A weak-intrusive stochastic finite element method for stochastic structural dynamics analysis

Zhibao Zheng[a,d,*], Michael Beer[b,e,f], Hongzhe Dai[c], Udo Nackenhorst[a,d]

[a]*Institute of Mechanics and Computational Mechanics, Leibniz Universität Hannover, Appelstraße 9a, 30167, Hannover, Germany*
[b]*Institute for Risk and Reliability, Leibniz Universität Hannover, Callinstraße 34, 30167 Hannover, Germany*
[c]*School of Civil Engineering, Harbin Institute of Technology, Harbin 150090, China*
[d]*International Research Training Group 2657, Leibniz Universität Hannover, Appelstraße 11/11a, 30167, Hannover, Germany*
[e]*Institute for Risk and Uncertainty and School of Engineering, University of Liverpool, Peach Street, Liverpool L69 7ZF, UK*
[f]*International Joint Research Center for Resilient Infrastructure & International Joint Research Center for Engineering Reliability and Stochastic Mechanics, Tongji University, 1239 Siping Road, Shanghai 200092, China*

Abstract

This paper presents a weak-intrusive stochastic finite element method for solving stochastic structural dynamics equations. In this method, the stochastic solution is decomposed into the summation of a series of products of random variables, spatial vectors and temporal functions. An iterative algorithm is proposed to compute each triplet of the random variable, spatial vector and temporal function one by one. The original stochastic dynamics problem is firstly transformed into spatial-temporal coupled problems (i.e. deterministic structural dynamics equations), which can be solved efficiently by existing FEM solvers. Based on the solution of the spatial-temporal coupled problem, the original problem is then transformed into stochastic-temporal coupled problems (i.e. one-dimensional second-order stochastic ordinary differential equations), which are solved by a proposed sampling method. All random sources are embedded into the stochastic-temporal coupled problems. The proposed sampling method can solve the stochastic-temporal problems of hundreds of dimensions with low computational costs. Thus the curse of dimensionality in high-dimensional stochastic spaces is avoided with great success. Three numerical examples, including low- and high-dimensional stochastic problems, are used to demonstrate the good accuracy and the high efficiency of the proposed method.

Keywords: Stochastic structural dynamics; Stochastic finite element method; Weak-intrusive

approach; Curse of dimensionality;

1. Introduction

Partial differential equations (PDEs) provide powerful tools to describe physical models in science and engineering. Discretization techniques usually perform numerical simulations of PDEs. One of the state-of-the-art methods is the finite element method (FEM). FEM offers a simple way to solve very high-resolution solutions in various physics models [1]. As predicting uncertainty propagation on the response of models has become an essential part of the analysis and design of practical engineering systems, a growing interest has been devoted to the propagation of uncertainties through physical models governed by stochastic PDEs [2, 3]. As an extension of FEM, the stochastic finite element method (SFEM) is also used to solve high-resolution stochastic solutions of physical models involving uncertainties [4, 5, 6].

In this paper, we consider that the governing equation with n degrees of freedom for linear stochastic structural dynamics analysis is given as

$$\mathbf{M}(\theta)\ddot{\mathbf{u}}(t,\theta) + \mathbf{C}(\theta)\dot{\mathbf{u}}(t,\theta) + \mathbf{K}(\theta)\mathbf{u}(t,\theta) = \mathbf{F}(t,\theta),\ t \in [0,T]\,, \tag{1}$$

where $\mathbf{M}(\theta),\ \mathbf{C}(\theta),\ \mathbf{K}(\theta) \in \mathbb{R}^{n\times n}$ and $\mathbf{F}(t,\theta)$ are obtained by the classical finite element discretization. Initial values are given by $\mathbf{u}(0,\theta) = \mathbf{u}_0(\theta)$ and $\dot{\mathbf{u}}(0,\theta) = \mathbf{u}_1(\theta)$. $\mathbf{M}(\theta),\ \mathbf{C}(\theta)$ and $\mathbf{K}(\theta)$, known as the stochastic mass, damp and stiffness matrices, are usually induced by stochastic material properties and $\mathbf{F}(t,\theta)$, known as the stochastic force vector, is induced by stochastic forces. The modelling of uncertainties is defined in a suitable probability space $(\Theta, \Xi, \mathcal{P})$, where Θ denotes the space of elementary events, Ξ is a σ-algebra defined on Θ and $\mathcal{P}$ is a probability measure. We consider the stochastic solution $\mathbf{u}\,(t,\theta)$ of Eq. (1) to be a stochastic function with value in a certain functional space. We focus on solving $\mathbf{u}\,(t,\theta)$ efficiently and accurately in this paper.

Over the last few decades, several methods have been developed to solve Eq. (1). They are usually extended from the methods for solving time-independent (or called static) stochastic problems. These methods are divided into non-intrusive and intrusive ways. A powerful non-intrusive

*Corresponding author
Email address: zhibao.zheng@ibnm.uni-hannover.de (Zhibao Zheng)

method is the Monte Carlo simulation (MCS) [7, 8, 9]. MCS is easy to be executed by repeatedly solving deterministic problems. It can be applied to solve high-dimensional stochastic problems due to the dimensionality-independent convergence. However, a large number of deterministic problems are solved to achieve a good accuracy, which is computationally expensive, especially for large-scale problems. Some improvements are developed to save the computational cost of MCS, e.g. quasi-MCS and multilevel MCS [8, 10, 11]. Other non-intrusive methods are also proposed to solve Eq. (1), e.g. adaptive sampling methods and response surface methods [12, 13, 14].

A kind of intrusive Galerkin spectral stochastic method and its extensions [3, 4, 15, 16, 17, 18] are proposed to solve time-independent stochastic finite element equations. It has been proven efficient for time-dependent stochastic problems. In this kind of method, the stochastic solution $\mathbf{u}(t, \theta)$ is decomposed into the summation of a series of products of polynomial chaos (PC) basis and time-dependent deterministic functions [4, 15]. The Galerkin projection is then used to transform Eq. (1) into an augmented deterministic dynamics equation whose size is much larger than the original problem. The size of the augmented equation increases dramatically as the stochastic dimension and the order of PC basis increase [19, 20]. The computational cost for solving the augmented equation is prohibitively expensive, especially for large-scale and high-dimensional stochastic problems. More often than not, this kind of method suffers from the curse of dimensionality when dealing with a large number of random parameters that are derived from the discretized input random fields [21, 22]. Although several improvements are presented to reduce computational requirements, e.g. Krylov-type iteration and sparse PC approximation [23, 24], it is still attractive to develop efficient and general-purpose methods to solve large-scale and high-dimensional stochastic dynamics problems in the context of PC-based methods. Also, other methods are also proposed to solve Eq. (1), e.g. ANOVA methods [25, 26], non-parametric methods [27, 28, 29], etc.

Another method that has received much attention is known as the proper generalized decomposition (PGD) method, which is firstly introduced in [30]. This kind of method explores a priori separated representations of the stochastic solutions and does not require prior knowledge of the stochastic solutions. PGD methods have been successfully applied to time-dependent problems [31, 32, 33, 34]. In the PGD method, the stochastic solution of time-dependent problems is

decomposed into the summation of a set of products of the triplet of stochastic, spatial and temporal components. Alternating minimization algorithms are used to compute corresponding random variables, spatial vectors and temporal function [35]. The stochastic version of PGD, named generalized spectral decomposition (GSD) method [17], has been used to solve stochastic transient heat diffusion problems and high-dimensional stochastic problems [20, 36, 37, 38]. PGD and GSD methods provide efficient and valuable ways for structural dynamics analysis. These methods have a few applications in stochastic structural dynamics analysis, e.g. PGD-based stochastic structural dynamic analysis in frequency domain [39] and comparative analysis between GSD-based stochastic structural dynamic analysis and the ANOVA method [26].

In this paper, we propose a PGD-like expansion of the stochastic solution and develop an alternating iterative algorithm to solve the components of the stochastic expansion. The iterative procedure of the classical PGD method is to fix two components in three components and then update the other. However, it is numerically unstable for some cases (this will be discussed in Section 2 and the numerical example in Section 6). To develop a numerically stable iteration, we fix one component in three components, and coupled equations are used to solve the other two components. Decoupling of the two coupled components is computed via singular value decomposition (SVD) of the coupled solution. Specifically, the proposed iterative procedure relies on two steps: The first step is to transform the original problem into a spatial-temporal coupled problem under a given random variable, and the spatial vector is computed via SVD of the spatial-temporal solution. The second step is to transform the original problem into a stochastic-temporal coupled problem under the spatial vector obtained by the first step, and the updated random variable is then computed via SVD of the stochastic-temporal solution. The stochastic-temporal problem is a one-dimensional second-order stochastic ordinary differential equation (SODE). An efficient sampling method is used to solve the SODE of hundreds of dimensions, which avoids the curse of dimensionality in high-dimensional stochastic spaces. The proposed method is considered to be a weak-intrusive method and it combines the advantages of intrusive and non-intrusive methods.

The paper is organized as follows: Approximations of the stochastic solution are introduced in Section 2, including the PC-based approximation and the PGD expansion. In Section 3, we present an alternating iterative algorithm to solve the components of each triplet of the stochastic

solution. High-dimensional stochastic problems are briefly discussed in Section 4. Following this, the algorithm implementation of the proposed method is elaborated in Section 5. In Section 6, three numerical examples of low- and high-dimensional cases are presented to demonstrate the performance of the proposed method, and conclusions and discussions follow in Section 7.

2. Approximation of stochastic solutions

To simplify the representation, we introduce the stochastic operator $\mathscr{L} = \mathbf{M}(\theta)\frac{\partial^2}{\partial t^2} + \mathbf{C}(\theta)\frac{\partial}{\partial t} + \mathbf{K}(\theta)$. Eq. (1) thus becomes

$$\mathscr{L}\mathbf{u}(t,\theta) = \mathbf{F}(t,\theta). \tag{2}$$

In the context of PC-based methods [4, 15], the stochastic solution $\mathbf{u}(t,\theta)$ is approximated in the form

$$\mathbf{u}(t,\theta) = \sum_{i=1}^{P} \Gamma_i(\theta)\mathbf{d}_i(t), \tag{3}$$

where $\{\Gamma_i(\theta)\}_{i=1}^P$ are a set of PC (or generalized PC) basis, $\{\mathbf{d}_i(t)\}_{i=1}^P$ are deterministic time-dependent vectors that need to be solved and $\mathbf{d}_i(t_j) \in \mathbb{R}^n$ for each time $t_j \in [0,T]$. The number P of the truncated term is given by $P = \frac{(r+p)!}{r!p!}$, where r is the number of random variables and p is the order of PC basis. Substitute Eq. (3) into Eq. (2) and apply the stochastic Galerkin projection [19] we get a system of equations about unknown $\{\mathbf{d}_i(t)\}_{i=1}^P$,

$$\sum_{i=1}^{P} \mathbb{E}\left\{\Gamma_j(\theta)\Gamma_i(\theta)\mathscr{L}\right\}\mathbf{d}_i(t) = \mathbb{E}\left\{\Gamma_j(\theta)\mathbf{F}(t,\theta)\right\},\ j = 1,\cdots,P, \tag{4}$$

which can be written as an augmented system

$$\begin{bmatrix} \mathscr{L}_{\Gamma,11} & \cdots & \mathscr{L}_{\Gamma,1P} \\ \vdots & \ddots & \vdots \\ \mathscr{L}_{\Gamma,P1} & \cdots & \mathscr{L}_{\Gamma,PP} \end{bmatrix} \begin{bmatrix} \mathbf{d}_1(t) \\ \vdots \\ \mathbf{d}_P(t) \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{\Gamma,1}(t) \\ \vdots \\ \mathbf{F}_{\Gamma,P}(t) \end{bmatrix}, \tag{5}$$

where the operator $\mathscr{L}_{\Gamma,ij} = \mathbb{E}\left\{\Gamma_i(\theta)\Gamma_j(\theta)\mathscr{L}\right\}$, the vector $\mathbf{F}_{\Gamma,j}(t) = \mathbb{E}\left\{\Gamma_j(\theta)\mathbf{F}(t,\theta)\right\}$, $\mathbb{E}\{\cdot\}$ is the expectation operator. It is noted that the size of the solution $\left[\mathbf{d}_1^T(t_j),\cdots,\mathbf{d}_P^T(t_j)\right]^T$ (at the time t_j) of Eq. (5) is $n \times P$, which is much larger than the original stochastic problem. For instance, the size

of Eq. (5) is 5.15×10^7 when $n = 1 \times 10^4$, $r = 100$ and $p = 2$, which leads to the curse of dimensionality and the computational effort for solving Eq. (5) is very high. Some methods have been proposed to reduce the computational effort of classical PC-based methods [23, 24, 40].

In this paper, we develop a new method to solve Eq. (2). Inspired by the expansion of PGD methods [33], we let the stochastic solution $\mathbf{u}(t,\theta)$ be a form $\mathbf{u}(t,\theta) = \sum_{i=1}^{\infty} \lambda_i(\theta)\,\mathbf{d}_i g_i(t)$ and truncate it at the k-th term,

$$\mathbf{u}_k(t,\theta) = \sum_{i=1}^{k} \lambda_i(\theta)\,\mathbf{d}_i g_i(t) = \mathbf{D}\mathbf{G}(t)\mathbf{\Lambda}(\theta), \tag{6}$$

where $\mathbf{D} = [\mathbf{d}_1, \cdots, \mathbf{d}_k] \in \mathbb{R}^{n\times k}$, $\mathbf{G}(t) = \mathrm{diag}\left[g_1(t), \cdots, g_k(t)\right]$ and $\mathbf{\Lambda}(\theta) = [\lambda_1(\theta), \cdots, \lambda_k(\theta)]^T \in \mathbb{R}^k$, $\lambda_i(\theta) \in \mathcal{S},\ i = 1, \cdots, k$. $\mathcal{S}$ is a stochastic space for real valued random variables with second moments.

A suitable way is to solve the triplet $\{\lambda_i(\theta), \mathbf{d}_i, g_i(t)\}$ one by one since it is not easy to determine $\{\lambda_i(\theta), \mathbf{d}_i, g_i(t)\}_{i=1}^{k}$ once time. Based on this idea, we assume that the first $k-1$ terms $\{\lambda_i(\theta), \mathbf{d}_i, g_i(t)\}_{i=1}^{k-1}$ have been determined and the goal is to solve the k-th triplet $\{\lambda_k(\theta), \mathbf{d}_k, g_k(t)\}$. We rewrite Eq. (6) as

$$\mathbf{u}_k(t,\theta) = \mathbf{u}_{k-1}(t,\theta) + \lambda_k(\theta)\,\mathbf{d}_k g_k(t), \tag{7}$$

where $\mathbf{u}_{k-1}(t,\theta) = \sum_{i=1}^{k-1} \lambda_i(\theta)\,\mathbf{d}_i g_i(t)$. A new stochastic dynamics equation about the unknown triplet $\{\lambda_k(\theta), \mathbf{d}_k, g_k(t)\}$ is obtained by transforming Eq. (2) into

$$\mathscr{L}\{\lambda_k \mathbf{d}_k g_k(t)\} = \mathbf{F}_k(t,\theta), \tag{8}$$

where $\mathbf{F}_k(t,\theta) = \mathbf{F}(t,\theta) - \mathscr{L}\mathbf{u}_{k-1}(t,\theta)$.

The classical PGD method is an available way to iteratively compute $\lambda_k(\theta)$, $\mathbf{d}_k$ and $g_k(t)$ in (8), which is corresponding to

Step 1: For a given random variable $\lambda_k(\theta)$ and a given spatial vector $\mathbf{d}_k$,

$$\left[\mathbf{d}_k^T \mathbb{E}\left\{\lambda_k^2(\theta)\mathscr{L}\right\}\mathbf{d}_k\right] g_k(t) = \mathbf{d}_k^T \mathbb{E}\{\lambda_k(\theta)\mathbf{F}(t,\theta)\} \tag{9}$$

is used to compute the temporal function $g_k(t)$. Eq. (9) is a second-order ODE and existing numerical techniques [41] can be used to solve it efficiently.

Step 2: For the given random variable $\lambda_k(\theta)$ (same value as that in *Step 1*) and the temporal function $g_k(t)$ obtained by *Step 1*,

$$\left[\int_0^T g_k(t)\mathbb{E}\left\{\lambda_k^2(\theta)\mathscr{L}\right\}g_k(t)dt\right]\mathbf{d}_k = \int_0^T g_k(t)\mathbb{E}\left\{\lambda_k(\theta)\mathbf{F}\left(t,\theta\right)\right\}dt \tag{10}$$

is used to compute the spatial vector $\mathbf{d}_k$. Eq. (10) is a time-independent finite element equation and classical FEM numerical methods [1, 42] can be used to solve it efficiently. It is noted that we only adopt FEM for the discretization in this paper but other discretization techniques can also be used, e.g. the finite volume method and the finite difference method.

Step 3: For the spatial vector $\mathbf{d}_k$ obtained by *Step 2* and the temporal function $g_k(t)$ obtained by *Step 1*,

$$\left[\mathbf{d}_k^T\int_0^T g_k(t)\mathscr{L}g_k(t)dt\mathbf{d}_k\right]\lambda_k(\theta) = \mathbf{d}_k^T\int_0^T g_k(t)\mathbf{F}\left(t,\theta\right)dt \tag{11}$$

is used to compute the updated random variable $\lambda_k(\theta)$. Eq. (11) is a time-independent one-dimensional stochastic algebraic equation and classical numerical techniques can be used to solve it, e.g. PC-based methods and sampling methods [43].

The final solution of the triplet $\{\lambda_k(\theta), \mathbf{d}_k, g_k(t)\}$ is obtained by iteratively solving Eq. (9), (10) and (11). However, the above PGD iteration maybe fail to convergence (see the example in Section 6.1 and a similar conclusion can also be found in [33] for the deterministic problems with non-symmetric differential operators). As far as the stochastic dynamic analysis is concerned in this paper, the deterministic matrix $\int_0^T g_k(t)\mathbb{E}\left\{\lambda_k^2(\theta)\mathscr{L}\right\}g_k(t)dt \in \mathbb{R}^{n\times n}$ may be ill-conditioned for solving the spatial vector $\mathbf{d}_k$. Also, the random variable $\mathbf{d}_k^T\left(\int_0^T g_k(t)\mathscr{L}g_k(t)dt\right)\mathbf{d}_k \in \mathcal{S}$ may take a value of zero (or very close to zero), which makes Eq. (11) unstable to compute the random variable $\lambda_k(\theta)$. In our experience, the integral operations of $g_k(t)$, $\dot{g}_k(t)$, $\ddot{g}_k(t)$ in Eq. (10) and (11) usually induce the ill conditions and unstableness, that is, $\int_0^T g_k(t)\dot{g}_k(t)dt > 0$ (or < 0) and $\int_0^T g_k(t)\ddot{g}_k(t)dt > 0$ (or < 0) cannot hold for some problems and their values may be zero.

3. A weak-intrusive SFEM for solving stochastic dynamics equations

In order to overcome the above shortcoming of the classical PGD iteration, a residual minimization method is used in [33]. In this paper, we introduce a new iterative procedure to avoid

the shortcoming. Specifically, we firstly consider a spatial-temporal coupled representation $\widetilde{\mathbf{d}}_k(t)$ instead of the separated form $\mathbf{d}_k g_k(t)$ in Eq. (8),

$$\mathscr{L}\left[\lambda_k(\theta)\widetilde{\mathbf{d}}_k(t)\right] = \mathbf{F}_k(t,\theta). \tag{12}$$

The stochastic Galerkin method is used to solve Eq. (12), that is, for a given random variable $\lambda_k(\theta)$, we solve $\widetilde{\mathbf{d}}_k(t)$ by

$$\mathscr{L}_k\widetilde{\mathbf{d}}_k(t) = \mathbf{f}_k(t), \tag{13}$$

where the deterministic operator $\mathscr{L}_k = \mathbb{E}\left\{\lambda_k^2(\theta)\mathscr{L}\right\} = \mathbb{E}\left\{\lambda_k^2(\theta)\mathbf{M}(\theta)\right\}\frac{\partial^2}{\partial t^2} + \mathbb{E}\left\{\lambda_k^2(\theta)\mathbf{C}(\theta)\right\}\frac{\partial}{\partial t} + \mathbb{E}\left\{\lambda_k^2(\theta)\mathbf{K}(\theta)\right\}$ and the deterministic force vector is given by $\mathbf{f}_k(t) = \mathbb{E}\{\lambda_k(\theta)\mathbf{F}_k(t,\theta)\} = \mathbb{E}\{\lambda_k(\theta)\left[\mathbf{F}(t,\theta) - \mathscr{L}\mathbf{u}_{k-1}(t,\theta)\right]\}$. Eq. (13) can be considered as a deterministic structural dynamics equation and it has good numerical stability. Existing numerical methods can be used to solve $\widetilde{\mathbf{d}}_k(t) \in \mathbb{R}^{n\times n_t}$ (n_t is the total time step) in Eq. (13) efficiently and accurately, e.g. the Newmark method and the central difference method [44].

Based on the solution $\widetilde{\mathbf{d}}_k(t)$ obtained by Eq. (13), the Eq. (11)-like way to compute the random variable $\lambda_k(\theta)$ still causes numerical unstableness. In order to avoid this problem, we decouple $\widetilde{\mathbf{d}}_k(t)$ in spatial and temporal spaces. $\widetilde{\mathbf{d}}_k(t)$ is approximated by the rank-one SVD,

$$\widetilde{\mathbf{d}}_k(t) \approx \mathbf{d}_k g_k(t),\ \mathbf{d}_k(t) \in \mathbb{R}^n,\ g_k(t) \in \mathbb{R}^{1\times n_t}, \tag{14}$$

which provides an optimal rank-one approximation of $\widetilde{\mathbf{d}}_k(t)$ [45]. A more accurate SVD approximation of $\widetilde{\mathbf{d}}_k(t)$ is $\widetilde{\mathbf{d}}_k(t) \approx \mathbf{d}_{k,1}g_{k,1}(t) + \mathbf{d}_{k,2}g_{k,2}(t) + \cdots$. In this paper, only the first term is retained and $\mathbf{d}_k$ in the stochastic solution Eq. (7) is approximated by $\mathbf{d}_{k,1}$. Although the single-term approximation in Eq. (14) has low accuracy, the accuracy of the stochastic solution $\mathbf{u}(t,\theta)$ still increases as the number k of retained terms increases. In other words, the accuracy of $\mathbf{u}(t,\theta)$ can be improved by summing a series of low-accuracy terms. In practical computations, we orthogonalize the solution $\mathbf{d}_k$ by Gram-Schmidt orthogonalization, which is corresponding to

$$\mathbf{d}_k = \mathbf{d}_k - \sum_{i=1}^{k-1}\frac{\mathbf{d}_k^T\mathbf{d}_i}{\mathbf{d}_i^T\mathbf{d}_i}\mathbf{d}_i = (\mathbf{I}_n - \mathbf{D}\mathbf{D}^T)\mathbf{d}_k, \tag{15}$$

where $\{\mathbf{d}_i\}_{i=1}^{k-1}$ are orthonormal vectors meeting $\mathbf{d}_i^T\mathbf{d}_j = \delta_{ij}$, δ_{ij} is the Kronecker delta function, $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ is the identity matrix.

Based on the spatial vector $\mathbf{d}_k$ obtained by Eq. (14), we consider a stochastic-temporal coupled form $\widetilde{g}_k(t,\theta)$ instead of the separated form $\lambda_k(\theta)g_k(t)$ in Eq. (8),

$$\mathscr{L}\left[\mathbf{d}_k\widetilde{g}_k(t,\theta)\right] = \mathbf{F}_k(t,\theta) \tag{16}$$

which is solved via a Galerkin procedure,

$$\mathscr{S}_k\widetilde{g}_k(t,\theta) = h_k(t,\theta), \tag{17}$$

where the stochastic operator $\mathscr{S}_k = \mathbf{d}_k^T\mathscr{L}\mathbf{d}_k = m_k(\theta)\frac{\partial^2}{\partial t^2} + c_k(\theta)\frac{\partial}{\partial t} + k_k(\theta)$, the random variables $m_k(\theta) = \mathbf{d}_k^T\mathbf{M}(\theta)\mathbf{d}_k,\ c_k(\theta) = \mathbf{d}_k^T\mathbf{C}(\theta)\mathbf{d}_k,\ k_k(\theta) = \mathbf{d}_k^T\mathbf{K}(\theta)\mathbf{d}_k \in \mathcal{S}$, $h_k(t,\theta) = \mathbf{d}_k^T\left[\mathbf{F}(t,\theta) - \mathscr{L}\mathbf{u}_{k-1}(t,\theta)\right]$. To solve Eq. (17), we introduce a sample-based method [43] and adopt the Newmark method for the explanation of this method. We remark that other methods can also be implemented by the proposed scheme, e.g. central difference method.

According to the Newmark method, the solution $\widetilde{g}_k(t,\theta)$ of Eq. (17) at the time $t+\Delta t$ is solved by

$$q_k(\theta)\widetilde{g}_k(t+\Delta t,\theta) = \widetilde{h}_k(t+\Delta t,\theta), \tag{18}$$

where the random variable $q_k(\theta)$ is

$$q_k(\theta) = \alpha_1 m_k(\theta) + \alpha_2 c_k(\theta) + k_k(\theta) \in \mathcal{S} \tag{19}$$

and the random variable $\widetilde{h}_k(t+\Delta t,\theta)$ is given by

$$\widetilde{h}_k(t+\Delta t,\theta) = h_k(t+\Delta t,\theta) + \alpha_{9,k}(\theta)\widetilde{g}_k(t,\theta) + \alpha_{10,k}(\theta)\dot{\widetilde{g}}_k(t,\theta) + \alpha_{11,k}(\theta)\ddot{\widetilde{g}}_k(t,\theta) \in \mathcal{S}. \tag{20}$$

The parameters $\alpha_1,\cdots,\alpha_8$ are inherited from the classical Newmark method. They are given by $\alpha_1 = \frac{1}{\beta\Delta t^2}$, $\alpha_2 = \frac{\gamma}{\beta\Delta t}$, $\alpha_3 = \frac{1}{\beta\Delta t}$, $\alpha_4 = \frac{1}{2\beta} - 1$, $\alpha_5 = \frac{\gamma}{\beta} - 1$, $\alpha_6 = \frac{\Delta t}{2}\left(\frac{\gamma}{\beta} - 2\right)$, $\alpha_7 = \Delta t\,(1-\gamma)$, $\alpha_8 = \gamma\Delta t$. These parameters are fixed for the given time discretization and the chosen parameters γ, β. The parameters $\alpha_{9,k}(\theta)$, $\alpha_{10,k}(\theta)$ and $\alpha_{11,k}(\theta)$ are random variables given by $\alpha_{9,k}(\theta) = \alpha_1 m_k(\theta) + \alpha_2 c_k(\theta)$, $\alpha_{10,k}(\theta) = \alpha_3 m_k(\theta) + \alpha_5 c_k(\theta)$, $\alpha_{11,k}(\theta) = \alpha_4 m_k(\theta) + \alpha_6 c_k(\theta)$. It is noted that $q_k(\theta) > 0$ holds for all $\theta \in \Theta$, thus Eq. (18) is numerically stable for solving the solution $\widetilde{g}_k(t,\theta)$.

In order to avoid the curse of dimensionality in high-dimensional stochastic spaces, we solve Eq. (18) by a sampling method,

$$\widetilde{g}_k\left(t+\Delta t,\theta^{(i)}\right) = \frac{\widetilde{h}_k\left(t+\Delta t,\theta^{(i)}\right)}{q_k\left(\theta^{(i)}\right)} \tag{21}$$

for $i = 1, \cdots, n_s$, which solves n_s solutions of Eq. (18). However, solving Eq. (18) for each sample $\theta^{(i)}$, $i = 1, \cdots, n_s$ is a bit time-consuming since the computational cost is strongly dependent on the sample size n_s. An accelerated method is to adopt a vector-based form,

$$\widetilde{g}_k(t + \Delta t, \boldsymbol{\theta}) = \widetilde{h}_k(t + \Delta t, \boldsymbol{\theta}) \oslash q_k(\boldsymbol{\theta}), \tag{22}$$

where $\boldsymbol{\theta} = \{\theta^{(i)}\}_{i=1}^{n_s}$ is the sample vector (or matrix), $\widetilde{g}_k(t + \Delta t, \boldsymbol{\theta})$, $h_k(t + \Delta t, \boldsymbol{\theta})$, $q_k(\boldsymbol{\theta}) \in \mathbb{R}^{n_s}$, the operator $\oslash$ represents the element-wise division of the vectors $\widetilde{h}_k(t + \Delta t, \boldsymbol{\theta})$ and $q_k(\boldsymbol{\theta})$, also known as the Hadamard division operator. Eq. (22) is easy to implement and its computational effort is weakly dependent on the sample size. Further, we can compute the first and second derivatives of $\widetilde{g}_k(t, \theta)$ by

$$\ddot{\widetilde{g}}_k(t + \Delta t, \theta) = \alpha_1 \left(\widetilde{g}_k(t + \Delta t, \theta) - \widetilde{g}_k(t, \theta) - \alpha_3 \dot{\widetilde{g}}_k(t, \theta) - \alpha_4 \ddot{\widetilde{g}}_k(t, \theta) \right) \tag{23}$$

and

$$\dot{\widetilde{g}}_k(t + \Delta t, \theta) = \dot{\widetilde{g}}_k(t, \theta) + \alpha_7 \ddot{\widetilde{g}}_k(t, \theta) + \alpha_8 \ddot{\widetilde{g}}_k(t + \Delta t, \theta), \tag{24}$$

which can also be computed by use of the sample vectors $\widetilde{g}_k(t, \boldsymbol{\theta})$, $\dot{\widetilde{g}}_k(t, \boldsymbol{\theta})$, $\ddot{\widetilde{g}}_k(t, \boldsymbol{\theta})$, $\widetilde{g}_k(t + \Delta t, \boldsymbol{\theta}) \in \mathbb{R}^{n_s}$.

Similar to Eq. (14), we decouple $\widetilde{g}_k(t, \theta)$ in stochastic and temporal spaces,

$$\widetilde{g}_k(t, \theta) \approx \lambda_k(\theta) g_k(t), \tag{25}$$

In the practical computation, it is calculated in a sample form $\widetilde{g}_k(t, \boldsymbol{\theta}) \in \mathbb{R}^{n_s \times n_t}$ by the rank-one SVD,

$$\widetilde{g}_k(t, \boldsymbol{\theta}) \approx \lambda_k(\boldsymbol{\theta}) g_k(t), \ \lambda_k(\boldsymbol{\theta}) \in \mathbb{R}^{n_s}, \tag{26}$$

where $\lambda_k(\boldsymbol{\theta}) \in \mathbb{R}^{n_s}$ is the sample vector of the random variable $\lambda_k(\theta)$ and $g_k(t) \in \mathbb{R}^{1 \times n_t}$ is the normalized deterministic vector. Eq. (26) provides a sample-based description for the random variable $\lambda_k(\theta)$ and the probability characteristic of $\lambda_k(\theta)$ can be computed via the random samples $\lambda_k(\boldsymbol{\theta}) \in \mathbb{R}^{n_s}$.

The above iterative procedure is summarized as following,

$$\boxed{\rightarrow \lambda_{k,j}(\boldsymbol{\theta}) \xrightarrow{\widetilde{\mathbf{d}}_{k,j}(t)} \mathbf{d}_{k,j} \xrightarrow{\widetilde{g}_{k,j}(t,\boldsymbol{\theta})} \begin{cases} \lambda_{k,j+1}(\boldsymbol{\theta}) \rightarrow \\ g_{k,j+1}(t) \end{cases}} \Rightarrow \{\lambda_k(\boldsymbol{\theta}), \mathbf{d}_k, g_k(t)\}, \tag{27}$$

where the iteration in the box is used to compute the triplet $\{\lambda_k(\theta), \mathbf{d}_k, g_k(t)\}$. The solutions $\widetilde{\mathbf{d}}_{k,j}(t)$ and $\widetilde{g}_{k,j}(t,\theta)$ of the j-th loop are intermediates. They are only used to compute $\mathbf{d}_{k,j}$ and $\lambda_{k,j+1}(\theta)$ and do not arise in the next iteration. For each iteration, only the random variable $\lambda_{k,j}(\theta)$ (or the vector $\mathbf{d}_{k,j}$ if starting from $\mathbf{d}_0$) is inherited from the previous iteration, which is very different from the classical PGD iteration. In the classical PGD method, two of $\lambda_{k,j}(\theta)$, $\mathbf{d}_{k,j}$, $g_{k,j}(t)$ are passed to the next iteration. It is also seen from Eq. (27) that the original stochastic dynamics equation is transformed into the deterministic dynamics equation and the one-dimensional SODE. The deterministic dynamics equation (i.e. Eq. (13)) is used to solve the solution $\widetilde{\mathbf{d}}_{k,j}(t)$ and only deterministic computations are involved, thus it is easily applied to large-scale problems. The one-dimensional SODE (i.e. Eq. (17)) is used to solve the stochastic solution $\widetilde{g}_{k,j}(t,\theta)$. The sampling method proposed to solve the one-dimensional SODE has low computational effort and it can be easily applied to high-dimensional stochastic problems.

By use of the above iteration, we compute each triplet $\{\lambda_k(\theta), \mathbf{d}_k, g_k(t)\}$ in a sequential way. However, the stochastic solution $\mathbf{u}_k(t,\theta)$ approximated by Eq. (8) does not exactly meet the original problem Eq. (2), and it may have low accuracy for some problems. We introduce a recalculation process to improve the accuracy of the stochastic solution $\mathbf{u}_k(t,\theta)$. We rewrite Eq. (2) under the approximation Eq. (6),

$$\mathscr{L}\left[\mathbf{D}\mathbf{G}(t)\mathbf{\Lambda}(\theta)\right] = \mathbf{F}(t,\theta). \tag{28}$$

Similar to Eq. (16), we solve a stochastic-temporal coupled solution $\widetilde{\mathbf{G}}(t,\theta) \in \mathbb{R}^{k\times n_t}$ based on the known matrix $\mathbf{D} \in \mathbb{R}^{n\times k}$ instead of computing the separated form $\mathbf{G}(t)\mathbf{\Lambda}(\theta)$ in Eq. (28). By use of the Galerkin approach we have

$$\left[\mathbf{D}^T\mathscr{L}\mathbf{D}\right]\widetilde{\mathbf{G}}(t,\theta) = \mathbf{D}^T\mathbf{F}(t,\theta), \tag{29}$$

which is rewritten as

$$\widetilde{\mathbf{m}}(\theta)\,\ddot{\widetilde{\mathbf{G}}}(t,\theta) + \widetilde{\mathbf{c}}(\theta)\,\dot{\widetilde{\mathbf{G}}}(t,\theta) + \widetilde{\mathbf{k}}(\theta)\,\widetilde{\mathbf{G}}(t,\theta) = \widetilde{\mathbf{f}}(t,\theta), \tag{30}$$

where $\widetilde{\mathbf{m}}(\theta) = \mathbf{D}^T\mathbf{M}(\theta)\mathbf{D}$, $\widetilde{\mathbf{c}}(\theta) = \mathbf{D}^T\mathbf{C}(\theta)\mathbf{D}$, $\widetilde{\mathbf{k}}(\theta) = \mathbf{D}^T\mathbf{K}(\theta)\mathbf{D} \in \mathbb{R}^{k\times k}$, $\widetilde{\mathbf{f}}(t,\theta) = \mathbf{D}^T\mathbf{F}(t,\theta) \in \mathbb{R}^k$. We solve Eq. (30) by using the sampling method, that is, the solution of Eq. (30) is solved for each random sample. Total computational costs are still low thanking to the fact that the size k of

Eq. (30) is usually small. The proposed method is similar to classical reduced-order methods, e.g. proper orthogonal decomposition methods [21, 46], but an efficient way is proposed to construct the reduced basis.

In order to obtain the Eq. (6)-like decoupled approximation, we expand the solution $\widetilde{\mathbf{G}}(t,\theta)$ in Eq. (29) by use of Karhunen–Loève (KL) expansion [22, 47, 48],

$$\widetilde{\mathbf{G}}(t,\theta) = \sum_{i=1}^{m} \lambda_i^*(\theta) g_i^*(t),\ g_i^*(t) \in \mathbb{R}^{k\times n_t}, \tag{31}$$

where m is the number of truncated terms, $\{\lambda_i^*(\theta)\}$ and $\{g_j^*(t)\}$ are eigenvalues and eigenvectors of the autocorrelation matrix $\mathbf{C}_{\widetilde{\mathbf{G}}\widetilde{\mathbf{G}}}$ of $\widetilde{\mathbf{G}}(t,\theta)$, respectively. They meet

$$\mathbb{E}\left\{\lambda_i^*(\theta)\,\lambda_j^*(\theta)\right\} = \mathbb{E}\left\{\lambda_i^{*2}(\theta)\right\}\delta_{ij},\ \mathrm{Tr}\left(g_i^{*T}(t)\,g_j^*(t)\right) = \delta_{ij}, \tag{32}$$

where $\mathrm{Tr}(\cdot)$ represents the trace of the matrix. In practice, the autocorrelation matrix $\mathbf{C}_{\widetilde{\mathbf{G}}\widetilde{\mathbf{G}}}$ is calculated by

$$\mathbf{C}_{\widetilde{\mathbf{G}}\widetilde{\mathbf{G}}} = \left[\mathbb{E}\left\{\widetilde{\mathbf{G}}(\mathbf{p}_i,\theta)\,\widetilde{\mathbf{G}}\left(\mathbf{p}_j,\theta\right)\right\}\right]_{i,j=1}^{kn_t} \in \mathbb{R}^{kn_t\times kn_t}, \tag{33}$$

where $\{\mathbf{p}_i\}_i$ is the discretization of the physical coordinate (k,t), $\mathbf{p}_i = \left(k^{(i)}, t^{(i)}\right)$, $k^{(i)} \in [1,\cdots,k]$, $t^{(i)} \in [t_1,\cdots,t_{n_t}]$. It is noted that the size of the eigenvectors of $\mathbf{C}_{\widetilde{\mathbf{G}}\widetilde{\mathbf{G}}}$ is $kn_t \times 1$ and $g_i^*(t) \in \mathbb{R}^{k\times n_t}$ in Eq. (31) is obtained by rearranging the i-th eigenvector according to the physical coordinate (k,t). Based on Eq. (31), we reconstruct the stochastic solution $\mathbf{u}(t,\theta)$ as

$$\mathbf{u}(t,\theta) = \mathbf{D}\widetilde{\mathbf{G}}(t,\theta) = \sum_{i=1}^{m} \lambda_i^*(\theta)\mathbf{D}g_i^*(t). \tag{34}$$

In the sense of minimizing the mean squared error, Eq. (34) provides the optimal stochastic-deterministic decomposition of the stochastic solution $\mathbf{u}(t,\theta)$ under the known matrix $\mathbf{D}$. Also, we can estimate the PDF of the stochastic solution $u\left(x_k,t_j,\theta\right)$ at the point $\left(x_k,t_j\right)$ by the random samples $\sum_{i=1}^{m} \lambda_i^*(\boldsymbol{\theta})\mathbf{D}(x_k)\,g_i^*\left(t_j\right) \in \mathbb{R}^{n_s}$. In this way, the proposed method provides a weak-intrusive approximation of PDF of the stochastic solution, which combines the high efficiency of intrusive methods and the weakly dimensionality-dependent property of non-intrusive methods.

Although we only consider the linear case in this paper, the proposed method also works on the stochastic nonlinear dynamics problems. The decoupling Eq. (14) and (25) are independent

of the type of problems and can be computed efficiently for both linear and nonlinear stochastic problems. However, for nonlinear problems, Eq. (13) becomes deterministic nonlinear dynamics equations and Eq. (17) becomes second-order nonlinear SODEs. Solving Eq. (13) and (17) of nonlinear stochastic dynamics problems are usually time-consuming.

4. High-dimensional stochastic problems

In this section, we show that the proposed method can be applied to high-dimensional stochastic problems without any modification. Without loss of generality, we assume that matrices $\mathbf{M}$, $\mathbf{C}$ and the force vector $\mathbf{F}(t)$ are deterministic and the stochastic matrix $\mathbf{K}(\theta)$ can be represented as a separated form

$$\mathbf{K}(\theta) = \sum_{j=0}^{r} \xi_j(\theta)\,\mathbf{K}_j, \tag{35}$$

where $\xi_0(\theta) \equiv 1$ and $\mathbf{K}_0$ is the deterministic component of the stochastic stiffness matrix $\mathbf{K}(\theta)$. For the unseparated forms, series expansion methods can be used to provide separated representations, e.g. KL expansion and PC expansion. In this paper, we introduce high-dimensional stochastic spaces by setting a large value r in Eq. (35).

We rewrite $\mathscr{L} = \mathscr{L}_1 + \mathbf{K}(\theta)$, where $\mathscr{L}_1 = \mathbf{M}\frac{\partial^2}{\partial t^2} + \mathbf{C}\frac{\partial}{\partial t}$ represents the deterministic component of the stochastic operator $\mathscr{L}$. We still adopt Eq. (13) to compute $\widetilde{\mathbf{d}}_k(t)$, which is corresponding to

$$\mathscr{L}_1\widetilde{\mathbf{d}}_k(t) + \left(\sum_{j=0}^{r}\beta_{jkk}\mathbf{K}_j\right)\widetilde{\mathbf{d}}_k(t) = \mathbf{f}_k(t), \tag{36}$$

where

$$\mathbf{f}_k(t) = \frac{\mathbb{E}\{\lambda_k(\theta)\}}{\mathbb{E}\left\{\lambda_k^2(\theta)\right\}}\mathbf{F}(t) - \sum_{i=1}^{k-1}\left(\mathscr{L}_1\left[\mathbf{d}_i g_i(t)\right] + \left(\sum_{j=0}^{r}\beta_{jik}\mathbf{K}_j\right)\mathbf{d}_i g_i(t)\right) \tag{37}$$

and

$$\beta_{ijk} = \frac{\mathbb{E}\left\{\xi_i(\theta)\,\lambda_j(\theta)\,\lambda_k(\theta)\right\}}{\mathbb{E}\left\{\lambda_k^2(\theta)\right\}}. \tag{38}$$

The total number of $\left\{\beta_{ijk}\right\}_{i,j}$ to be computed is $(r+1)\times k$. By adopting a sample-based strategy, we

compute all coefficients $\left\{\beta_{ijk}\right\}_{i,j}$ at once, which is corresponding to

$$\boldsymbol{\beta}_k = \frac{\boldsymbol{\xi}(\boldsymbol{\theta})^T \left(\boldsymbol{\lambda}(\boldsymbol{\theta}) \odot \lambda_k(\boldsymbol{\theta})\right)}{\lambda_k(\boldsymbol{\theta})^T \lambda_k(\boldsymbol{\theta})} = \begin{bmatrix} \beta_{01k} & \cdots & \beta_{0kk} \\ \beta_{11k} & \cdots & \beta_{1kk} \\ \vdots & \ddots & \vdots \\ \beta_{r1k} & \cdots & \beta_{rkk} \end{bmatrix} \in \mathbb{R}^{(r+1)\times k}, \tag{39}$$

where $\boldsymbol{\beta}_k$ is a matrix including all coefficients $\left\{\beta_{ijk}\right\}_{i,j}$, the sample matrices $\boldsymbol{\xi}(\boldsymbol{\theta})$, $\boldsymbol{\lambda}(\boldsymbol{\theta})$ and the sample vector $\lambda_k(\boldsymbol{\theta})$ are given by

$$\begin{aligned} \boldsymbol{\xi}(\boldsymbol{\theta}) &= [\xi_0(\boldsymbol{\theta}), \xi_1(\boldsymbol{\theta}), \cdots, \xi_r(\boldsymbol{\theta})] = \begin{bmatrix} 1 & \xi_1(\theta_1) & \cdots & \xi_r(\theta_1) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \xi_1(\theta_{n_s}) & \cdots & \xi_r(\theta_{n_s}) \end{bmatrix} \in \mathbb{R}^{n_s\times(r+1)}, \\ \lambda_k(\boldsymbol{\theta}) &= \left[\lambda_k(\theta_1), \cdots, \lambda_k(\theta_{n_s})\right]^T \in \mathbb{R}^{n_s\times 1}, \\ \boldsymbol{\lambda}(\boldsymbol{\theta}) &= [\lambda_1(\boldsymbol{\theta}), \cdots, \lambda_k(\boldsymbol{\theta})] \in \mathbb{R}^{n_s\times k}, \end{aligned} \tag{40}$$

and the operator $\odot$ represents the element-by-element multiplication, which meets

$$\boldsymbol{\lambda}(\boldsymbol{\theta}) \odot \lambda_k(\boldsymbol{\theta}) = \begin{bmatrix} \lambda_1(\theta_1)\lambda_k(\theta_1) & \cdots & \lambda_k(\theta_1)\lambda_k(\theta_1) \\ \vdots & \ddots & \vdots \\ \lambda_1(\theta_{n_s})\lambda_k(\theta_{n_s}) & \cdots & \lambda_k(\theta_{n_s})\lambda_k(\theta_{n_s}) \end{bmatrix} \in \mathbb{R}^{n_s\times k}. \tag{41}$$

In this way, all random variables $\left\{\xi_j(\theta)\right\}_{j=0}^{r}$ are embedded into the sample matrix $\boldsymbol{\xi}(\boldsymbol{\theta})$. The matrix $\boldsymbol{\xi}(\boldsymbol{\theta})^T \left(\boldsymbol{\lambda}(\boldsymbol{\theta}) \odot \lambda_k(\boldsymbol{\theta})\right)$ can thus be computed efficiently for hundreds or even more stochastic dimensions. Similarly, we recall the Eq. (19) and (20) for solving $\widetilde{g}_k(t,\theta)$. All random variables are embedded into the random variables $\widetilde{h}_k(t+\Delta t,\theta)$ and $q_k(\theta) = \alpha_1 m_k + \alpha_2 c_k + k_k(\theta)$, where $m_k = \mathbf{d}_k^T \mathbf{M} \mathbf{d}_k$, $c_k = \mathbf{d}_k^T \mathbf{C} \mathbf{d}_k$, $k_k(\theta) = \sum_{j=0}^{r} \xi_j(\theta)\, \mathbf{d}_k^T \mathbf{K}_j \mathbf{d}_k \in \mathcal{S}$. Both $\widetilde{h}_k\left(t+\Delta t, \{\xi_i(\theta)\}_{j=1}^{r}, \theta\right)$ and $q_k\left(\left\{\xi_j(\theta)\right\}_{j=1}^{r}, \theta\right)$ are multivariable functions. It is usually not a simple matter to analyze multivariate functions, although some methods, e.g. sparse PC approach and high-dimensional model representations [24, 49], can be used for this purpose. From another view of point, both $\widetilde{h}_k\left(t+\Delta t, \left\{\xi_j(\theta)\right\}_{j=1}^{r}, \theta\right)$ and $q_k\left(\left\{\xi_j(\theta)\right\}_{j=1}^{r}, \theta\right)$ are random variables. The random sample vectors $\widetilde{h}_k\left(t+\Delta t, \left\{\xi_j(\boldsymbol{\theta})\right\}_{j=1}^{r}, \boldsymbol{\theta}\right) \in \mathbb{R}^{n_s}$ and $q_k\left(\left\{\xi_j(\boldsymbol{\theta})\right\}_{j=1}^{r}, \boldsymbol{\theta}\right) \in \mathbb{R}^{n_s}$ can be computed efficiently even for very high dimensions and they are almost dimensionality-independent.

5. Algorithm implementation

The above iterative procedure for approximating the stochastic solution $\mathbf{u}(t,\theta)$ of linear stochastic structural dynamics equations is summarized in Algorithm 1, which consists of outer and inner loop procedures. The inner loop, which is from step 3 to 11, is used to determine the triplet of $\{\lambda_k(\theta), \mathbf{d}_k, g_k(t)\}$. Before executing the inner loop, we initialize the random samples $\lambda_{k,0}(\boldsymbol{\theta}) \in \mathbb{R}^{n_s}$ in step 2. In our experience, the initialization has little influence on the computational accuracy and efficiency of the proposed method. In practical implementation, any nonzero vector of size n_s can be used as the initial random samples. With the initial random samples, the time-dependent solution $\widetilde{\mathbf{d}}_{k,j}(t) \in \mathbb{R}^{n\times n_t}$ is determined in step 4 by solving linear deterministic structural dynamics

Algorithm 1 Algorithm for solving linear stochastic structural dynamics equations

1: **while** $\varepsilon_{g,k} > \varepsilon_g$ **do**
2: Initialize the random samples $\lambda_{k,0}(\boldsymbol{\theta}) \in \mathbb{R}^{n_s}$
3: **while** $\varepsilon_{l,k,j} > \varepsilon_{l,k}$ **do**
4: Compute $\widetilde{\mathbf{d}}_{k,j}(t) \in \mathbb{R}^{n\times n_t}$ by solving Eq. (13)
5: Compute $\mathbf{d}_{k,j} \in \mathbb{R}^{n}$ via the rank-one SVD Eq. (14)
6: Orthogonalize $\mathbf{d}_{k,j} \perp \{\mathbf{d}_i\}_{i=1}^{k-1}$ by Eq. (15) and normalize $\mathbf{d}_{k,j}$
7: Compute $\widetilde{g}_{k,j}(t,\boldsymbol{\theta}) \in \mathbb{R}^{n_s\times n_t}$ by solving Eq. (17)
8: Compute $\lambda_{k,j+1}(\boldsymbol{\theta}) \in \mathbb{R}^{n_s}$ and $g_{k,j+1}(t) \in \mathbb{R}^{1\times n_t}$ via the rank-one SVD Eq. (26)
9: Compute the local error $\varepsilon_{l,k,j}$
10: $j \leftarrow j+1$
11: **end**
12: Update $\mathbf{u}_k(t,\theta) = \sum_{i=1}^{k-1} \lambda_i(\theta)\mathbf{d}_i g_i(t) + \lambda_k(\theta)\mathbf{d}_k g_k(t)$
13: Compute the global error $\varepsilon_{g,k}$
14: $k \leftarrow k+1$
15: **end**
16: Solve $\widetilde{\mathbf{G}}(t,\theta)$ by Eq. (29) based on the known matrix $\mathbf{D}$
17: Expand $\widetilde{\mathbf{G}}(t,\theta) = \sum_{i=1}^{m} \lambda_i^*(\theta) g_i^*(t)$

equations and step 5 provides the time-independent vector $\mathbf{d}_{k,j} \in \mathbb{R}^n$ via the rank-one SVD of $\widetilde{\mathbf{d}}_{k,j}(t)$. It is noted that $\mathbf{d}_{k,j}$ requires to be orthogonalized and normalized along the whole process. We implement this computation using the Gram-Schmidt orthogonalization Eq. (15) in step 6. With the obtained vector $\mathbf{d}_{k,j}$, one-dimensional linear stochastic dynamics equations (i.e. second-order SODEs) are used to solve the time-dependent solution $\widetilde{g}_{k,j}(t,\boldsymbol{\theta}) \in \mathbb{R}^{n_s \times n_t}$ in step 7. The random samples $\lambda_{k,j+1}(\boldsymbol{\theta}) \in \mathbb{R}^{n_s}$ and $g_{k,j+1}(t) \in \mathbb{R}^{1\times n_t}$ are then computed via SVD in step 8. The outer loop, which is from step 1 to 15, corresponds to recursively build the set of triplets such that the approximate solution in step 12 satisfies Eq. (2). Based on the determined matrix $\mathbf{D}$, a new solution $\widetilde{\mathbf{G}}(t,\theta)$ is solved by k-dimensional linear stochastic structural dynamics equations. Following that, a stochastic-deterministic decomposition of $\widetilde{\mathbf{G}}(t,\theta)$ is executed by KL expansion in step 17. It is noted that we can adopt different sample sizes in step 2 and step 16. In practice, a small size is firstly used in step 2 and a large sample size is used after obtaining the matrix $\mathbf{D}$, which saves a lot of computational costs.

There are two iterative stopping criteria in Algorithm 1, $\varepsilon_{g,k}$ in step 13 and $\varepsilon_{l,k,j}$ in step 9. The globally iterative stopping criterion $\varepsilon_{g,k}$ is defined as

$$\begin{aligned}\varepsilon_{g,k} &= \frac{\|\mathbf{u}_k(t,\theta) - \mathbf{u}_{k-1}(t,\theta)\|}{\|\mathbf{u}_k(t,\theta)\|} = \frac{\mathbb{E}\left\{\lambda_k^2(\theta)\right\}\mathbf{d}_k^T\mathbf{d}_k \int_0^T g_k^2(t)\,dt}{\sum_{i,j=1}^{k}\mathbb{E}\left\{\lambda_i(\theta)\lambda_j(\theta)\right\}\mathbf{d}_i^T\mathbf{d}_j\int_0^T g_i(t)\,g_j(t)\,dt} \\ &= \frac{\mathbb{E}\left\{\lambda_k^2(\theta)\right\}}{\sum_{i=1}^{k}\mathbb{E}\left\{\lambda_i^2(\theta)\right\}},\end{aligned} \tag{42}$$

where $\|\square\| = \mathbb{E}\{\square^T\square\}$. Eq. (42) measures the contribution of the k-th component $\{\lambda_k(\theta), \mathbf{d}_k, g_k(t)\}$ to the stochastic solution $\mathbf{u}_k(t,\theta)$. The approximate solution $\mathbf{u}_k(t,\theta)$ in step 12 converges to the final solution when $\varepsilon_{g,k}$ achieves a specified accuracy ε_g. Similarly, the locally iterative stopping criterion $\varepsilon_{l,k,j}$ is defined as

$$\varepsilon_{l,k,j} = \frac{\left\|\mathbf{d}_{k,j} - \mathbf{d}_{k,j-1}\right\|}{\left\|\mathbf{d}_{k,j}\right\|} = 2 - 2\mathbf{d}_{k,j}^T\mathbf{d}_{k,j-1}, \tag{43}$$

which measures the difference between $\mathbf{d}_{k,j}$ and $\mathbf{d}_{k,j-1}$ and the calculation is stopped when $\mathbf{d}_{k,j}$ is almost the same as $\mathbf{d}_{k,j-1}$. It is noted that $\lambda_{k,j}(\theta)$ converges as $\mathbf{d}_{k,j}$ converges and $\mathbf{d}_{k,j}$ also converges

as $\lambda_{k,j}(\theta)$ converges, thus the stopping criterion only involving one of $\mathbf{d}_{k,j}$ and $\lambda_{k,j}(\theta)$ is reasonable. We can also adopt the stopping criterion $\varepsilon_{l,k,j} = \left\|\lambda_{k,j}(\boldsymbol{\theta}) - \lambda_{k,j-1}(\boldsymbol{\theta})\right\| / \left\|\lambda_{k,j}(\boldsymbol{\theta})\right\|$, which measures the difference between $\lambda_{k,j}(\theta)$ and $\lambda_{k,j-1}(\theta)$.

In the sense of minimizing the mean squared error, the approximation in step 12 in Algorithm 1 is not the optimal stochastic-deterministic decomposition of the stochastic solution $\mathbf{u}_k(t,\theta)$. The iterative error defined in Eq. (42) may fail to check the convergence of the stochastic solution. The true error that is calculated by substituting the random variables $\left\{\lambda_i^*(\theta)\right\}_{i=1}^k$ in Eq. (34) into Eq. (42) is a better error estimator, but we can only compute the true error after solving the final stochastic solution. In order to check the convergence of the stochastic solution in a prior way, we propose an improved iterative error estimation that can be calculated along the iterative process. We name the iterative error calculated by Eq. (42) as the original error and the improved iterative error estimation as the indicator error.

To compute the indicator error, we recall Eq. (6) and rewrite $\mathbf{u}_k(t,\theta)$ as

$$\mathbf{u}_k(t,\theta) = \sum_{i=1}^{k} \lambda_i(\theta)\,\mathbf{d}_i g_i(t) = \mathbf{D}^{\sqcap}(t)\boldsymbol{\Lambda}(\theta), \tag{44}$$

where $\mathbf{D}^{\sqcap}(t) = [\mathbf{d}_1 g_1(t), \cdots, \mathbf{d}_k g_k(t)]$ (with a slight abuse of notation, here $\{g_i(t)\}_{i=1}^k$ are treated as continuous functions instead of discrete vectors) and it meets

$$\int_0^T \mathbf{D}_i^{\sqcap T}(t)\mathbf{D}_j^{\sqcap}(t)dt = \mathbf{d}_i^T\mathbf{d}_j \int_0^T g_i(t)g_j(t)dt = \delta_{ij}. \tag{45}$$

The autocorrelation matrix of the random vector $\boldsymbol{\Lambda}(\theta)$ in Eq. (44) is

$$\mathbf{C}_{\boldsymbol{\Lambda}\boldsymbol{\Lambda}} = \mathbb{E}\{\boldsymbol{\Lambda}(\theta)\boldsymbol{\Lambda}(\theta)^T\}, \tag{46}$$

which can be decomposed into

$$\mathbf{C}_{\boldsymbol{\Lambda}\boldsymbol{\Lambda}} = \mathbf{Q}\boldsymbol{Z}\mathbf{Q}^T \tag{47}$$

by use of the spectral decomposition, where $\mathbf{Q} \in \mathbb{R}^{k\times k}$ is an orthonormal matrix and $\boldsymbol{Z} \in \mathbb{R}^{k\times k}$ is a diagonal matrix. The stochastic solution $\mathbf{u}_k(t,\theta)$ is thus rewritten as

$$\mathbf{u}_k(t,\theta) = \mathbf{D}^{\sqcap}(t)\mathbf{Q}\mathbf{Q}^T\boldsymbol{\Lambda}(\theta) = \overline{\mathbf{D}^{\sqcap}}(t)\overline{\boldsymbol{\Lambda}}(\theta), \tag{48}$$

where

$$\overline{\mathbf{D}^{\sqcap}}(t) = \mathbf{D}^{\sqcap}(t)\mathbf{Q},\ \overline{\mathbf{\Lambda}}(\theta) = \mathbf{Q}^T\mathbf{\Lambda}(\theta) = \left[\overline{\lambda}_1(\theta), \cdots, \overline{\lambda}_k(\theta)\right]^T, \tag{49}$$

which meet

$$\int_0^T \overline{\mathbf{D}^{\sqcap}}^T(t)\overline{\mathbf{D}^{\sqcap}}(t)dt = \mathbf{Q}^T\left(\int_0^T \mathbf{D}^{\sqcap T}(t)\mathbf{D}^{\sqcap}(t)dt\right)\mathbf{Q} = \mathbf{I}_k, \tag{50}$$

and

$$\mathbb{E}\{\overline{\mathbf{\Lambda}}(\theta)\overline{\mathbf{\Lambda}}(\theta)^T\} = \mathbf{Q}^T\mathbb{E}\{\mathbf{\Lambda}(\theta)\mathbf{\Lambda}(\theta)^T\}\mathbf{Q} = \boldsymbol{Z}. \tag{51}$$

The indicator error is calculated by substituting the improved random vector $\overline{\mathbf{\Lambda}}(\theta)$ into Eq. (42)

$$\varepsilon_{g,k}^{(\mathrm{im})} = \frac{\mathbb{E}\left\{\overline{\lambda}_k^2(\theta)\right\}}{\sum_{i=1}^{k}\mathbb{E}\left\{\overline{\lambda}_i^2(\theta)\right\}} = \frac{\mathbb{E}\left\{\overline{\lambda}_k^2(\theta)\right\}}{\sum_{i=1}^{k}\mathbb{E}\left\{\mathbf{\Lambda}(\theta)^T\mathbf{\Lambda}(\theta)\right\}} = \frac{\boldsymbol{Z}_k}{\mathrm{Tr}(\boldsymbol{Z})}, \tag{52}$$

where $\boldsymbol{Z}_k$ is the last element on the diagonal of the matrix $\boldsymbol{Z}$, i.e. $\boldsymbol{Z}_k = \boldsymbol{Z}(k,k)$.

Eq. (48) is the optimal stochastic-deterministic decomposition of the stochastic solution $\mathbf{u}_k(t,\theta)$ in step 12 in Algorithm 1, thus it provides a better error estimation. In practice, one does not need to implement the above computations fully. Only Eq. (46) and (47) are necessary to compute the diagonal matrix $\mathbf{Z}$. The computational effort is very low since the size of $\mathbf{C}_{\mathbf{\Lambda}\mathbf{\Lambda}} \in \mathbb{R}^{k\times k}$ is usually small. In the numerical examples, we will show that compared to the original error, the indicator error is a better measure of the true error. We can use it to estimate the convergence of the stochastic solution.

6. Numerical examples

The numerical implementation of the proposed method is illustrated with the aid of three examples, including dynamics analysis of a n-dof stiffness-mass system with stochastic material properties and a stochastic force, elasto-dynamic analysis of a two-dimensional plate and a three-dimensional tuning fork with stochastic material properties. The convergence error ε_g and $\varepsilon_{l,k}$ in Algorithm 1 are set as 1×10^{-6} and 1×10^{-3}. For all three examples, the sample size in step 2 in Algorithm 1 is set as $n_s = 1\times10^3$. After solving the matrix $\mathbf{D}$, the sample size in step 16 is reset as $n_s = 1\times10^5$ for solving $\widetilde{\mathbf{G}}(t,\theta)$. All the numerical implementations are tested on a laptop (dual-core, Intel Core i7, 2.40 GHz).

6.1. Dynamics analysis of a n-dof stiffness-mass system

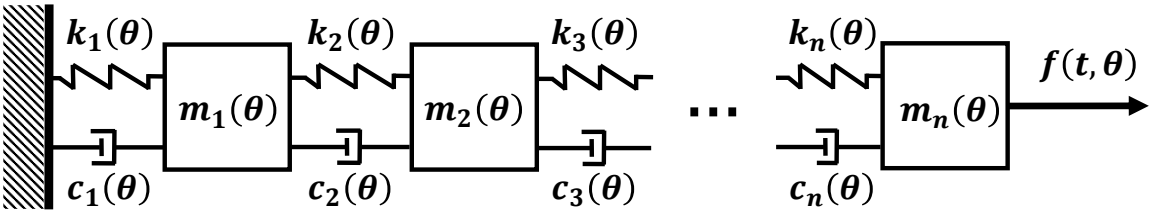


Figure 1: A n-dof stiffness-mass system.

In this example, we consider a n-dof spring-mass system shown in Fig. 1. The stochastic mass, stiffness and damping at each dof are $m_i(\theta) = m_0(1 + h_0\xi_i(\theta))$, $c_i(\theta) = c_0(1 + h_0\xi_{n+i}(\theta))$ and $k_i(\theta) = k_0(1 + h_0\xi_{2n+i}(\theta))$, $i = 1, \cdots, n$, where the parameters are set as $n = 50$, $m_0 = 1.0$kg, $c_0 = 1.0\text{N} \cdot \text{s/m}$, $k_0 = 20\text{N/m}$, $h_0 = 0.2$ and $\{\xi_i(\theta)\}_{i=1}^{3n}$ are uniformly distributed random variables on $[-1, 1]$. The stochastic mass, damping and stiffness matrices of the system are given as

$$\mathbf{M}(\theta) = \begin{bmatrix} m_1(\theta) & & \\ & \ddots & \\ & & m_n(\theta) \end{bmatrix} = \mathbf{M}_0 + \sum_{i=1}^{n} \xi_i(\theta)\mathbf{M}_i \in \mathbb{R}^{n\times n},$$

$$\mathbf{C}(\theta) = \begin{bmatrix} c_1(\theta) + c_2(\theta) & -c_2(\theta) & & & \\ -c_2(\theta) & c_2(\theta) + c_3(\theta) & -c_3(\theta) & & \\ & \ddots & \ddots & \ddots & \\ & & -c_{n-1}(\theta) & c_{n-1}(\theta) + c_n(\theta) & -c_n(\theta) \\ & & & -c_n(\theta) & c_n(\theta) \end{bmatrix}$$

$$= \mathbf{C}_0 + \sum_{i=1}^{n} \xi_{n+i}(\theta)\mathbf{C}_i \in \mathbb{R}^{n\times n},$$

$$\mathbf{K}(\theta) = \begin{bmatrix} k_1(\theta) + k_2(\theta) & -k_2(\theta) & & & \\ -k_2(\theta) & k_2(\theta) + k_3(\theta) & -k_3(\theta) & & \\ & \ddots & \ddots & \ddots & \\ & & -k_{n-1}(\theta) & k_{n-1}(\theta) + k_n(\theta) & -k_n(\theta) \\ & & & -k_n(\theta) & k_n(\theta) \end{bmatrix}$$

$$= \mathbf{K}_0 + \sum_{i=1}^{n} \xi_{2n+i}(\theta)\mathbf{K}_i \in \mathbb{R}^{n\times n}, \tag{53}$$

where $\mathbf{M}_0$, $\mathbf{C}_0$, $\mathbf{K}_0$ are the deterministic components of the stochastic matrix $\mathbf{M}(\theta)$, $\mathbf{C}(\theta)$ and $\mathbf{K}(\theta)$, and $\mathbf{M}_i$, $\mathbf{C}_i$ and $\mathbf{K}_i$ are the deterministic matrices corresponding to the random variables $\xi_i(\theta)$, $\xi_{n+i}(\theta)$ and $\xi_{2n+i}(\theta)$.

The stochastic force $f(t,\theta)$, $t \in [0,T]$ is considered as a Gaussian random process with the mean function $f_0(t) = 0.2-\exp(-at)\sin(bt)$ and the covariance function $C_{ff}(t_1,t_2) = \sigma_f^2 \exp^{-|t_1-t_2|/l_t}$, where $a = 1$, $b = 10\pi$, $\sigma_f = 1$, $l_t = 0.1$, $T = 2$s. We consider that $f(t,\theta)$ is expanded by use of KL expansion

$$f(t,\theta) = f_0(t) + \sum_{i=1}^{r} \xi_{3n+i}(\theta)\sqrt{\kappa_i} f_i(t), \tag{54}$$

where $\{\xi_{3n+i}(\theta)\}_{i=1}^{r}$ are standard normal random variables, $\{f_i(t)\}$ and $\{\kappa_i\}$ are eigenvectors and eigenvalues of the covariance function $C_{ff}(t_1,t_2)$, respectively. In this case, the stochastic force vector $\mathbf{F}(t,\theta)$ in the right side of the stochastic dynamics equation (2) is obtained by the time discretization $dt = 0.01$ and setting $r = 50$ in Eq. (54).

6.1.1. Results

To illustrate the applicability of the classical PGD method, we firstly adopt it to solve this example. Eq. (9), (10) and (11) are used to compute $\{\lambda_i(\theta), \mathbf{d}_i, g_i(t)\}$ and the results of probability density functions (PDFs) of the first two random variables $\lambda_1(\theta)$ and $\lambda_2(\theta)$ are shown in Fig. 2. The

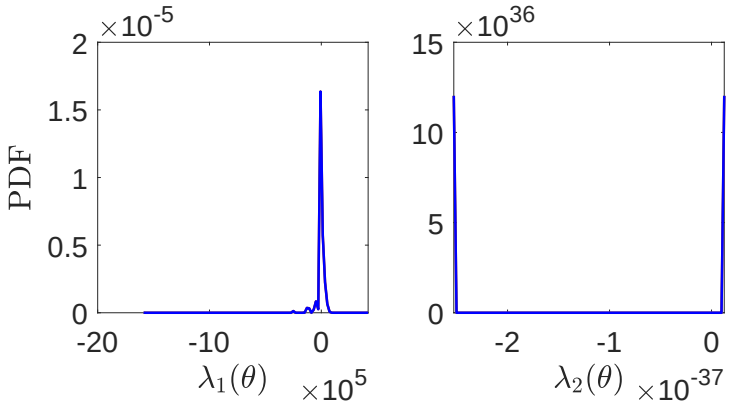


Figure 2: PDFs of the first two random variables $\lambda_1(\theta)$ and $\lambda_2(\theta)$ obtained by the classical PGD iteration.

PDF of the random variable $\lambda_2(\theta)$ is lack of numerical stabilities and it fails to converge (or results a wrong solution) in this case, which indicates the classical PGD method is out of work for this problem.

We can solve this problem efficiently and accurately by using the proposed method. Iterative errors of different numbers of the retained item k are shown in Fig. 3, where the indicator error is calculated by Eq. (52), and the original error and the true error are calculated by Eq. (42). We compute the indicator error as the stopping criterion and then compute the original error and the true error for comparisons. Thirteen retained items $k = 13$ are necessary to achieve the specified precision, which indicates that the proposed Algorithm 1 has good convergence. The original error is poor to check the convergence, while the indicator error has a good agreement with the true error (the true error converges slightly better than the indicator error). Since the true error is a posterior estimation, it cannot determine the number of retained items in a simple way. Thus we adopt the indicator error instead of the true error to check the convergence in practical computations and subsequent examples.

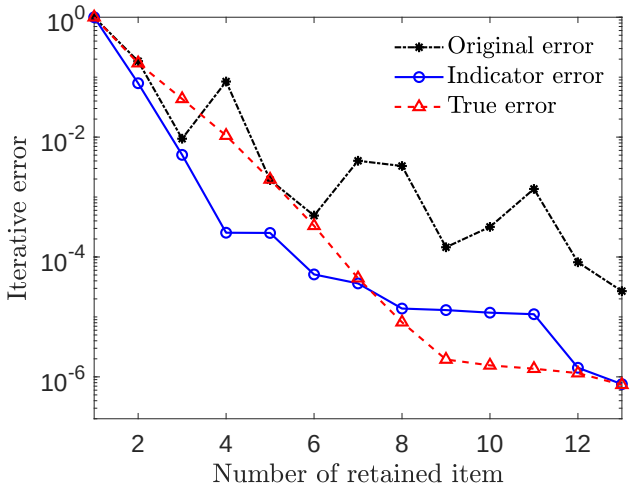
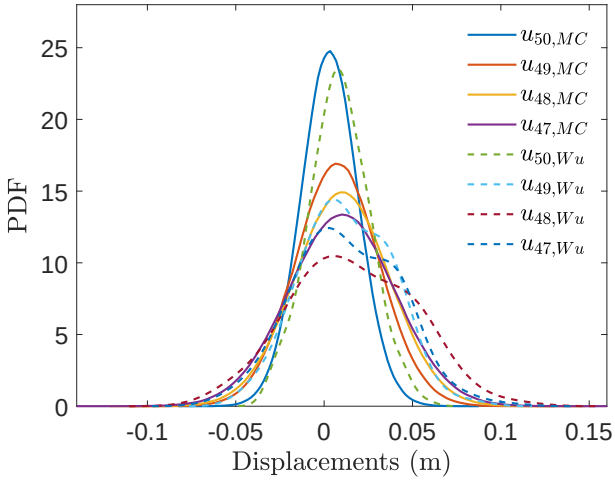
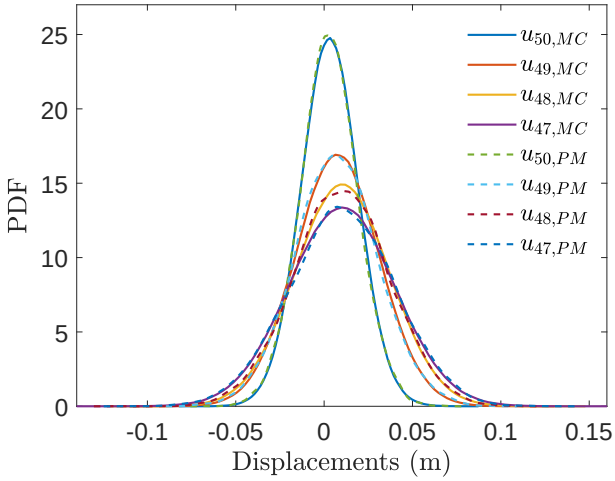


Figure 3: Iterative errors of different numbers of the retained item k.

In Fig. 4 we test the accuracy of the proposed method. PDFs of $u_n(t = 1, \theta)$ at the node $n = 47,\ 48,\ 49,\ 50$ are computed by the proposed method and 1×10^6 Monte Carlo simulations, where $u_{n,Wu}$, $u_{n,PM}$ and $u_{n,MC}$ represent the solution without update (i.e. the stochastic solution obtained



(a) PDFs of $u_{47\sim50}(t = 1, \theta)$ without update and the MCS reference solutions.



(b) PDFs of $u_{47\sim50}(t = 1, \theta)$ with update and the MCS reference solutions.

Figure 4: PDFs of the stochastic solution $u_n(t = 1, \theta)$ of the node $n = 47 \sim 50$: (a) the solutions without update $u_{n,Wu}$ and the reference solutions $u_{n,MC}$ obtained by 1×10^6 MCS, (b) the solutions with update $u_{n,PM}$ and the reference solutions $u_{n,MC}$.

by the sequential iteration from step 1 to 15 in Algorithm 1), the solution with update (i.e. the stochastic solution obtained by Eq. (34)) and the solution obtained by MCS, respectively. Fig. 4a shows that PDFs of $u_{n,Wu}$ have poor accuracy compared to MCS, while PDFs of $u_{n,PM}$ in Fig. 4b

are in very good accordance with MCS reference solutions, which indicates the proposed method has comparable accuracy to MCS. The comparison of PDFs between $u_{n,Wu}$ and $u_{n,PM}$ demonstrates the recomputation step 16 in Algorithm 1 is essential to improve the accuracy of the stochastic solution.

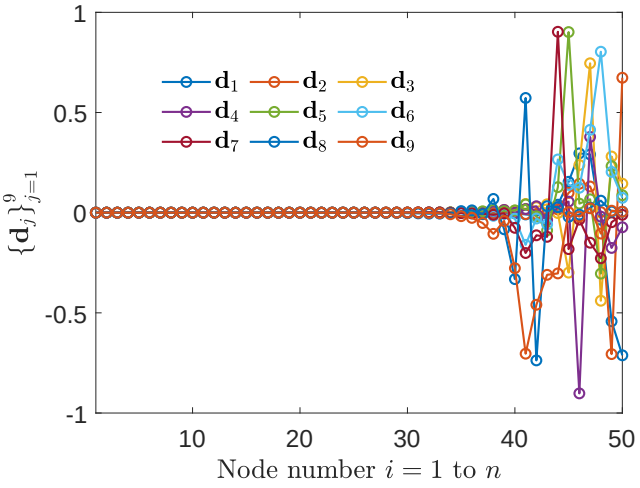


Figure 5: Solutions of first nine terms $\left\{\mathbf{d}_j\right\}_{j=1}^9$.

Table 1: Computational costs of different sample sizes n_s.

Sample size	2×10^2	1×10^3	5×10^3	1×10^6 MCS
Solving costs	17.57	72.21	202.04	
Updating costs	43.80	47.26	37.42	
Total costs (seconds)	61.37	119.47	239.46	9.41×10^3

Fig. 5 shows the solutions of first nine terms $\left\{\mathbf{d}_j\right\}_{j=1}^9$, which indicates that only a small part of the solutions of nodes is nonzero. The proposed method can capture the dominative modes of the solutions. The PDFs of the random variables $\{\lambda_i(\theta)\}_{i=1}^9$ are seen from Fig. 6, where $\{\lambda_i(\theta)\}_i$ (without update, improved and update) represent the original random variables $\{\lambda_i(\theta)\}_i$ obtained in step 12 in Algorithm 1, the improved random variables $\{\overline{\lambda}_i(\theta)\}_i$ obtained by Eq. (49) and the recalculated random variables $\left\{\lambda_i^*(\theta)\right\}_i$ obtained by Eq. (34), respectively. Fig. 6 indicates that most improved random variables are very different from the original random variables. As the number of retained

items k increases, the mean square value $\mathbb{E}\left\{\lambda_k^2(\theta)\right\}$ of the original random variables does not keep decreasing, thus the original error in Fig. 3 is not monotonically decreasing. While the decreasing mean square values of improved and updated random variables make the indicator error and the true error decrease monotonously.

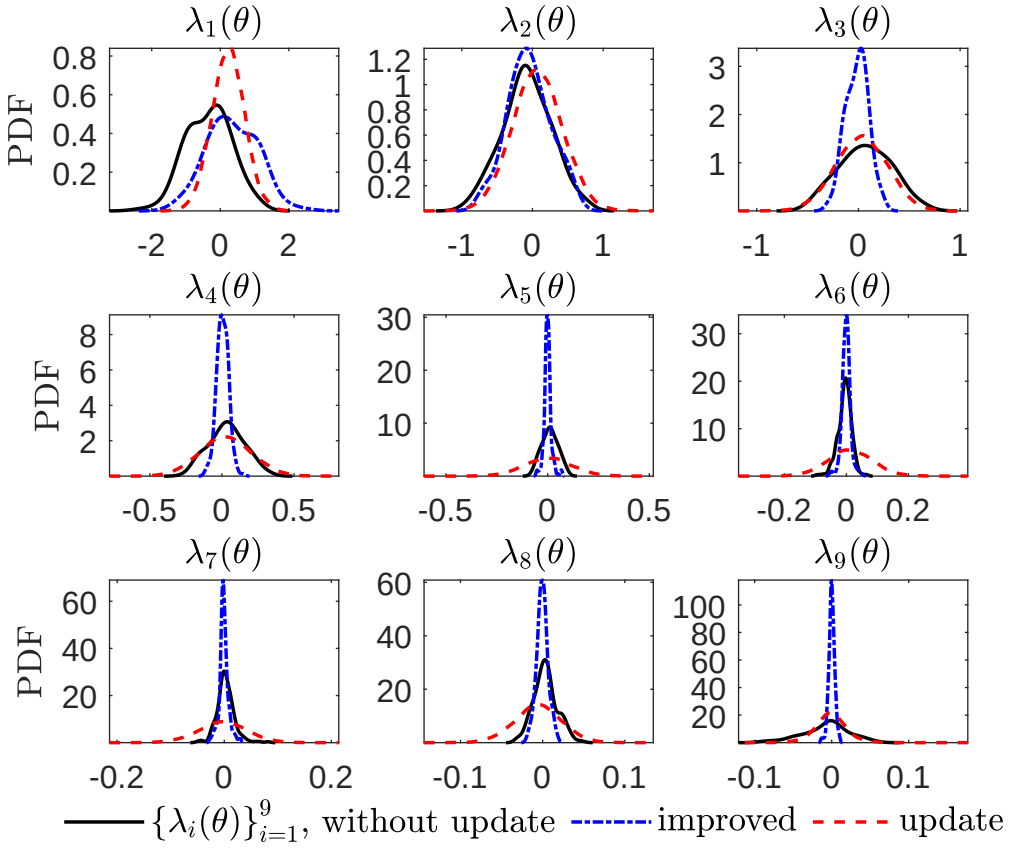


Figure 6: PDFs of first nine random variables $\{\lambda_i(\theta)\}_{i=1}^9$ (without update), $\{\lambda_i(\theta)\}_{i=1}^9$ (improved) and $\{\lambda_i(\theta)\}_{i=1}^9$ (update).

Tab. 1 shows the influence of sample size on computational efficiency. We adopt different sample sizes $n_s = 2 \times 10^2$, 1×10^3 and 5×10^3 in step 2 of Algorithm 1 and reset the sample size $n_s = 1 \times 10^5$ in step 16. All tested sample sizes achieve comparable accuracy, thus we only test the computational efficiencies. Tab. 1 indicates that the total cost increases as the sample size increases, and total costs of all tested sample sizes are much lower than MCS. The solving costs (i.e. the cost of the iteration from step 1 to 15 in Algorithm 1) are mainly from the solution of Eq. (13) and the SVD in Eq. (26). They can be accelerated by high-performance equipment and

parallel computing. The updating costs (i.e. the cost of the recomputation step 16 in Algorithm 1) of different sample sizes are close since they have close numbers of retained items. However, it is still an open problem to choose a suitable sample size for different problems to achieve good accuracy and high efficiencies simultaneously.

6.2. Elasto-dynamic analysis of a two-dimensional bracket

In this example, we consider the elasto-dynamic analysis of a two-dimensional steel bracket [50] shown in Fig. 7. The bracket is fixed at the left boundary Γ_D and is forced by $F(t)$ at the up boundary. The finite element mesh for this problem includes 390 nodes and 672 triangle elements. Material, geometry and load parameters of the model are given by Poisson ratio $\nu = 0.3$, mass density $\rho = 8\times10^3\text{kg/m}^3$, $H = 0.2\text{m}$ and the force $F(t) = 10 - 10\exp(-5t)\sin(4\pi t)\ \text{kN/m},\ t \in [0, T],\ T = 1\text{s}$. The Young's modulus $E(x, y, \theta)$ is a Gaussian random field with the mean function $E_0(x, y) = 2.1\times10^8\text{N/m}^2$ and the covariance function $C_{EE}(x_1, y_1; x_2, y_2) = \sigma_E^2 \exp(-|x_1 - x_2|/l_x - |y_1 - y_2|/l_y)$, where the standard deviation $\sigma_E = 0.1E_0(x, y)$, the correlation lengths $l_x = l_y = 1$. It is noted that the covariance function used in this example (also in example 6.1 and 6.3) is non-differentiable and it may suffer from some obstacles in modeling uncertainties. Differentiable covariance functions are suggested to be better choices in many practical applications [51, 52].

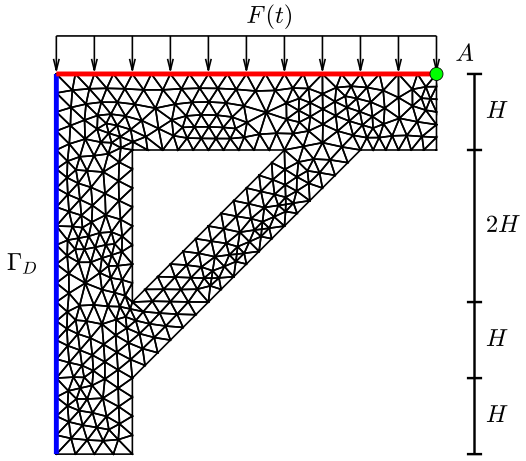


Figure 7: The steel bracket and its finite element mesh.

We expand the Young's modulus $E(x, y, \theta)$ by use of KL expansion with the r-term truncation

$$E(x, y, \theta) = E_0(x, y) + \sum_{i=1}^{r} \xi_i(\theta) \sqrt{\kappa_i} E_i(x, y), \tag{55}$$

where $\{\xi_i(\theta)\}_{i=1}^{r}$ are standard normal random variables, $\{E_i(x, y)\}$ and $\{\kappa_i\}$ are eigenvectors and eigenvalues of the covariance function $C_{EE}(x_1, y_1; x_2, y_2)$, respectively. We set $r = 10$ in this example and Fig. 8 shows the first nine terms $\{E_i(x, y)\}_{i=1}^{9}$. It is noted that the Gaussian random field assumption for the Young's modulus may violate physics since negative values may take. In practice, the samples $\theta^{(i)}$ such that $\min_{x,y} E\left(x, y, \theta^{(i)}\right) < 1 \times 10^{-3}$ will be dropped out, thus $E(x, y, \theta)$ in Eq. (55) is considered a truncated Gaussian random field in the numerical implementation.

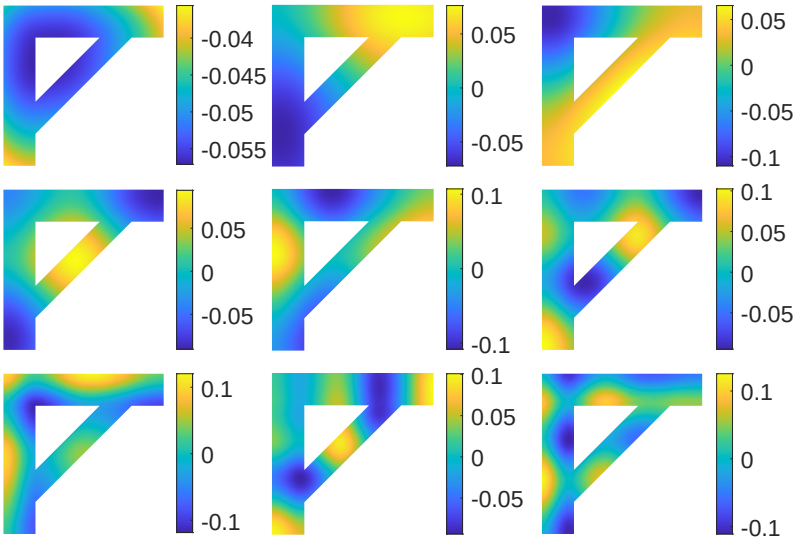


Figure 8: The first nine eigenvectors $\{E_i(x, y)\}_{i=1}^{9}$ of the bracket.

6.2.1. Results

We adopt Algorithm 1 to solve this problem and a Rayleigh damping $\mathbf{C}(\theta) = \alpha_1 \mathbf{M}(\theta) + \alpha_2 \mathbf{K}(\theta)$ is used in this analysis, where the coefficients are $\alpha_1 = 10$ and $\alpha_2 = 0$. Fig. 9 shows the indicator error and the true error of the proposed method. The indicator error still has a good agreement with the true error. Only six items $k = 6$ are retained to achieve the specified precision, which verifies the fast convergence of Algorithm 1 again. Corresponding six solution components $\{\mathbf{d}_i(x, y)\}_{i=1}^{6}$ are seen from Fig. 10, where Fig. 10a and 10b are the solution components $\{\mathbf{d}_{i,x}\}_{i=1}^{6}$ in the x (i.e.

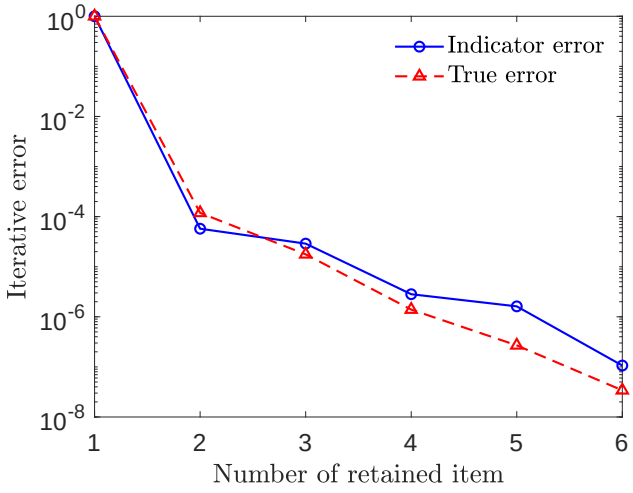
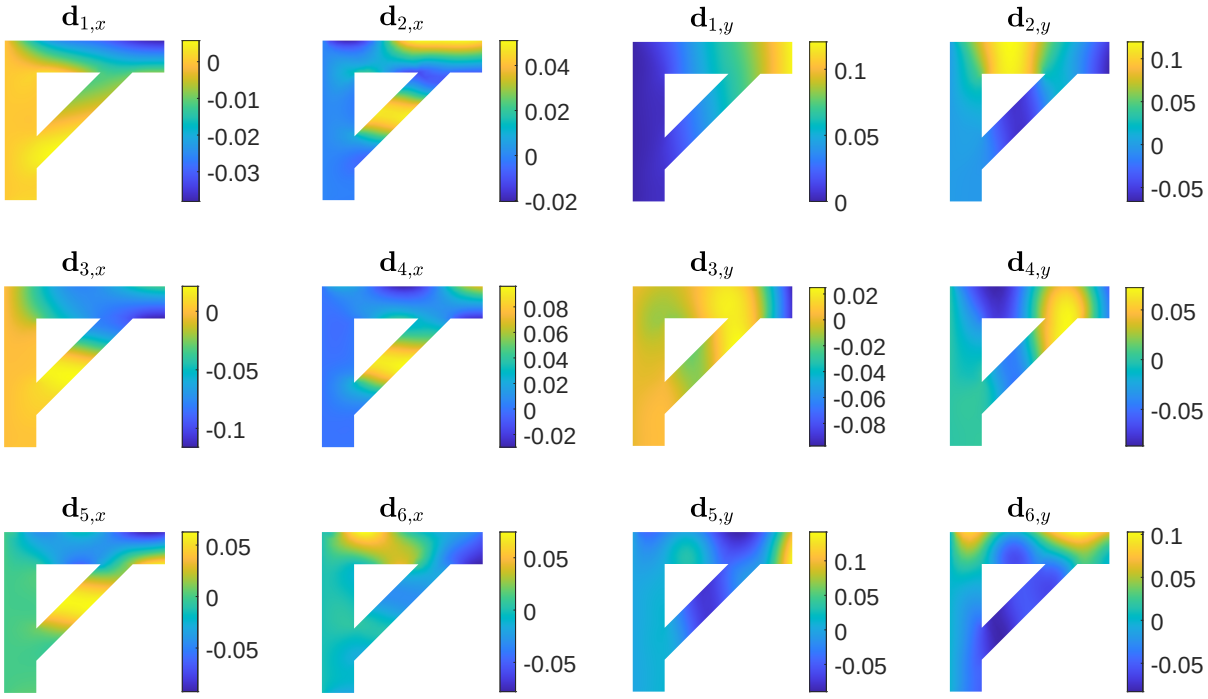


Figure 9: Iterative errors of different numbers of the retained item k.



(a) Displacement components $\{\mathbf{d}_{i,x}\}_{i=1}^{6}$. (b) Displacement components $\left\{\mathbf{d}_{i,y}\right\}_{i=1}^{6}$.

Figure 10: Solutions of $\{\mathbf{d}_i(x,y)\}_{i=1}^{6}$: (a) displacement components $\{\mathbf{d}_{i,x}\}_{i=1}^{6}$ in x direction and (b) displacement components $\left\{\mathbf{d}_{i,y}\right\}_{i=1}^{6}$ in y direction.

horizontal) direction and $\left\{\mathbf{d}_{i,y}\right\}_{i=1}^{6}$ in the y (i.e. vertical) direction, respectively. In the x direction, the displacements are much smaller than that in the y direction since the force $F(t)$ works on the y direction. Fig. 11 shows PDFs of six random variables $\{\lambda_i(\theta)\}_{i=1}^{6}$ (without update, improved and update) that are calculated by step 12 in Algorithm 1, Eq. (49) and Eq. (34), respectively. As the number of retained items k increases, the mean square values of improved and updated random variables still keep monotonically decreasing.

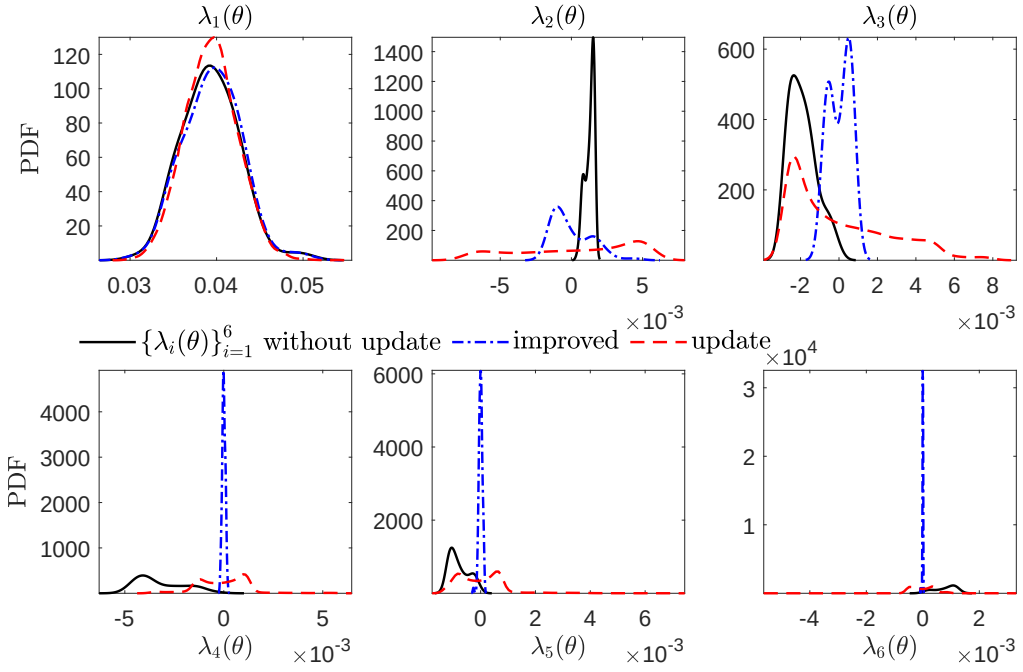


Figure 11: PDFs of six random variables $\{\lambda_i(\theta)\}_{i=1}^{6}$ (without update), $\{\lambda_i(\theta)\}_{i=1}^{6}$ (improved) and $\{\lambda_i(\theta)\}_{i=1}^{6}$ (update).

PDFs of the solution $u(t = 0.5, \theta)$ in the y direction of the point A (shown in Fig. 7) are seen from Fig. 12. The PDF of the solution with update $u_{A,PM}(t = 0.5, \theta)$ is in very good accordance with the reference solution $u_{A,MC}(t = 0.5, \theta)$ obtained by 1×10^6 MCS, while the solution without update $u_{A,Wu}(t = 0.5, \theta)$ achieves a wrong result. Also, we compare the proposed method and the PC-based spectral stochastic finite element method. Two-order Hermite PC basis of $r = 10$ standard normal random variables are adopted and the total number of PC basis is 66. The size of the PC-based derived finite element equation is about 5.15×10^4, which is much larger than the original stochastic finite element equation. As shown in Fig. 12, the PC-based solution has a poor

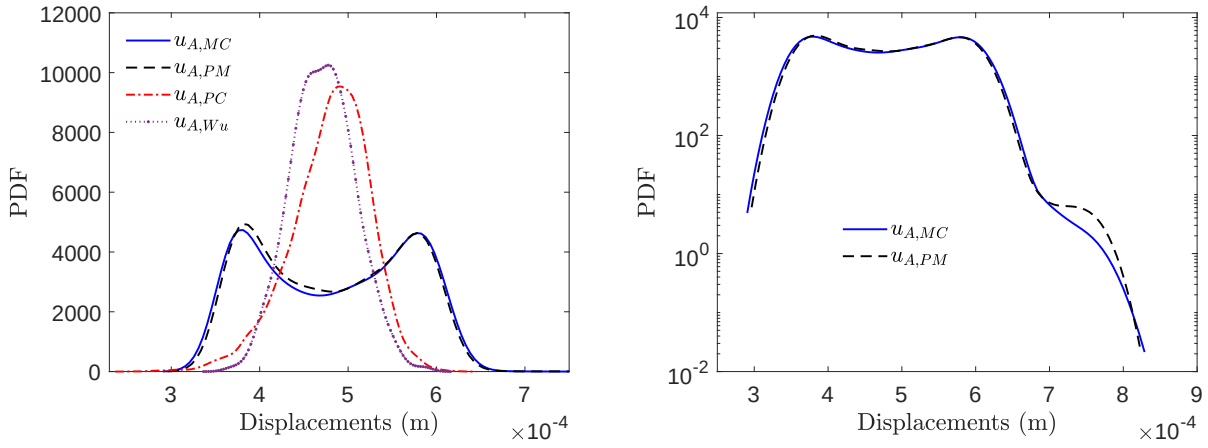


Figure 12: PDFs of the solution without update $u_{A,Wu}(t = 0.5, \theta)$, the solution with update $u_{A,PM}(t = 0.5, \theta)$, the PC solution $u_{A,PC}(t = 0.5, \theta)$ and the reference solution $u_{A,MC}(t = 0.5, \theta)$ obtained by 1×10^6 MCS: regular scale (left) and logarithmic scale (right).

accuracy compared to the proposed method and MCS, and it cannot capture the bimodal mode of the PDF. Further, the proposed method is more convenient compared to PC-based methods since it does not need to choose the type and the order of PC basis.

For the low-dimensional case $r = 10$, the computational times of the proposed method, the PC-based method and 1×10^6 MCS are 44.19s, 219.47s and 1.06×10^5s, respectively, which

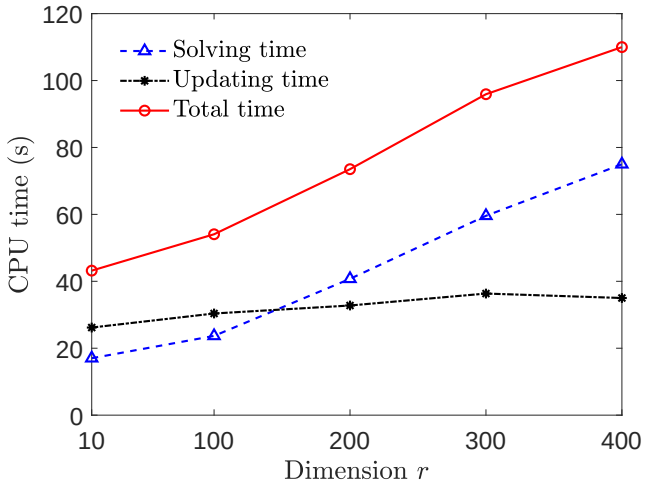


Figure 13: Computational costs of different stochastic dimensions $r = 10, 100 \sim 400$.

demonstrates the high efficiency of the proposed method. We also test the computational efficiency of different stochastic dimensions. The computational times of stochastic dimensions $r = 10, 100, 200, 300, 400$ are shown in Fig. 13. The proposed method avoids the curse of dimensionality with great success. The total computational cost of the proposed method increases linearly as the stochastic dimension increases. The solving and updating times represent the time cost of the iteration from step 1 to 15 in Algorithm 1 and the time cost of the recomputation step 16, respectively. Fig. 13 indicates that the solving time increases linearly as the stochastic dimension increases. The updating time hardly increases since the retained item k changes slightly. The solving time dominates the total computational cost, thus the methods that reduce the solving time can accelerate the proposed method.

Further, we test the influence of different node sizes n_p on the computational cost. Tab. 2 demonstrates that the total computational cost will not increase dramatically as the node size increases. The solving time strongly depends on the computational effort of the solution of Eq. (13) and more efficient solvers for solving linear dynamics equations can save the solving times. The updating time hardly increases as the node size increases since the retained item k hardly changes for different node sizes. It is seen from Fig. 13 and Tab. 2, the computational cost weakly depends on the coupling of the node size and the stochastic dimension, which makes the proposed method very efficient for solving large-scale and high-dimensional stochastic dynamics problems.

Table 2: Computational costs of different node size n_p.

Node size	111	390	1452	5592
Element size	168	672	2688	10752
Solving costs	6.75	17.01	68.18	779.08
Updating costs	26.75	27.18	32.99	31.33
Total costs (seconds)	33.0	44.19	101.99	810.41

6.3. Elasto-dynamic analysis of a three-dimensional tuning fork

In this example, we consider the elasto-dynamic analysis of a three-dimensional tuning fork shown in Fig. 14, which includes 1322 nodes and 3956 tetrahedron elements. The tuning fork

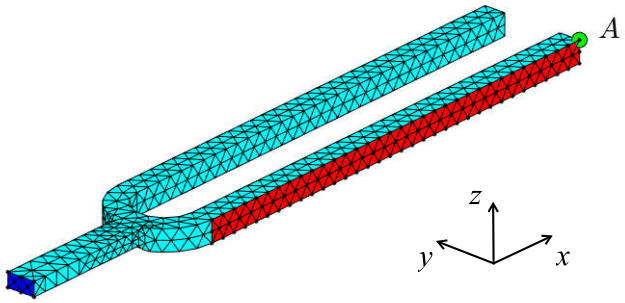


Figure 14: Model of the tuning fork and its finite element mesh.

is fixed at the end of the handle (blue region) and is forced by $F(t)$ at a part of the surface (red region), where $F(t) = 5 \times 10^4 \mathrm{N/m}^2$ for $t \in [0, t_0]$ and 0 for $t \in (t_0, T]$, $t_0 = 0.05\mathrm{s}$, $T = 1\mathrm{s}$. Material parameters are given by Poisson ratio $\nu = 0.3$ and mass density $\rho = 8 \times 10^3 \mathrm{kg/m}^3$. The Young's modulus $E(x, y, \theta)$ is a random field with the mean function $E_0(x, y) = 2.1 \times 10^{11} \mathrm{N/m}^2$ and the covariance function $C_{EE}(x_1, y_1, z_1; x_2, y_2, z_2) = \sigma_E^2 \exp(-|x_1 - x_2|/l_x - |y_1 - y_2|/l_y - |z_1 - z_2|/l_z)$, where the standard deviation $\sigma_E = 0.2E_0(x, y)$, the correlation lengths $l_x = l_y = l_z = 0.1$. The Young's modulus $E(x, y, \theta)$ is approximated by the r-term KL expansion in Eq. (55). In this example, $\{\xi_i(\theta)\}_{i=1}^r$ in the expansion Eq. (55) are uniformly distributed random variables on $[-1, 1]$.

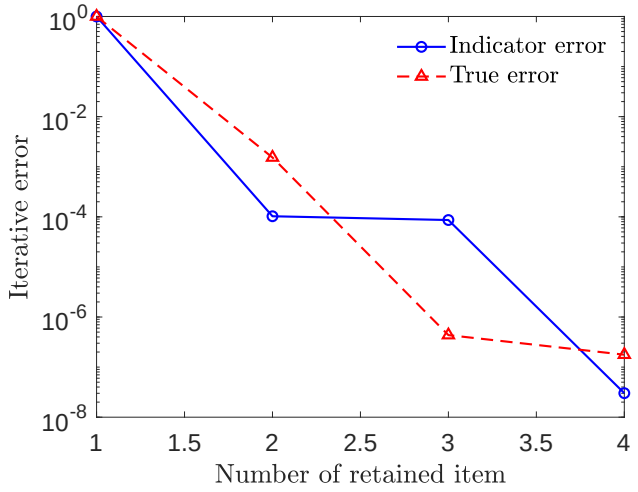


Figure 15: Iterative errors of different numbers of the retained item k.

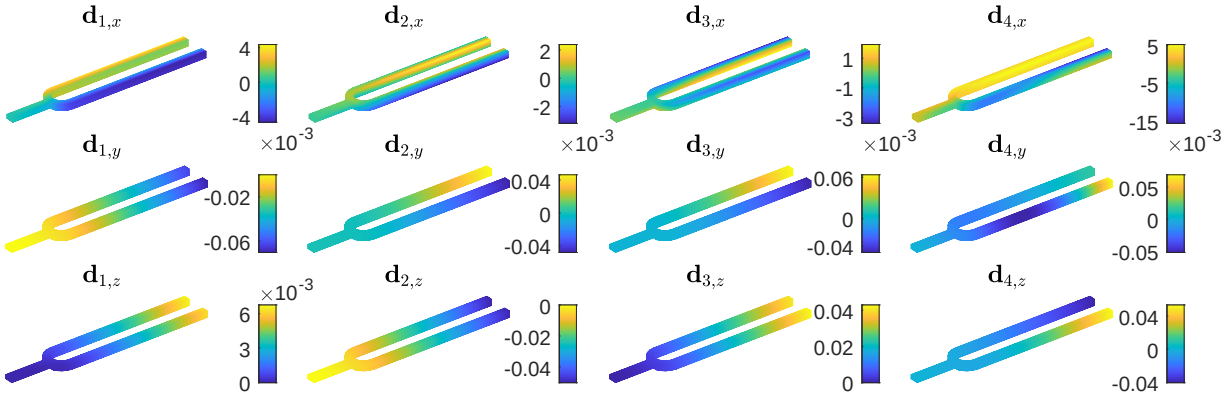


Figure 16: Solutions of four terms $\{\mathbf{d}_i(x,y,z)\}_{i=1}^4$: displacement components $\{\mathbf{d}_{i,x}\}_{i=1}^4$ in x direction (the first row), displacement components $\{\mathbf{d}_{i,y}\}_{i=1}^4$ in y direction (the second row) and displacement components $\{\mathbf{d}_{i,z}\}_{i=1}^4$ in z direction (the third row).

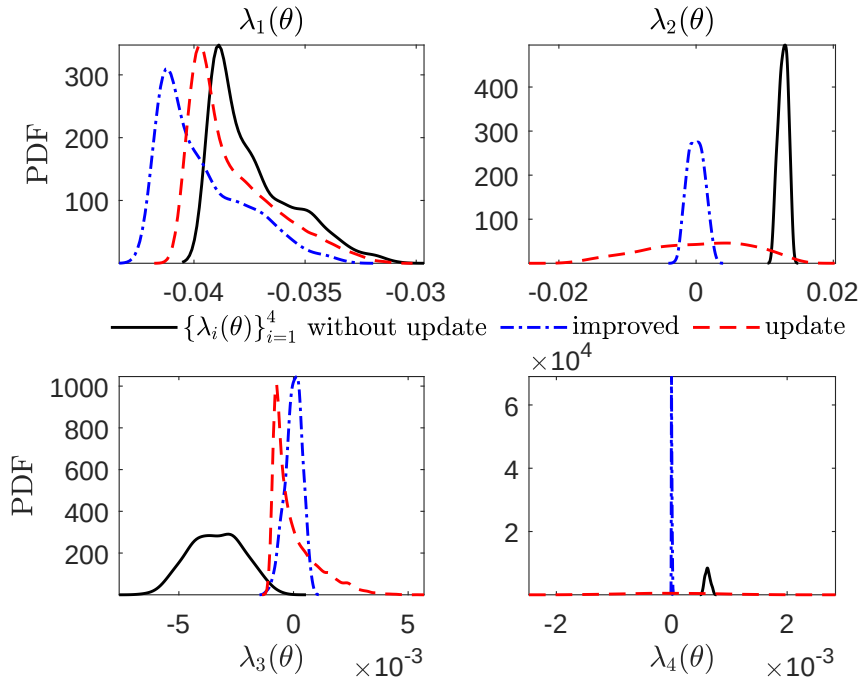


Figure 17: PDFs of four random variables $\{\lambda_i(\theta)\}_{i=1}^4$ (without update), $\{\lambda_i(\theta)\}_{i=1}^4$ (improved) and $\{\lambda_i(\theta)\}_{i=1}^4$ (update).

6.3.1. Results

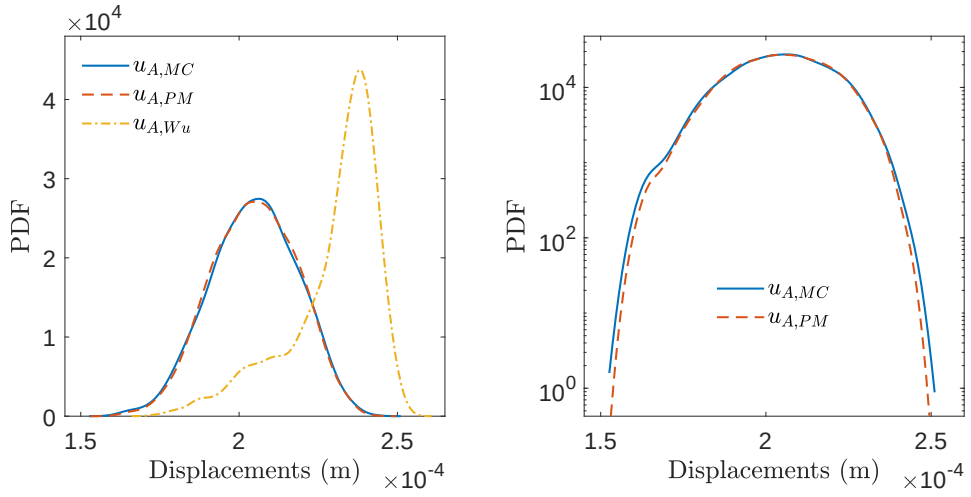
The Rayleigh damping $\mathbf{C} = 10\mathbf{M}$ in Example 6.2 is also used in this example and the stochastic dimension is set as $r = 10$. Four retained items $k = 4$ meet the convergence criterion. As the number of retained terms increases, both the indicator error and the true error tend to converge.

Fig. 15 demonstrates that the indicator error still has a good agreement with the true error. The variability of the stochastic solution is not too large due to the small variability of the random field $E(x, y, \theta)$, thus both Example 6.2 and this example only require small numbers of retained terms to achieve the specified precision. It usually requires more retained terms for the problems with large variabilities, as illustrated in Example 6.1. Four solution components, including the solutions $\{\mathbf{d}_{i,x}\}_{i=1}^4$ in the x direction, $\left\{\mathbf{d}_{i,y}\right\}_{i=1}^4$ in the y direction and $\{\mathbf{d}_{i,z}\}_{i=1}^4$ in the z direction, are seen from Fig. 16, and Fig. 17 shows PDFs of corresponding random variables $\{\lambda_i(\theta)\}_{i=1}^4$ (without update, improved and update) calculated by step 12 in Algorithm 1, Eq. (49) and Eq. (34), respectively. It is noted that Eq. (48) does not improve the accuracy of the stochastic solution $\mathbf{u}_k(t, \theta)$ but only provides an optimal decoupled representation. The accuracy of the stochastic solution approximated by the improved random variables in Eq. (48) is still lower than the final stochastic solution approximated by Eq. (34). It is seen from Fig. 17 that the improved random variables are very different from the update random variables since the improved random variables are not obtained from the final stochastic solution while the update random variables are computed by the final stochastic solution.

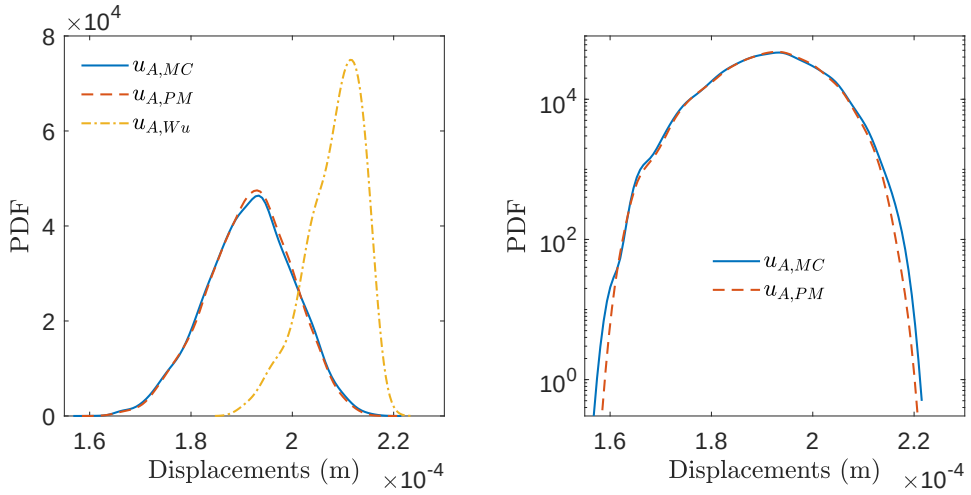
Table 3: Computational costs of the stochastic dimensions $r = 10$ and 100.

Dimension r	10 ($\sigma_E = 0.2E_0$)		100 ($\sigma_E = 0.1E_0$)	
Method	SFEM	MCS	SFEM	MCS
Solving costs	126.50		135.78	
Updating costs	33.75		36.21	
Total costs (second)	160.25	4.69×10^6	171.99	6.48×10^6

Further, we test the computational efficiency and accuracy of different stochastic dimensions. Computational costs of $r = 10$, $\sigma_E = 0.2E_0(x, y)$ and $r = 100$, $\sigma_E = 0.1E_0(x, y)$ (a larger σ_E will cause the matrix $\mathbf{K}(\theta)$ to be non-positive definite) are shown in Tab. 3 and they have similar time costs. The solving cost is the main component of the total cost, especially for large-scale stochastic problems since solving Eq. (13) of large-scale problems is more time-consuming. The updating cost weakly depends on degrees of freedom of the finite element discretization and is only related



(a) PDFs of the solutions when $r = 10$: regular scale (left) and logarithmic scale (right).



(b) PDFs of the solutions when $r = 100$: regular scale (left) and logarithmic scale (right).

Figure 18: PDFs of the solution without update $u_{A,Wu}(t = 0.5, \theta)$, the solution with update $u_{A,PM}(t = 0.5, \theta)$ and the reference solution $u_{A,MC}(t = 0.5, \theta)$ obtained by 1×10^6 MCS: (a) the stochastic dimension $r = 10$, (b) the stochastic dimension $r = 100$.

to the number of retained terms, thus it has cheap computational costs even for very complex stochastic problems. PDFs of the solutions $u(t = 0.5, \theta)$ in the y direction of the point A (shown in Fig. 14) are seen from Fig. 18. As shown in Fig. 18a (left) and 18b (left), for both the low- and high-dimensional cases, PDFs of the solution without update $u_{A,Wu}(t = 0.5, \theta)$ deviate significantly from the reference solutions $u_{A,MC}(t = 0.5, \theta)$ obtained by 1×10^6 MCS, while PDFs of the solution

with update $u_{A,PM}(t = 0.5, \theta)$ keep in very good accordance with the MCS reference solutions even for the high-dimensional case, which demonstrates the high accuracy of the proposed method. Also, as shown in Fig. 18a (right) and 18b (right), we compare the solutions with update $u_{A,PM}(t = 0.5, \theta)$ and the MCS reference solutions on the logarithmic scale, which demonstrates that the distribution ranges of the stochastic solutions for both low- and high-dimensional cases are well captured by the proposed method.

6.4. A n-dof system subjected to a seismic ground motion

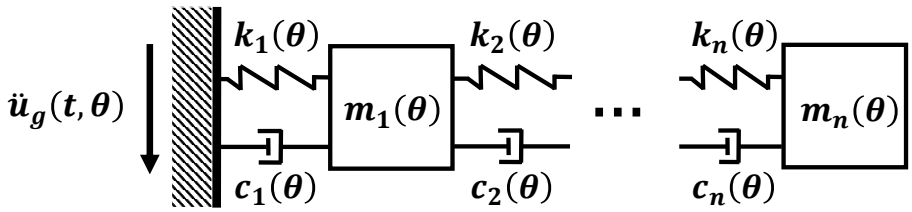


Figure 19: A n-dof system subjected to a seismic ground acceleration (rotated in 90° to save the space).

In this example, we consider a similar physical model in Fig. 1. As shown in Fig. 19, the system is subjected to a seismic ground acceleration $\ddot{u}_g(t, \theta)$. All parameters are the same as Example 6.1 except that $m_i(\theta) = 1.0$kg are deterministic and $\{\xi_{n+i}(\theta))\}_{i=1}^{n}$ in $c_i(\theta) = c_0(1 + h_0\xi_{n+i}(\theta))$ (should be shear damping in this case) are modeled as n independent Weibull random variables, the scale and shape parameters of which are 0.25 and 1.5. The seismic ground acceleration $\ddot{u}_g(t, \theta)$ is a zero-mean Gaussian stochastic process described by use of the Clough-Penzien spectrum [53, 54, 55],

$$S_{u_g u_g}(\omega) = \frac{\omega^4 \left[\omega_g^4 + \left(2\zeta_g\omega_g\omega\right)^2\right] S_0}{\left[\left(\omega_g^2 - \omega^2\right)^2 + \left(2\zeta_g\omega_g\omega\right)^2\right]\left[\left(\omega_f^2 - \omega^2\right)^2 + \left(2\zeta_f\omega_f\omega\right)^2\right]}, \tag{56}$$

where the parameters are given as $\omega_g = 4\pi$ rad/s, $\omega_f = 0.4\pi$ rad/s, $\zeta_g = \zeta_f = 0.7$, $S_0 = 3 \times 10^{-4}$ m^2/s^3. The autocorrelation function $C_{u_g u_g}$ is then calculated based on the Clough-Penzien spectrum in Eq. (56). Thus we can approximate the time history of $\ddot{u}_g(t, \theta)$ using the autocorrelation function and the KL expansion. In this example, the time duration is 20s and 2×10^3 time steps are adopted. 1066 truncated items are retained to achieve $\kappa_{1066}/\sum_{i=1}^{\infty}\kappa_i \geq 95\%$ (κ_i is the eigenvalues

of the autocorrelation function $C_{u_g u_g}$ referring Eq. (55)). A total of 1166 random variables are thus involved, including 50 uniform random variables for the stiffness parameters, 50 Weibull random variables for the damping parameters and 1066 Gaussian random variables for the excitation.

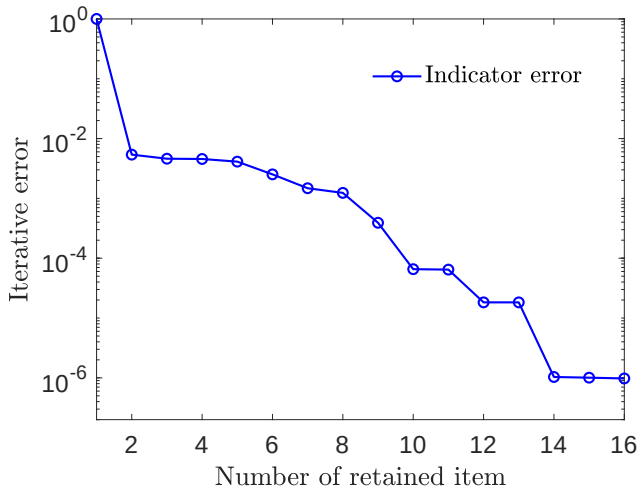


Figure 20: Iterative errors of different numbers of the retained item k.

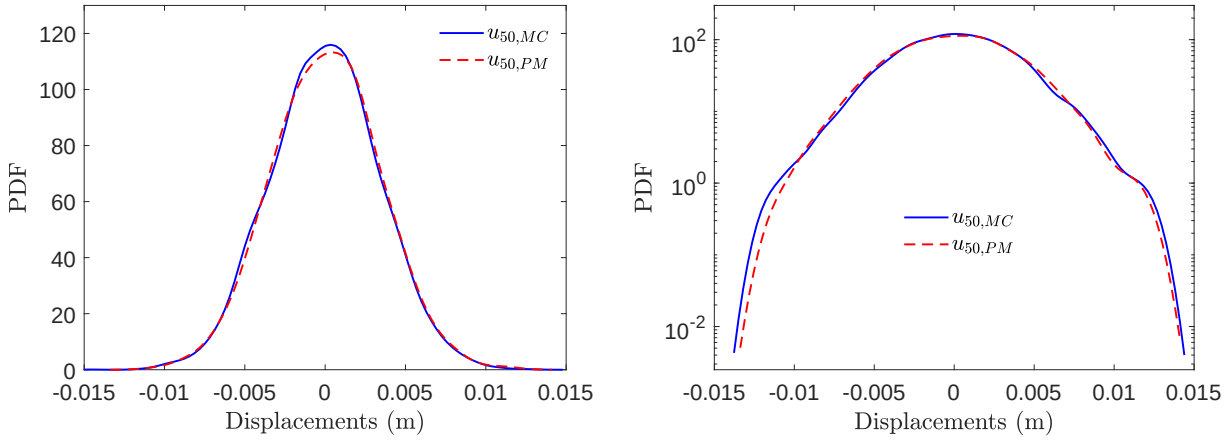


Figure 21: PDFs of the stochastic solution $u_{50}(t = 10, \theta)$ obtained by the proposed method and 1×10^6 MCS: regular scale (left) and logarithmic scale (right).

The iterative errors are depicted in Fig. 20 and 16 items are retained to achieve the specified accuracy. Compared to Example 6.1, more items are needed to capture a high-accuracy stochastic solution. As shown in Fig. 21, the PDF of the stochastic solution of the 50th node at time $t = 10$s

obtained by the proposed method is compared with the corresponding PDF obtained by 1×10^6 MCS, which indicates that the proposed method still has comparable accuracy to MCS in this case. The total computational costs of the proposed method are 260.98s, including 177.04s for the solving costs and 83.94s for the updating costs. Compared to 6.36×10^4s for the MCS costs, the proposed method saves a lot of computational costs.

7. Conclusions

This paper develops an efficient stochastic finite element method for linear stochastic structural dynamics analysis and illustrates its accuracy and efficiency by three numerical examples. By constructing a universal form of stochastic solutions and developing a dedicated iterative algorithm, stochastic dynamics problems are decoupled into stochastic and deterministic analyses that are executed in their individual spaces. For the deterministic analyses, existing FEM solvers can be readily incorporated into the computational procedure without any modification. All stochastic analyses are embedded into one-dimensional stochastic ordinary differential equations that can be solved efficiently by the proposed sampling method. In this way, the curse of dimensionality in high-dimensional stochastic spaces is avoided with great success, which has been illustrated by the numerical example of up to 1166 dimensions. In these senses, the proposed method is particularly appropriate for the stochastic structural dynamics analysis of practical interests.

Acknowledgments

The authors are grateful to the Alexander von Humboldt Foundation and the International Research Training Group 2657 (IRTG 2657) funded by the German Research Foundation (DFG) (Grant reference number 433082294).

References

[1] T. J. Hughes, The finite element method: linear static and dynamic finite element analysis, Courier Corporation, 2012.

[2] H. N. Najm, Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics, Annual Review of Fluid Mechanics 41 (2009) 35–52.

[3] O. Le Maître, O. M. Knio, Spectral methods for uncertainty quantification: with applications to computational fluid dynamics, Springer Science & Business Media, 2010.

[4] R. G. Ghanem, P. D. Spanos, Stochastic finite elements: a spectral approach, Courier Corporation, 2003.

[5] B. Sudret, A. Der Kiureghian, Stochastic finite element methods and reliability: a state-of-the-art report, Department of Civil and Environmental Engineering, University of California, 2000.

[6] G. Stefanou, The stochastic finite element method: past, present and future, Computer Methods in Applied Mechanics Engineering 198 (2009) 1031–1051.

[7] M. Papadrakakis, V. Papadopoulos, Robust and efficient methods for stochastic finite element analysis using Monte Carlo simulation, Computer Methods in Applied Mechanics Engineering 134 (1996) 325–340.

[8] R. E. Caflisch, Monte carlo and quasi-monte carlo methods, Acta Numerica 7 (1998) 1–49.

[9] C. Robert, G. Casella, Monte Carlo statistical methods, Springer Science & Business Media, 2013.

[10] S. Brooks, A. Gelman, G. Jones, X. Meng, Handbook of markov chain monte carlo, CRC press, 2011.

[11] M. B. Giles, Multilevel Monte Carlo methods, Acta Numerica 24 (2015) 259–328.

[12] J. N. Fuhg, A. Fau, U. Nackenhorst, State-of-the-art and comparative review of adaptive sampling methods for kriging, Archives of Computational Methods in Engineering (2020) 1–59.

[13] A. I. Khuri, S. Mukhopadhyay, Response surface methodology, Wiley Interdisciplinary Reviews: Computational Statistics 2 (2010) 128–149.

[14] A. Doostan, A. Validi, G. Iaccarino, Non-intrusive low-rank separated approximation of high-dimensional stochastic models, Computer Methods in Applied Mechanics and Engineering 263 (2013) 42–55.

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

[16] H. G. Matthies, A. Keese, Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations, Computer Methods in Applied Mechanics Engineering 194 (2005) 1295–1331.

[17] A. Nouy, A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations, Computer Methods in Applied Mechanics and Engineering 196 (2007) 4521–4537.

[18] I. Babuška, P. Chatzipantelidis, On solving elliptic stochastic partial differential equations, Computer Methods in Applied Mechanics Engineering 191 (2002) 4093–4122.

[19] D. Xiu, Numerical methods for stochastic computations: a spectral method approach, Princeton University Press, 2010.

[20] A. Nouy, Proper generalized decompositions and separated representations for the numerical solution of high dimensional stochastic problems, Archives of Computational Methods in Engineering 17 (2010) 403–434.

[21] M. Frangos, Y. Marzouk, K. Willcox, B. van Bloemen Waanders, Surrogate and reduced-order modeling: a comparison of approaches for large-scale statistical inverse problems [chapter 7] (2010).

[22] Z. Zheng, H. Dai, Simulation of multi-dimensional random fields by Karhunen–Loève expansion, Computer

Methods in Applied Mechanics and Engineering 324 (2017) 221–247.

[23] M. F. Pellissetti, R. G. Ghanem, Iterative solution of systems of linear equations arising in the context of stochastic finite elements, Advances in Engineering Software 31 (2000) 607–616.

[24] G. Blatman, B. Sudret, An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis, Probabilistic Engineering Mechanics 25 (2010) 183–197.

[25] C. Audouze, P. B. Nair, Anchored ANOVA Petrov–Galerkin projection schemes for parabolic stochastic partial differential equations, Computer Methods in Applied Mechanics and Engineering 276 (2014) 362–395.

[26] L. Gao, C. Audouze, P. B. Nair, Anchored analysis of variance Petrov–Galerkin projection schemes for linear stochastic structural dynamics, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 471 (2015) 20150023.

[27] C. Soize, A comprehensive overview of a non-parametric probabilistic approach of model uncertainties for predictive models in structural dynamics, Journal of Sound and Vibration 288 (2005) 623–652.

[28] D. Moens, M. Hanss, Non-probabilistic finite element analysis for parametric uncertainty treatment in applied mechanics: Recent advances, Finite Elements in Analysis and Design 47 (2011) 4–16.

[29] S. Adhikari, Matrix variate distributions for probabilistic structural dynamics, AIAA Journal 45 (2007) 1748–1762.

[30] P. Ladevèze, Nonlinear computational structural mechanics: new approaches and non-incremental methods of calculation, Springer Science & Business Media, 2012.

[31] F. Chinesta, A. Ammar, E. Cueto, Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models, Archives of Computational Methods in Engineering 17 (2010) 327–350.

[32] A. Nouy, A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations, Computer Methods in Applied Mechanics and Engineering 199 (2010) 1603–1626.

[33] F. Chinesta, R. Keunings, A. Leygue, The proper generalized decomposition for advanced numerical simulations: a primer, Springer Science & Business Media, 2013.

[34] B. Favoretto, C. De Hillerin, O. Bettinotti, V. Oancea, A. Barbarulo, Reduced order modeling via PGD for highly transient thermal evolutions in additive manufacturing, Computer Methods in Applied Mechanics and Engineering 349 (2019) 405–430.

[35] L. Giraldi, D. Liu, H. G. Matthies, A. Nouy, To be or not to be intrusive? The solution of parametric and stochastic equations—Proper Generalized Decomposition, SIAM Journal on Scientific Computing 37 (2015) A347–A368.

[36] A. Nouy, Generalized spectral decomposition method for solving stochastic finite element equations: invariant subspace problem and dedicated algorithms, Computer Methods in Applied Mechanics and Engineering 197 (2008) 4718–4736.

[37] A. Nouy, O. P. Le Maître, Generalized spectral decomposition for stochastic nonlinear problems, Journal of Computational Physics 228 (2009) 202–235.

[38] M. Billaud-Friess, A. Nouy, Dynamical model reduction method for solving parameter-dependent dynamical systems, SIAM Journal on Scientific Computing 39 (2017) A1766–A1792.

[39] M. Chevreuil, A. Nouy, Model order reduction based on proper generalized decomposition for the propagation of uncertainties in structural dynamics, International Journal for Numerical Methods in Engineering 89 (2012) 241–268.

[40] N. Lüthen, S. Marelli, B. Sudret, Sparse polynomial chaos expansions: Literature survey and benchmark, SIAM/ASA Journal on Uncertainty Quantification 9 (2021) 593–649.

[41] J. C. Butcher, Numerical methods for ordinary differential equations, John Wiley & Sons, 2016.

[42] J. N. Reddy, An introduction to nonlinear finite element analysis: with applications to heat transfer, fluid mechanics, and solid mechanics, OUP Oxford, 2014.

[43] Z. Zheng, H. Dai, Structural stochastic responses determination via a sample-based stochastic finite element method, Computer Methods in Applied Mechanics and Engineering 381 (2021) 113824.

[44] M. Paz, Structural dynamics: theory and computation, Springer Science & Business Media, 2012.

[45] D. Kalman, A singularly valuable decomposition: the SVD of a matrix, The College Mathematics Journal 27 (1996) 2–23.

[46] G. Kerschen, J.-C. Golinval, A. F. Vakakis, L. A. Bergman, The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview, Nonlinear Dynamics 41 (2005) 147–169.

[47] W. Betz, I. Papaioannou, D. Straub, Numerical methods for the discretization of random fields by means of the Karhunen–Loève expansion, Computer Methods in Applied Mechanics and Engineering 271 (2014) 109–129.

[48] Z. Zheng, H. Dai, Y. Wang, W. Wang, A sample-based iterative scheme for simulating non-stationary non-Gaussian stochastic processes, Mechanical Systems and Signal Processing 151 (2021) 107420.

[49] H. Rabitz, Ö. F. Aliş, General foundations of high-dimensional model representations, Journal of Mathematical Chemistry 25 (1999) 197–233.

[50] M. G. Larson, F. Bengzon, The finite element method: theory, implementation, and applications, volume 10, Springer Science & Business Media, 2013.

[51] P. D. Spanos, M. Beer, J. Red-Horse, Karhunen–Loève expansion of stochastic processes with a modified exponential covariance kernel, Journal of Engineering Mechanics 133 (2007) 773–779.

[52] M. G. Faes, M. Broggi, P. D. Spanos, M. Beer, Elucidating appealing features of differentiable auto-correlation functions: A study on the modified exponential kernel, Probabilistic Engineering Mechanics (2022) 103269.

[53] Z. Liu, W. Liu, Y. Peng, Random function based spectral representation of stationary and non-stationary stochastic processes, Probabilistic Engineering Mechanics 45 (2016) 115–126.

[54] G. Chen, D. Yang, Direct probability integral method for stochastic response analysis of static and dynamic structural systems, Computer Methods in Applied Mechanics and Engineering 357 (2019) 112612.

[55] P. Ni, D. J. Jerez, V. C. Fragkoulis, M. G. Faes, M. A. Valdebenito, M. Beer, Operator norm-based statistical linearization to bound the first excursion probability of nonlinear structures subjected to imprecise stochastic loading, ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering 8 (2022) 04021086.