungartz H, Griebel M. Sparse grids. *Acta Numerica* 2004; **13**:1–123.
21. Xiu D. Efficient collocational approach for parametric uncertainty analysis. *Communications in Computational Physics* 2007; **2**(2):293–309.
22. Nobile F, Tempone R, Webster C. A sparse grid collocation method for elliptic partial differential equations with random inputs. *SIAM Journal on Numerical Analysis* 2008; **45**:2309–2345.
23. Agarwal N, Aluru NR. Stochastic analysis of electrostatic MEMS subjected to parameter variations. *Journal of Microelectromechanical Systems* 2009; **18**(6):1454–1468.
24. Eldred MS, Burkardt J. Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification. *AIAA Paper 2009-0976*, 2009; 1–20.
25. Golub GH, Welsch JH. Calculation of gauss quadrature rules. *Mathematics of Computation* 1969; **23**(106):221–230.
26. Foo J, Wan X, Karniadakis GE. The multi-element probabilistic collocation method (ME-PCM): error analysis and applications. *Journal of Computational Physics* 2008; **227**(22):9572–9595.

DOI: 10.1002/nme

27. Ma X, Zabaras N. An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *Journal of Computational Physics* 2009; **228**(8):3084–3113.
28. Agarwal N, Aluru NR. A domain adaptive stochastic collocation approach for analysis of MEMS under uncertainties. *Journal of Computational Physics* 2009; **228**(20):7662–7688.
29. Rosenblatt M. Remarks on a multivariate transformation. *Annals of Mathematical Statistics* 1952; **23**(3):470–472.
30. Hickernell FJ, Sloan IH, Wasilkowski GW. A piecewise constant algorithm for weighted L_1 approximation over bounded or unbounded regions in $\mathbb{R}^d$. *SIAM Journal on Numerical Analysis* 2005; **43**(3):1003–1020.
31. Gajda P. Smolyak's algorithm for weighted L_1-approximation of multivariate functions with bounded rth mixed derivatives over $\mathbb{R}^d$. *Numerical Algorithms* 2005; **40**:401–414.
32. Loève M. *Probability Theory*. Springer: Berlin, 1977.
33. Babuska I, Liu K, Tempone R. Solving stochastic partial differential equations based on the experimental data. *Mathematical Models and Methods in Applied Sciences* 2003; **13**(3):415–444.
34. Ganapathysubramanian B, Zabaras N. A seamless approach towards stochastic modeling: sparse grid collocation and data-driven input models. *Finite Elements in Analysis and Design* 2008; **44**:298–320.
35. Ghanem R, Doostan A. On the construction and analysis of stochastic models: characterization and propagation of the errors associated with limited data. *Journal of Computational Physics* 2006; **217**:63–81.
36. Desceliers C, Ghanem R, Soize C. Maximum likelihood estimation of stochastic chaos representations from experimental data. *International Journal for Numerical Methods in Engineering* 2006; **66**:978–1001.
37. Das S, Ghanem R, Spall JC. Asymptotic sampling distributions for polynomial chaos representations from data: a maximum entropy and fisher information approach. *SIAM Journal on Scientific Computing* 2008; **30**(5):2207–2234.
38. Genz AC. A package for testing multiple integration subroutines. In *Numerical Integration*, Keast P, Fairweather G (eds). Kluwer: Dordrecht, The Netherlands, 1987; 337–340.
39. Agarwal N, Aluru NR. A data-driven stochastic collocation approach for uncertainty quantification in MEMS. *International Journal for Numerical Methods in Engineering* 2010; **83**(5):575–597.
40. Wan X, Karniadakis GE. An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *Journal of Computational Physics* 2005; **209**:617–642.

DOI: 10.1002/nme