

MULTI-ELEMENT GENERALIZED POLYNOMIAL CHAOS FOR ARBITRARY PROBABILITY MEASURES*

XIAOLIANG WAN[†] AND GEORGE EM KARNIADAKIS[†]

Abstract. We develop a multi-element generalized polynomial chaos (ME-gPC) method for arbitrary probability measures and apply it to solve ordinary and partial differential equations with stochastic inputs. Given a stochastic input with an arbitrary probability measure, its random space is decomposed into smaller elements. Subsequently, in each element a new random variable with respect to a conditional probability density function (PDF) is defined, and a set of orthogonal polynomials in terms of this random variable is constructed numerically. Then, the generalized polynomial chaos (gPC) method is implemented element-by-element. Numerical experiments show that the cost for the construction of orthogonal polynomials is negligible compared to the total time cost. Efficiency and convergence of ME-gPC are studied numerically by considering some commonly used random variables. ME-gPC provides an efficient and flexible approach to solving differential equations with random inputs, especially for problems related to long-term integration, large perturbation, and stochastic discontinuities.

Key words. uncertainty, polynomial chaos, stochastic differential equations, random inputs

AMS subject classifications. 65C20, 65C30

DOI. 10.1137/050627630

1. Introduction. *Polynomial chaos* (PC) has been used extensively recently to model uncertainty in physical applications. It originated from *homogenous chaos* first defined by Wiener as the span of Hermite polynomial functionals of a Gaussian process [31]. Other names, such as Wiener-chaos and Hermite-chaos, are also used in literature. Ghanem and Spanos first combined Hermite-chaos with a finite element method to model uncertainty in solid mechanics [14, 12, 13]. A more general extension, termed *generalized polynomial chaos* (gPC), was proposed in [32] by Xiu and Karniadakis based on the correspondence between the probability density functions (PDFs) of certain random variables and the weight functions of orthogonal polynomials of the Askey scheme. The family of gPC includes Hermite-chaos as a subset and supplies optimal bases for stochastic processes represented by random variables of commonly used distributions, such as uniform distribution and Beta distribution. In [18, 19], PC was combined with wavelets to deal with discontinuities for uniform random inputs for which standard PC or gPC fails to converge.

To solve differential equations with stochastic inputs following the procedure established by Ghanem and Spanos, the random solution is expanded spectrally by PC, and a Galerkin projection scheme is subsequently used to transform the original stochastic problem into a deterministic one with a larger dimensional parameter [14, 32, 7, 21, 20]. On the other hand, Deb, Babuška, and Oden have proposed employing finite elements in the random space to approximate the stochastic dependence of the solution [4]; this approach also reduces a stochastic differential equation to a high-dimensional deterministic one. This method was later studied theoretically within the framework of a deterministic finite element method in [1]. Since a finite

*Received by the editors March 25, 2005; accepted for publication (in revised form) December 13, 2005; published electronically June 19, 2006. This work was supported by NSF, AFOSR, and ONR.
<http://www.siam.org/journals/sisc/28-3/62763.html>

[†]Division of Applied Mathematics, Brown University, Providence, RI 02912 (xlwan@dam.brown.edu, gk@dam.brown.edu).

element method is generally used to solve the obtained deterministic PDE system, the above methods are called *stochastic Galerkin finite element methods* in [1], while the scheme in [14, 32] is classified as the $p \times h$ version and the scheme in [4] as the $k \times h$ version, where p denotes the polynomial order of PC, k the element size in the random space, and h the element size in the physical space. Both schemes use finite elements in the physical space. The $p \times h$ version relies on the spectral representation in the entire random space by PC, while the $k \times h$ version is based on the discretization of the random space using the same basis as the deterministic finite element method to approximate the random field locally. For simplicity, here we refer to the scheme in [14, 32] as a PC method of p -version and the scheme in [4] as a k -version. Both the concepts and the terminology introduced here share similarities with those of the spectral/ hp element method for deterministic problems [17, 25].

In this work, we present a multi-element generalized polynomial chaos (ME-gPC) method, which can deal with stochastic processes represented by random variables of arbitrary distributions and achieve kp -convergence in the random space. In the ME-gPC method, we discretize the random space and use a spectral expansion locally to represent the desired random field. However, the PC basis is, in general, not orthogonal in a random element since the PDF is also discretized simultaneously. The only exception is the uniform distribution which has been considered so far in previous works [14, 4, 32, 28]. To overcome this difficulty, we reconstruct the orthogonal basis numerically in each random element. From the practical standpoint, it is more effective to employ a relatively low polynomial order ($p = 5$ to $p = 8$) for the ME-gPC method. Thus, we can do this reconstruction *on-the-fly* with high accuracy for most continuous or discrete arbitrary probability measures.

In certain nonlinear stochastic differential equations, random inputs can give rise to singularities as discussed in [18, 19] or the Kraichnan–Orszag problem [24], where the gPC method may converge very slowly or even fail to converge. By noting that singularities usually occur at a lower-dimensional manifold in the random space for such problems, we aim to “capture” the singular region by random elements related to a very small probability measure, and implement the gPC method in the regular region. Hence, the error in the singular region will be dominated by the size (k -convergence) of the random elements, while the error in the regular region decreases fast (p -convergence)—possibly exponentially fast. To reduce the total computational cost, we also present a criterion to implement the decomposition of random space adaptively.

2. Multi-element generalized polynomial chaos (ME-gPC) method.

2.1. Overview. The original PC was first proposed by Wiener [31]. It employs the Hermite polynomials in terms of Gaussian random variables as the trial basis to expand stochastic processes in the random space. According to the theorem by Cameron and Martin [3] such expansion converges for any second-order processes in the L_2 sense. gPC was proposed in [33] and employs more types of orthogonal polynomials from the Askey family. It is a generalization of the original Wiener’s Hermite-chaos and can deal with non-Gaussian random inputs more efficiently.

Let (Ω, \mathcal{F}, P) be a probability space, where Ω is the sample space, \mathcal{F} is the σ -algebra of subsets of Ω , and P is a probability measure. Let

$$(2.1) \quad \boldsymbol{\xi} = (\xi_1, \dots, \xi_d) : (\Omega, \mathcal{F}) \rightarrow (\mathbb{R}^d, \mathcal{B}^d)$$

be an \mathbb{R}^d -valued continuous random variable, where $d \in \mathbb{N}$ and \mathcal{B}^d is the σ -algebra of Borel subsets of \mathbb{R}^d . A general second-order random process $R(\omega) \in L_2(\Omega, \mathcal{F}, P)$ can

be expressed by gPC as

$$(2.2) \quad R(\omega) = \sum_{i=0}^{\infty} \hat{a}_i \Phi_i(\boldsymbol{\xi}(\omega)),$$

where ω is the random event and $\Phi_i(\boldsymbol{\xi}(\omega))$ denotes the gPC basis of degree p in terms of the random variable $\boldsymbol{\xi}$. The family $\{\Phi_i\}$ is an orthogonal basis in $L_2(\Omega, \mathcal{F}, P)$ with orthogonality relation

$$(2.3) \quad \mathbb{E}[\Phi_i \Phi_j] = \mathbb{E}[\Phi_i^2] \delta_{ij},$$

where δ_{ij} is the Kronecker delta and \mathbb{E} denotes the expectation with respect to the probability measure $dP(\omega) = f(\boldsymbol{\xi}(\omega))d\omega$. The index in (2.2) and $d \in \mathbb{N}$ are, in general, infinite. In practice, both limits will be truncated at a certain level. For example, the dimension d of the stochastic input is usually determined by the Karhunen–Loeve (K-L) expansion based on the decay rate of eigenvalues [14].

For a certain random variable $\boldsymbol{\xi}$, the orthogonal gPC basis $\{\Phi_i\}$ can be chosen in such a way that its weight function has the same form as the PDF $f(\boldsymbol{\xi})$ of $\boldsymbol{\xi}$. The corresponding types of classical orthogonal polynomials $\{\Phi_i\}$ and their associated random variable $\boldsymbol{\xi}$ can be found in [32]. In this paper we generalize the idea of gPC and construct numerically the orthogonal polynomials with respect to an *arbitrary* probability measure.

2.2. ME-gPC formulation. We next present the scheme of an adaptive ME-gPC method, which consists of the decomposition of random space, the construction of orthogonal polynomials, and an adaptive procedure.

2.2.1. Decomposition of random space. We assume that $\boldsymbol{\xi}$ is a random variable defined on $B = \times_{i=1}^d [a_i, b_i]$, where a_i and b_i are finite or infinite in \mathbb{R} and the components of $\boldsymbol{\xi}$ are independent identically-distributed (i.i.d.) random variables. We define a decomposition \mathbf{D} of B as

$$(2.4) \quad \mathbf{D} = \begin{cases} B_k = [a_{k,1}, b_{k,1}) \times [a_{k,2}, b_{k,2}) \times \cdots \times [a_{k,d}, b_{k,d}), \\ B = \bigcup_{k=1}^N B_k, \\ B_{k_1} \cap B_{k_2} = \emptyset \text{ if } k_1 \neq k_2, \end{cases}$$

where $k, k_1, k_2 = 1, 2, \dots, N$. Based on the decomposition \mathbf{D} , we define the following indicator random variables:

$$(2.5) \quad I_{B_k} = \begin{cases} 1 & \text{if } \boldsymbol{\xi} \in B_k, \\ 0 & \text{otherwise.} \end{cases} \quad k = 1, 2, \dots, N.$$

Thus, $\Omega = \cup_{k=1}^N I_{B_k}^{-1}(1)$ is a decomposition of the sample space Ω , where

$$(2.6) \quad I_{B_i}^{-1}(1) \cap I_{B_j}^{-1}(1) = \emptyset \text{ for } i \neq j.$$

Given any point $\mathbf{q} = (q_1, q_2, \dots, q_d)$, we use $\boldsymbol{\xi} \leq \mathbf{q}$ to denote $\xi_i \leq q_i$ for $i = 1, 2, \dots, d$. According to the law of total probability, we can obtain

$$(2.7) \quad \Pr(\boldsymbol{\xi} \leq \mathbf{q}) = \sum_{k=1}^N \Pr(\boldsymbol{\xi} \leq \mathbf{q} | I_{B_k} = 1) \Pr(I_{B_k} = 1).$$

Using Bayes’s rule, (2.7) implies that we can define a new random variable $\boldsymbol{\xi}_k : I_{B_k}^{-1}(1) \mapsto B_k$ on the probability space $(I_{B_k}^{-1}(1), \mathcal{F} \cap I_{B_k}^{-1}(1), P(\cdot | I_{B_k} = 1))$ subject to a conditional PDF

$$(2.8) \quad \hat{f}_k(\boldsymbol{\xi}_k | I_{B_k} = 1) = \frac{f(\boldsymbol{\xi}_k)}{\Pr(I_{B_k} = 1)}$$

in each random element B_k , where $\Pr(I_{B_k} = 1) > 0$.

Let $u(\boldsymbol{x}, t; \boldsymbol{\xi}) \in L_2(\Omega, \mathcal{F}, P)$ denote a *second-order* space-time related random field. For simplicity, we may drop \boldsymbol{x} and t .

PROPOSITION 2.1. *Let $\mathcal{P}_M u(\boldsymbol{\xi})$ denote the Galerkin projection of $u(\boldsymbol{\xi})$ onto the polynomial chaos basis $\{\Phi_i(\boldsymbol{\xi})\}$ up to polynomial order M , $i = 1, 2, \dots, M$. If $\mathcal{P}_M u(\boldsymbol{\xi})$ converges to $u(\boldsymbol{\xi})$ in the L_2 sense with respect to the PDF $f(\boldsymbol{\xi})$, then $\mathcal{P}_M u(\boldsymbol{\xi}_k)$ converges to $u(\boldsymbol{\xi}_k)$ in the L_2 sense with respect to the conditional PDF $\hat{f}_k(\boldsymbol{\xi}_k | I_{B_k} = 1)$, $k = 1, 2, \dots, N$.*

Proof. According to the assumption, we know that

$$\mathbb{E}[(u(\boldsymbol{\xi}) - \mathcal{P}_M u(\boldsymbol{\xi}))^2] = \int_B (u(\boldsymbol{\xi}) - \mathcal{P}_M u(\boldsymbol{\xi}))^2 f(\boldsymbol{\xi}) d\boldsymbol{\xi} \rightarrow 0 \quad \text{as } M \rightarrow \infty.$$

By noting

$$(2.9) \quad \mathbb{E}[(u(\boldsymbol{\xi}) - \mathcal{P}_M u(\boldsymbol{\xi}))^2] = \sum_{k=1}^N \Pr(I_{B_k} = 1) \int_{B_k} (u(\boldsymbol{\xi}_k) - \mathcal{P}_M u(\boldsymbol{\xi}_k))^2 \hat{f}_k(\boldsymbol{\xi}_k | I_{B_k} = 1) d\boldsymbol{\xi}_k,$$

we obtain

$$\int_{B_k} (u(\boldsymbol{\xi}_k) - \mathcal{P}_M u(\boldsymbol{\xi}_k))^2 \hat{f}_k(\boldsymbol{\xi}_k | I_{B_k} = 1) d\boldsymbol{\xi}_k \rightarrow 0 \quad \text{as } M \rightarrow \infty,$$

since both $\Pr(I_{B_k} = 1)$ and the integrand on the right-hand side of (2.9) are positive for $k = 1, 2, \dots, N$. \square

To this end, we know that there exists a local polynomial approximation for each random element B_k , which converges in the L_2 sense with respect to the local conditional PDF. For the orthogonal basis $\{\Phi_i(\boldsymbol{\xi})\}$ on the entire random space, $u(\boldsymbol{\xi}) - \mathcal{P}_M u(\boldsymbol{\xi})$ is orthogonal to the space $V(M, \boldsymbol{\xi}) := \text{span}\{\Phi_i(\boldsymbol{\xi}) : i \leq M\}$. We note here that although $\mathcal{P}_M u(\boldsymbol{\xi}_k)$ converges to $u(\boldsymbol{\xi}_k)$ in the L_2 sense within random element B_k , $u(\boldsymbol{\xi}_k) - \mathcal{P}_M u(\boldsymbol{\xi}_k)$ is not orthogonal to the space $V(M, \boldsymbol{\xi}_k)$ with respect to the conditional PDF $\hat{f}_k(\cdot | I_{B_k} = 1)$, since the orthogonality $\mathbb{E}[\Phi_i \Phi_j] = \delta_{ij}$ is, in general, valid only on the entire random space of $\boldsymbol{\xi}$ with respect to the PDF $f(\boldsymbol{\xi})$. Due to the efficiency of orthogonal polynomials in representing stochastic processes, we will reconstruct the local polynomial chaos modes numerically to make them mutually orthogonal with respect to the local conditional PDF $\hat{f}_k(\boldsymbol{\xi}_k | I_{B_k} = 1)$. According to [26, theorem 2.1.1], such an orthogonal polynomial system $\{\Phi_{k,i}(\boldsymbol{\xi}_k)\}$ always exists. Since in each random element we perform a spectral expansion as in gPC, we call this method “*multi-element generalized polynomial chaos*” (ME-gPC).

To approximate a random field $u(\boldsymbol{\xi})$ using ME-gPC, we expand the random field spectrally in each element B_k , then reconstruct the entire random field by the following proposition.

PROPOSITION 2.2. Let $\hat{u}_k(\boldsymbol{\xi}_k)$ be the local polynomial chaos expansion in element B_k . The approximation on the entire random field can be defined as

$$(2.10) \quad u^r(\boldsymbol{\xi}) = \sum_{k=1}^N \hat{u}_k(\boldsymbol{\xi}) I_{B_k} = \sum_{k=1}^N \sum_{j=0}^M \hat{u}_{k,j} \Phi_{k,j}(\boldsymbol{\xi}) I_{B_k},$$

which converges to $u(\boldsymbol{\xi})$ in the L_2 sense; in other words,

$$(2.11) \quad \int_B (u^r(\boldsymbol{\xi}) - u(\boldsymbol{\xi}))^2 f(\boldsymbol{\xi}) d\boldsymbol{\xi} \rightarrow 0 \quad \text{as } M \rightarrow \infty.$$

Proof. It is easy to see that

$$\begin{aligned} & \int_B (u^r(\boldsymbol{\xi}) - u(\boldsymbol{\xi}))^2 f(\boldsymbol{\xi}) d\boldsymbol{\xi} \\ &= \sum_{k=1}^N \Pr(I_{B_k} = 1) \int_{B_k} (u^r(\boldsymbol{\xi}_k) - u(\boldsymbol{\xi}_k))^2 \hat{f}_k(\boldsymbol{\xi}_k | I_{B_k} = 1) d\boldsymbol{\xi}_k. \end{aligned}$$

Since $u^r(\boldsymbol{\xi}_k) = \hat{u}_k(\boldsymbol{\xi}_k)$ and the spectral expansion $\hat{u}_k(\boldsymbol{\xi}_k)$ converges locally in the L_2 sense with respect to $\hat{f}_k(\boldsymbol{\xi}_k | I_{B_k} = 1)$, it is clear that the right-hand side of the above equation goes to zero as $M \rightarrow \infty$. The conclusion follows immediately. \square

By Bayes's rule and the law of total probability, any statistics can be obtained as

$$(2.12) \quad \int_B g(u(\boldsymbol{\xi})) f(\boldsymbol{\xi}) d\boldsymbol{\xi} \approx \sum_{k=1}^N \Pr(I_{B_k} = 1) \int_{B_k} g(\hat{u}_k(\boldsymbol{\xi}_k)) \hat{f}_k(\boldsymbol{\xi}_k | I_{B_k} = 1) d\boldsymbol{\xi}_k,$$

where $g(\cdot) \in L_1(\Omega, \mathcal{F}, P)$ is a functional of random field $u(\boldsymbol{\xi})$.

In the appendix, we demonstrate for the interested reader some details on how to solve PDEs with stochastic coefficients using ME-gPC.

2.2.2. Construction of orthogonal polynomials. We next discuss the numerical construction of orthogonal polynomials with respect to a conditional PDF $\hat{f}_k(\cdot | I_{B_k} = 1)$. For simplicity, here we discuss only the construction of one-dimensional orthogonal polynomials, since the high-dimensional basis can be obtained using tensor products of one-dimensional basis.

It is a distinctive feature of orthogonal polynomials, compared to other orthogonal systems, that they satisfy a three-term recurrence relation,

$$(2.13) \quad \begin{aligned} \pi_{i+1}(\tau) &= (\tau - \alpha_i)\pi_i(\tau) - \beta_i\pi_{i-1}(\tau), \quad i = 0, 1, \dots, \\ \pi_0(\tau) &= 1, \quad \pi_{-1}(\tau) = 0, \end{aligned}$$

where $\{\pi_i(\tau)\}$ is a set of (monic) orthogonal polynomials,

$$(2.14) \quad \pi_i(\tau) = \tau^i + \text{lower-degree terms}, \quad i = 0, 1, \dots,$$

and the coefficients α_i and β_i are uniquely determined by a positive measure, which corresponds to a probability measure in the construction we propose herein.

For a continuous measure $m(\tau)$ there are two classical methods to compute the recurrence coefficients α_i and β_i : the Stieltjes procedure and the modified Chebyshev

algorithm [8]. The Stieltjes procedure uses the fact that the coefficients α_i and β_i can be expressed by the following simple formulas:

$$(2.15) \quad \alpha_i = \frac{(\tau\pi_i, \pi_i)}{(\pi_i, \pi_i)}, \quad i = 0, 1, 2, \dots$$

and

$$(2.16) \quad \beta_0 = (\pi_0, \pi_0), \quad \beta_i = \frac{(\pi_i, \pi_i)}{(\pi_{i-1}, \pi_{i-1})}, \quad i = 1, 2, \dots,$$

where (\cdot, \cdot) denotes the inner product in terms of the measure $m(\tau)$. The above two formulas together with the recurrence relation (2.13) can be used to calculate recursively as many coefficients α_i and β_i as desired.

The modified Chebyshev algorithm is a generalization of the Chebyshev algorithm [8]. The Chebyshev algorithm relies on the fact that the first n pairs of recurrence coefficients α_i and β_i , $i = 0, 1, \dots, n - 1$, can be uniquely determined by the first $2n$ moments μ_i :

$$(2.17) \quad \mu_i = \int_B \tau^i dm(\tau), \quad i = 0, 1, \dots, 2n - 1.$$

Analytical formulas are known which express α_i and β_i in terms of Hankel determinants in these moments. However, this algorithm is not reliable for a big n due to the increasing sensitivity of these formulas to small errors. The modified Chebyshev algorithm replaces the power τ^i with a properly chosen polynomial $h_i(\tau)$ of degree i :

$$(2.18) \quad \nu_i = \int_B h_i(\tau) dm(\tau), \quad i = 0, 1, \dots, 2n - 1.$$

Generally, we can assume that $h_i(\tau)$ are monic orthogonal polynomials satisfying a three-term relation

$$(2.19) \quad \begin{aligned} h_{i+1}(\tau) &= (\tau - \hat{\alpha}_i)h_i(\tau) - \hat{\beta}_i h_{i-1}(\tau), \quad i = 0, 1, \dots, \\ h_0(\tau) &= 1, \quad h_{-1}(\tau) = 0. \end{aligned}$$

Using the $2n$ modified moments in (2.18) and the $2n - 1$ pairs of recurrence coefficients $\hat{\alpha}_i$ and $\hat{\beta}_i$, $i = 0, 1, \dots, 2n - 2$, in (2.19), the first n desired pairs of recurrence coefficients α_i and β_i , $i = 0, 1, \dots, n - 1$, can be generated [8].

For a discrete measure

$$(2.20) \quad dm_M(\tau) = \sum_{i=1}^M w_i \delta(\tau - \tau_i) d\tau, \quad i = 0, 1, \dots, M$$

with δ being the Dirac delta function, we have another choice: the Lanczos algorithm [2, 8]. Given (2.20), there exists an orthogonal matrix $Q^{(M+1) \times (M+1)}$ with the first column being $[1, 0, \dots, 0]^T \in \mathbb{R}^{(M+1) \times 1}$ such that

$$(2.21) \quad Q^T A_M Q = J_M,$$

where

$$(2.22) \quad A_M = \begin{bmatrix} 1 & \sqrt{w_1} & \sqrt{w_2} & \cdots & \sqrt{w_M} \\ \sqrt{w_1} & \tau_1 & 0 & \cdots & 0 \\ \sqrt{w_2} & 0 & \tau_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sqrt{w_M} & 0 & 0 & \cdots & \tau_M \end{bmatrix},$$

and J_M is the Jacobian matrix

$$(2.23) \quad J_M = \begin{bmatrix} 1 & \sqrt{\beta_0} & 0 & \cdots & 0 \\ \sqrt{\beta_0} & \alpha_0 & \sqrt{\beta_1} & \cdots & 0 \\ 0 & \sqrt{\beta_1} & \alpha_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \alpha_{M-1} \end{bmatrix}.$$

The above algorithms were presented in [8], and a Fortran package based on these algorithms was proposed in [9]. To implement the Stieltjes procedure and the modified Chebyshev algorithm we need to evaluate the inner product with high precision. In practice, we usually employ a Gauss-type quadrature rule. Such a quadrature rule can be regarded as a discrete measure that yields the corresponding discrete versions of these two methods [8]. The stability of the Stieltjes procedure is not completely clear. One interesting aspect about this procedure is that it is reliable in many cases where a discretization for the inner product of a continuous measure is employed, while it breaks down in some cases where the measure itself is discrete. In [6], the Stieltjes procedure for discrete measures is discussed and improved. The stability of the modified Chebyshev algorithm is determined by the condition of the map from the $2n$ modified moments to the n pairs of recurrence coefficients. Such a condition was studied in [8, 5]. The Lanczos algorithm has good stability properties but it may be considerably slower than the Stieltjes procedure. In this work, we use the Stieltjes procedure and the Lanczos algorithm.

It can be seen that the basic operation in the above procedures is the inner product with respect to a given measure $m(\tau)$. Since an explicit formula is generally unavailable, we discretize the inner product using an appropriate quadrature rule. We know that an n -point Gauss-type quadrature rule can reach the highest algebraic accuracy of order $2n - 1$. However, such quadrature points and the corresponding weights are not in general explicitly known. In this work, we use the interpolatory quadrature rule [23] relative to the Legendre weight function on $[-1, 1]$. An n -point interpolatory quadrature rule has an algebraic precision $n - 1$. Compared to the Gauss–Legendre quadrature rule, the advantage of the interpolatory quadrature rule is that the nodes and weights are explicitly known; thus the expensive Newton–Raphson iterations are avoided.

Using the Stieltjes procedure or the Lanczos algorithm, the recurrence coefficients α_i and β_i are computed iteratively using the following stopping criterion [8]:

$$(2.24) \quad |\beta_i^s - \beta_i^{s-1}| \leq \epsilon \beta_i^s, \quad i = 0, 1, \dots, n - 1,$$

where s is the iteration step and ϵ is the relative error.

If a measure $m(\tau)$ is given and orthogonal polynomials up to p th order are needed, we first compute the recurrence coefficients α_i and β_i , $i = 0, 1, \dots, p - 1$. Once the recurrence coefficients are obtained, the orthogonal polynomials are uniquely determined. From the recurrence coefficients, the Gauss-type quadrature points and the corresponding integration weights can be efficiently derived [9].

It is well known that the gPC method usually relies on a three-dimensional table $\mathbb{E}[\pi_l \pi_m \pi_n]$ for the Galerkin projection [14]. Here we use the notation $\mathbb{E}[\cdot]$ freely to denote the ensemble average with respect to a given measure $m(\tau)$ since $m(\tau)$ can be regarded as a probability measure in this work. If the recurrence coefficients are obtained—in other words, the orthogonal polynomials are uniquely determined—we

can use any numerical integration formula to calculate the table $\mathbb{E}[\pi_l \pi_m \pi_n]$. Here we consider the quadrature rule. We know that for the measure $m(\tau)$ there exists, for each $M \in \mathbb{N}$, a quadrature rule

$$(2.25) \quad \int_B f(\tau) d\sigma(\tau) = \sum_{i=1}^M w_i f(\tau_i) + R_M(f),$$

where $R_M(f) = 0$ if f is a polynomial of degree $\leq 2M - c$. The value of c is determined by the type of quadrature used, which can be either classical Gauss ($c = 1$), Gauss–Radau ($c = 2$), or Gauss–Lobatto ($c = 3$). If we do not consider the numerical accuracy of the recurrence coefficients, we need

$$(2.26) \quad M = \left\lceil \frac{3p + c}{2} \right\rceil$$

quadrature points to get the three-term integration $\mathbb{E}[\pi_l \pi_m \pi_n]$ with zero error, where $\lceil * \rceil$ denotes the smallest integer no less than $*$. Thus, we need to compute all the recurrence coefficients α_i and β_i with $i \leq M - 1$ although we employ polynomials only up to order p .

2.2.3. An adaptive procedure. Adaptivity is necessary for many cases, such as problems related to long-term integration or discontinuity. In this work we present a heuristic adaptivity criterion based on the relative local errors similar to the adaptivity criteria employed in the spectral element method for deterministic problems [22, 15].

We assume that the gPC expansion of a random field in element k is

$$(2.27) \quad \hat{u}_k(\boldsymbol{\xi}_k) = \sum_{j=0}^{N_p} \hat{u}_{k,j} \Phi_{k,j}(\boldsymbol{\xi}_k),$$

where p is the highest order of PC and N_p denotes the total number of basis modes given by

$$(2.28) \quad N_p = \frac{(p + d)!}{p!d!} - 1,$$

where d is the dimension of $\boldsymbol{\xi}_k$. From the orthogonality of gPC we can easily obtain the local variance given by PC with order p :

$$(2.29) \quad \sigma_{k,p}^2 = \sum_{j=1}^{N_p} \hat{u}_{k,j}^2 \mathbb{E}[\Phi_{k,j}^2].$$

The *approximate* global mean \bar{u} and variance $\bar{\sigma}^2$ can be expressed as

$$(2.30) \quad \bar{u} = \sum_{k=1}^N \hat{u}_{k,0} \Pr(I_{B_k} = 1), \quad \bar{\sigma}^2 = \sum_{k=1}^N \left[\sigma_{k,p}^2 + (\hat{u}_{k,0} - \bar{u})^2 \right] \Pr(I_{B_k} = 1).$$

Compared to the error of variance, the error of mean is usually much smaller in the gPC method. Thus, we can express the *exact* global variance as

$$(2.31) \quad \sigma^2 \approx \bar{\sigma}^2 + \sum_{k=1}^N \gamma_k \Pr(I_{B_k} = 1),$$

where η_k denotes the error of local variance. We define the decay rate of relative error of PC in each element as follows:

$$(2.32) \quad \eta_k = \frac{\sum_{i=N_{p-1}+1}^{N_p} \hat{u}_{k,i}^2 \mathbb{E}[\Phi_{k,i}^2]}{\sigma_{k,p}^2}.$$

Based on η_k and the scaled parameter $\Pr(I_{B_k} = 1)$, we implement *k-type refinement*, in other words, decompose the current random element into smaller ones, if the criterion

$$(2.33) \quad \eta_k^\gamma \Pr(I_{B_k} = 1) \geq \theta_1, \quad 0 < \gamma < 1,$$

is satisfied, where γ and θ_1 are prescribed constants. A discussion about γ and θ_1 can be found in [28] for random inputs with *uniform* distributions.

The error of the variance should decrease exponentially (for continuous random inputs) given sufficient smoothness of the solutions; furthermore, the error of $\hat{u}_{k,0}$ is usually much smaller than the error of local variance. Thus, we can use η_k as an indicator of the true error for any orthogonal chaos basis, and the criterion (2.33) is a general one. For high-dimensional random inputs, we can choose the most sensitive random dimensions for refinement. We define the sensitivity for each random dimension as

$$(2.34) \quad r_i = \frac{(\hat{u}_{i,p})^2 \mathbb{E}[\Phi_{i,p}^2]}{\sum_{j=N_{p-1}+1}^{N_p} \hat{u}_j^2 \mathbb{E}[\Phi_j^2]}, \quad i = 1, 2, \dots, d,$$

where we drop the subscript k for clarity and the subscript $_{*,i,p}$ denotes the one-dimensional mode of ξ_i with polynomial order p . Let us consider a d -dimensional mode $\phi_{(j_1, j_2, \dots, j_d)}(\boldsymbol{\xi}) = \phi_{j_1}(\xi_1)\phi_{j_2}(\xi_2)\cdots\phi_{j_d}(\xi_d)$ with $\sum_{n=1}^d j_n \leq p$, which is a tensor product of one-dimensional modes $\phi_{j_n}(\xi_n)$, $n = 1, 2, \dots, d$. The subscript $_{*,i,p}$ means that $\phi_{i,p}(\boldsymbol{\xi}) = \phi_p(\xi_i)$, $i = 1, 2, \dots, d$. All random dimensions which satisfy

$$(2.35) \quad r_i \geq \theta_2 \cdot \max_{j=1, \dots, d} r_j, \quad 0 < \theta_2 < 1, \quad i = 1, 2, \dots, d$$

will be split into two *equal* random elements in the next time step while all other random dimensions will remain unchanged. Hence, we can reduce the total element number while gaining efficiency.

2.2.4. Implementation. Next we discuss issues related to the numerical implementation. Based on the decomposition of random space, it is straightforward to reconstruct orthogonal polynomials with respect to the conditional PDFs of the local random variables $\boldsymbol{\xi}_k$. However, such a procedure is not robust in practice because (π_i, π_i) may decay fast and cause a potential problem of overflow or underflow in the computer.

PROPOSITION 2.3. *For any orthogonal polynomials defined on $[a, b]$, where a and b are positive, we have*

$$(2.36) \quad (\pi_{i+1}, \pi_{i+1}) \leq (b - a)^2 (\pi_i, \pi_i).$$

Proof. From (2.15), we can obtain that $a \leq \alpha_i \leq b$. Using (2.13) and the Cauchy-Schwarz inequality, we know that

$$(2.37) \quad (\pi_{i+1}, \pi_{i+1}) = ((\tau - \alpha_i)\pi_i, \pi_{i+1}) \leq ((\tau - \alpha_i)\pi_i, (\tau - \alpha_i)\pi_i)^{1/2} (\pi_{i+1}, \pi_{i+1})^{1/2},$$

which yields that

$$(2.38) \quad (\pi_{i+1}, \pi_{i+1}) \leq ((\tau - \alpha_i)\pi_i, (\tau - \alpha_i)\pi_i).$$

The proof concludes by noting that $a \leq \alpha_i \leq b$. \square

Thus, if the element is small, the L_2 norm of π_i will lead to underflow quickly. In practice, we rescale the random elements with *finite* boundaries by the following linear transformation:

$$(2.39) \quad \xi_{k,i} = \frac{b_{k,i} - a_{k,i}}{2} Y_{k,i} + \frac{b_{k,i} + a_{k,i}}{2},$$

where we map the random variable ξ_k defined in element k to a random variable \mathbf{Y}_k defined in $[-1, 1]^d$. The PDF of \mathbf{Y}_k can be obtained as

$$(2.40) \quad \bar{f}(\mathbf{y}_k) = \det \left| \frac{\partial \xi_k}{\partial \mathbf{y}_k} \right| \hat{f}(\xi_k(\mathbf{y}_k)) = \frac{f(\xi_k(\mathbf{y}_k))}{\Pr(I_{B_k} = 1)} \prod_{i=1}^d \frac{b_{k,i} - a_{k,i}}{2}.$$

Compared to ξ_k , \mathbf{Y}_k are much more tractable for the numerical construction of orthogonal polynomials as demonstrated in section 3.1.1. After such a transformation is employed, we can apply the ME-gPC scheme with respect to the new random variable \mathbf{Y}_k instead of the random variable ξ_k . Note that such a mapping is usually unnecessary for random elements with at least one *infinite* boundary because these elements can be related to a small probability $\Pr(I_{B_k} = 1)$.

We now summarize the overall ME-gPC algorithm.

ALGORITHM 2.1 (ME-gPC).

- Step 1: build a stochastic ODE system by gPC
 - Step 2: perform the decomposition of random space adaptively
 - time step i : from 1 to M
 - loop all the random elements
 - if $\eta_k^\gamma \Pr(I_{B_k} = 1) \geq \theta_1$ in element k , then
 - if $r_n \geq \theta_2 \cdot \max_{j=1, \dots, d} r_j$, then
 - split random dimension ξ_n into two equal ones $\xi_{n,1}$ and $\xi_{n,2}$
 - map $\xi_{n,1}$ and $\xi_{n,2}$ to $Y_{n,1}$ and $Y_{n,2}$ defined on $[-1, 1]$
 - construct one-dimensional orthogonal polynomials for $Y_{n,1}$ and $Y_{n,2}$
 - end if
 - construct d -dimensional orthogonal polynomials using tensor products
 - map information to children elements
 - end if
 - update the information by the gPC method
 - end element loop
 - end time step
- Step 3: postprocessing stage

Remark 2.1 (C^0 continuity in random space). In the ME-gPC scheme, we deal with the stochastic problem locally without considering the interface between two random elements; in other words, we split the original problem into N independent ones. The reason is that all the statistics are in L_q sense, which means that the statistics are determined by up to q th moments of the desired random field. Thus, the continuity

$$(2.41) \quad u_{B_1}(\xi) = u_{B_2}(\xi), \quad \xi \in \bar{B}_1 \cap \bar{B}_2,$$

where \bar{B}_1 and \bar{B}_2 indicate the closure of two adjacent random elements, respectively, is not required since the measure of the interface is zero.

Remark 2.2 (construction cost of π_i). In practice, the K-L decomposition related to a random variable ξ is often used to reduce the dimensionality of colored noise. Since all the components of ξ are assumed to have identical distributions, we can maintain a table for one-dimensional orthogonal polynomials which have already been constructed. When the decomposition of random space is activated, we first look up the table to check if the edges of new random elements are in the table or not and construct orthogonal polynomials only for the missing edges. This way, the time cost for construction of orthogonal polynomials is almost independent of the random dimension.

Remark 2.3 (complexity of ME-gPC). In Algorithm 2.1, we decompose a stochastic problem into N independent ones, since C^0 continuity is unnecessary in random space. Thus, if polynomials of the same order are used in both gPC and ME-gPC, the cost of ME-gPC will increase linearly by a factor N compared to the cost of gPC, since the cost of numerical orthogonal polynomials is negligible as demonstrated in section 3.1.4.

3. Numerical results.

3.1. Construction of orthogonal polynomials. We first discuss the construction of orthogonal polynomials for random variables with some commonly used distributions. Due to the nice properties of uniform distribution, the orthogonality of Legendre-chaos can be naturally inherited in the decomposition of random space, which means that the polynomial construction is unnecessary for the uniform distribution. Here we focus on the Beta and Gaussian distributions, for which we need the on-the-fly polynomial construction.

3.1.1. Problem of overflow and underflow. We consider a random variable defined in $[-1, 1]$ with the following Beta distribution:

$$(3.1) \quad f(x) = \frac{(1-x)^\alpha(1+x)^\beta}{2^{\alpha+\beta+1}B(\alpha+1, \beta+1)}, \quad -1 \leq x \leq 1.$$

In a random element $[0, 0.001]$, we define a random variable X with a conditional PDF

$$(3.2) \quad \hat{f}(x) = \frac{f(x)}{\int_0^{0.001} f(x)dx},$$

and map X to another random variable Y defined in $[-1, 1]$ with a PDF

$$(3.3) \quad \bar{f}(y) = \frac{0.0005f(x(y))}{\int_0^{0.001} f(x)dx},$$

where

$$(3.4) \quad x(y) = 0.0005y + 0.0005.$$

Here we let $\alpha = 1$ and $\beta = 4$. In Figure 3.1, we show the orthogonal polynomials for random variable X from first order up to fourth order. We can see that the maximum values of these polynomials decrease very quickly approaching the machine accuracy. In Figure 3.2, the corresponding orthogonal polynomials for random variable Y are shown. It can be seen that their shapes are the same as those of the orthogonal polynomials for X but are well scaled.

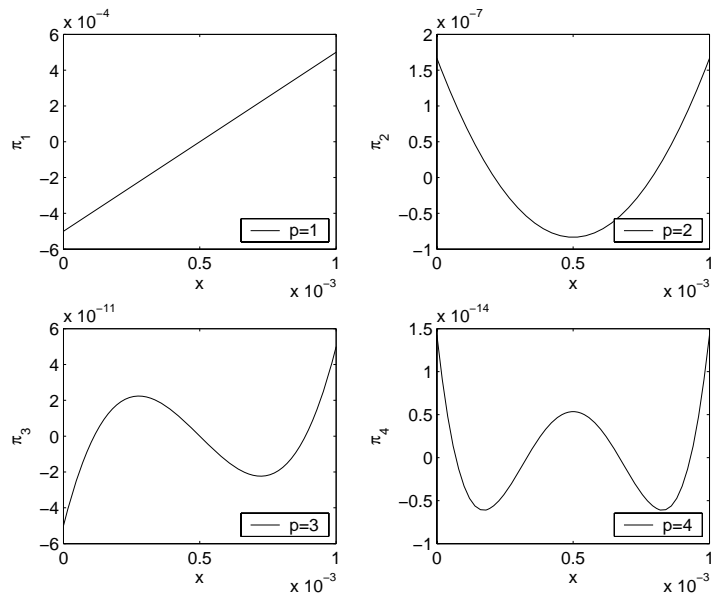


FIG. 3.1. Orthogonal polynomials for random variable X , π_1 to π_4 .

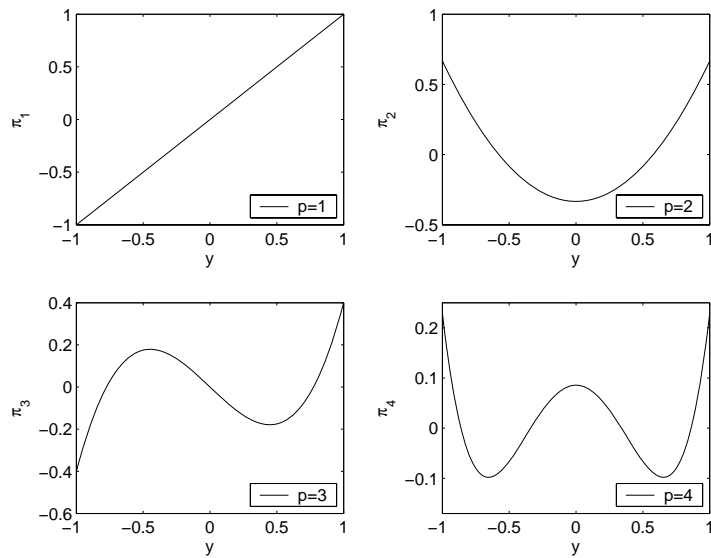


FIG. 3.2. Orthogonal polynomials for random variable Y , π_1 to π_4 .

3.1.2. Orthogonal polynomials for Beta distribution. From the ME-gPC scheme, we know that the orthogonal basis in each random element depends on a particular part of the PDF of the random inputs. We now demonstrate such a dependence using a Beta-type random variable X with $\alpha = 1$ and $\beta = 4$. For simplicity, we consider only two random elements: $[-1, 0]$ and $[0, 1]$. We define a random variable

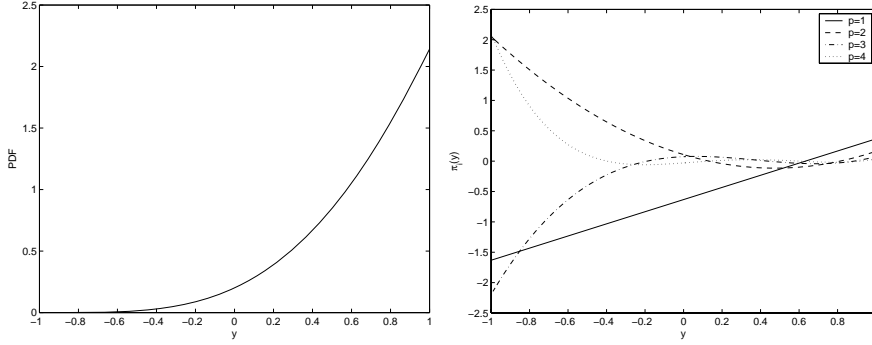


FIG. 3.3. Left: PDF of random variable Y_l induced by Beta distribution with $\alpha = 1$ and $\beta = 4$. Right: orthogonal polynomials for Y_l .

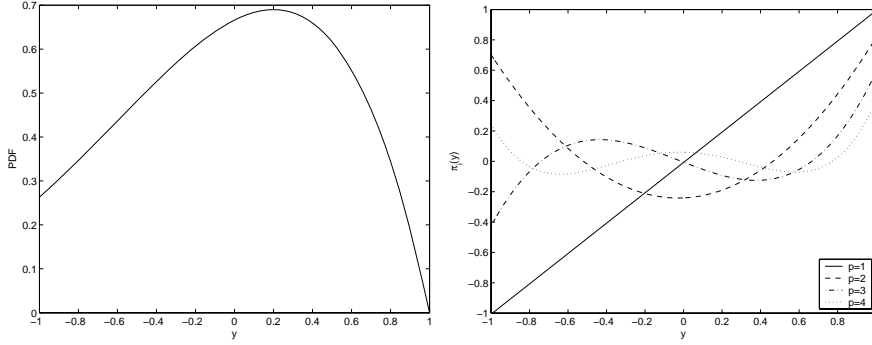


FIG. 3.4. Left: PDF of random variable Y_r induced by Beta distribution with $\alpha = 1$ and $\beta = 4$. Right: orthogonal polynomials for Y_r .

X_l in $[-1, 0]$ with a conditional PDF

$$(3.5) \quad f_l(x_l) = \frac{f(x_l)}{\int_{-1}^0 f(x)dx}$$

and another random variable X_r in $[0, 1]$ with a conditional PDF

$$(3.6) \quad f_r(x_r) = \frac{f(x_r)}{\int_0^1 f(x)dx},$$

where $f(x)$ is the PDF of X . Due to the potential problem of overflow (underflow), we will not construct the orthogonal basis according to the above two conditional PDFs, but first map X_l and X_r to Y_l and Y_r , respectively, using the transformation

$$(3.7) \quad X_l = \frac{1}{2}Y_l - \frac{1}{2}, \quad Y_r = \frac{1}{2}Y_r + \frac{1}{2},$$

and then construct orthogonal polynomials in terms of Y_l and Y_r . In Figures 3.3 and 3.4, we show the PDFs of Y_l and Y_r on the left and the corresponding orthogonal polynomials on the right. It is clear that these two sets of orthogonal polynomials are quite different due to their different weight functions (i.e., local PDFs).

3.1.3. Orthogonal polynomials for Gaussian distribution. Some random distributions, e.g., Gaussian distribution and Gamma distribution, have long tails. Next, we demonstrate the decomposition of random space and the corresponding orthogonal polynomials for this type of random variable. Given a Gaussian random variable $X \sim N(0, 1)$, we decompose the random space into three random elements $(-\infty, -a]$, $[-a, a]$, and $[a, +\infty)$, where a is a positive constant. In the middle element $[-a, a]$, we define a random variable X_m with a conditional PDF

$$(3.8) \quad f_m(x_m) = \frac{e^{-\frac{1}{2}x_m^2}}{\int_{-a}^a e^{-\frac{1}{2}x^2} dx}.$$

In one of the tail elements, say, $[a, +\infty)$, we define a random variable X_∞ with a conditional PDF

$$(3.9) \quad f_\infty(x_\infty) = \frac{e^{-\frac{1}{2}x_\infty^2}}{\int_a^\infty e^{-\frac{1}{2}x^2} dx}.$$

We choose a in such a way that $\Pr(X \geq a) < \epsilon$, where ϵ is a small positive constant. From (2.31), we know that the error contribution of tail elements can be effectively weakened by the small value of $\Pr(X \geq a)$; thus we usually do not decompose the tail elements in applications, but rather the middle element $[-a, a]$. Furthermore, there is no overflow or underflow problem for the tail elements due to the infinite boundary. Here we take $a = 5$, which yields $\Pr(X \geq 5) = 2.87 \times 10^{-7}$. In Figure 3.5, we show the PDF of the rescaled random variable $Y_m = X_m/a$ on the left and the corresponding orthogonal polynomial basis on the right. The orthogonal polynomials have similar shapes with Hermite polynomials; however, they are bounded now. In Figure 3.6, the conditional PDF of X_∞ is shown on the left and the corresponding orthogonal basis on the right. Due to the infinite boundary the orthogonal polynomials look almost like Laguerre polynomials.

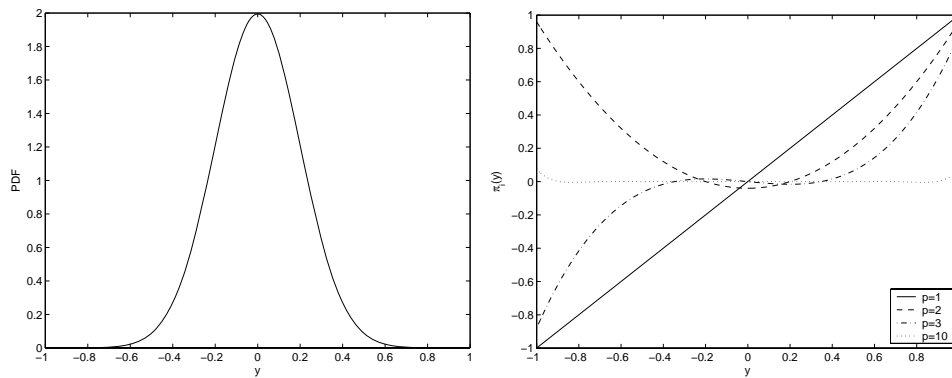


FIG. 3.5. Left: PDF of random variable Y_m induced by Gaussian distribution. Right: orthogonal polynomials for Y_m .

3.1.4. Efficiency of the solver for numerical orthogonal polynomials. It is clear that the ME-gPC method relies heavily on the numerical orthogonal polynomials. We now test the solver for the recurrence coefficients, quadrature points, and integration weights. Due to the absence of explicit formulas, we use the following

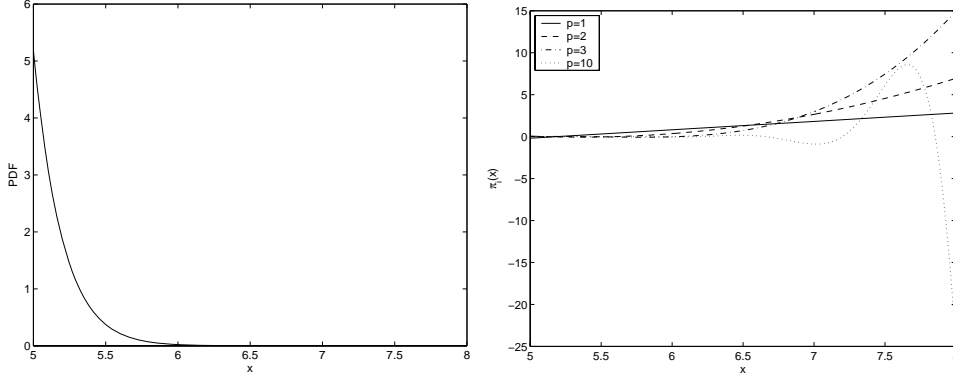


FIG. 3.6. Left: PDF of random variable X_∞ induced by Gaussian distribution. Right: orthogonal polynomials for X_∞ .

example. Given a weight function $w(x)$ defined in $[a, b]$, we decompose $[a, b]$ into small elements and construct numerical orthogonal polynomials in these elements. Then we map a simple function $g(x) = c_1x^p + c_2$ onto such elements numerically and compute the errors on the quadrature points in each element. In the k th element $[a_k, b_k]$, $g(x) = c_1x^p + c_2$ should have the form

$$(3.10) \quad g_k(\tau) = c_1 \left(\frac{b_k - a_k}{2} \tau + \frac{b_k + a_k}{2} \right)^p + c_2,$$

since all the orthogonal polynomials in element k are defined in $[-1, 1]$. Expressed by the numerical orthogonal polynomials, $g_k(\tau)$ has another form:

$$(3.11) \quad \hat{g}_k(\tau) = \sum_{i=0}^p \nu_{k,i} \pi_{k,i}(\tau),$$

where the coefficients $\nu_{k,i}$ can be obtained by the inner product

$$(3.12) \quad \nu_{k,i} = (g_k(\tau), \pi_{k,i}) / (\pi_{k,i}, \pi_{k,i}).$$

Next, we define the L_∞ error on the quadrature points over all the elements

$$(3.13) \quad \epsilon_{max} = \max \left| \frac{g_k(\tau_j) - \hat{g}_k(\tau_j)}{g_k(\tau_j)} \right|, \quad k = 1, 2, \dots, N, \quad j = 1, 2, \dots, n,$$

where N is the number of elements and n is the number of quadrature points in each element. In Table 3.1, we show ϵ_{max} and the time cost for different uniform meshes. We record the time used for the computation of recurrence coefficients, quadrature points, integration weights, and the table $\mathbb{E}[\pi_{k,i} \pi_{k,j} \pi_{k,m}]$, which represents the total set-up requirement for the standard Galerkin gPC method. We can see that the biggest error ϵ_{max} occurs in a one-element mesh. As the number of elements increases, ϵ_{max} is almost of the same order as the relative error of recurrence coefficients, which implies that the numerical polynomials can achieve good orthogonality and high accuracy. Furthermore, numerical experiments show that the time cost for the numerical polynomials is very small. For example, 10^5 sets of orthogonal polynomials up to 10th order can be computed within less than one-and-a-half minutes on a 1.5GHz

TABLE 3.1

Error ϵ_{max} and time cost for a uniform mesh with N elements. $p = 10$ and $g(x) = x^{10} + 1$. The relative error for the recurrence coefficients was set to be 10^{-13} . $\alpha = 1$ and $\beta = 4$ for Beta distribution. The middle element $[-6, 6]$ was used for Gaussian distribution. The computations were performed on a 1.5GHz AMD CPU.

| N | Beta distribution | | Gaussian distribution | |
|--------|-------------------|------------|-----------------------|------------|
| | Error | Time (sec) | Error | Time (sec) |
| 1 | 7.98e-11 | <0.01 | 1.50e-10 | <0.01 |
| 2 | 2.58e-12 | <0.01 | 4.94e-11 | <0.01 |
| 4 | 2.38e-13 | <0.01 | 1.53e-11 | <0.01 |
| 6 | 4.58e-13 | <0.01 | 4.66e-13 | <0.01 |
| 8 | 3.53e-13 | <0.01 | 3.60e-13 | <0.01 |
| 10 | 7.45e-13 | 0.01 | 1.10e-12 | 0.01 |
| 10^2 | 9.45e-13 | 0.08 | 8.28e-13 | 0.08 |
| 10^3 | 1.00e-12 | 0.79 | 9.37e-13 | 0.84 |
| 10^4 | 3.69e-12 | 7.92 | 4.51e-13 | 7.87 |
| 10^5 | 9.28e-13 | 76.43 | 4.09e-13 | 78.74 |

AMD CPU. In practice, we do not need so many elements and the time cost for the construction of orthogonal polynomials actually can be ignored compared to the time used by the gPC solver.

3.2. Accuracy and convergence of ME-gPC. We next show the accuracy and convergence of ME-gPC by several numerical examples. We use a simple stochastic ODE model to demonstrate the kp -convergence of the ME-gPC method and the Kraichnan–Orszag problem to show how to deal with discontinuities introduced by random inputs using the *adaptive* ME-gPC method.

3.2.1. Basic stochastic ODE model. We consider the performance of the ME-gPC method for the following simple ODE equation studied with the original gPC in [32]:

$$(3.14) \quad \frac{du}{dt} = -\lambda h(\zeta(\omega))u, \quad u(0; \omega) = u_0,$$

where $\zeta(\omega)$ is a random variable, h is a function of $\zeta(\omega)$, and λ is constant. The exact solution can be easily found as

$$(3.15) \quad u(t; \omega) = u_0 e^{-\lambda h(\zeta(\omega))t}.$$

Using ME-gPC with uniform meshes, we study the following two cases:

- (i) ζ is a Beta-type random variable. $\alpha = 1$, $\beta = 4$, and $h(\zeta(\omega)) = \zeta(\omega)$.
- (ii) ζ is a Gaussian random variable. $h(\zeta(\omega)) = \zeta(\omega)$.

In Figure 3.7, we show the convergence of ME-gPC at $t = 5$ for case (i) with p -convergence on the left and k -convergence on the right. It can be seen that ME-gPC exhibits exponential convergence while the index of algebraic convergence goes asymptotically to a constant. The convergence of the ME-gPC method at $t = 5$ for case (ii) is shown in Figure 3.8. Note here for the Gaussian random variable we start with a three-element mesh: $(-\infty, -6]$, $[-6, 6]$, and $[6, +\infty)$. We refine only the middle element $[-6, 6]$ because the error contribution of the tail elements is very small compared with that of the middle element. We can see that the error curve of such a three-element mesh matches exactly with that given by the Hermite-chaos. The plot of p -convergence also shows that the four-element mesh works better than

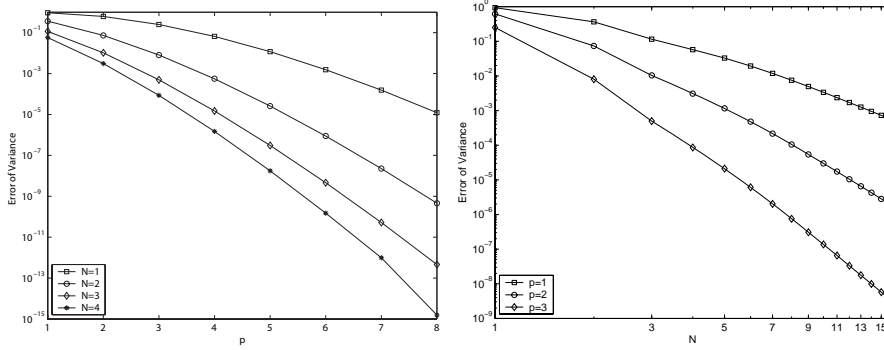


FIG. 3.7. Convergence of ME-gPC for the simple ODE with ζ being a Beta-type random variable. $\lambda = 1$. Left: p -convergence. Right: k -convergence.

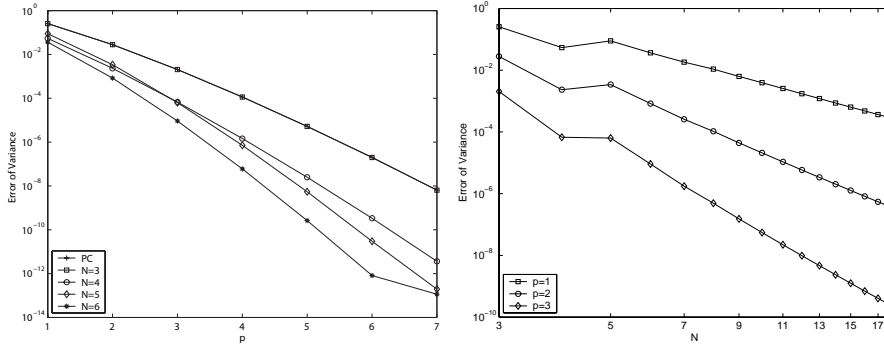


FIG. 3.8. Convergence of ME-gPC for the simple ODE with ζ being a Gaussian random variable. $\lambda = 0.1$. Left: p -convergence. Right: k -convergence.

the five-element mesh when $p < 3$. The behavior of (3.14) may be very different if $\zeta(\omega)$ is located in different random elements, which gives rise to a nonuniform error distribution in each element. For instance, when $p = 2$, $\zeta(\omega) < 0$ in element $[-6, 0]$ and the error of local variance is $\epsilon_{[-6,0]} = 5.06 \times 10^{-3}$; $\zeta(\omega) > 0$ in element $[0, 6]$ and $\epsilon_{[0,6]} = 2.48 \times 10^{-3}$.

From (2.31), we know that the error of global variance is of $O((\epsilon_{[-6,0]} + \epsilon_{[0,6]})/2)$, where we neglect the error contribution of the tail elements. For a five-element mesh the error contribution comes mainly from element $[-2, 2]$ in which $\Pr(I_{[-2,2]} = 1) = 0.95$. Thus, the error of global variance is of $O(\epsilon_{[-2,2]})$, where $\epsilon_{[-2,2]} = 5.39 \times 10^{-3}$. It can be seen that $\epsilon_{[-2,2]} > (\epsilon_{[-6,0]} + \epsilon_{[0,6]})/2$, which corresponds to the behavior of p -convergence and k -convergence in Figure 3.8. As the polynomial order p or the number N of the random elements increases, p -convergence or k -convergence will be dominant and such a phenomenon will disappear. In Table 3.2, we present the estimated indices of algebraic convergence for both case (i) and case (ii). It can be seen that for both mean and variance the index of algebraic convergence is close to $-2(p + 1)$, which is consistent with the error estimate given in [4]. From the theory of the deterministic finite element method [4], it can be shown that the index of algebraic convergence will go to $-2(p + 1)$ for any PDF. When computing the recurrence coefficients α_i and β_i iteratively, a relative error 10^{-13} is imposed for both case (i) and case (ii). From the plots of p -convergence, it can be seen that the maximum accuracy ME-gPC can

TABLE 3.2

The indices of algebraic convergence of the basic stochastic ODE model. The algebraic indices are computed by the errors given by $N = 14$ and $N = 15$. $t = 5$.

| | Beta distribution (case (i)) | | Gaussian distribution (case (ii)) | |
|---------|------------------------------|----------|-----------------------------------|----------|
| | Mean | Variance | Mean | Variance |
| $p = 1$ | -3.95 | -3.84 | -3.90 | -3.88 |
| $p = 2$ | -5.96 | -5.87 | -5.91 | -5.90 |
| $p = 3$ | -7.96 | -7.88 | -7.59 | -7.90 |

TABLE 3.3

A comparison between gPC, ME-gPC, and the standard Monte Carlo method for a fixed cost denoted by the number of floating point operations n . Beta distribution (case (i)) is considered and normalized errors are used. The first row with $N = 1$ corresponds to the gPC results.

| $n = 200$ | $\lambda = 1.0$ | | $\lambda = 1.5$ | |
|------------------|-----------------|-------------|-----------------|-------------|
| | Error (mean) | Error (var) | Error (mean) | Error (var) |
| $N = 1, p = 9$ | 4.27e-12 | 7.84e-7 | 6.67e-9 | 1.95e-4 |
| $N = 4, p = 4$ | 2.84e-9 | 1.49e-6 | 1.20e-7 | 3.98e-5 |
| $N = 25, p = 1$ | 5.88e-6 | 9.91e-5 | 2.95e-5 | 4.59e-4 |
| $N = 100, p = 0$ | 4.16e-4 | 1.73e-3 | 9.36e-4 | 3.76e-3 |
| Monte Carlo | 2.75e-1 | 5.13e-1 | 2.77e-1 | 8.31e-1 |

achieve is of $O(10^{-13})$, which implies that the constructed orthogonal polynomials are very accurate.

An obvious question that arises here is how well the kp -convergence of ME-gPC compares with the k -convergence or p -convergence. The answer is very dependent on the problem under consideration, since the properties of the solution, such as smoothness, may be significantly different. Given enough smoothness in the random space and a fixed time cost, p -convergence is usually a better choice than k -convergence or kp -convergence; however, kp -convergence can be better than p -convergence for many cases. In the following we will measure computational complexity through n that indicates the number of operations. Using a standard gPC procedure with a Galerkin projection, we obtain the following deterministic equations for the coefficients of the polynomial chaos expansion of $u(\zeta)$:

$$\frac{du_i}{dt} = \frac{1}{\mathbb{E}[\Phi_i^2(\zeta)]} \sum_{j=0}^{p_h} \sum_{k=0}^p h_j u_k \mathbb{E}[\Phi_i(\zeta)\Phi_j(\zeta)\Phi_k(\zeta)], \quad i = 0, 1, \dots, p,$$

where p_h and p indicate the highest polynomial orders of polynomial chaos expansions for $h(\zeta)$ and $u(\zeta)$, respectively. Then we can use $n = N(p_h + 1)(p + 1)^2$ to measure the cost of ME-gPC, which is proportional to the number of operations of each time step if the Runge–Kutta method is employed for the integration in time. In Table 3.3 we present the accuracy of ME-gPC for different k and p while the cost is fixed ($n = 200$). We note that $p_h = 1$. It can be seen that p -convergence is superior for the case $\lambda = 1.0$. However, for a larger perturbation $\lambda = 1.5$, the ME-gPC with $N = p = 4$ achieves a better accuracy of the variance than gPC. This implies that for a large perturbation the kp -convergence of ME-gPC can enhance the accuracy and efficiency significantly. From (2.39), it can be seen that the degree of perturbation is reduced in ME-gPC from $O(\lambda)$ to $O(\frac{b_i - a_i}{2}\lambda)$. Due to the fast (exponential) convergence of local gPC in each random element B_k , the overall performance of ME-gPC can be more efficient than that of gPC. We also expect that ME-gPC should be more robust than gPC for a large perturbation due to the decomposition of the random space. Furthermore, if

the solution is not smooth in the random space, the p -convergence may be destroyed and we need to resort to the (adaptive) kp -convergence, as demonstrated in the next section.

3.2.2. Kraichnan–Orszag (K-O) problem (stochastic discontinuity). It is well known that gPC fails for the K-O problem [24]. The K-O problem with uniform random inputs was studied in [28]. We now reconsider this problem with random inputs of Beta and Gaussian distributions. The transformed K-O problem can be expressed as [28]

$$(3.16) \quad \begin{aligned} \frac{dy_1}{dt} &= y_1 y_3, \\ \frac{dy_2}{dt} &= -y_2 y_3, \\ \frac{dy_3}{dt} &= -y_1^2 + y_2^2 \end{aligned}$$

subject to initial conditions

$$(3.17) \quad y_1(0) = y_1(0; \omega), \quad y_2(0) = y_2(0; \omega), \quad y_3(0) = y_3(0; \omega).$$

The deterministic solutions of this problem are periodic, and the period will go to infinity if the initial points are located at the planes $y_1 = 0$ and $y_2 = 0$. It was shown in [28] that gPC fails if the initial random inputs can pass through these two planes. We study the following three different kinds of initial conditions:

- (i) $y_1(0) = 1$, $y_2(0) = 0.1\xi_1$, and $y_3(0) = 0$;
- (ii) $y_1(0) = 1$, $y_2(0) = 0.1\xi_1$, and $y_3(0) = 0.1\xi_2$;
- (iii) $y_1(0) = c\xi_1$, $y_2(0) = c\xi_2$, and $y_3(0) = c\xi_3$,

where c is constant and ξ_i are random variables of Beta or Gaussian distribution.

For all three cases, the relative error of recurrence coefficients α_i and β_i is set to be 10^{-12} . For case (i) we show the convergence of adaptive ME-gPC in Tables 3.4 and 3.5, where ξ_1 is of Beta distribution and Gaussian distribution, respectively. It can be seen that ME-gPC converges as θ_1 decreases. For all the cases in Tables 3.4 and 3.5, we have recorded the time used for the construction of orthogonal PC, which is less than 0.15% of the time used by the ME-gPC solver. Thus, the cost of the polynomial construction can be ignored compared with the total cost. In Figure 3.9, we show the adaptive meshes for Beta distribution and Gaussian distribution. Due to the symmetry of distribution, the mesh for Gaussian distribution is symmetric in contrast to the unsymmetric one for Beta distribution with $\alpha = 1$ and $\beta = 4$. From the adaptive criterion (2.33), we know that the element size is controlled by two factors: the relative error η_k and $\Pr(I_{B_k} = 1)$. We can see that around the discontinuity region ($\xi_1 = 0$) the mesh is refined because the error of gPC goes to $O(1)$, and the mesh is coarser where $\Pr(I_{B_k} = 1)$ is smaller. For the Gaussian distribution we refine only the middle element $[-6, 6]$ as before. In Figure 3.10 we show the speedup of ME-gPC for case (i) compared to the Monte Carlo (MC) method; n is defined as the number of operations. If the data in Figure 3.10 are approximated by a first-order polynomial in a least-squares sense, the accuracy of ME-gPC can be obtained as $O(n^{-3.23})$, $O(n^{-3.85})$, and $O(n^{-5.55})$, corresponding to $p = 3$, $p = 4$, and $p = 5$, respectively, for Beta distribution and $O(n^{-3.07})$, $O(n^{-5.16})$, and $O(n^{-6.33})$ for Gaussian distribution. It can be seen that the adaptive ME-gPC converges much faster than the Monte Carlo method and the speedup increases with the polynomial

TABLE 3.4

Maximum normalized errors of the variance of y_1 , y_2 , and y_3 at $t = 20$ for case (i) of the K-O problem. $\alpha = 1/2$ and ξ_1 is of Beta distribution with $\alpha = 1$ and $\beta = 4$. (The results given by ME-gPC with $\theta_1 = 10^{-7}$ and polynomial order $p = 7$ are used as exact solutions.)

| | $\theta_1 = 10^{-2}$ | | $\theta_1 = 10^{-3}$ | | $\theta_1 = 10^{-4}$ | | $\theta_1 = 10^{-5}$ | |
|---------|----------------------|---------|----------------------|---------|----------------------|---------|----------------------|---------|
| | N | Error | N | Error | N | Error | N | Error |
| $p = 3$ | 24 | 2.98e-2 | 77 | 1.85e-3 | 236 | 3.61e-5 | 704 | 1.18e-6 |
| $p = 4$ | 21 | 1.54e-2 | 45 | 1.05e-3 | 93 | 3.21e-4 | 209 | 4.62e-6 |
| $p = 5$ | 16 | 6.25e-2 | 34 | 3.30e-3 | 57 | 1.50e-4 | 104 | 2.56e-6 |

TABLE 3.5

Maximum normalized errors of the variance of y_1 , y_2 , and y_3 at $t = 20$ for case (i) of the K-O problem. $\alpha = 1/2$ and ξ_1 is of normal distribution. (The results given by ME-gPC with $\theta_1 = 10^{-7}$ and polynomial order $p = 7$ are used as exact solutions.)

| | $\theta_1 = 10^{-2}$ | | $\theta_1 = 10^{-3}$ | | $\theta_1 = 10^{-4}$ | | $\theta_1 = 10^{-5}$ | |
|---------|----------------------|---------|----------------------|---------|----------------------|---------|----------------------|---------|
| | N | Error | N | Error | N | Error | N | Error |
| $p = 3$ | 38 | 2.79e-2 | 98 | 2.21e-3 | 292 | 6.58e-5 | 878 | 1.40e-6 |
| $p = 4$ | 32 | 4.02e-2 | 62 | 9.07e-3 | 128 | 4.34e-4 | 282 | 1.46e-6 |
| $p = 5$ | 28 | 4.63e-2 | 42 | 3.28e-3 | 78 | 6.46e-5 | 138 | 2.61e-6 |

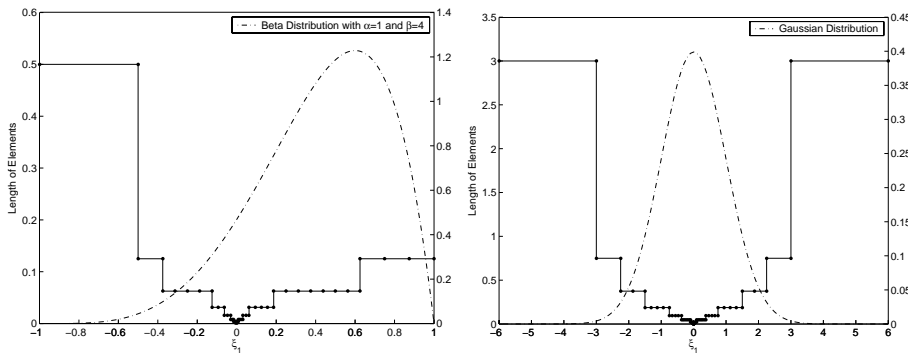


FIG. 3.9. Adaptive mesh and corresponding random distribution for the K-O problem with one-dimensional random inputs. $p = 5$ and $\theta_1 = 10^{-3}$. Left: Beta distribution with $\alpha = 1$ and $\beta = 4$. Right: Gaussian distribution.

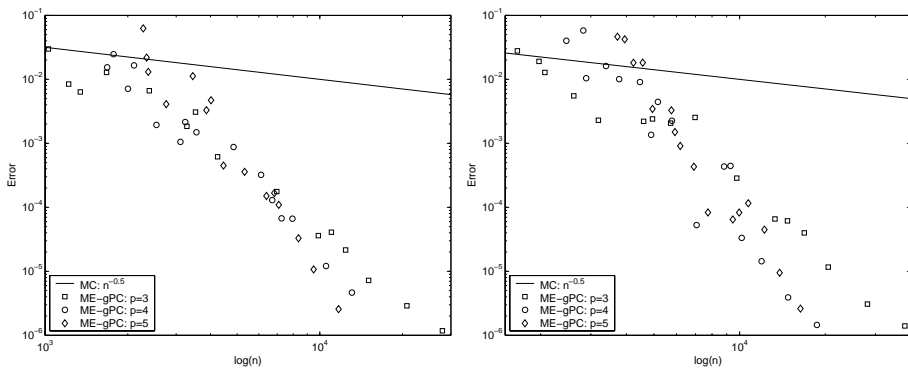


FIG. 3.10. Speedup for the K-O problem with one-dimensional random inputs at $t = 20$. Left: Beta distribution with $\alpha = 1$ and $\beta = 4$. Right: Gaussian distribution.

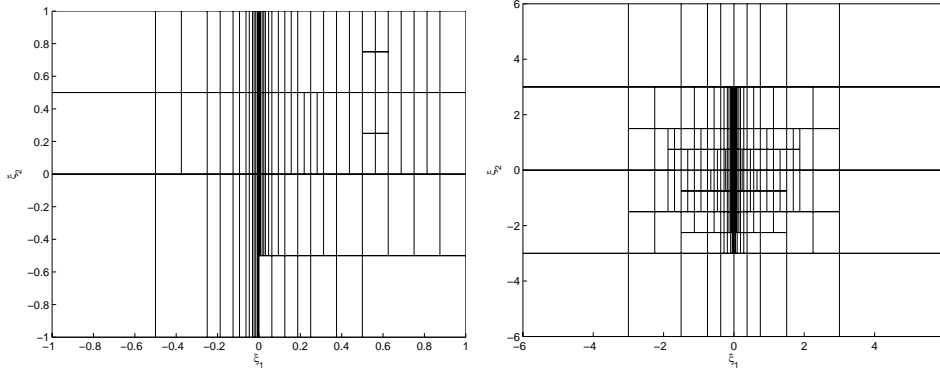


FIG. 3.11. Adaptive meshes for case (ii) of the *K-O* problem. $p = 4$, $\theta_1 = 10^{-4}$, and $\theta_2 = 10^{-2}$. Left: Beta distribution with $\alpha = 1$ and $\beta = 4$. Right: Gaussian distribution.

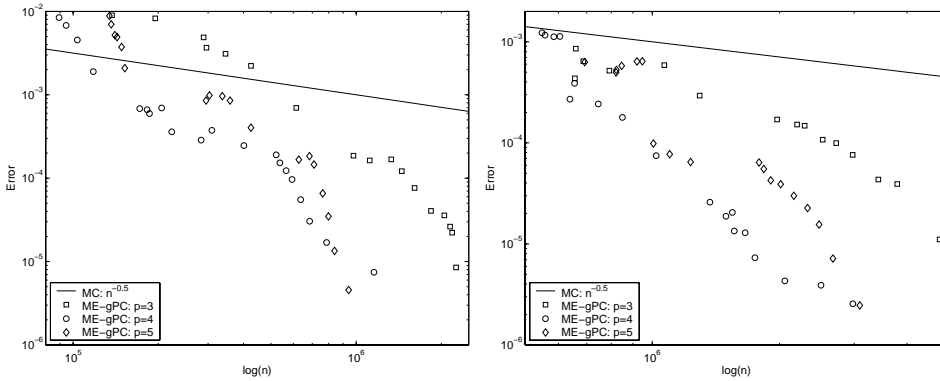


FIG. 3.12. Speedup for case (ii) of the *K-O* problem at $t = 10$. Left: Beta distribution with $\alpha = 1$ and $\beta = 4$. Right: Gaussian distribution.

order. However, since the error of gPC increases with time, the above speedups will decrease with time. The long-term behavior of gPC and ME-gPC was studied in [29].

In Figure 3.11, we show the adaptive meshes of case (ii) for Beta and Gaussian distributions. Since the discontinuity occurs at the line $\xi_1 = 0$ in the random space for this case, it can be seen that the meshes are well refined along the line $\xi_1 = 0$. It is not surprising that the random elements are bigger where the PDF is smaller because $\Pr(I_{B_k} = 1)$ is relatively smaller in these elements. In Figure 3.12, the speedup for case (ii) is shown. We can see that the Monte Carlo method is competitive for low accuracy and ME-gPC can achieve a good speedup for high accuracy.

In Figure 3.13, the evolution of y_1 in case (iii) is shown. We take $c = 1$ for the Beta distribution and 0.3 for the Gaussian distribution. For the purpose of comparison, we include the results of gPC. For both cases, gPC with $p = 3$ begins to fail at $t \approx 1$. Since increasing polynomial order does not improve gPC [28], the results of gPC with a higher order are not shown. ME-gPC can maintain convergence by increasing random elements adaptively. However, for a comparable accuracy of $O(10^{-3})$, the Monte Carlo method is about twice as fast as ME-gPC for the Beta distribution and about four times as fast as ME-gPC for the Gaussian distribution. This is mainly due to two factors. First, the discontinuity for case (iii) is strong because the discontinuity

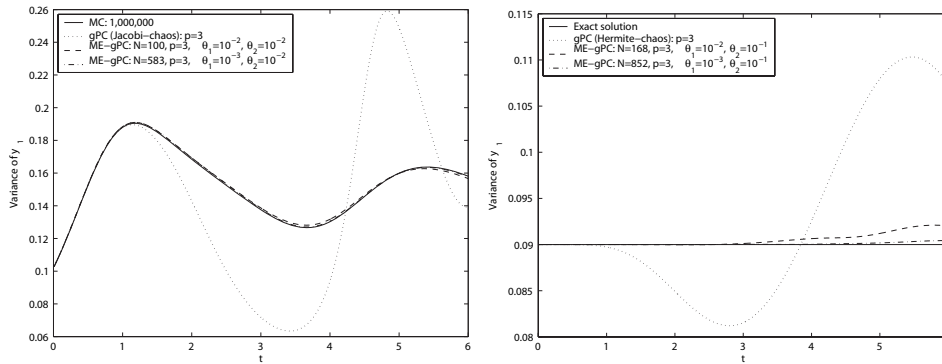


FIG. 3.13. Evolution of the variance of y_1 for case (iii) of K - O problem. Left: Beta distribution with $\alpha = 1$ and $\beta = 4$. $c = 1$. Right: Gaussian distribution. $c = 0.3$.

region consists of two planes: $\xi_1 = 0$ and $\xi_2 = 0$. Second, the cost of gPC increases very quickly with the random dimension. Since ME-gPC is a dimension dependent method, its efficiency decreases as the random dimension increases. The Monte Carlo method is still a better choice for obtaining a moderate accuracy for high-dimensional cases unless a high accuracy is really required.

3.2.3. ME-gPC for PDEs: Heat transfer in a grooved channel. Next, we demonstrate the use of ME-gPC to engineering applications, by simulating the heat transfer enhancement in a two-dimensional grooved channel subjected to stochastic excitation. The phenomenon of heat transfer enhancement in a grooved channel has been widely studied numerically and experimentally (see [11, 16] and references therein). The geometry to be considered is depicted in Figure 3.14. We use exactly the same geometry as in [11], which corresponds to $L = 6.6666$, $l = 2.2222$, and $a = 1.1111$.

The flow is assumed to be fully developed in the x (flow) direction. For the velocity field we have the incompressible Navier–Stokes equations

$$(3.18a) \quad \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = -\nabla \Pi + \nu \nabla^2 \mathbf{v},$$

$$(3.18b) \quad \nabla \cdot \mathbf{v} = 0,$$

where $\mathbf{v}(\mathbf{x}, t; \omega)$ is the stochastic velocity, Π is the pressure, and ν is the kinematic viscosity. The fully developed boundary conditions for the velocity are

$$(3.19a) \quad \mathbf{v}(\mathbf{x}, t; \omega) = 0 \quad \text{on } \partial D,$$

$$(3.19b) \quad \mathbf{v}(x + L, y, t; \omega) = \mathbf{v}(x, y, t; \omega),$$

corresponding to no-slip and periodicity, respectively. The uncertainty is introduced by the following imposed flow-rate condition:

$$(3.20) \quad Q(t; \omega) = \int_{\partial D_B}^{\partial D_T} u(x, y, t; \omega) dy = \frac{4}{3} (1 + 0.2 \xi_H(\omega) \sin 2\pi \Omega_F t),$$

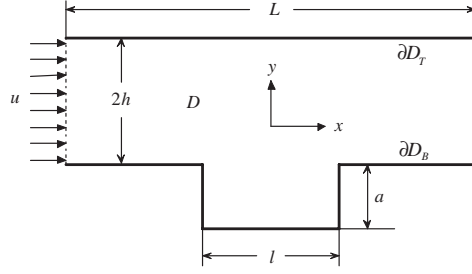


FIG. 3.14. Schematic of the geometry of a grooved channel.

where $\xi_H(\omega)$ is a random variable and Ω_F is a prescribed frequency. Due to the periodic boundary condition for the velocity, we require

$$(3.21a) \quad \Pi(\mathbf{x}, t; \omega) = -h(t; \omega)x + \tilde{\Pi}(\mathbf{x}, t; \omega),$$

$$(3.21b) \quad \tilde{\Pi}(x, y, t; \omega) = \tilde{\Pi}(x + L, y, t; \omega)$$

for the pressure; otherwise, a trivial solution for the velocity will be obtained. The convection equation for the temperature $T(\mathbf{x}, t)$ is given as

$$(3.22) \quad \frac{\partial T}{\partial t} + (\mathbf{v} \cdot \nabla)T = \alpha_t \nabla^2 T,$$

where α_t is the assumed constant thermal diffusivity. For the temperature boundary conditions on the top and bottom walls we have

$$(3.23a) \quad \kappa \nabla T \cdot \mathbf{n} = 0 \quad \text{on } \partial D_T,$$

$$(3.23b) \quad \kappa \nabla T \cdot \mathbf{n} = 1 \quad \text{on } \partial D_B,$$

where κ is the thermal conductivity of the fluid and \mathbf{n} refers to the outward normal on the boundary. As for the inlet and outlet thermal boundary conditions, we first subtract a linear term from T and use a periodic boundary condition for the other contribution

$$(3.24a) \quad T(\mathbf{x}, t; \omega) = \theta(\mathbf{x}, t; \omega) + \phi(\omega)x,$$

$$(3.24b) \quad \theta(x, y, t; \omega) = \theta(x + L, y, t; \omega),$$

where the coefficient $\phi(\omega)$ can be determined by the global energy balance [10]. The numerical schemes for the aforementioned equations, which can be found in [27], are based on the deterministic splitting schemes for the Navier–Stokes equations [17, 10].

In this work we demonstrate only the k -convergence of ME-gPC by the computation of an averaged global Nusselt number. Using the same notation as in [11], we define a Nusselt number Nu as

$$(3.25) \quad Nu = L \left(1 + \frac{2a}{L}\right)^2 \bigg/ \int_0^{L+2a} \langle \theta - \theta_b \rangle ds|_{\partial D_B},$$

TABLE 3.6

k-convergence of the ME-gPC method in terms of the variance of the Nusselt number in a grooved channel. Uniform meshes are used in the random space. The reference variance of the Nusselt number is 0.0394, which is given by the ME-gPC method with $N = 10$ and $p = 6$.

| gPC | Variance | Δ_{var} | ME-gPC | Variance | Δ_{var} |
|------------------|----------|----------------|--------------------------|----------|----------------|
| $p = 1, n_p = 2$ | 0.0719 | 8.26e-1 | $N = 1, p = 1, n_p = 2$ | 0.0719 | 8.26e-1 |
| $p = 3, n_p = 4$ | 0.0617 | 5.67e-1 | $N = 3, p = 1, n_p = 6$ | 0.0325 | 1.75e-1 |
| $p = 5, n_p = 6$ | 0.0435 | 1.06e-1 | $N = 5, p = 1, n_p = 10$ | 0.0355 | 9.84e-2 |
| $p = 7, n_p = 8$ | 0.0430 | 8.41e-2 | $N = 7, p = 1, n_p = 14$ | 0.0402 | 4.82e-3 |

where θ_b is a reference temperature taken to be the mixed-mean temperature at $x = 0$,

$$(3.26) \quad \theta_b = \left\langle \frac{3}{4} \int_{-1}^1 u(x=0, y, t; \omega) \theta(x=0, y, t; \omega) dy \right\rangle,$$

and $\langle \cdot \rangle$ refers to the time average over one period of the flow, $t < t' < +\Omega_F^{-1}$.

The Reynolds number is defined as $Re = 3\langle Q(t) \rangle / 4\nu$, and the Prandtl number as $Pr = \nu/\alpha$. We take $Re = 525$ and $Pr = 1$ in this work. We let $\Omega_F = 0.15$ and let ξ_H be a random variable of Beta distribution with $\alpha = 1$ and $\beta = 4$.

The convergence of gPC for the Navier–Stokes equation and the advection-diffusion equation has been studied and verified in [34, 30]. The solution of (3.18) and (3.22) converges to a steady state. We use the variance obtained by the ME-gPC method with $N = 10$, $p = 6$, and $t = 4000$ as the reference R^* . We define a relative normalized error for the variance as

$$(3.27) \quad \Delta_{var} = \left| \frac{R - R^*}{R^*} \right|,$$

where R is the variance given by gPC or ME-gPC. Let $n_p = N(p + 1)$ indicate the number of unknowns in the PC expansion. In Table 3.6, we show the k -convergence of the ME-gPC method. We note that both the mean and the variance obtained from a low-order ME-gPC are converging to the reference results as the number of elements N increases, which yields a k -convergence. The decomposition of random space accelerates the convergence significantly, but at this (relatively low) Reynolds number we need a slightly larger n_p for ME-gPC to obtain comparable accuracy with gPC. However, at higher Reynolds numbers we have larger stochastic perturbations and ME-gPC is more efficient than gPC; this is consistent with the cost comparison results of the first example presented earlier. In fact, for large values of the polynomial order p , gPC yields stochastic solutions that exhibit artificial oscillations at the tails of the corresponding PDFs; such artifacts are not present with ME-gPC [27]. Finally, compared to high-order gPC, an important cost advantage of ME-gPC for large-scale numerical simulations in fluid mechanics is that it can be easily parallelized as the local problems in each random element are independent.

In Figure 3.15, we show the standard deviation of the temperature at $t = 4000$. It can be seen that the standard deviation increases from the channel region to the groove part of the domain. The local maxima are inside the cavity and close to the top right corner of the cavity, which reflects the established flow communication between the groove and the channel. This, in turn, is the reason that gives rise to enhanced heat transfer [11].

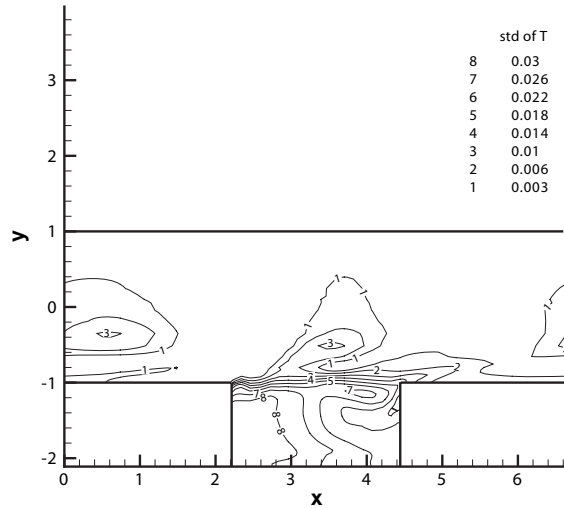


FIG. 3.15. Standard deviation of T at $t = 4000$ obtained by ME-gPC with $N = 10$ and $p = 6$.

4. Summary. In this paper we present an ME-gPC method for stochastic processes represented by random variables of arbitrary distributions. The ME-gPC method combines the two PC schemes of k -version and p -version by the numerical reconstruction of orthogonal polynomials, which yields a $k \times p$ version. Numerical experiments show that the numerical reconstruction of high accuracy can be obtained with negligible cost. Hence, the ME-gPC method inherits the advantages of previous PC methods and can efficiently achieve kp -convergence for stochastic differential equations. This is shown by a simple ODE model and a heat transfer problem in a grooved channel. The application of ME-gPC depends on the smoothness of the solution in the random space. Due to the fast (exponential) convergence, gPC is preferred if the p -convergence can be achieved; otherwise, the convergence can be improved by the kp -convergence of ME-gPC. The adaptive ME-gPC scheme shows a good performance in convergence for problems related to discontinuities induced by random inputs because it can weaken the influence of discontinuities by small random elements and maintain the fast (exponential) convergence of gPC in regular elements, which is clearly shown by the adaptive meshes for the K-O problem. Thus, the ME-gPC method is more flexible and efficient than previous polynomial chaos schemes in many cases.

In [1], the asymptotic numerical complexity of stochastic Galerkin finite element methods of $k \times h$ version and $p \times h$ version is compared with the Monte Carlo method. Note that the comparison is valid only for the optimal case. For some cases, especially when the random dimension is high and the desired accuracy is moderate, the Monte Carlo method is still preferred in practice because it is random dimension independent while all PC methods including the ME-gPC method depend on the random dimension. If strong discontinuities exist together with a high random dimension as shown in case (iii) of section 3.2.2, polynomial chaos methods can lose efficiency further or even fail to converge. However, if the required error is sufficiently small, the PC method, especially the ME-gPC method, will be more efficient than the Monte

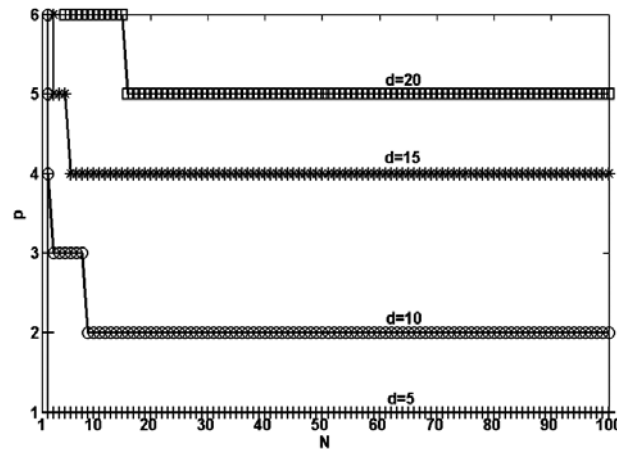


FIG. 4.1. Cost comparison between ME-gPC and the standard Monte Carlo method for fixed accuracy. Here p is the order of gPC in each element, N is the number of elements and d is the number of dimensions. The symbol-lines are iso-cost lines.

Carlo method due to the k - p convergence as shown in cases (i) and (ii) of section 3.2.2. In Figure 4.1, we present a comparison between the k -convergence $O(N^{-2(p+1)})$ of ME-gPC and the convergence $O(n^{-1/2})$ of the standard Monte Carlo method; see [1]. For the same accuracy and different random dimension numbers, the lines show the cases where the cost of the standard Monte Carlo method is equal to that of ME-gPC. For a certain random dimension number (denoted by “ d ”), the region below the line is where MC is more efficient; the region above the line is where ME-gPC is more efficient.

In this work we use heuristically the decay rate of relative error of variance as the indicator for the k -type refinement. A more rigorous a posteriori error estimate is still needed for the kp -adaptivity. These issues will be addressed in future publications.

Appendix (application of ME-gPC to a stochastic elliptic problem).

Here we briefly elaborate on how to apply ME-gPC to solve differential equations with stochastic coefficients using the following stochastic linear boundary value problem: Find a stochastic function, $u : \bar{D} \times \Omega \rightarrow \mathbb{R}$, such that almost surely the following equation holds:

$$(4.1) \quad \begin{aligned} -\nabla \cdot (a(\mathbf{x}; \omega) \nabla u(\mathbf{x}; \omega)) &= f(\mathbf{x}; \omega) \quad \text{on } D, \\ u(\mathbf{x}; \omega) &= 0 \quad \text{on } \partial D, \end{aligned}$$

where D is an open domain in the physical space with Lipschitz boundaries, $a(\mathbf{x}; \omega)$ and $f(\mathbf{x}; \omega)$ are second-order random processes. We assume $0 < \alpha_1 \leq a(\mathbf{x}; \omega) \leq \alpha_2 < \infty$ almost everywhere on $\bar{D} \times \Omega$. Using the K-L expansion and following the procedure in [1], we assume that $a(\mathbf{x}; \omega) \approx a(\mathbf{x}; \boldsymbol{\xi}(\omega))$ and $f(\mathbf{x}; \omega) \approx f(\mathbf{x}; \boldsymbol{\xi}(\omega))$, where the same \mathbb{R}^d -valued random variable $\boldsymbol{\xi}$ with i.i.d. components is used for both $a(\mathbf{x}; \omega)$ and $f(\mathbf{x}; \omega)$ in their K-L expansions. We note here that the “i.i.d.” assumption is strong since it is sufficient for the random variables in a K-L expansion to be mutually uncorrelated.

In each random element B_k we naturally have the following problem:

$$(4.2) \quad \begin{aligned} -\nabla \cdot (a(\mathbf{x}; \boldsymbol{\xi}_k) \nabla u(\mathbf{x}; \boldsymbol{\xi}_k)) &= f(\mathbf{x}; \boldsymbol{\xi}_k) \quad \text{on } D, \\ u(\mathbf{x}; \boldsymbol{\xi}_k) &= 0 \quad \text{on } \partial D. \end{aligned}$$

All previous assumptions on $a(\mathbf{x}; \boldsymbol{\xi}_k)$ and $f(\mathbf{x}; \boldsymbol{\xi}_k)$ are satisfied with respect to a conditional PDF $\hat{f}_k(\boldsymbol{\xi}_k | I_{B_k} = 1)$. We note that such a system is complete on each random element B_k .

We define a bilinear form $\mathcal{B}(\cdot, \cdot)$ as

$$(4.3) \quad \mathcal{B}(u, v) = \int_{B_k} \hat{f}_k(\boldsymbol{\xi}_k | I_{B_k} = 1) \int_D a(\mathbf{x}; \boldsymbol{\xi}_k) \nabla u(\mathbf{x}; \boldsymbol{\xi}_k) \cdot \nabla v(\mathbf{x}; \boldsymbol{\xi}_k) d\mathbf{x} d\boldsymbol{\xi}_k$$

and a linear form $\mathcal{L}(\cdot)$ as

$$(4.4) \quad \mathcal{L}(v) = \int_{B_k} \hat{f}_k(\boldsymbol{\xi}_k | I_{B_k} = 1) \int_D f(\mathbf{x}; \boldsymbol{\xi}_k) v(\mathbf{x}; \boldsymbol{\xi}_k) d\mathbf{x} d\boldsymbol{\xi}_k.$$

Due to the assumption on $a(\mathbf{x}; \boldsymbol{\xi}_k)$ the space

$$(4.5) \quad W(D, B_k, \hat{f}_k) = \{v(\mathbf{x}; \boldsymbol{\xi}_k) | \mathcal{B}(v, v) = \|v\|_W^2 < \infty \quad \forall v \in L_2(B_k) \otimes H_0^1(D)\}$$

is a Hilbert space. Using the Lax–Milgram theorem, we can claim that a unique solution $\hat{u}_k(\mathbf{x}; \boldsymbol{\xi}_k)$ exists for the problem

$$(4.6) \quad \mathcal{B}(u, v) = \mathcal{L}(v) \quad \forall v \in W.$$

To this end, the gPC procedure can be readily employed in each element B_k on a truncated subspace $V^r \otimes V^p \subset W$ to solve (4.2) [14, 32], where V^r indicates the local orthogonal basis $\{\Phi_{k,i}(\boldsymbol{\xi}_k)\}$ in the random space and V^p denotes the piecewise polynomial basis in the physical space. Once the local approximation is obtained, the global approximation can be constructed by Proposition 2.2.

Using a similar procedure ME-gPC can be readily applied to any differential equations with stochastic coefficients.

REFERENCES

- [1] I. BABUŠKA, R. TEMPONE, AND G. E. ZOURARIS, *Galerkin finite element approximations of stochastic elliptic partial differential equations*, SIAM J. Numer. Anal., 42 (2004), pp. 800–825.
- [2] D. BOLEY AND G. H. GOLUB, *A survey of matrix inverse eigenvalue problems*, Inverse Problems, 3 (1987), pp. 595–622.
- [3] R. CAMERON AND W. MARTIN, *The orthogonal development of non-linear functionals in series of Fourier–Hermite functionals*, Ann. of Math. (2), 48 (1947), pp. 385–392.
- [4] M. K. DEB, I. BABUŠKA, AND J. T. ODEN, *Solution of stochastic partial differential equations using Galerkin finite element techniques*, Comput. Methods Appl. Mech. Engrg., 190 (2001), pp. 6359–6372.
- [5] H. J. FISCHER, *On the condition of orthogonal polynomials via modified moments*, Z. Anal. Anwendungen, 15 (1996), pp. 1–18.
- [6] H. J. FISCHER, *On generating orthogonal polynomials for discrete measures*, Z. Anal. Anwendungen, 17 (1998), pp. 183–205.
- [7] P. FRAUENFELDER, C. SCHWAB, AND R. A. TODOR, *Finite elements for elliptic problems with stochastic coefficients*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 205–228.
- [8] W. GAUTSCHI, *On generating orthogonal polynomials*, SIAM J. Sci. Statist. Comput., 3 (1982), pp. 289–317.

- [9] W. GAUTSCHI, *Algorithm 726: ORTHPOL—A package of routines for generating orthogonal polynomials and Gauss-type quadrature rules*, ACM Trans. Math. Software, 20 (1994), pp. 21–62.
- [10] N. K. GHADDAR, G. E. KARNIADAKIS, AND A. T. PATERA, *A conservative isoparametric spectral element method for forced convection: Application to fully developed flow in periodic geometries*, Num. Heat Transfer, 9 (1986), pp. 277–300.
- [11] N. K. GHADDAR, M. MAGEN, B. B. MIKIC, AND A. T. PATERA, *Numerical investigation of incompressible flow in grooved channels. II. Resonance and oscillatory heat-transfer enhancement*, J. Fluid Mech., 168 (1986), pp. 541–567.
- [12] R. G. GHANEM, *Ingredients for a general purpose stochastic finite element formulation*, Comput. Methods Appl. Mech. Engrg., 168 (1999), pp. 19–34.
- [13] R. G. GHANEM, *Stochastic finite elements for heterogeneous media with multiple random non-Gaussian properties*, ASCE J. Engrg. Mech., 125 (1999), pp. 26–40.
- [14] R. GHANEM AND P. SPANOS, *Stochastic Finite Elements: A Spectral Approach*, Springer-Verlag, New York, 1991.
- [15] R. D. HENDERSON, *Dynamic refinement algorithms for spectral element methods*, Comput. Methods Appl. Mech. Engrg., 175 (1999), pp. 395–411.
- [16] C. HERMAN AND E. KANG, *Heat transfer enhancement in a grooved channel with curved vanes*, Int. J. Heat Mass Transfer, 45 (2002), pp. 3741–3757.
- [17] G. KARNIADAKIS AND S. SHERWIN, *Spectral/hp Element Methods for CFD*, Oxford University Press, New York, 1999.
- [18] O. P. LE MAITRE, H. N. NJAM, R. G. GHANEM, AND O. M. KNIO, *Uncertainty propagation using Wiener–Haar expansions*, J. Comput. Phys., 197 (2004), pp. 28–57.
- [19] O. P. LE MAITRE, H. N. NJAM, R. G. GHANEM, AND O. M. KNIO, *Multi-resolution analysis of Wiener-type uncertainty propagation schemes*, J. Comput. Phys., 197 (2004), pp. 502–531.
- [20] H. G. MATTHIES AND C. G. BUCKER, *Finite elements for stochastic media problems*, Comput. Methods Appl. Mech. Engrg., 168 (1999), pp. 3–17.
- [21] H. G. MATTHIES AND A. KEESE, *Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 1295–1331.
- [22] C. MAVRIPLIS, *Adaptive mesh strategies for the spectral element method*, Comput. Methods Appl. Mech. Engrg., 116 (1994), pp. 77–86.
- [23] S. E. NOTARIS, *Interpolatory quadrature formulae with Chebyshev abscissae*, J. Comput. Appl. Math., 133 (2001), pp. 507–517.
- [24] S. A. ORSZAG AND L. R. BISSONNETTE, *Dynamical properties of truncated Wiener-Hermite expansions*, Physics Fluids, 10 (1967), pp. 2603–2613.
- [25] C. SCHWAB, *p- and hp- Finite Element Methods: Theory and Applications in Solid and Fluid Mechanics*, The Clarendon Press, Oxford University Press, New York, 1998.
- [26] G. SZEGÖ, *Orthogonal Polynomials*, American Mathematical Society, Providence, RI, 1959.
- [27] X. WAN, *Multi-Element Generalized Polynomial Chaos for Differential Equations with Random Inputs: Algorithms and Applications*, Ph.D. thesis, Division of Applied Mathematics, Brown University, Providence, RI, in preparation.
- [28] X. WAN AND G. E. KARNIADAKIS, *An adaptive multi-element generalized polynomial chaos method for stochastic differential equations*, J. Comput. Phys., 209 (2005), pp. 617–642.
- [29] X. WAN AND G. E. KARNIADAKIS, *Long-term behavior of polynomial chaos in stochastic flow simulations*, Comput. Methods Appl. Mech. Engrg., in press.
- [30] X. WAN, D. XIU, AND G. E. KARNIADAKIS, *Stochastic solutions for the two-dimensional advection-diffusion equation*, SIAM J. Sci. Comput., 26 (2004), pp. 578–590.
- [31] N. WIENER, *The homogeneous chaos*, Amer. J. Math., 60 (1938), pp. 897–936.
- [32] D. XIU AND G. E. KARNIADAKIS, *The Wiener–Askey polynomial chaos for stochastic differential equations*, SIAM J. Sci. Comput., 24 (2002), pp. 619–644.
- [33] D. XIU AND G. E. KARNIADAKIS, *Modeling uncertainty in flow simulations via generalized polynomial chaos*, J. Comput. Phys., 187 (2003), pp. 137–167.
- [34] D. XIU AND G. E. KARNIADAKIS, *A new stochastic approach to transient heat conduction modeling with uncertainty*, Int. J. Heat Mass Transfer, 46 (2003), pp. 4681–4693.