Journal of Computational Physics 228 (2009) 7662–7688

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A domain adaptive stochastic collocation approach for analysis of MEMS under uncertainties

Nitin Agarwal, 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, United States

ARTICLE INFO

Article history:
Received 3 July 2008
Received in revised form 29 April 2009
Accepted 14 July 2009
Available online 26 July 2009

Keywords:
Multiphysics
Stochastic collocation
Stochastic Galerkin method
Sparse grids
Adaptive sampling
Reliability
Geometrical uncertainty
Uncertainty propagation

ABSTRACT

This work proposes a domain adaptive stochastic collocation approach for uncertainty quantification, suitable for effective handling of discontinuities or sharp variations in the random domain. The basic idea of the proposed methodology is to adaptively decompose the random domain into subdomains. Within each subdomain, a sparse grid interpolant is constructed using the classical Smolyak construction [S. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, Soviet Math. Dokl. 4 (1963) 240–243], to approximate the stochastic solution locally. The adaptive strategy is governed by the hierarchical surpluses, which are computed as part of the interpolation procedure. These hierarchical surpluses then serve as an error indicator for each subdomain, and lead to subdivision whenever it becomes greater than a threshold value. The hierarchical surpluses also provide information about the more important dimensions, and accordingly the random elements can be split along those dimensions. The proposed adaptive approach is employed to quantify the effect of uncertainty in input parameters on the performance of micro-electromechanical systems (MEMS). Specifically, we study the effect of uncertain material properties and geometrical parameters on the pull-in behavior and actuation properties of a MEMS switch. Using the adaptive approach, we resolve the pull-in instability in MEMS switches. The results from the proposed approach are verified using Monte Carlo simulations and it is demonstrated that it computes the required statistics effectively.

1. Introduction

Micro-electromechanical systems (MEMS) have been used in widespread applications such as micro-switches, gyroscopes, accelerometers, etc. In order to design and analyze such devices it is required to accurately model the interaction of various physical fields such as mechanical, electrical and fluidic. In recent years, advances in numerical simulation methods have increased the ability to accurately model these devices [2–5]. These simulation methods assume that the material properties and various geometrical parameters of the device are known in a deterministic sense. However, low cost manufacturing processes used for MEMS often result in significant uncertainties in these parameters which may lead to large variation in the device performance. Thus, in order to design reliable and efficient electrostatic MEMS devices, it is required to quantify the effect of uncertain input parameters on various relevant performance parameters.

* Corresponding author.
E-mail address: aluru@uiuc.edu (N.R. Aluru).
URL: http://www.uiuc.edu/~aluru (N.R. Aluru).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.07.014

Uncertainties can be described using stochastic quantities – uncertain parameters can be modeled using random variables and uncertain spatial functions are represented as random fields. Using this, the original governing equations can be reformulated as stochastic partial differential equations (SPDEs). Traditionally, sampling based methods such as Monte Carlo (MC) method has been used for systems with random input parameters. It involves 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 (LHS) [6], the quasi-Monte Carlo (QMC) method [7] and the Markov Chain Monte Carlo (MCMC) method [8], etc.

An important non-sampling approach is based on stochastic Galerkin method, where the basic idea is to project the unknown stochastic solution onto a stochastic space spanned by complete orthogonal polynomials. The stochastic Galerkin method was initially developed by Ghanem and Spanos [9] using Wiener–Hermite polynomial chaos expansion [10], where the orthogonal polynomials are chosen as global hermite polynomials in terms of Gaussian random variables. This idea was further generalized by Xiu and Karniadakis [11], to obtain exponentially converging algorithms even for non-Gaussian random variables. We developed a stochastic Lagrangian framework based on generalized polynomial chaos (GPC) in [12], to handle the uncertain electromechanical interaction. It was demonstrated that the stochastic framework can be effectively used to quantify the effect of uncertain input parameters on the performance of MEMS devices, as long as the solution is smooth in the random domain.

The stochastic Galerkin method 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. Also, 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 complicated form, such as nonlinear terms and coupled multiphysics.

In recent years, another class of methods known as stochastic collocation method [13–15] has been explored. The stochastic collocation method provides high resolution as stochastic Galerkin method, as well as easy implementation as the sampling based methods. This approach is based on approximating the unknown stochastic solution by constructing sparse grid interpolants in the multi-dimensional random domain, based on the Smolyak algorithm [1]. Using this algorithm, interpolation schemes can be constructed with orders of magnitude reduction in the number of support nodes to give the same level of approximation (up to a logarithmic factor) as the usual tensor product.

The stochastic Galerkin and collocation approaches provide fast converging approximations as compared to the sampling based methods, assuming that the unknown stochastic solution is sufficiently smooth in the random domain. However, in many physical systems, small variations in the uncertain parameters may lead to jumps in the solution. For example, in MEMS actuators, because of the nonlinear nature of the electrostatic actuation force, small variation in material properties and geometrical parameters may lead to a well known phenomenon known as *pull-in*. This pull-in instability is manifested as a discontinuity in the switch displacement in the random domain. In order to accurately compute the statistics of the stochastic solution in such situations, it is important to correctly capture these discontinuities in the random domain. To this end, several efforts have been made using the Galerkin approach, such as the wavelet based Weiner–Haar basis functions [16,17] and the multi-element GPC (ME-GPC) method [18,19]. The basic idea of ME-GPC is to adaptively decompose the random domain into a set of random elements, and then to employ a GPC expansion within each element to locally approximate the stochastic solution. This leads to a set of coupled deterministic equations that need to be solved within each random element. An adaptive sparse grid collocation methodology was presented in [15], based on the dimensional adaptive quadrature algorithm given in [20], to study the equilibrium jumps encountered during the stochastic modeling of natural convection problems. This approach automatically detects the more important dimensions and the sparse sampling is appropriately biased in those dimensions.

This work proposes a *domain adaptive stochastic collocation approach* to effectively handle discontinuities or sharp variations in the random domain. The basic idea of the proposed methodology is to adaptively decompose the random domain into subdomains. Within each subdomain, we then construct the sparse grid interpolant using the classical Smolyak construction in a hierarchical fashion, to approximate the stochastic solution locally. The adaptive strategy is governed by the *hierarchical surpluses*, which are computed as part of the interpolation procedure. These hierarchical surpluses then serve as an error indicator for each subdomain, and lead to subdivision whenever it becomes greater than a threshold value. The hierarchical surpluses also provide information about the more important dimensions, and accordingly the random elements can be split along those dimensions.

During the preparation of this manuscript, the authors came across two recent methods which also deal with problems with limited regularity in the stochastic domain. The first approach, multi-element probabilistic collocation method (ME-PCM) proposed by Foo et al. [21], discretizes the parametric space, and prescribes a collocation/cubature grid on each element. Although, both the ME-PCM method and our approach, adaptively decompose the parametric space into elements, the construction of the interpolant within each random element, and, more importantly, the computation of local error indicators, which ultimately leads to adaptive refinement, are significantly different. For the benefit of readers, we summarize the key differences between the two approaches as follows:

- The ME-PCM method constructs sparse/tensor interpolants within each random element using *nodal* interpolation. However, in our method, we employ hierarchical sparse grid interpolation within each random element. The construction of the hierarchical interpolation is different than the usual procedure using nodal basis functions. This hierarchical construction is central to our adaptive refinement procedure and has been explained in detail in Section 4.
- Once the interpolant is constructed within each random element, the ME-PCM method then projects this interpolant onto the GPC bases, in order to obtain the coefficients, using which the adaptivity criterion is evaluated exactly as described by the ME-GPC framework [19]. On the other hand, in our method, the construction of the sparse interpolant within each element, requires us to compute the so called hierarchical surpluses, which is the difference between the actual function value and the value obtained using the interpolant from the previous level. As described in Section 5, these hierarchical surpluses automatically yield an estimate for the interpolation error within each element, and can be appropriately used to decide whether or not to refine a particular element and also along which random dimensions. Thus, the ways in which the two approaches compute the adaptivity criterion are entirely different.

The second approach, introduced by Ma and Zabaras [22], relies on resolving the discontinuities using piecewise-multilinear basis functions, such that, as the adaptive refinement proceeds, functions with smaller support are inserted in parts of the domain with limited regularity, as determined using the hierarchical surpluses. This method does not involve any explicit decomposition of the random domain. On the other hand, in our approach, the discontinuities are resolved by adaptively decomposing the parametric space into elements. As the adaptive refinement proceeds, only elements which do not meet the prescribed error tolerance are subdivided, to locally approximate the solution with greater accuracy. Also, in our approach, the interpolation within each random element can be constructed using piecewise-multilinear or Lagrange polynomials based on suitable 1D rule, as their support is controlled by the size of the element.

We employ the proposed adaptive approach to study the effect of uncertain material properties and geometrical parameters on the pull-in behavior and actuation properties of a MEMS switch. Using the adaptive approach, we resolve the *pull-in instability* in MEMS switches, which was not possible using the framework developed earlier in [12]. The results from the proposed approach are verified using Monte Carlo simulations and it is demonstrated that it computes the required statistics effectively. The rest of the paper is organized as follows: In Section 2 we present the deterministic and stochastic formulation for the coupled electromechanical problem, applicable to static analysis of electrostatic MEMS. In Section 3 we briefly present the Monte Carlo and the stochastic Galerkin framework for the stochastic electromechanical problem. We then explain the stochastic collocation approach based on classical Smolyak construction in Section 4, using which the adaptive stochastic collocation framework is developed in Section 5. In Section 6 we consider the numerical example of a MEMS switch and study the effect of uncertain parameters on its pull-in behavior and actuation properties. We finally conclude the discussion in Section 7.

2. Problem formulation

2.1. Deterministic formulation

Physical level analysis of electrostatic MEMS requires a self-consistent solution of the coupled mechanical and electrostatic equations. A framework for the deterministic analysis is presented in [4], which uses a Lagrangian description both for the mechanical and the electrostatic domains. The mechanical deformation of the MEM structures is obtained by performing a 2D geometrically nonlinear elasticity analysis [23]. Let Ω represent the undeformed configuration with boundary $d\Omega = d\Omega_g \cup d\Omega_h$. The governing equations for the deformation of the MEM structures in the absence of body force are given as,

$$\nabla \cdot (\mathbf{FS}) = 0 \quad \text{in } \Omega, \tag{1}$$

$$\mathbf{u} = \mathbf{G} \quad \text{on } d\Omega_g, \tag{2}$$

$$\mathbf{P} \cdot \mathbf{N} = \mathbf{H} \quad \text{on } d\Omega_h, \tag{3}$$

where $\mathbf{u}$ is the displacement vector, $\mathbf{F}$ is the deformation gradient, $\mathbf{P}$ and $\mathbf{S}$ are the first and second Piola–Kirchoff stress tensors, respectively. $\mathbf{H}$ is the electrostatic pressure acting on the surface of the structures and $\mathbf{N}$ is the unit outward normal vector in the undeformed configuration. The prescribed displacement is given by $\mathbf{G}$. The constitutive law can be written as,

$$\mathbf{S} = \mathcal{C}\mathbf{E}, \quad \mathbf{E} = \frac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{I}), \tag{4}$$

where $\mathcal{C}$ is the material tensor and $\mathbf{E}$ is the Green–Lagrangian strain.

The electrostatic analysis is done using a Lagrangian boundary integral form as described in [24]. The *Lagrangian boundary integral equations* are given by:

$$\phi(P) = \int_{d\Omega} G(p(P), \quad q(Q))\sigma(q(Q))\gamma(Q)d\Gamma_Q + C, \tag{5}$$

$$C_T = \int_{d\Omega} \sigma(q(Q))\gamma(Q)d\Gamma_Q, \quad \gamma(Q) = [\mathbf{T}(Q) \cdot \mathbf{C}(Q)\mathbf{T}(Q)]^{\frac{1}{2}}, \tag{6}$$

where σ is the unknown surface charge density and C is an unknown constant that needs to be computed. P and Q refer to the positions of source and field points, respectively, in the undeformed configuration. G is the Green's function, which in 2D is given as $G(P,Q) = -\frac{1}{2\pi\epsilon}\ln|P - Q + \mathbf{u}_P - \mathbf{u}_Q|$, where ϵ is the dielectric constant of the medium and $|P - Q + \mathbf{u}_P - \mathbf{u}_Q|$ represents the distance between the source and field points in the deformed configuration. C_T represents the total charge of the system, which is set to be zero. $\mathbf{C}(Q) = \mathbf{F}^T(Q)\mathbf{F}(Q)$ and $\mathbf{T}(Q)$ is the unit tangential vector at the field point Q in the undeformed configuration. The electrostatic pressure acting on the conductors in the undeformed configuration can be computed from the surface charge density as,

$$\mathbf{H} = P_e J\mathbf{F}^{-T}\mathbf{N}, \tag{7}$$

where $P_e = \frac{\sigma^2}{2\epsilon}$ is the electrostatic pressure acting normal to the surface of the conductors and $J = det(\mathbf{F})$. Eq. (7) represents the nonlinear coupling between the mechanical and electrostatic energy domains, and a relaxation or Newton scheme can be used to obtain self-consistent solutions, as described in [4].

We can represent the coupled electromechanical system (Eqs. (1)–(3), (5) and (6)) as

$$\mathcal{L}(\mathbf{u},\sigma;\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega. \tag{8}$$

Such a system can be solved easily using finite element method (FEM) and boundary element method (BEM) [25]. State-of-the-art design methodologies for MEMS are based on such deterministic approaches where the geometrical and physical properties of the device are assumed to be known precisely. However, in practice, such devices may be subjected to severe stochastic variations in these parameters, which must be considered during modeling.

2.2. Stochastic formulation

Let $(\Theta, B, \mathcal{P})$ denote a probability space, where Θ is the set of elementary events, B is the σ-algebra of events and $\mathcal{P}$ 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. For the stochastic modeling of the coupled electromechanical system given by Eq. (8), we seek displacement $\mathbf{u}(\mathbf{x},\theta) : \overline{\Omega} \times \Theta \to \mathbb{R}$ and surface charge density $\sigma(\mathbf{x},\theta) : d\Omega \times \Theta \to \mathbb{R}$, such that for $\mathcal{P}$-almost everywhere $\theta \in \Theta$, the following holds

$$\mathcal{L}(\mathbf{u},\sigma;\mathbf{x},\theta) = 0, \quad (\mathbf{x},\theta) \in \Omega \times \Theta. \tag{9}$$

In order to solve the above problem numerically, we first need to reduce the infinite dimensional probability space into a finite dimensional space. This can be accomplished by characterizing the input random parameters in terms of a finite number of random variables. For example, the input random processes can be decomposed into a set of uncorrelated random variables using the Karhunen–Loève (KL) expansion [26]. Representing the set of random variables as $\boldsymbol{\xi} = \{\xi_i(\theta)\}_{i=1}^n$, we can write $\mathcal{L}(\mathbf{u},\sigma;\mathbf{x},\theta) = \mathcal{L}(\mathbf{u},\sigma;\mathbf{x},\boldsymbol{\xi})$. Further, we assume that the random variables $\boldsymbol{\xi}$ 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 fact, the problem of representing non-Gaussian random processes in terms of independent random variables is an area of active research [27].

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

$$\rho(\boldsymbol{\xi}) = \prod_{i=1}^{n} \rho_i(\xi_i) \quad \forall \boldsymbol{\xi} \in \Gamma, \tag{10}$$

where $\Gamma = \prod_{i=1}^n \Gamma_i$ represents the support of the set of random variables. Using this, we can rewrite Eq. (9) as: we seek random displacement $\mathbf{u}(\mathbf{x},\boldsymbol{\xi})$ and surface charge density $\sigma(\mathbf{x},\boldsymbol{\xi})$, such that

$$\mathcal{L}(\mathbf{u},\sigma;\mathbf{x},\boldsymbol{\xi}) = 0, \quad (\mathbf{x},\boldsymbol{\xi}) \in \Omega \times \Gamma, \tag{11}$$

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.

3. Numerical solution of stochastic systems

In this section, we briefly present the two most widely used approaches for the numerical solution of stochastic systems (such as Eq. (11)), namely, Monte Carlo (MC) method and stochastic Galerkin method.

3.1. Monte Carlo method

Monte Carlo simulation has been traditionally used for systems with random input parameters. It involves generating various realizations of the input parameters according to the underlying probability distribution, and repeatedly employing the deterministic solver for each realization. Eq. (11) can be easily solved using MC method as follows:

1. For the given number of realizations N, generate independent and identically distributed (iid) random variables $\{\boldsymbol{\xi}^j\} = [\xi_1(\theta_j), \ldots, \xi_n(\theta_j)]$, for $j = 1, \ldots, N$.
2. For each of the realizations, solve the deterministic problem $\mathcal{L}(\mathbf{u}^j, \sigma^j; \mathbf{x}, \boldsymbol{\xi}^j) = 0$ and obtain the solution $(\mathbf{u}^j, \sigma^j)$, for $j = 1, \ldots, N$.
3. Compute the required statistics such as mean μ and variance v, for example

$$\mu(\mathbf{u}) = \frac{1}{N}\sum_{j=1}^{N} \mathbf{u}^j, \quad v(u) = \frac{1}{N}\sum_{j=1}^{N} (\mathbf{u}^j - \mu(\mathbf{u}))^2. \tag{12}$$

The amount of work required for a MC simulation to converge to a given accuracy ϵ is $\epsilon(N) = \mathcal{O}\left(N^{-\frac{1}{2}}\right)$, which is independent of the number of random variables n. The convergence rate is quite low and it can be prohibitively expensive to achieve high accuracy. The convergence can be slightly improved using quasi-Monte Carlo (QMC) method [7], where the deterministic problem is solved at structurally determined points, and not at random points as in MC method. The convergence rate for QMC method is given as $\epsilon(N) = \mathcal{O}(N^{-1}(\log N)^n)$, which is almost half an order better than the MC approach. However, the convergence rate for QMC method depends weakly on the number of random variables n, and performance may suffer for high dimensional problems. For MEMS, the Monte Carlo method incorporated in the ANSYS probabilistic design system (ANSYS/PDS) has been used to study the effect of various geometrical features on the design of a comb drive [28].

3.2. Stochastic Galerkin method

The stochastic Galerkin method is based on representing the unknown random process $w(\mathbf{x}, \theta)$ in terms of the orthogonal basis functions $\{\Psi_i(\boldsymbol{\xi}(\theta))\}$ in multi-dimensional random variables as,

$$w(\mathbf{x}, \theta) = \sum_{i=0}^{\infty} w_i(\mathbf{x})\Psi_i(\boldsymbol{\xi}(\theta)), \tag{13}$$

where the coefficients $\{w_i\}$ are deterministic and are computed using Galerkin projections in the space spanned by the basis functions $\{\Psi_i\}$. The functions $\{\Psi_i\}$ form an orthogonal basis in the probability space, with the orthogonality relation

$$\langle \Psi_i, \Psi_j \rangle = \delta_{ij}\langle \Psi_i^2 \rangle, \tag{14}$$

where δ_{ij} is the Kronecker delta and $\langle \cdot, \cdot \rangle$ denotes the ensemble average, which is the inner product given as

$$\langle \Psi_i, \Psi_j \rangle = \int_{\Theta} \Psi_i(\boldsymbol{\xi})\Psi_j(\boldsymbol{\xi}) d\mathcal{P}. \tag{15}$$

The stochastic Galerkin method was initially developed using Wiener–Hermite polynomial chaos expansion [10], where the basis functions $\{\Psi_i\}$ are chosen as global hermite polynomials in terms of Gaussian random variables. 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 [11], wavelet based Weiner–Haar basis to deal with sharp or even discontinuous variation in the random domain [16,17] and piecewise polynomial expansions [29,30].

We note that the expansion in Eq. (13) is usually truncated for numerical implementation. Using this for the stochastic coupled electromechanical system given by Eq. (11), we first represent the random displacement field $\mathbf{u}(\mathbf{x}, \boldsymbol{\xi})$ and the surface charge density $\sigma(\mathbf{x}, \boldsymbol{\xi})$ as,

$$\mathbf{u}(\mathbf{x}, \boldsymbol{\xi}) = \sum_{i=0}^{N} \mathbf{u}_i(\mathbf{x})\Psi_i(\boldsymbol{\xi}), \text{ and } \quad \sigma(\mathbf{x}, \boldsymbol{\xi}) = \sum_{i=0}^{N} \sigma_i(\mathbf{x})\Psi_i(\boldsymbol{\xi}), \tag{16}$$

where $N+1$ is the total number of terms used. Using this in Eq. (11) we get,

$$\mathcal{L}\left(\sum_{i=0}^{N} \mathbf{u}_i\Psi_i, \sum_{i=0}^{N} \sigma_i\Psi_i; \mathbf{x}, \boldsymbol{\xi}\right) = 0. \tag{17}$$

The coefficients $\{\mathbf{u}_i\}$ and $\{\sigma_i\}$ are determined using Galerkin projections onto each of the basis function $\{\Psi_j, j = 0, \ldots, N\}$,

$$\left\langle \mathcal{L}\left(\sum_{i=0}^{N} \mathbf{u}_i\Psi_i, \sum_{i=0}^{N} \sigma_i\Psi_i; \mathbf{x}, \boldsymbol{\xi}\right), \Psi_j \right\rangle = 0. \tag{18}$$

Using the orthogonality of the basis functions $\{\Psi_i\}$, this leads to $N+1$ coupled set of deterministic equations which can be used to solve for the modes $\{\mathbf{u}_i\}$ and $\{\sigma_i\}$. The details of this approach for the stochastic modeling of electrostatic MEMS can be found in [12].

The stochastic Galerkin method provides high accuracy and faster convergence rate and has been successfully applied to various problems such as computational mechanics [31,32], diffusion [33], fluid flow [34,35] and heat conduction [36,37].

However, as the number of stochastic dimensions of the problem increases, the number of basis functions $(N+1)$ that need to be considered increases rapidly, which reduces the efficiency. Also, 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 requires substantial effort to convert a deterministic code to a stochastic implementation. Moreover, when the governing equations take complicated form, such as nonlinear terms [38–40] and coupled multiphysics [12,41], the numerical implementation of the stochastic Galerkin method may not be straightforward and may require special techniques, such as those described in [42].

In recent years, another class of methods known as stochastic collocation method [13–15] has been explored. The stochastic collocation method provides high resolution as stochastic Galerkin method, as well as easy implementation as the sampling based methods, and is described next.

4. Stochastic collocation method

The stochastic collocation method is based on constructing polynomial interpolants in the multi-dimensional random space. It only requires the *deterministic* solution at a pre-determined set of points in the random domain, using which the multi-dimensional interpolant is constructed which approximates the unknown stochastic solution. The stochastic collocation method has been described in [13] and has been used in [15] for stochastic natural convection problem. In this section, we briefly present this framework as given in [13,15], as it is later used to develop the *adaptive stochastic collocation method*.

4.1. Formulation

Let Π_n denote the space of all n-variate polynomials with real coefficients and $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_n]$ be any point in the random space $\Gamma \subset \mathbb{R}^n$. The Lagrange interpolation problem can then be stated as: Given a set of pre-determined points $\mathcal{X}_n = \{\boldsymbol{\xi}^i\}_{i=1}^N$ in the random domain Γ and a smooth function $f : \mathbb{R}^n \to \mathbb{R}$, find a polynomial $\mathcal{I}f \in \Pi_n$, such that $\mathcal{I}f(\boldsymbol{\xi}^i) = f(\boldsymbol{\xi}^i), \forall i = 1, \ldots, N$. The polynomial interpolation $\mathcal{I}f$, using Lagrange interpolation functions can be given as,

$$\mathcal{I}f(\boldsymbol{\xi}) = \sum_{i=1}^{N} f(\boldsymbol{\xi}^i) L_i(\boldsymbol{\xi}), \tag{19}$$

where $L_i(\boldsymbol{\xi}^j) = \delta_{ij}, (i,j) \in [1,N]$. The interpolated value of the function at any point $\boldsymbol{\xi}$ is then simply given as $\mathcal{I}f(\boldsymbol{\xi})$. Using Eq. (19) for the stochastic electromechanical problem, we denote the interpolated displacement $\hat{\mathbf{u}}(\mathbf{x}, \boldsymbol{\xi})$ and surface charge density $\hat{\sigma}(\mathbf{x}, \boldsymbol{\xi})$ as,

$$\hat{\mathbf{u}}(\mathbf{x}, \boldsymbol{\xi}) = \sum_{i=1}^{N} \mathbf{u}(\mathbf{x}, \boldsymbol{\xi}^i) L_i(\boldsymbol{\xi}), \quad \hat{\sigma}(\mathbf{x}, \boldsymbol{\xi}) = \sum_{i=1}^{N} \sigma(\mathbf{x}, \boldsymbol{\xi}^i) L_i(\boldsymbol{\xi}). \tag{20}$$

Using this in Eq. (11), the collocation procedure gives,

$$\mathcal{L}(\hat{\mathbf{u}}, \hat{\sigma}; \mathbf{x}, \boldsymbol{\xi})|_{\boldsymbol{\xi}^k} = 0 \quad \forall\ k \in 1, \ldots, N. \tag{21}$$

Using the property of the Lagrange interpolation polynomials $L_i(\boldsymbol{\xi}^j) = \delta_{ij}$, this immediately leads to: for $k = 1, \ldots, N$,

$$\mathcal{L}(\mathbf{u}(\mathbf{x}, \boldsymbol{\xi}^k), \sigma(\mathbf{x}, \boldsymbol{\xi}^k); \mathbf{x}, \boldsymbol{\xi}^k) = 0, \quad \mathbf{x} \in \Omega. \tag{22}$$

Thus, the stochastic collocation procedure reduces to solving N deterministic systems, at each nodal point $\boldsymbol{\xi}^k, k = 1, \ldots, N$ in a given set of points $\mathcal{X}_n$. We also note that these deterministic systems are naturally decoupled and existing deterministic solver can be readily used, unlike the stochastic Galerkin procedure where one needs to solve a set of coupled deterministic equations. Once the deterministic solution is computed at all the collocation points, the statistics of the stochastic solution can be easily computed using,

$$\mathbb{E}[g(\hat{\mathbf{u}})](\mathbf{x}) = \sum_{i=1}^{N} g(\mathbf{u}(\mathbf{x}, \boldsymbol{\xi}^i)) \int_{\Gamma} L_i(\boldsymbol{\xi}) \rho(\boldsymbol{\xi}) d\boldsymbol{\xi} = \sum_{i=1}^{N} g(\mathbf{u}(\mathbf{x}, \boldsymbol{\xi}^i)) w_i, \quad w_i = \int_{\Gamma} L_i(\boldsymbol{\xi}) \rho(\boldsymbol{\xi}) d\boldsymbol{\xi}, \tag{23}$$

where $\mathbb{E}[\cdot]$ denotes the expectation operator and the weights $\{w_i\}_{i=1}^N$ can be pre-computed using appropriate quadrature scheme and stored for later use. The function $g(\cdot)$ can be chosen appropriately, by noticing that the mean $\mu(\hat{\mathbf{u}}) = \mathbb{E}[\hat{\mathbf{u}}]$ and the variance $v(\hat{\mathbf{u}}) = \mathbb{E}[(\hat{\mathbf{u}} - \mu(\hat{\mathbf{u}}))^2]$. We note that in writing Eq. (23) we have assumed that the function $g(\hat{\mathbf{u}})$ can also be approximated as,

$$g(\hat{\mathbf{u}})(\mathbf{x}, \boldsymbol{\xi}) = \sum_{i=1}^{N} g(\mathbf{u}(\mathbf{x}, \boldsymbol{\xi}^i)) L_i(\boldsymbol{\xi}), \tag{24}$$

where the coefficients $g(\mathbf{u}(\mathbf{x}, \boldsymbol{\xi}^i))$ are easily computed using the sampled values of the unknown solution at the collocation points.

The construction of a Lagrange interpolation polynomial as given by Eq. (19) is central to the stochastic collocation approach. There exists a well-developed theory of univariate Lagrange interpolation. However, to construct such an interpolation in higher dimensions is a non-trivial task, and is considered next.

4.2. Multi-variate sparse grid interpolation

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 $\mathcal{X}_n$. Thus, the key issue for the stochastic collocation procedure is the *selection* of this set of nodes $\mathcal{X}_n$, such that using the minimal number of nodes one achieves a good approximation (to the desired accuracy level) by Lagrange interpolation. One such possible choice proposed in [13,15] is based on the sparse grids generated using the Smolyak algorithm [1]. In this section, we describe the construction of these sparse grids based on linear or higher degree interpolation using the notation as given in [43]. Without loss of generality, 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 a n-hypercube.

4.2.1. Hierarchical univariate interpolation

Let $f : [0,1] \to \mathbb{R}$ be a function in 1D, which is approximated using a sequence of interpolation formulas given as,

$$\mathcal{I}^k(f)(\xi) = \sum_{j=1}^{m_k} f(\xi_j^k) l_j^k(\xi), \quad \text{for each } k \geqslant 1, \tag{25}$$

with the set of support nodes $\mathcal{X}^k = \{\xi_j^k | \ \xi_j^k \in [0,1], j = 1,\ldots,m_k\}$, and interpolation basis functions $l^k = \{l_j^k | \ l_j^k \in C[0,1], j = 1,\ldots,m_k\}$, 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. For this choice, the set of equidistant nodes $\mathcal{X}^k$ can be described using,

$$m_k = \begin{cases} 1, & \text{if } k = 1, \\ 2^{k-1} + 1, & \text{if } k > 1, \end{cases} \tag{26}$$

$$\xi_j^k = \begin{cases} 0.5, & \text{for } j = 1 \text{ if } m_k = 1, \\ \frac{j-1}{m_k - 1}, & \text{for } j = 1,\ldots,m_k \text{ if } m_k > 1. \end{cases} \tag{27}$$

Using this set of points, the piecewise linear basis functions can be defined as follows,

$$l_j^k(\xi) = 1, \quad \text{for } k = 1, \text{ and}, \tag{28}$$

$$l_j^k(\xi) = \begin{cases} 1 - (m_k - 1)|\xi - \xi_j^k|, & \text{if } |\xi - \xi_j^k| < \frac{1}{m_k - 1} \\ 0, & \text{otherwise}, \end{cases} \tag{29}$$

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

In the case of smooth objective functions, Lagrange polynomial interpolation offers faster error decay with increasing number of nodes as compared to the piecewise linear interpolation. The approximation quality of the Lagrange interpolant depends on the node distribution. Several node distributions which are known to perform better include Gauss quadrature points, and the extrema of the Chebyshev polynomials (also known as Chebyshev Gauss–Lobatto nodes (CGL)). The set of CGL nodes $\mathcal{X}^k = \{\xi_j^k\}_{j=1}^{m_k}$ are given as,

$$m_k = \begin{cases} 1, & \text{if } k = 1, \\ 2^{k-1} + 1, & \text{if } k > 1, \end{cases} \tag{30}$$

$$\xi_j^k = \begin{cases} 0.5, & \text{for } j = 1 \text{ if } m_k = 1, \\ \left(1 - \cos\left(\frac{(j-1)\pi}{m_k - 1}\right)\right)/2, & \text{for } j = 1,\ldots,m_k \text{ if } m_k > 1. \end{cases} \tag{31}$$

The Lagrange polynomial basis functions are given as,

$$l_j^k(\xi) = \begin{cases} 1, & \text{for } k = 1, \text{ and} \\ \prod\limits_{i=1, i\neq j}^{m_k} \frac{\xi - \xi_i^k}{\xi_j^k - \xi_i^k} & \text{for } k > 1 \text{ and } j = 1,\ldots,m_k. \end{cases} \tag{32}$$

We must note that both the set of nodes, namely equidistant and CGL, are nested and satisfy $\mathcal{X}^k \subset \mathcal{X}^{k+1}, k \geqslant 1$. Using this property, we can write the nodal interpolation formula given by Eq. (25), in a hierarchical fashion. We first define $\mathcal{I}^0 = 0$, and the incremental interpolant $\Delta^k = \mathcal{I}^k - \mathcal{I}^{k-1}, \ \forall\, k \geqslant 1$ [43–45]. Thus,

$$\Delta^k(f) = \mathcal{I}^k(f) - \mathcal{I}^{k-1}(f), \tag{33}$$

where, $\mathcal{I}^k(f) = \sum_{\xi_j^k \in \mathcal{X}^k} f(\xi_j^k) l_j^k$ and $\mathcal{I}^{k-1}(f) = \mathcal{I}^k(\mathcal{I}^{k-1}(f))$. Using this, we obtain (see [43] for details),

$$\Delta^k(f) = \sum_{\xi_j^k \in \mathcal{X}^k} f(\xi_j^k) l_j^k - \sum_{\xi_j^k \in \mathcal{X}^k} \mathcal{I}^{k-1}(f)(\xi_j^k) l_j^k \tag{34}$$

$$= \sum_{\xi_j^k \in \mathcal{X}^k} \left(f(\xi_j^k) - \mathcal{I}^{k-1}(f)(\xi_j^k) \right) l_j^k, \tag{35}$$

and, since $f(\xi_j^k) - \mathcal{I}^{k-1}(f)(\xi_j^k) = 0 \ \forall \ \xi_j^k \in \mathcal{X}^{k-1}$,

$$\Delta^k(f) = \sum_{\xi_j^k \in \mathcal{X}_\Delta^k} \left(f(\xi_j^k) - \mathcal{I}^{k-1}(f)(\xi_j^k) \right) l_j^k, \tag{36}$$

where $\mathcal{X}_\Delta^k = \mathcal{X}^k \setminus \mathcal{X}^{k-1}$, with $\mathcal{X}^0 = \emptyset$, represents the set of nodes added to the set of support nodes in going from depth $k-1$ to k. Clearly, $\mathcal{X}_\Delta^k$ has $m_k^\Delta = m_k - m_{k-1}$ elements, and numbering them consecutively as $\mathcal{X}_\Delta^k = \{\xi_j^k, j = 1 \cdots, m_k^\Delta\}$, we can rewrite Eq. (36) as,

$$\Delta^k(f) = \sum_{j=1}^{m_k^\Delta} \underbrace{\left(f(\xi_j^k) - \mathcal{I}^{k-1}(f)(\xi_j^k) \right)}_{w_j^k} l_j^k. \tag{37}$$

As in [43], we define w_j^k as the 1D hierarchical surpluses, which is the difference between the actual function value and the value obtained using the interpolant from the previous level. We also define the set of functions l_j^k as the hierarchical basis functions. Using the incremental interpolant, we can rewrite the nodal interpolant in Eq. (25) in a hierarchical fashion as,

$$\mathcal{I}^k(f) = \mathcal{I}^{k-1}(f) + \Delta^k(f) = \sum_{i=1}^{k} \Delta^i(f), \tag{38}$$

using the set of support nodes $\mathcal{X}^k = \bigcup_{i=1}^k \mathcal{X}_\Delta^i$. The 1D nodal and hierarchical support nodes and basis functions for the case of piecewise linear interpolation are plotted in Fig. 1.

4.2.2. Univariate to multi-variate interpolation

Given the univariate interpolation formula as in Eq. (25), to obtain an interpolation formula for the multi-variate case, one could simply use tensor product, given as

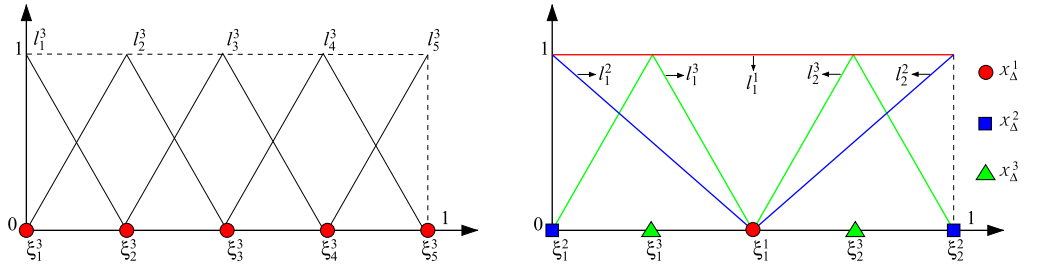
$$(\mathcal{I}^{k_1} \otimes \cdots \otimes \mathcal{I}^{k_n})(f) = \sum_{\xi_{j_1}^{k_1} \in \mathcal{X}^{k_1}} \cdots \sum_{\xi_{j_n}^{k_n} \in \mathcal{X}^{k_n}} f(\xi_{j_1}^{k_1}, \ldots, \xi_{j_n}^{k_n}) \cdot (l_{j_1}^{k_1} \otimes \cdots \otimes l_{j_n}^{k_n}), \tag{39}$$

where $\mathbf{k} = [k_1, \ldots, k_n]$ represents the depth of interpolation used in each direction. Using the hierarchical representation for the 1D formulas (Eq. (38)), this tensor product can be rewritten as,

$$(\mathcal{I}^{k_1} \otimes \cdots \otimes \mathcal{I}^{k_n})(f) = \sum_{i_1=1}^{k_1} \cdots \sum_{i_n=1}^{k_n} (\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_n})(f). \tag{40}$$

Clearly, the tensor product formula requires a very high number of support nodes $N = m_{k_1} \cdots m_{k_n}$. Even for a poor approximation employing just two nodes ($m_{k_i} = 2, \ \forall i$) in each direction, this number $N = 2^n$ grows rapidly for high dimensions $n \gg 1$. Thus, although the tensor product formula easily extends the univariate interpolation formula to the multi-dimensional case, it rapidly gets expensive as the number of dimensions grows.

The Smolyak algorithm [1] provides a way to extend the univariate interpolation formula to higher dimensions using the minimal number of support nodes. The algorithm employs tensor products in a special way such that it leads to orders of



(a) Nodal basis functions l_j^k and support nodes ξ_j^k, $j = 1, \ldots, m_k$, for depth $k = 3$.

(b) Hierarchical basis functions l_j^k and support nodes $\xi_j^k \in \mathcal{X}_\Delta^k$, $j = 1, \ldots, m_k^\Delta$, for $k = 1, 2, 3$.

Fig. 1. Univariate nodal and hierarchical basis functions and support nodes for linear interpolation.

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 [44]. The algorithm was proposed in [1] and has been extensively explored, such as in [44,46,47]. Using the 1D incremental interpolant defined in Eq. (33), the sparse interpolant $A_{q,n}$, where q is the depth of interpolation ($q \geqslant 0, q \in \mathbb{N}_0$) and n is the number of stochastic dimensions, is given by the Smolyak algorithm as,

$$A_{q,n}(f) = \sum_{|\mathbf{k}| \leqslant n+q} (\Delta^{k_1} \otimes \cdots \otimes \Delta^{k_n})(f) = A_{q-1,n}(f) + \underbrace{\sum_{|\mathbf{k}|=n+q} (\Delta^{k_1} \otimes \cdots \otimes \Delta^{k_n})(f)}_{\Delta A_{q,n}(f)}, \tag{41}$$

with $A_{-1,n} = 0$, and $|\mathbf{k}| = k_1 + \ldots + k_n$. Comparing this to the tensor product formula given by Eq. (40), it can be seen that the sparse interpolant $A_{q,n}$ only utilizes support nodes and basis functions with the restriction $|\mathbf{k}| \leqslant n+q$, which leads to the reduced sum. Using the 1D incremental interpolant given by Eq. (37), the n-dimensional incremental sparse interpolant $\Delta A_{q,n}(f)$, can be written as,

$$\Delta A_{q,n}(f) = \sum_{|\mathbf{k}|=n+q} \sum_{\mathbf{j}} \underbrace{(l_{j_1}^{k_1} \otimes \cdots \otimes l_{j_n}^{k_n})}_{l_{\mathbf{j}}^{\mathbf{k}}} \cdot \underbrace{(f(\xi_{j_1}^{k_1}, \cdots, \xi_{j_n}^{k_n}) - A_{q-1,n}(f)(\xi_{j_1}^{k_1}, \cdots, \xi_{j_n}^{k_n}))}_{w_{\mathbf{j}}^{\mathbf{k}}}, \tag{42}$$

where $\mathbf{j} = (j_1, \ldots, j_n)$ denotes the multi-index, such that $j_e = 1, \ldots, m_{k_e}^{\Delta}$ and $e = 1, \ldots, n$. As for the 1D case, the coefficients $\{w_{\mathbf{j}}^{\mathbf{k}}\}$ are defined as *hierarchical surpluses*, which is the difference between the actual function value and the value obtained using the interpolant from the previous level. For the case of continuous functions, the hierarchical surpluses tend to zero as the depth of interpolation q is increased. The adaptive stochastic collocation approach described later, employs these hierarchical surpluses as error indicators to govern the adaptive refinement procedure.

From Eq. (41) we can also observe that the Smolyak construction is hierarchical, which means that the accuracy of the interpolation can be improved by increasing the parameter q, and without having to discard previous results. In order to construct the sparse interpolant $A_{q,n}$, one needs to evaluate the function at sparse grid points $\mathcal{H}_{q,n}$ given by,

$$\mathcal{H}_{q,n} = \bigcup_{|\mathbf{k}| \leqslant n+q} (\mathcal{X}_{\Delta}^{k_1} \times \cdots \times \mathcal{X}_{\Delta}^{k_n}), \tag{43}$$

recalling that the univariate interpolation nodes are selected in a nested fashion, $X^{k-1} \subset \mathcal{X}^k$, and $\mathcal{X}_{\Delta}^k = \mathcal{X}^k \setminus \mathcal{X}^{k-1}$. The sparse grid $\mathcal{H}_{q,n}$ can also be written in a hierarchical fashion as,

$$\mathcal{H}_{q,n} = \mathcal{H}_{q-1,n} \cup \Delta\mathcal{H}_{q,n}, \quad \Delta\mathcal{H}_{q,n} = \bigcup_{|\mathbf{k}|=n+q} \left(\mathcal{X}_{\Delta}^{k_1} \times \cdots \times \mathcal{X}_{\Delta}^{k_n}\right), \tag{44}$$

with $\mathcal{H}_{-1,n} = \emptyset$. Thus, in order to increase the depth of interpolation from $q-1$ to q in n-dimensions, one only needs to evaluate the function at newly added set of points given by $\Delta\mathcal{H}_{q,n}$. The tensor and sparse grids for the case of piecewise linear interpolation using depth $k_i = 3,\ i = 1, 2$, are shown in Fig. 2, which illustrates a reduction in number of support nodes.

Remark 1. We must note that while constructing the interpolation hierarchically, the coefficients of the interpolation basis functions are not the nodal function values, but are hierarchical surpluses (see Eq. (42)). Thus, when we compute the variance using Eqs. (23) and (24), the coefficients $g(\mathbf{u}(\mathbf{x}, \boldsymbol{\xi}^i))$ also denote the hierarchical surpluses corresponding to the hierarchical interpolation for $g(\mathbf{u})$, which can be trivially constructed using the sampled solution at the collocation nodes.

4.2.3. Interpolation error estimates

Using the univariate error bounds, *a priori* error estimators for Smolyak construction are derived in [44]. For a n-variate objective function $f : [0,1]^n \rightarrow \mathbb{R}$, we first define,

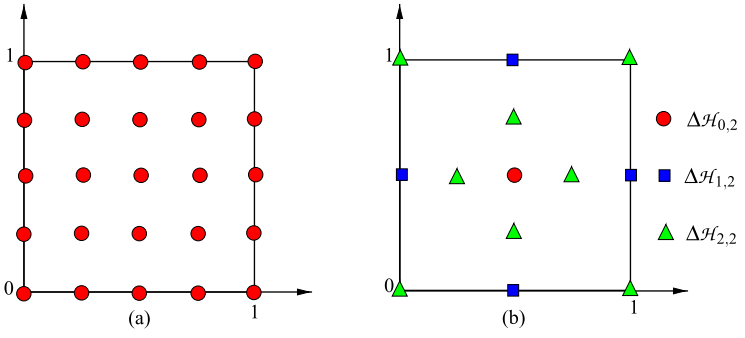


Fig. 2. Set of support nodes for piecewise linear interpolation. (a) Tensor grid $\mathcal{X}^3 \otimes \mathcal{X}^3$. (b) Sparse grid using Smolyak algorithm $\mathcal{H}_{2,2} = \bigcup_{q=0}^{2} \Delta\mathcal{H}_{q,2}$.

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

with $\beta \in \mathbb{N}_0^n$ and $|\beta| = \sum_{i=1}^{n} \beta_i$. Then, for $f \in F_n^2$, the error bound for the Smolyak construction using piecewise linear basis functions is given as,

$$\|f - A_{q,n}(f)\|_\infty \leqslant c_n \cdot (N^{-2} \cdot |\log_2 N|^{3(n-1)}), \tag{45}$$

where $c_n > 0$ is a constant which only depends on n, $A_{q,n}$ denotes the sparse interpolant of f, and $N = \dim(\mathcal{H}_{q,n})$ is the total number of support nodes. Similarly for $f \in F_n^r$, the error bound using higher-order Lagrange polynomials can be given as,

$$\|f - A_{q,n}(f)\|_\infty \leqslant c_{n,r} \cdot (N^{-r} \cdot |\log_2 N|^{(r+2)(n-1)+1}). \tag{46}$$

where $c_{n,r} > 0$ is a constant which depends only on n and r.

From Eqs. (45) and (46) we can observe that for sufficiently smooth functions, the convergence rate of the stochastic collocation method using either piecewise linear or polynomial interpolation would be orders of magnitude faster than the Monte Carlo (MC) or quasi-Monte Carlo (QMC) method, which were mentioned in Section 3.1. The performance of the sparse grid method would suffer with increasing the number of dimensions (unlike MC method) because of the weak dependence on the dimension in the logarithmic term. However, it has been shown in [13] that for the number of random dimensions as high as 50, the sparse grid method is far more efficient than the brute force MC method. At this point, we also notice that the sparse grid method offers exponential convergence only when the considered function is sufficiently smooth or the mixed derivatives are continuous. In later sections, we demonstrate through numerical examples that the convergence rate is drastically reduced for non-smooth functions. Based on this observation, we propose an adaptive stochastic collocation approach which can be effectively used for situations where one expects jumps or discontinuities in the random domain.

4.3. Numerical examples

In this section we consider a numerical example to demonstrate the sparse grid interpolation procedure using piecewise linear and polynomial basis functions based on the Chebyshev extrema. This example also serves as a motivation for the adaptive sparse grid interpolation approach. In this work, we have used the sparse grid interpolation toolbox developed by Klimke [45,48], where we have either modified the existing functions or added new subroutines as required. Consider a two-dimensional function as follows:

$$f(\xi_1, \xi_2) = \begin{cases} 0, & \text{if } \xi_1 > \alpha_1, \xi_2 > \alpha_2, \\ \sin(\pi\xi_1)\sin(\pi\xi_2), & \text{otherwise}, \end{cases} \tag{47}$$

where $0 \leqslant \xi_1, \xi_2 \leqslant 1$. In the following presentation, we denote the n-variate Smolyak interpolant of depth q, based on piecewise linear interpolation functions as $A_{q,n}^L(f)$ and the corresponding sparse grid as $\mathcal{H}_{q,n}^L$. Similarly, the n-variate sparse interpolant of depth q constructed using Lagrange interpolation functions based on Chebyshev nodes is represented as $A_{q,n}^C(f)$ and the corresponding sparse grid as $\mathcal{H}_{q,n}^C$.

4.3.1. Interpolation of smooth functions

For $\alpha_1 = \alpha_2 = 1, f(\xi_1, \xi_2)$ represents a smooth function in domain $[0,1]^2$. We construct the sparse interpolants $A_{q,2}^L(f)$ and $A_{q,2}^C(f)$, where the accuracy of interpolation increases with increasing q. The approximate function f as given by $A_{9,2}^L(f)$ is shown in Fig. 3(a). We compute the interpolation error as,

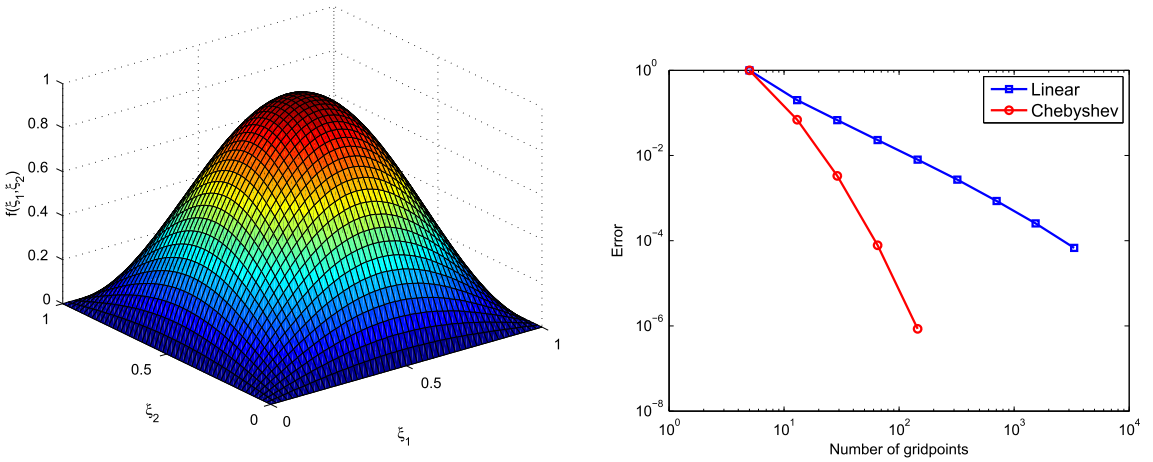
$$e_{q,n}^{L,C}(f) = \max_{\vec{\xi}^j \in \mathcal{X}_f} |f(\vec{\xi}^j) - A_{q,n}^{L,C}(f)(\vec{\xi}^j)|, \tag{48}$$

where $\mathcal{X}_f$ represents a sufficiently fine grid, employed to compute the error. The interpolation error $e_{q,2}^L(f)$ and $e_{q,2}^C(f)$ are plotted in Fig. 3(b) with increasing depth of interpolation q. As expected from Eqs. (45) and (46), the interpolation error decays much faster using higher-order Lagrange interpolation as compared to piecewise linear interpolation. The linear interpolation leads to an accuracy of $\sim 1 \times 10^{-4}$ for $q = 9$ using 3329 support nodes, as compared to an accuracy of $\sim 1 \times 10^{-6}$ for $q = 5$ using 145 nodes. The corresponding sparse grids are shown in Fig. 4.

In addition to the interpolation error, we also consider the error in the moments, such as mean and variance, since ultimately we are interested in approximating the statistics of the random solution. The relative error in mean $e_{q,n}^{L,C}(\mu)$ and variance $e_{q,n}^{L,C}(v)$ are computed as,

$$e_{q,n}^{L,C}(\mu) = \frac{|\mu_0 - \mu(A_{q,n}^{L,C}(f))|}{|\mu_0|}, \quad e_{q,n}^{L,C}(v) = \frac{|v_0 - v(A_{q,n}^{L,C}(f))|}{v_0}, \tag{49}$$

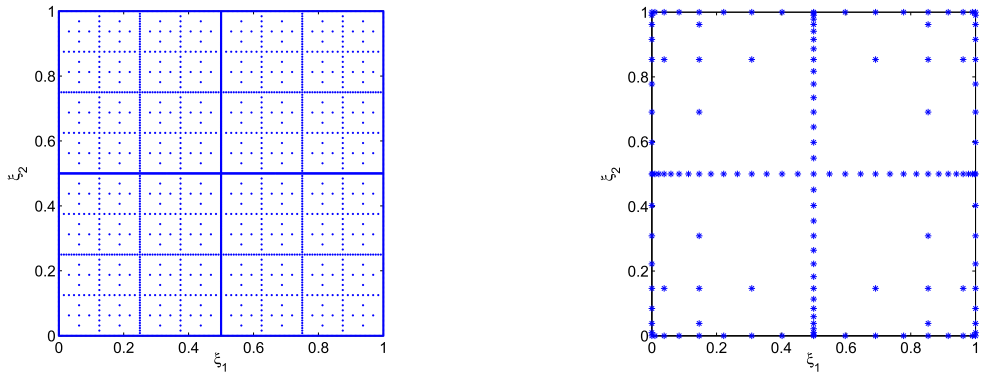
where μ_0 and v_0 are the actual mean and variance (computed analytically for the considered test functions), respectively, and $\mu(A_{q,n}^{L,C}(f))$ and $v(A_{q,n}^{L,C}(f))$ are the approximate mean and variance, respectively, which can be computed using Eq. (23). The relative error in mean and variance, with increasing level of interpolation q, is plotted in Fig. 5. As expected, the error in the statistics also decays faster using polynomial interpolation as compared to linear interpolation.



(a) Interpolated function using piecewise linear basis functions, $N = 3329$.

(b) Interpolation error using piecewise linear and polynomials basis functions.

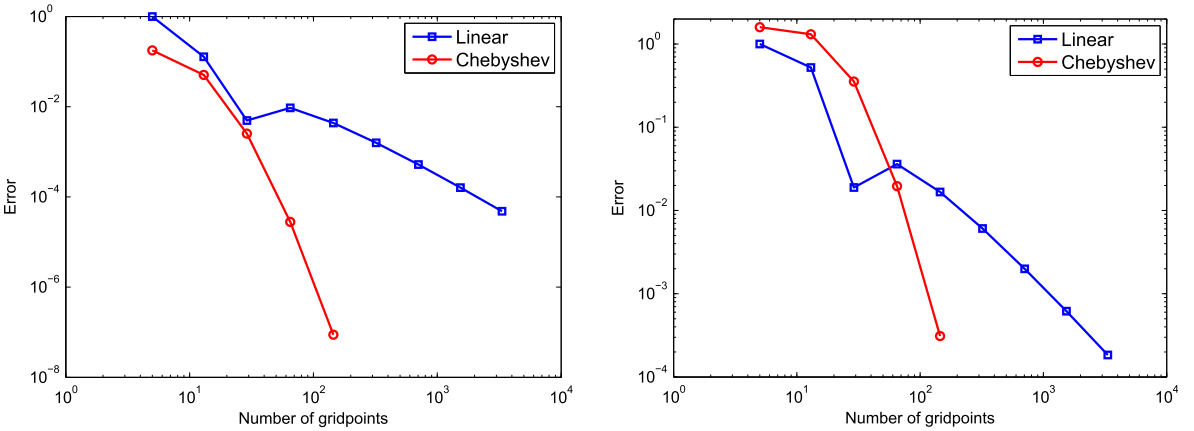
Fig. 3. Interpolation of a smooth function using sparse grid interpolation procedure.



(a) Support nodes for piecewise linear interpolation, $\#\mathcal{H}^{L}_{9,2} = 3329$.

(b) Support nodes for polynomial interpolation, $\#\mathcal{H}^{C}_{5,2} = 145$.

Fig. 4. Sparse grid collocation nodes for a two-dimensional problem.



(a) Relative error in mean.

(b) Relative error in variance.

Fig. 5. Error in the statistics using piecewise linear and polynomial interpolation.

4.3.2. Interpolation of non-smooth functions

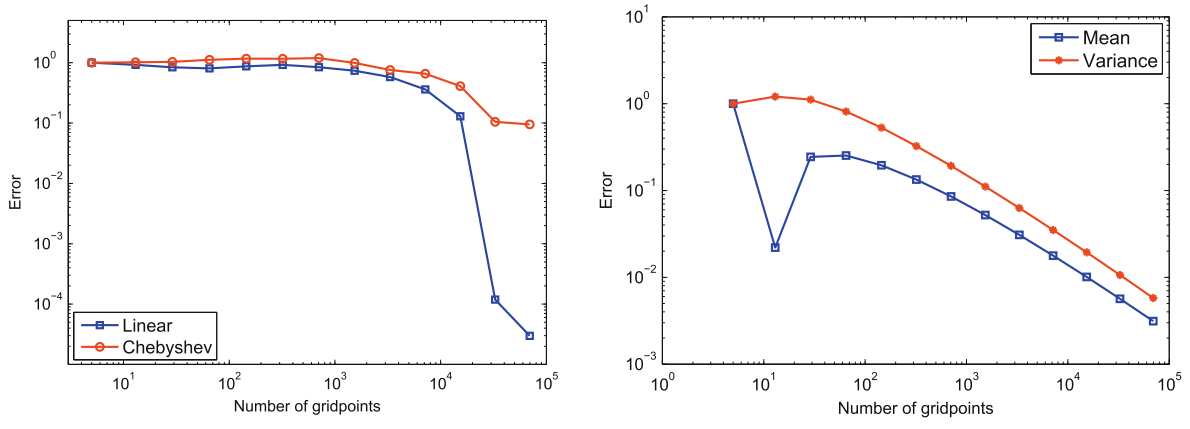
In many physical systems, small variations in the uncertain parameters may lead to jumps in the solution. For example, in MEMS switches, small variation in material properties and geometrical parameters may lead to *pull-in*, which is manifested as a discontinuity in the switch displacement in the random domain. In order to accurately compute the statistics of the stochastic solution using sparse grid interpolation technique, it is important to correctly capture these discontinuities in the random domain. We again consider the objective function as given by Eq. (47), for $\alpha_1 = \alpha_2 = 0.5$. The function f has sharp discontinuities at $\xi_1 = 0.5$ and $\xi_2 = 0.5$ planes. The sparse grid interpolation is used to construct an interpolant for this function.

In Fig. 6(a) we plot the interpolation error obtained using linear and polynomial basis functions with increasing level of interpolation. We observe that the convergence rate for both linear and polynomial interpolation are sub-linear for $q \leqslant 11$. The linear interpolation leads to an accuracy in the range 1×10^{-4} for $q = 13$ which includes 69,633 grid points. For the same number of support nodes, the polynomial interpolation could only provide an accuracy in the range 1×10^{-1}. This suggests that for discontinuous functions it is better to use piecewise linear interpolation as compared to polynomial interpolation. We also notice that the convergence rate for the discontinuous function is much slower as compared to the continuous case (shown in Fig. 3(b)), since the function no longer satisfies the regularity condition as required for error bounds given in Eqs. (45) and (46). In Fig. 6(b) we observe same trend for relative error in mean and variance, where we obtain an accuracy in the range 1×10^{-2} using 69,633 support nodes for the discontinuous function, as compared to an accuracy of 1×10^{-4} using 3329 nodes for the continuous function case. The interpolated function and corresponding sparse grids using linear interpolation for various levels of interpolation are shown in Fig. 7.

5. Adaptive stochastic collocation method

The performance of the sparse grid interpolation deteriorates in the presence of discontinuities, as expected from Eqs. (45) and (46) and shown by the numerical example considered earlier. A dimension adaptive sparse grid interpolation algorithm has been developed in [20] and further explored in [15,45], where important dimensions are sampled differently and ultimately it leads to employing larger number of support nodes in those dimensions. However, such an approach may not significantly improve the performance of the classical Smolyak construction in situations where most of the considered random dimensions are important. Such a situation can occur more frequently for physical systems where one may need to consider only a moderate number of random parameters, each of which may lead to a discontinuity in the stochastic solution.

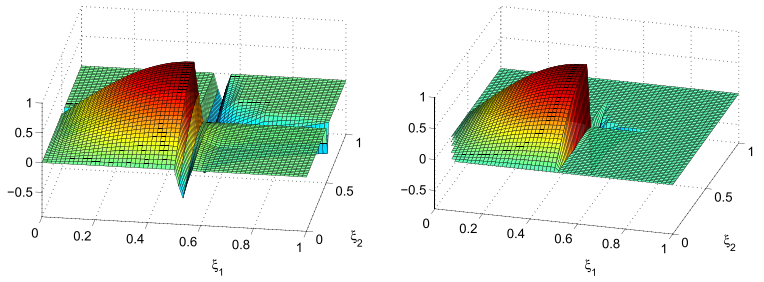
In order to effectively handle such situations, we propose an adaptive stochastic collocation approach. The adaptive refinement proceeds as follows: we begin with the entire random domain, which is subdivided into several subdomains or elements during the course of the refinement procedure. For the random domain, we construct a coarse interpolant using the classical Smolyak algorithm. Such a construction involves computing the hierarchical surpluses, using which we estimate the local contribution of the random element towards the global error. In addition, the hierarchical surpluses are also used to estimate the more sensitive random dimensions. Based on these error estimates, we subdivide the random domain into subdomains along the most sensitive dimensions, as explained in the next section. The refinement procedure is then repeated for each subdomain until the required error tolerance is achieved. Finally, the required statistics such as mean and variance can be easily computed using the local statistics computed within each subdomain.



(a) Interpolation error using piecewise linear and polynomial interpolation.

(b) Error in mean and variance using linear interpolation.

Fig. 6. Error in interpolation and statistics for discontinuous function.



(a) Linear interpolant $A^{L}_{5,2}(f)$.

5.1. Adaptivity criterion

We recall that the random domain was characterized by n mutually independent random variables $\boldsymbol{\xi} = [\xi_i]_{i=1}^{n}$, where $\boldsymbol{\xi} : \Theta \rightarrow \Gamma = [0,1]^n$. During the adaptive refinement procedure, the random domain Γ is decomposed into N_d non-overlapping subdomains, such that $\Gamma = \bigcup_{s=1}^{N_d} \Gamma^s$. For each of the subdomains s, we define the indicator function $I_s(\boldsymbol{\xi})$ as,

$$I_s(\boldsymbol{\xi}) = \begin{cases} 1 & \text{if } \boldsymbol{\xi} \in \Gamma^s, \\ 0 & \text{otherwise}. \end{cases} \tag{50}$$

Let us assume that the approximate solution in the sth-subdomain is constructed using Smolyak interpolant of depth q_s. We note that in order to employ the basis functions and sparse grids defined on the hypercube $[0,1]^n$, we need to map the subdomain Γ^s appropriately. We recall that the interpolation requires computation of hierarchical surpluses $w_{\mathbf{j}}^{\mathbf{k}}$ (Eq. (42)) where $\mathbf{k} = [k_1, \cdots, k_n]$ represents the depth of interpolation used in each direction, and $\mathbf{j} = [j_1, \cdots, j_n]$ denotes the multi-index, such that $j_e = 1, \ldots, m^{\Delta}_{k_e}$ and $e = 1, \ldots, n$. These hierarchical surpluses can be used to effectively guide the adaptive refinement strategy. We note that the incremental sparse interpolant $\Delta A_{q,n}$ corrects the interpolation $A_{q-1,n}$ at the newly added grid points $\Delta\mathcal{H}_{q,n}$. Therefore, the interpolation error in the random element s using an interpolant of depth q_s can be estimated as,

$$\beta_s = \max_{|\mathbf{k}|=n+q_s} (|w_{\mathbf{j}}^{\mathbf{k}}|), \tag{51}$$

which is the maximum of the hierarchical surpluses corresponding to newly added nodes given by grid $\Delta\mathcal{H}_{q_s,n}$. Using this, we will split a random element s if the following criterion is satisfied,

$$\beta_s J_s \geqslant \tau_1, \tag{52}$$

where $J_s = \Pr(I_s = 1)$ is the probability that $\boldsymbol{\xi}$ lies in sth-subdomain, and τ_1 is the prescribed error tolerance.

Furthermore, in order to identify the more sensitive random dimensions, we define another error measure $\gamma_i,\ i = 1, \ldots, n$ as,

$$\gamma_i = \sum_{\mathbf{j},\mathbf{k}=\mathbf{k}_i} (w_{\mathbf{j}}^{\mathbf{k}})^2, \quad i = 1, \cdots, n, \quad \mathbf{k}_i = \{\mathbf{k} : k_i = q_s + 1, k_j = 1\ \forall\ j \neq i\}. \tag{53}$$

We recall from Eq. (42) that the incremental sparse interpolant $\Delta A_{q_s,n}$ is constructed by employing tensor product of the 1D interpolation formulas which satisfy $|\mathbf{k}| = n + q_s$. Each of the error measures γ_i, includes contribution only from those newly added nodes which correspond to employing an interpolation of maximum allowable depth $k_i = q_s + 1$ in the ith-dimension and coarsest interpolation with depth $k_j = 1,\ \forall\ j \neq i$, in all other directions. Thus, each of the error measures γ_i gives an estimate of the error along the ith-dimension and can be used to identify the more sensitive random dimensions. We split each random element into two equal elements along all random directions which satisfy,

$$\gamma_i \geqslant \tau_2 \cdot (\max_{j=1,\ldots,n} \gamma_j), \quad 0 < \tau_2 < 1, i = 1, \ldots, n, \tag{54}$$

where τ_2 is a tunable parameter, which is chosen as 0.5 in this work.

We illustrate the refinement procedure for the case of piecewise linear interpolation using depth $q_s = 1$, in Fig. 8. We begin with the random element as shown in Fig. 8(a), and for simplicity, denote the hierarchical surplus corresponding to each node as $\{w_i\}_{i=1}^5$. Using this, the error estimates can be computed as,

$$\beta_s = \max_{i \in [2,5]}(w_i), \quad \gamma_1 = w_2^2 + w_3^2, \quad \text{and } \gamma_2 = w_4^2 + w_5^2. \tag{55}$$

The new random elements and the corresponding support nodes which are created based on the adaptivity criterion are shown in Fig. 8(b–d).

We must note that when a random element s is split into subdomains, the local interpolant of depth q_s, which had already been computed, is no longer used, and such an interpolant needs to be constructed again for each of the subdomains. This implies that the functional evaluations at the support nodes of the grid corresponding to an interpolant of depth q_s are wasted at each refinement step. However, this additional cost is not very significant since we employ a very coarse interpolant $(q_s = q_0 = 1)$ within each subdomain. In order to represent the actual cost of the adaptive algorithm, we report both the actual number of grid points N at any adaptive iteration step, as well as the total number of functional evaluations $\widetilde{N}$, for the numerical examples. In Fig. 9 we illustrate a few steps of the adaptive refinement procedure for a two-dimensional problem.

5.2. Computation of moments

After we obtain the local approximation within each subdomain, we can construct the global moments as described by Wan and Karniadakis for the case of ME-GPC in [18]. In each of the subdomains, we define a random vector $\boldsymbol{\eta}^s : I_s^{-1}(1) \to \Gamma^s$, subject to the conditional PDF $\rho^s(\boldsymbol{\eta}^s)$, given as,

$$\rho^s(\boldsymbol{\eta}^s) = \frac{\rho(\boldsymbol{\eta}^s)}{\Pr(I_s = 1)}. \tag{56}$$

Using this, we can approximate the mth-moment of the stochastic solution $u(\boldsymbol{\xi})$ on the entire random domain, using law of total probability [26] as,

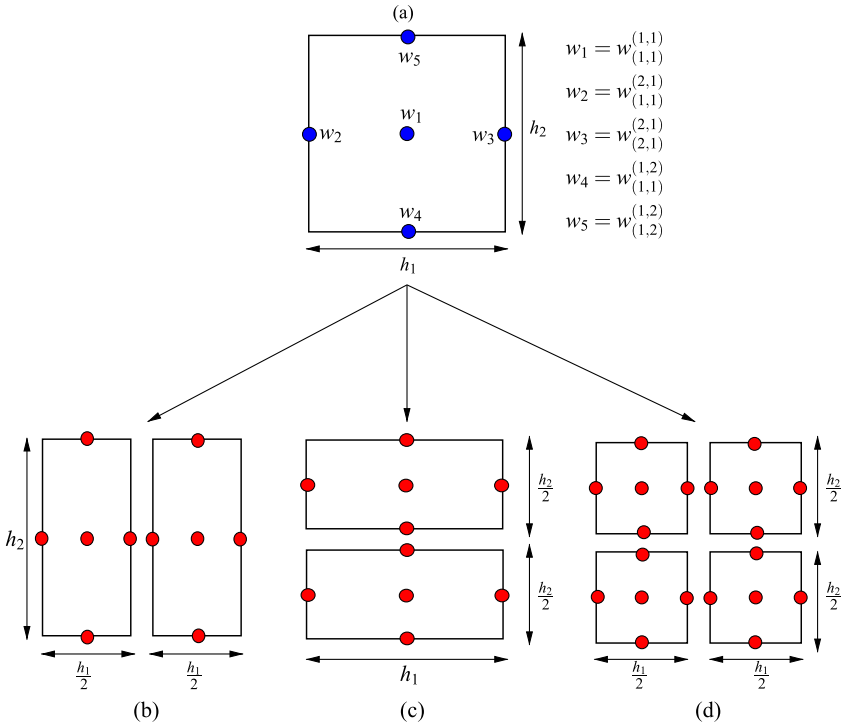


Fig. 8. Decomposition of a random element based on the adaptivity criterion. (a) Original random element s, (b) $\gamma_2 < \tau_2 \cdot \gamma_m \leqslant \gamma_1$, (c) $\gamma_1 < \tau_2 \cdot \gamma_m \leqslant \gamma_2$, (d) $\gamma_{1,2} \geqslant \tau_2 \cdot \gamma_m$, $(q_s = 1, \gamma_m = \max_{i=1,2}\gamma_i)$.

$$\langle u^m(\xi)\rangle \approx \int_{\Gamma} \hat{u}^m(\xi)\rho(\xi)d\xi = \sum_{s=1}^{N_d} \Pr(I_s = 1) \int_{\Gamma^s} \hat{u}^m(\eta^s)\rho^s(\eta^s)d\eta^s. \tag{57}$$

In this work, since we assume ξ to be a vector of n mutually independent uniformly distributed random variables in $[0,1]$, $\rho(\xi) = 1$ and $\Pr(I_s = 1) = \text{Volume}(\Gamma^s)$. Also, the integral over Γ^s in Eq. (57), can be mapped to the standard domain $[0,1]^n$, by introducing another random vector $\xi^s = g_s(\eta^s) : I_s^{-1}(1) \to [0,1]^n$, with a PDF $\rho_\xi^s = 1$, where,

$$g_s(\eta^s) : \eta_i^s = (b_i^s - a_i^s)\xi_i^s + a_i^s, \quad i = 1,\ldots,n, \tag{58}$$

and $(a_i^s, b_i^s), i = 1,\ldots,n$ define the subdomain $\Gamma^s = [a_1^s, b_1^s) \times [a_2^s, b_2^s) \ldots \times [a_n^s, b_n^s)$. Using this, Eq. (57) can be rewritten as,

$$\langle u^m(\xi)\rangle \approx \sum_{s=1}^{N_d} \Pr(I_s = 1) \int_{[0,1]^n} \hat{u}_s^m(\xi^s)d\xi^s, \tag{59}$$

where $\hat{u}_s(\xi^s)$ represents the approximate solution in sth-subdomain.

5.3. Error estimate

Following [18] it can be shown that the error in the L_2 norm of the mth-moment of the random field $u(\xi)$ can be written as a weighted sum of the local L_2 error in the mth-moment in each random element,

$$\epsilon = \int_{\Gamma} (u^m(\xi) - \hat{u}^m(\xi))^2 \rho(\xi)d\xi = \left(\sum_{s=1}^{N_d} J_s \epsilon_s^2\right)^{\frac{1}{2}}, \tag{60}$$

where ϵ and ϵ_s denote the global and local error, respectively, and $J_s = \Pr(I_s = 1)$ denotes the weight. The basic idea behind the adaptive sparse grid collocation approach is to *capture* the discontinuity into smaller random elements, such that its contribution towards the global error is minimal. For example, if the discontinuity lies in the element s^*, we may not be able to reduce the local error ϵ_{s^*} to the desired accuracy level by increasing the depth of interpolation q_{s^*}, but we can certainly reduce the product $J_{s^*}\epsilon_{s^*}^2$ by further refining the element s^*. This error reduction strategy makes sense, since in the random domain we are only interested in computing the moments of the stochastic solution accurately. Although, we have not proved any error bounds for the local error ϵ_s for the case of discontinuous functions, we expect it to be bounded as the size of the random element is adaptively reduced. We demonstrate the validity of such an assumption for the case of using piecewise linear interpolation within each subdomain through numerical examples. The adaptive stochastic collocation methodology is detailed in Algorithm 1.

Algorithm 1. Adaptive sparse grid collocation scheme

1: **Pre-processing**. Identify uncertain parameters (material properties and geometrical parameters) and represent them in terms of independent random variables $\xi : \Theta \to \Gamma = [\xi_1, \xi_2, \ldots, \xi_n]^T$, such that n represents the dimension of the random domain Γ.

2: **Sampling**. Sample the stochastic solution at the set of sparse grid points generated adaptively, as follows:

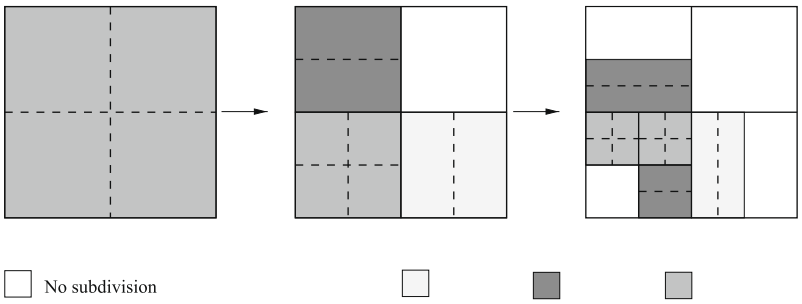
a. *Initialization*. Set number of subdomains $N_d = 1$, depth of interpolation in each subdomain $q_s = q_0, s = 1,\ldots,N_d$. Specify desired error tolerance τ_1 and parameter τ_2.

b. **Repeat**.

c. Loop over all subdomains $s = 1,\ldots,N_d$.
Solve the deterministic problem at the sparse grid nodes $\mathcal{H}_{q_s,n}$, and compute the error estimates β_s and $\gamma_i, i = 1,\ldots,n$.
If $(\beta_s J_s \geqslant \tau_1)$, **then**
If $\gamma_i \geqslant \tau_2 \cdot (\max_{j=1,\ldots,n}\gamma_j), i = 1,\ldots,n$, **then**.
Add dimension i to the index set $\mathbb{I}_s$.
End If
Split the random element s into two equal elements along all directions contained in index set $\mathbb{I}_s$. Update the number of subdomain N_d.
Else Do Nothing for element s.
End If.

d. **Until** $(\beta_s J_s < \tau_1), s = 1,\ldots,N_d$.

3: **Post-processing**. Compute the statistics (such as mean and standard deviation) of the deformation $\mathbf{u}(\mathbf{x},\xi)$ and surface charge density $\sigma(\mathbf{x},\xi)$.



Remarks. In this work, we have applied the proposed adaptive collocation framework to problems with discontinuities in the random domain. This framework can be easily extended to develop effective algorithms for certain other situations such as sharp local variations or skewed joint PDFs, by noting the following points.

- Here, we developed the adaptive collocation strategy using Smolyak interpolant of fixed depth q_s based on piecewise linear basis functions. In the presence of discontinuities in the stochastic domain, such a strategy offers significant reduction in the computational effort, as demonstrated by numerical examples in the next section. However, for situations where we do not expect discontinuities in the stochastic domain but sharp local variations, the use of liner interpolation may be restrictive, as the higher-order interpolation offers faster convergence rate. For such cases, we can easily extend our adaptive framework by using higher-order polynomial basis functions and choosing between subdivision or increasing the depth of interpolation q_s at each refinement step, just like the *hp-refinement* for the physical space.
- The statistics such as mean and variance over the entire random domain are computed as a weighted sum of the local statistics, which are obtained by independently solving the local approximation problem within each subdomain. This feature of our adaptive framework can be used to develop hybrid strategies, where for each subdomain, one can estimate the local statistics by any method of choice in addition to stochastic collocation (SC), such as Monte Carlo (MC) or generalized polynomial chaos (GPC), and employ these to obtain the global statistics.
- The standard stochastic collocation method works by first approximating the stochastic solution with high precision, by increasing the depth of interpolation, and then by computing the moments as given by Eq. (23). Such a framework may not result in an optimal node distribution with respect to computing the statistics, especially for the case of skewed joint probability distribution. However, the proposed adaptive collocation framework should work well in such situations, since the random elements are subdivided according to the criterion $\beta_s \Pr(I_s = 1) \geqslant \tau_1$, where the local error contribution is automatically weighed according to the probability density. Even for the case of smooth stochastic solution, such a refinement strategy would ultimately result in higher number of support nodes in regions with higher probability density.

5.4. Numerical illustration

5.4.1. Function with line singularity

We now revisit the interpolation problem considered in Section 4.3.2, for the non-smooth function f given by Eq. (47), for $\alpha_1 = \alpha_2 = 0.5$, and employ the adaptive interpolation procedure. In Fig. 10 we plot the interpolation error obtained using classical Smolyak construction and the adaptive procedure based on piecewise linear basis functions. The interpolation error is computed as before, given by Eq. (48). We present the interpolation error for the adaptive procedure using various values for the parameters τ_1 and q_0. We note that τ_1 represents the error tolerance parameter and governs how frequently the random elements are decomposed, while q_0 denotes the depth of interpolation, and hence determines the accuracy within each random element. We first fix $q_0 = 5$ and decrease the required error tolerance level $\tau_1 = 10^{-3}, 10^{-4}$ and 10^{-5}. As can be seen from Fig. 10, since lower value of τ_1 results in more frequent decomposition of random elements, it also leads to lower interpolation error. Also using $\tau_1 = 10^{-5}$ and $q_0 = 5$, the adaptive procedure leads to an accuracy in the range 1×10^{-4} using 15,805 nodes, as opposed to 69,633 nodes using the classical construction.

We now fix $\tau_1 = 10^{-5}$ and compare the interpolation error obtained using $q_0 = 1$ and 5. Since $q_0 = 5$ employs more accurate interpolation within each random element as opposed to $q_0 = 1$, it also leads to lower overall interpolation error. Thus, in order to drive the interpolation error to the desired accuracy level, we need to carefully choose both q_0 and τ_1.

For the purpose of uncertainty quantification, we are more interested in reducing the error in the moments of the unknown solution. We conducted several numerical experiments, which show that choosing a coarse interpolant within each random element, such as $q_0 = 1$ or 2, which leads to a reasonable approximation within each element, the error in the moments can be reduced to the desired accuracy level by simply decreasing the error tolerance level τ_1. In Fig. 11

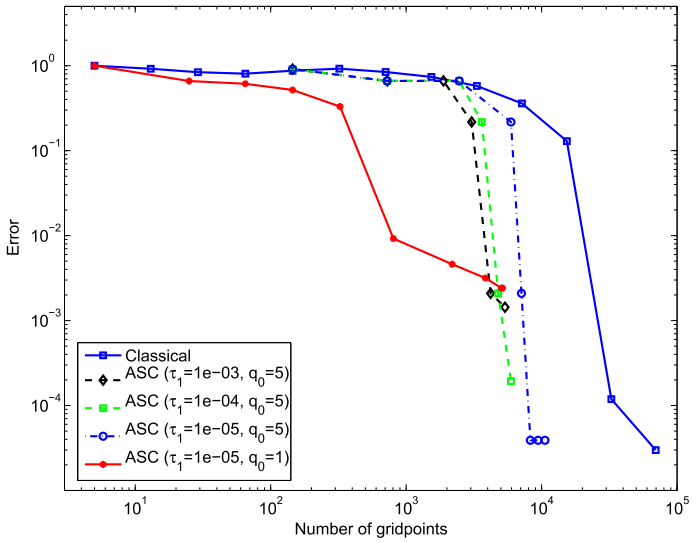
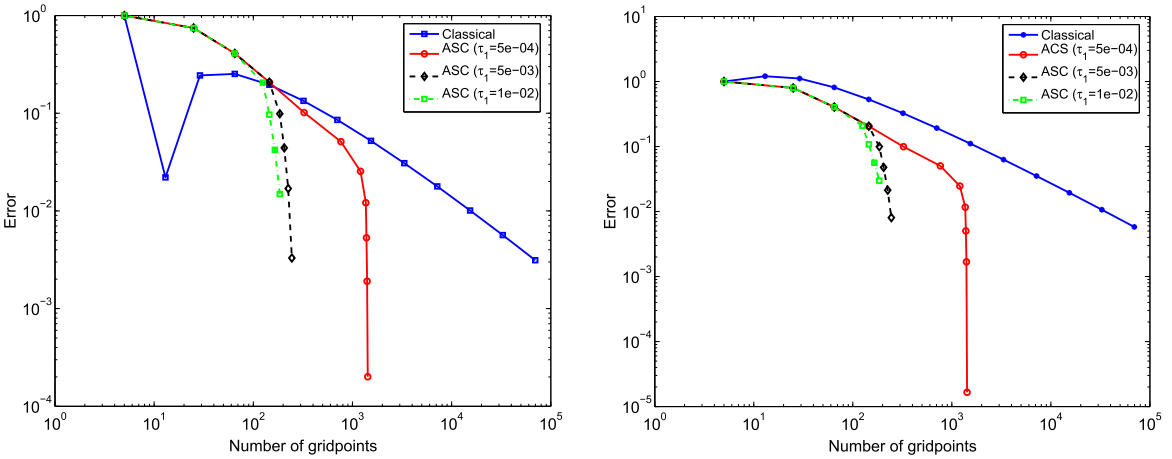


Fig. 10. Interpolation error for the non-smooth function using classical and adaptive interpolation procedure based on piecewise linear basis functions.

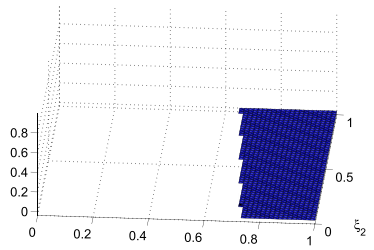


(a) Relative error in mean. (b) Relative error in variance.

Fig. 11. Error in statistics using classical and adaptive sparse grid interpolation, $q_0 = 1$.

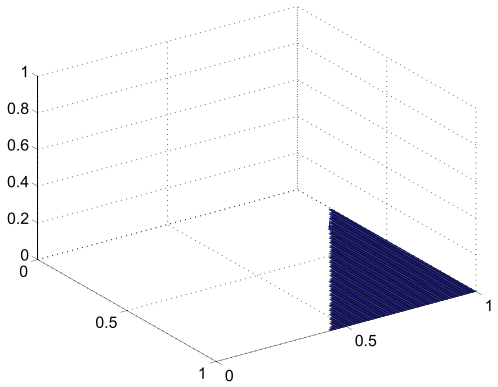
we plot the relative error in mean and variance obtained using the classical and adaptive interpolation procedure based on piecewise linear interpolation. As we reduce the error tolerance parameter τ_1 from 10^{-2} to 5×10^{-4}, the relative error in mean decreases from 1.49×10^{-2} to 2.00×10^{-4}, and that in variance from 2.98×10^{-2} to 1.66×10^{-5}. This shows that we can obtain the required accuracy by reducing the parameter τ_1. We also note that the adaptive procedure provides significant improvement, as it leads to an accuracy in the range $\sim 10^{-4}$ in mean and variance, using just 1425 functional evaluations, as opposed to an accuracy $\sim 10^{-2}$ using 69,633 nodes for the classical construction. The evolution of the interpolated function, corresponding random elements and sparse grid nodes are plotted in Fig. 12. We also compare the computational cost and accuracy of the moments obtained using adaptive stochastic collocation (ASC) procedure and the Monte Carlo (MC) method in Table 1. As can be easily seen, using 1425 functional evaluations, the ASC procedure approximately leads to 2 and 1 orders of magnitude reduction in the error in mean and variance, respectively, as compared to the MC method.

From Fig. 11 one should notice that towards the later stages of the adaptive refinement procedure, we obtain a sharp decrease in the error in mean and variance. We would like to point out that this steep decrease is the result of the fact that towards the last few adaptive iterations, the algorithm accurately captures the discontinuity along the element boundaries, and hence adding only a few tens of nodes results in a sharp reduction in the error. However, in general the stochastic discontinuity may not always lie along the direction of domain decomposition, and hence such a steep decrease in the error would not be expected. In order to demonstrate this, we consider the following function



$$f(\xi_1, \xi_2) = \begin{cases} \sin(\pi\xi_1)\sin(\pi\xi_2), & \text{if } \xi_1 + \xi_2 \leqslant 1, \\ 0, & \text{otherwise}, \end{cases} \tag{61}$$

where the discontinuity lies along the line $\xi_1 + \xi_2 = 1$, as shown in Fig. 13(a). The adaptively refined domain and the corresponding sparse grid are shown in Fig. 13(b). We can immediately see that the adaptive procedure automatically places more nodes along the discontinuity. In Fig. 14 we also plot the error in mean and variance obtained using classical and adaptive procedure. As can be seen, for this case we do not obtain any steep decrease in the error. However, we note that the adaptive procedure still leads to significant improvement over the classical Smolyak algorithm.



In order to demonstrate that the adaptive sparse grid interpolation provides significant advantage even for higher number of random dimensions ($n < 10$), we consider the function f such that,

$$f(\boldsymbol{\xi}) = \begin{cases} 0, & \text{if } \xi_i > 0.5, i = 1, \ldots, n \\ \prod_{i=1}^{n} \sin(\pi \xi_i), & \text{otherwise.} \end{cases} \tag{62}$$

We plot the relative error in mean and variance for $n = 4$, 6 and 8 in Fig. 15 and conclude that even for moderate number of random dimensions, the adaptive approach performs significantly better as compared to the standard Smolyak construction.

5.4.2. Kraichnan–Orszag (K–O) problem

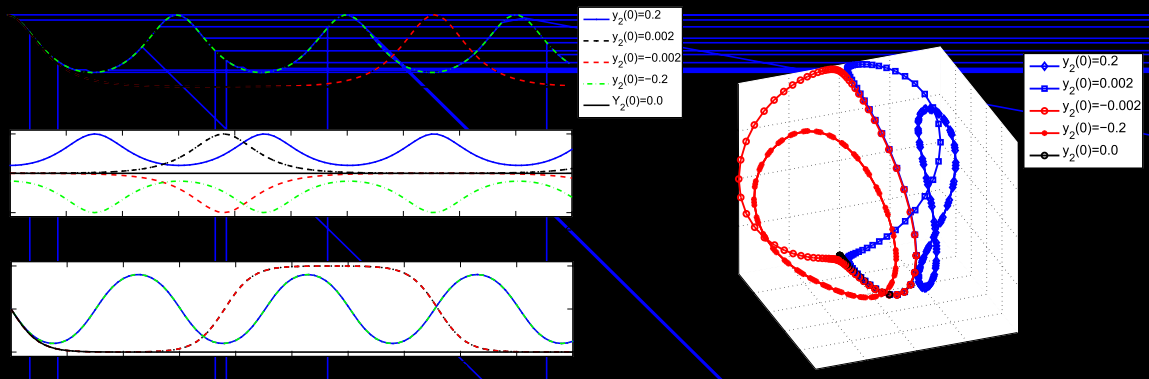
In order to further demonstrate the accuracy and efficiency of the proposed adaptive approach, we now consider the three-mode Kraichnan–Orszag (K–O) problem, which has also been studied in [21,22]. The transformed K–O problem is represented by a coupled nonlinear ODE system given as [18]:

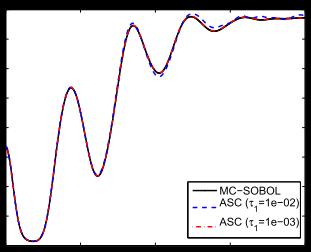
$$\frac{dy_1}{dt} = y_1 y_3, \tag{63}$$

$$\frac{dy_2}{dt} = -y_2 y_3, \tag{64}$$

$$\frac{dy_3}{dt} = -y_1^2 + y_2^2, \tag{65}$$

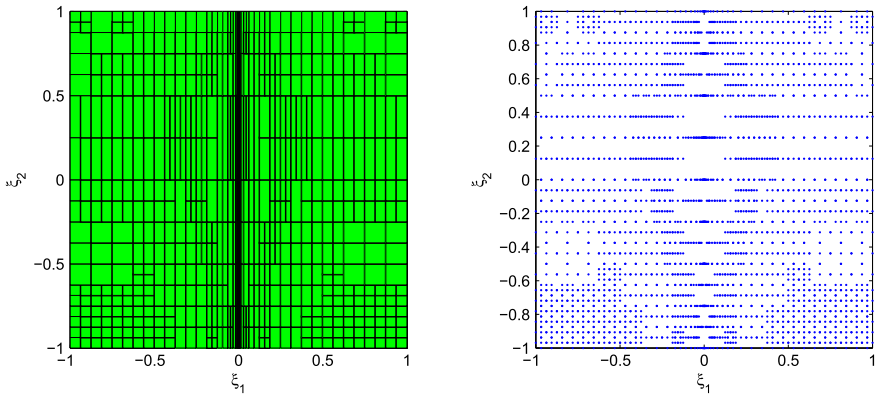
subjected to random initial conditions,


y2(0)=0.2
y2(0)=0.002
y2(0)=−0.002
y2(0)=−0.2
Y2(0)=0.0
y2(0)=0.2
y2(0)=0.002
y2(0)=−0.002
y2(0)=−0.2
y2(0)=0.0



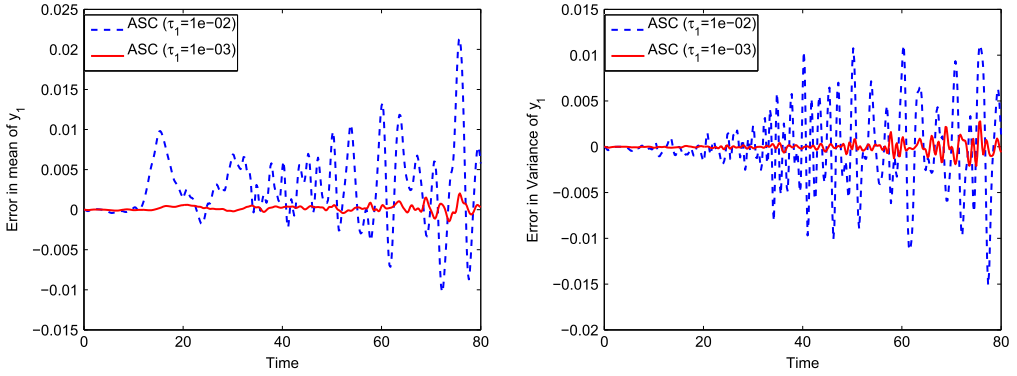
olerance levels τ_1, $\tau_2 = 0.5,\ q_0 = 1$.

-mean	Error-variance
$\times 10^{-2}$	8.28×10^{-2}
$\times 10^{-3}$	1.01×10^{-2}
$\times 10^{-4}$	2.12×10^{-3}
$\times 10^{-4}$	3.37×10^{-4}



(a) Number of elements $N_d = 594$. (b) Gridpoints $N = 2970, \tilde{N} = 4495$.

Fig. 19. Adaptively refined random domain and the corresponding sparse grid for $t = 10,\ \tau_1 = 10^{-3},\ q_0 = 1$.



(a) Error in mean. (b) Error in variance.

Fig. 20. Long term error behavior: Error in mean and variance of mode y_1 for $\tau_1 = 10^{-2}(\tilde{N} = 2575)$ and $\tau_1 = 10^{-3}(\tilde{N} = 17,295)$.

obtaining an accuracy in the range $\sim 10^{-4}$ for variance, the adaptive procedure requires 19,495 nodes as compared to 69,633 nodes for the Smolyak algorithm. The adaptively refined random domain and the corresponding sparse grid are shown in Fig. 19. From the adaptive grid we can see that more nodes are placed along the line $\xi_1 = 0$, since the discontinuity lies at the line $y_2(0) = 0$.

We also investigate the long term behavior of the error in the moments of K–O modes obtained using ASC approach. In Fig. 20 we plot the error in mean and variance for mode y_1 within the time interval $[0, 80]$ for various levels of error tolerance τ_1. It can be seen that we obtain an accuracy in the range $\sim 10^{-2}$ for $\tau_1 = 10^{-2}$ and $\sim 10^{-3}$ for $\tau_1 = 10^{-3}$.

6. Numerical results: MEMS switch

In this section we employ the adaptive sparse grid collocation approach to study the effect of uncertain material properties and geometrical parameters on the performance of MEMS devices. Specifically, we consider a micro-switch, modeled as a cantilever beam which is 80 μm long, 1 μm thick and 10 μm wide, located over a ground plane at a distance g, as shown in Fig. 21.

As the applied potential difference V between the beam and the ground plane is increased, the beam increasingly deflects towards the ground plane. At a voltage V_p (known as the pull-in voltage), when the electrostatic force can no longer be balanced by the restoring elastic force, the beam finally collapses onto the ground plane, which signifies the *ON* state of the switch. This phenomenon, known as *pull-in instability*, is caused due to the nonlinear nature of the electrostatic actuation force, and has been well studied for electrostatic MEMS devices. For certain other applications, where pull-in may be undesirable, this instability limits the travel distance of the microstructures to about $1/3$ of the initial gap g. Thus, both the pull-in voltage V_p and the allowable travel distance, measured by the maximum vertical tip deflection ν_L (just before pull-in), are important performance parameters for electrostatic MEMS. In order to design effective and reliable devices, it is necessary to consider the effect of uncertain input parameters on these two output parameters.

We consider the effect of uncertain Young's modulus E and the initial gap g between the beam and the ground plane, which are given as:

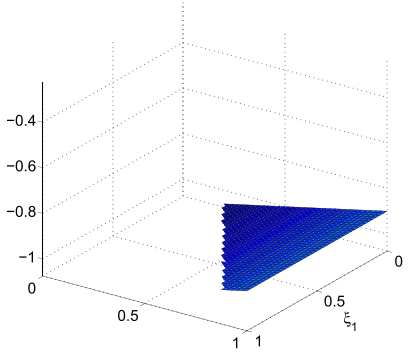
$$E = E_0(1 + v_E(2\xi_1 - 1)), \quad g = g_0(1 - v_g(2\xi_2 - 1)), \tag{67}$$

where $E_0 = 169$ GPa and $g_0 = 1\ \mu m$ are the mean values, $v_E = v_g = 0.1$, which represents a 10% variation in both parameters, and $\xi_{1,2}$ are independent uniformly distributed random variables in $[0, 1]$. We employ the adaptive stochastic collocation approach as detailed in Algorithm 1, to compute the statistics of the vertical displacement of the beam.

6.1. Verification using MC

The results obtained using adaptive stochastic collocation (ASC) approach are verified using Monte Carlo (MC) simulations. For MC, we use N_{mc} samples each for ξ_1 and ξ_2 generated according to the uniform distribution, such that ξ_1 and ξ_2 are independent. For each (ξ_1^i, ξ_2^i) pair, we solve the deterministic coupled electromechanical problem, and compute the values $\{v_L^i\}$. After several numerical experiments, it was determined that MC simulations converge for $N_{mc} = 10,000$, and hence the moments obtained using 10,000 MC samples are treated as "exact", in order to compute the error in the moments obtained using ASC approach.

We first consider the case when applied voltage $V = 10.0$ V, and compute the mean and variance for the vertical tip deflection using ASC and MC. In Table 3 we show the error in moments obtained using ASC approach while decreasing the error tolerance parameter from $\tau_1 = 10^{-2}$ to 10^{-4} to ensure convergent results. For $\tau_1 = 10^{-4}$ we obtain an accuracy



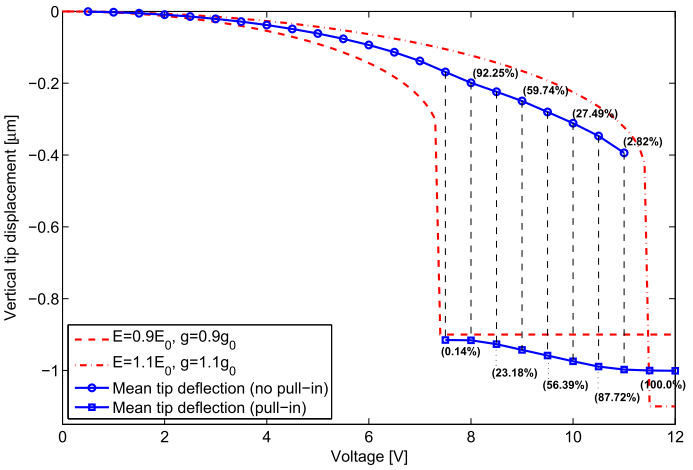


Fig. 24. Worst-case and mean pull-in behavior of the switch.

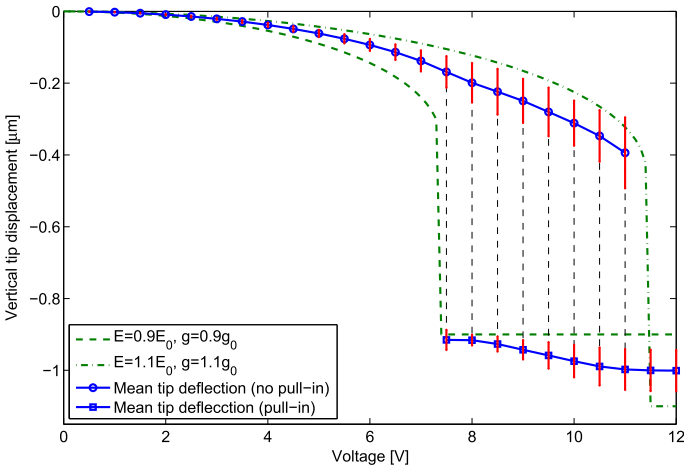


Fig. 25. Mean vertical tip displacement with error bars.

gives the mean actuation range available in the presence of uncertainties. On the other hand, the mean vertical tip displacement for realizations that lead to pull-in, denoted as $\mu(v_L^p)$, signifies the mean value of the initial gap g for samples that would lead to pull-in. This quantity increases from $-0.9\ \mu m$ to $-1.0\ \mu m$ as the probability of pull-in increases from 0 at $V = 7.3$ V to 1 at $V = 11.4$ V, indicating that as the applied potential difference increases, samples with larger value of gap g also lead to pull-in.

Having obtained an approximate interpolation formula for tip displacement $\hat{v}_L(\xi)$, we can efficiently compute $\mu(v_L^n)$ and $\mu(v_L^p)$. We generate $N = 20,000$ realizations for ξ and obtain $\{v_L^j\}_{j=1}^N$. Then, $\mu(v_L^n)$ and $\mu(v_L^p)$ can simply be obtained as the mean of those samples which satisfy $(v_L^j > -0.9)$ and $(-1.1 \leqslant v_L^j \leqslant -0.9)$, respectively. Using this, in addition to computing the mean, we also compute the probability of attaining that mean, and the same has been shown in Fig. 24. We plot the mean $\mu(v_L^n)$ and $\mu(v_L^p)$ with error bars, based on standard deviation, in Fig. 25. We plotted the PDFs for vertical tip displacement obtained using stochastic collocation for various applied voltages in Fig. 23. As expected, for voltages $7.3\ \text{V} \leqslant V \leqslant 11.4\ \text{V}$ we obtain two peaks in the PDF, signifying a non-zero probability of pull-in. For $V > 11.4$ V, since all realizations would lead to pull-in, the vertical tip displacement would simply be uniformly distributed between $-0.9\ \mu m$ and $-1.1\ \mu m$.

6.3. Design under uncertainties

The objective of the stochastic analysis is twofold – firstly, to quantify the effect of various uncertain design parameters on relevant performance parameters. Secondly, to employ that uncertainty quantification information in order to design reliable and effective devices, which provide the desired performance. In [12] we concluded using the sensitivity analysis, that the variations in gap are more critical to the performance of the switch as compared to the variations in Young's modulus.

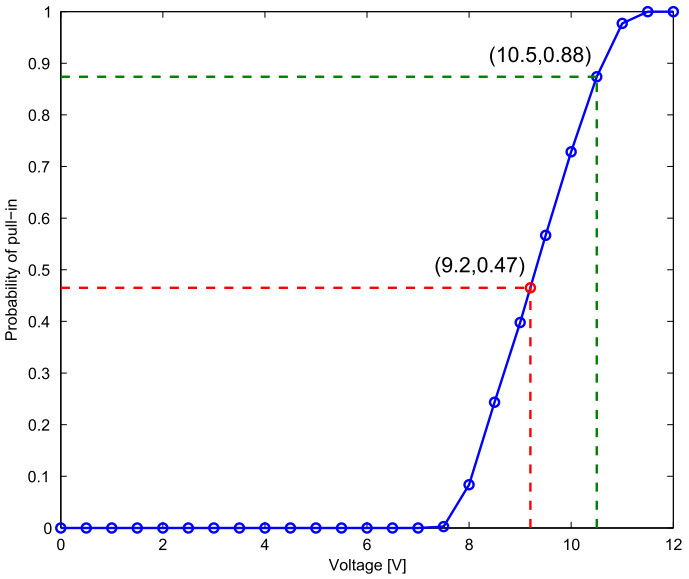


Fig. 26. Probability of pull-in curve.

Such observations can be used to identify critical design parameters, which can then be effectively controlled during fabrication.

In Fig. 26 we plot the probability of pull-in with various applied voltages obtained using the stochastic collocation approach. As mentioned earlier, as the applied voltage increases from 7.3 V to 11.4 V, the probability increases from 0 to 1. From a design viewpoint, we observe that the probability of pull-in for $V = 9.2$ V is 0.47. Using this, one can argue that if the switch was designed for the mean values of the parameters (given by the realization S_0), the probability that the switch may fail to operate is 53%. It can also be said that if the operating voltage of the switch is specified as $V = 10.5$ V, the probability that the switch will operate successfully is 88%, which may be acceptable. In the absence of such information, one may have no choice but to operate the switch at $V = 11.4$ V, which will lead to larger actuation voltage. Such observations can be effectively used to design reliable and efficient MEMS devices.

7. Conclusions

This work presented a domain adaptive stochastic collocation framework for uncertainty quantification, suitable for effectively handling discontinuities or sharp variations in the random domain. The developed framework was employed for stochastic modeling of the coupled electromechanical problem. The proposed stochastic collocation approach is based on an adaptive decomposition of the random domain into subdomains or elements. Within each random element, a coarse sparse grid interpolant is constructed using the classical Smolyak construction, which estimates the local error contribution and the most sensitive dimensions. Based on these error estimates, each random element is further subdivided, until the required error tolerance is achieved. We demonstrated through numerical examples that in the presence of discontinuities in the random domain, this approach leads to significant reduction in the number of support nodes required to achieve the same level of accuracy as the classical Smolyak construction. Moreover, similar to the classical approach, this framework also leads to a set of decoupled deterministic equations at each adaptive iteration, which can be solved in a parallel fashion.

We demonstrated that the developed framework can be effectively used to quantify the effect of uncertain parameters on the performance of electrostatic MEMS devices. Specifically, we considered a MEMS switch and studied the effect of uncertain Young's modulus and gap between the electrodes on the actuation properties and pull-in behavior of the switch. The results from the proposed approach were verified using MC simulations. We noted that in the presence of discontinuities, this approach provides the required statistics easily. Using the adaptive approach we resolved the discontinuity caused due to the pull-in instability in MEMS switches, which was not possible earlier. In addition to computing the statistics such as mean and standard deviation for the switch displacement, we also computed quantities such as probability of pull-in. Such observations can be used to design reliable and effective MEMS devices.

Acknowledgement

This work is supported by the National Science Foundation under Grant Number 0601479, by DARPA/MTO, and by DOE.

References

[1] S. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, Soviet Math. Dokl. 4 (1963) 240–243.
[2] S.D. Senturia, N.R. Aluru, J. White, Simulating the behavior of MEMS devices: computational methods and needs, IEEE Comput. Sci. Eng. 4 (1) (1997) 30–43.
[3] G. Li, N.R. Aluru, Efficient mixed-domain analysis of electrostatic MEMS, IEEE Trans. Comput. Aided Des. Integr. Circ. Syst. 22 (9) (2003) 1228–1242.
[4] S.K. De, N.R. Aluru, Full-Lagrangian schemes for dynamic analysis of electrostatic MEMS, J. Microelectromech. Syst. 13 (5) (2004) 737–758.
[5] S.K. De, N.R. Aluru, Coupling of hierarchical fluid models with electrostatic and mechanical models for the dynamic analysis of MEMS, J. Micromech. Microeng. 16 (8) (2006) 1705–1719.
[6] M.D. McKay, R.J. Beckman, W.J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 2 (1979) 239–245.
[7] H. Niederreiter, Random Number Generation and Quasi-Monte Carlo Methods, SIAM, Philadelphia, 1992.
[8] W. Gilks, S. Richardson, D. Spiergelhalter, Markov Chain Monte Carlo in Practice, Chapman & Hall, London, 1995.
[9] R.G. Ghanem, P. Spanos, Stochastic Finite Elements:A Spectral Approach, Springer, 1991.
[10] N. Wiener, The homogeneous chaos, Am. J. Math. 60 (1938) 897–936.
[11] D. Xiu, G.E. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations, SIAM J. Sci. Comput. 24 (2) (2002) 619–644.
[12] N. Agarwal, N.R. Aluru, Stochastic modeling of coupled electromechanical interaction for uncertainty quantification in electrostatically actuated MEMS, Comput. Methods Appl. Mech. Eng. 197 (43–44) (2008) 3456–3471.
[13] D. Xiu, J.S. Hesthaven, High-order collocation methods for differential equations with random inputs, SIAM J. Sci. Comput. 27 (3) (2005) 1118–1139.
[14] D. Xiu, Efficient collocational approach for parametric uncertainty analysis, Commun. Comput. Phys. 2 (2) (2007) 293–309.
[15] B. Ganapathysubramanian, N. Zabaras, Sparse grid collocation schemes for stochastic natural convection problems, J. Comput. Phys. 225 (2007) 652–685.
[16] O.P. LeMâitre, R.G. Ghanem, O.M. Knio, H.N. Najm, Uncertainty propagation using Wiener–Haar expansions, J. Comput. Phys. 197 (2004) 28–57.
[17] O.P. LeMâitre, H.N. Najm, R.G. Ghanem, O.M. Knio, Multi-resolution analysis of Wiener-type uncertainty propagation schemes, J. Comput. Phys. 197 (2004) 502–531.
[18] X. Wan, G.E. Karniadakis, An adaptive multi-element generalized polynomial chaos method for stochastic differential equations, J. Comput. Phys. 209 (2005) 617–642.
[19] X. Wan, G.E. Karniadakis, Multi-element generalized polynomial chaos for arbitrary probability measures, SIAM J. Sci. Comput. 28 (3) (2006) 901–928.
[20] T. Grestner, M. Griebel, Dimension adaptive tensor product quadrature, Computing 71 (2003) 65–87.
[21] J. Foo, X. Wan, G.E. Karniadakis, The multi-element probabilistic collocation method (ME-PCM): error analysis and applications, J. Comput. Phys. 227 (22) (2008) 9572–9595.
[22] X. Ma, N. Zabaras, An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations, J. Comput. Phys. 228 (8) (2009) 3084–3113.
[23] D.S. Chandrasekharaiah, L. Debnath, Continuum Mechanics, Academic, New York, 1994.
[24] G. Li, N.R. Aluru, A Lagrangian approach for electrostatic analysis of deformable conductors, J. Microelectromech. Syst. 11 (2002) 245–254. June.
[25] N.R. Aluru, J. White, An efficient numerical technique for electromechanical simulation of complicated microelectromechanical structures, Sens. Actuators, Phys. A 58 (1997) 1–11.
[26] M. Loève, Probability Theory, Springer, 1977.
[27] S. Sakamoto, R. Ghanem, Simulation of multi-dimensional non-Gaussian non-stationary random fields, Prob. Eng. Mech. 17 (2002) 162–176.
[28] S. Reh, P. Lethbridge, D.F. Ostergaard, Quality based design and design for reliability of micro electro mechanical systems (MEMS) using probabilistic methods, in: Proceedings of the 2000 International Conference on Modeling and Simulation of Microsystems, San Diego, CA, MSM, 2000.
[29] M. Deb, I. Babuska, J. Oden, Solution of stochastic partial differential equations using Galerkin finite element techniques, Comput. Methods Appl. Mech. Eng. 190 (2001) 6359–6372.
[30] I. Babuska, R. Tempone, G. Zourais, Galerkin finite element approximations of stochastic elliptic differential equations, SIAM J. Numer. Anal. 42 (2004) 800–825.
[31] G. Schueller, A state-of-the-art report on computational stochastic mechanics, Probab. Eng. Mech. 12 (1997) 197–321.
[32] G. Schueller, Computational stochastic mechanics – recent advances, Comput. Struct. 82 (2004) 1007–1020.
[33] D. Xiu, G.E. Karniadakis, Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos, Comput. Methods Appl. Mech. Eng. 191 (2002) 4927–4948.
[34] O.P. LeMâitre, O.M. Knio, H.N. Najm, R.G. Ghanem, A stochastic projection method for fluid flow I. Basic formulation, J. Comput. Phys. 173 (2001) 481–511.
[35] O.P. LeMâitre, M.T. Reagan, H.N. Najm, R.G. Ghanem, O.M. Knio, A stochastic projection method for fluid flow II. Random process, J. Comput. Phys. 181 (2002) 9–44.
[36] D. Xiu, G.E. Karniadakis, A new stochastic approach to transient heat conduction modeling with uncertainty, Int. J. Heat Mass Trans. 46 (2003) 4681–4693.
[37] R.G. Ghanem, Higher order sensitivity of heat conduction problems to random data using the spectral stochastic finite element method, ASME J. Heat Trans. 121 (1999) 290–299.
[38] N. Agarwal, N.R. Aluru, A stochastic Lagrangian approach for geometrical uncertainties in electrostatics, J. Comput. Phys. 226 (1) (2007) 156–179.
[39] S. Acharjee, N. Zabaras, Uncertainty propagation in finite deformations – a spectral stochastic Lagrangian approach, Comput. Methods Appl. Mech. Eng. 195 (2006) 2289–2312.
[40] Q.Y. Chen, D. Gottlieb, J. Hesthaven, Uncertainty analysis for the steady-state flows in a dual throat nozzle, J. Comput. Phys. 204 (2005) 387–398.
[41] B.J. Debusschere, H.N. Najm, A. Matta, O.M. Knio, R.G. Ghanem, O.P. LeMâitre, Protein labeling reactions in electrochemical microchannel flow: numerical simulation and uncertainty propagation, Phys. Fluids 15 (8) (2003) 2238–2250.
[42] B.J. Debusschere, H.N. Najm, P.P. Pebay, O.M. Knio, R.G. Ghanem, O.P. LeMâitre, Numerical challenges in the use of polynomial chaos representations for stochastic processes, SIAM J. Sci. Comput. 26 (2) (2004) 698–719.
[43] A. Klimke, B. Wholmuth, Algorithm 847: spinterp: piecewise multilinear hierarchical sparse grid interpolation in MATLAB, ACM Trans. Math. Software 31 (4) (2005) 561–579.
[44] V. Barthelmann, E. Novak, K. Ritter, High dimensional polynomial interpolation on sparse grid, Adv. Comput. Math. 12 (1999) 273–288.
[45] A. Klimke, Uncertainty Modeling Using Fuzzy Arithmetic and Sparse Grids, Ph.D. Thesis, Universität Stuttgart, Shaker Verlag, Aachen, 2006.
[46] E. Novak, K. Ritter, High dimensional integration of smooth functions over cubes, Numer. Math. 75 (1996) 79–97.
[47] E. Novak, K. Ritter, Simple cubature formulas with high polynomial exactness, Construct. Approx. 15 (1999) 499–522.
[48] A. Klimke, Sparse Grid Interpolation Toolbox – User's Guide, University of Stuttgart, IANS Report 2006/01, 2006.