



Stochastic virtual element methods for uncertainty propagation of stochastic linear elasticity

Zhibao Zheng¹ · Udo Nackenhorst¹

Received: 23 April 2023 / Accepted: 13 August 2023
© The Author(s) 2023

Abstract

This paper presents stochastic virtual element methods for propagating uncertainty in linear elastic stochastic problems. We first derive stochastic virtual element equations for 2D and 3D linear elastic problems that may involve uncertainties in material properties, external forces, boundary conditions, etc. A stochastic virtual element space that couples the deterministic virtual element space and the stochastic space is constructed for this purpose and used to approximate the unknown stochastic solution. Two numerical frameworks are then developed to solve the derived stochastic virtual element equations, including a Polynomial Chaos approximation based approach and a weakly intrusive approximation based approach. In the Polynomial Chaos based framework, the stochastic solution is approximated using the Polynomial Chaos basis and solved via an augmented deterministic virtual element equation that is generated by applying the stochastic Galerkin procedure to the original stochastic virtual element equation. In the weakly intrusive approximation based framework, the stochastic solution is approximated by a summation of a set of products of random variables and deterministic vectors, where the deterministic vectors are solved via converting the original stochastic problem to deterministic virtual element equations by the stochastic Galerkin approach, and the random variables are solved via converting the original stochastic problem to one-dimensional stochastic algebraic equations by the classical Galerkin procedure. This method avoids the curse of dimensionality in high-dimensional stochastic problems successfully since all random inputs are embedded into one-dimensional stochastic algebraic equations whose computational effort weakly depends on the stochastic dimension. Numerical results on 2D and 3D problems with low- and high-dimensional random inputs demonstrate the good performance of the proposed methods.

Keywords Stochastic virtual element method · Polynomial Chaos expansion · Weakly intrusive approximation · Curse of dimensionality · Uncertainty quantification

1 Introduction

Numerical techniques for solving complex partial differential equations are continuously developing at an incredible rate, e.g. finite difference methods, finite volume methods, finite element methods, spectral element methods, etc [23]. As a generalization of the finite element method, the virtual element method (VEM) has been proposed and received a lot of attention in the last decade [4, 20, 30, 31]. Compared to the classical finite element method, VEM can discretize 2D or 3D geometric domains utilizing arbitrary polygons

or polyhedrons and is not limited to the regular elements used in the finite element method, which is thus highly flexible and mesh insensitive. In VEM, the shape functions can be non-polynomial. It does not require constructing explicit shape functions on elements since all numerical integration is transferred to edges rather than performed on the elements. Several limitations of finite element methods are also avoided, such as convex elements and element degradation caused by small edges and interior angles. Applications of VEM to various problems have been extensively studied, such as linear elastic problems [5, 11, 14], large deformation problems [9, 34], contact problems [2, 32, 33], fracture and crack propagation problems [1, 7, 17], topology optimization [3, 10], etc. Although extensive studies have been conducted, there is still a gap in using VEM to deal with problems with uncertainties. In many practical engineering problems, the inherent or epistemic uncertainty of systems

✉ Zhibao Zheng
zhibao.zheng@ibnm.uni-hannover.de

¹ Leibniz Universität Hannover, Institute of Mechanics and Computational Mechanics & International Research Training Group 2657, Appelstraße 9a, 30167 Hannover, Germany

are unavoidable. Predicting uncertainty propagation on the physical models has become an important part of the analysis of systems [26], which leads to the development of dedicated numerical methods for uncertainty analysis.

In this paper, we focus on extending the deterministic VEM to stochastic VEMs (SVEMs) for the uncertainty analysis of 2D and 3D linear elastic stochastic problems that may involve random material properties and stochastic external forces, etc., which is currently still lacking. Our main contributions in this paper consist of two parts: the first contribution is to extend the deterministic virtual element discretization to stochastic cases and derive corresponding stochastic virtual element equations (SVEEs), and the second contribution is to present two numerical methods to solve the derived SVEEs efficiently and accurately. For the first contribution, we extend the deterministic virtual element space to a stochastic virtual element space that couples the classical virtual element space and the stochastic space, which can provide (stochastic) virtual element approximations for both deterministic and stochastic functions in the space. The constructed stochastic virtual element space can be considered as a deterministic virtual element space parameterized by random inputs. For each sample realization of the random input, it degenerates into a deterministic virtual element space and inherits all properties of classical virtual element spaces. Thus, we can simply approximate the stochastic solution using a linear combination of deterministic virtual basis functions with random coefficients (i.e. the unknown stochastic solution). Further, numerical techniques for calculating the gradients of virtual basis functions and the stabilization term in the deterministic VEM [5, 10, 14, 31] can still be used to the stochastic discretization with slight modifications. In this way we can obtain SVEEs by assembling the stochastic stiffness matrix and the stochastic force vector with a complexity similar to the deterministic VEM.

Similar to VEM being a generalization of the finite element method, SVEM is also considered to be a generalization of the stochastic finite element method [28]. Thus, numerical algorithms for solving the derived SVEEs can benefit from stochastic finite element solution algorithms, such as Monte Carlo simulation (MCS) and its improvements [16, 21], spectral stochastic finite element methods [15, 36], stochastic collocation methods [6, 35], response surface and kriging methods [13, 19], etc. For the second contribution, we develop two numerical methods to solve the derived SVEE, including a Polynomial Chaos expansion based SVEM (PC-SVEM) and a Weakly INtrusive approximation based SVEM (WIN-SVEM). The other methods mentioned above can also be extended to SVEMs in a similar way to this paper. The PC-SVEM is a natural extension of the spectral stochastic finite element method [15]. In this method, the stochastic solution is decomposed into a summation of a set of products of PC basis and deterministic vectors. By the use of stochas-

tic Galerkin procedure, the original SVEE is transformed into an augmented deterministic equation whose size is much larger than the original SVEE. Also, the size increases dramatically as the degree of freedom of physical models, the stochastic dimension and the expansion order of PC basis increase, which leads to the curse of dimensionality when dealing with large-scale and/or high-dimensional stochastic problems. To address this issue, we further present the WIN-SVEM, which is an extension of our previous work for solving stochastic finite element equations [41, 42]. In this method, the stochastic solution is approximated by a summation of a set of products of random variables and deterministic vectors. Different from the PC-SVEM, both random variables and deterministic vectors are not known a priori. To this end, we solve them using a dedicated iteration. The deterministic vectors are solved via a few deterministic equations that are obtained by a similar stochastic Galerkin process used to PC-SVEM. The random variables are solved via one-dimensional stochastic algebraic equations that are obtained by applying the classical Galerkin procedure to the original SVEE. In this way, all random inputs are embedded into these one-dimensional stochastic algebraic equations, and their efficient and accurate solutions are achieved using a non-intrusive sampling method with weak dimension dependence. The proposed WIN-SVEM thus avoids the curse of dimensionality in high-dimensional stochastic problems successfully.

The paper is organized as follows: Sect. 2 presents the stochastic virtual element discretization for linear elastic stochastic problems and stochastic virtual element equations are then derived. In Sect. 3, the PC-SVEM is developed to solve the derived stochastic systems. Following that, the WIN-SVEM is proposed in Sect. 4 to solve the derived stochastic systems efficiently, with special emphasis on high-dimensional stochastic problems. 2D and 3D numerical examples involving low- and high-dimensional random inputs are given in Sect. 5 to demonstrate the performance of the proposed methods. Conclusions and outlook follow in Sect. 6.

2 Stochastic virtual element equations

2.1 Stochastic elastic equations

Let $(\Theta, \mathfrak{E}, \mathcal{P})$ be a suitable probability space, where Θ denotes the space of elementary events, \mathfrak{E} is a σ -algebra defined on Θ and \mathcal{P} is a probability measure. In this paper, we consider the following elastic stochastic equation

$$\begin{cases} -\nabla \cdot \boldsymbol{\sigma}(\mathbf{x}, \theta) = \mathbf{f}(\mathbf{x}, \theta) & \text{in } \Omega \\ \boldsymbol{\sigma}(\mathbf{x}, \theta) \cdot \mathbf{n} = \mathbf{g}(\mathbf{x}, \theta) & \text{on } \Gamma_N, \\ \mathbf{u}(\mathbf{x}, \theta) = \mathbf{u}_D(\mathbf{x}, \theta) & \text{on } \Gamma_D \end{cases} \quad (1)$$

where the domain $\Omega \subset \mathbb{R}^d$ with the boundary $\partial\Omega$, the spatial dimension may be $d = 2, 3$ and the d -dimensional spatial coordinate is given by $\mathbf{x} = (x_1, \dots, x_d) \in \Omega$, $\nabla \cdot (\cdot)$ denotes the divergence operator, $\boldsymbol{\sigma}(\mathbf{x}, \theta)$ is the stochastic stress tensor, the vector-valued displacement field $\mathbf{u}(\mathbf{x}, \theta) = [u_1(\mathbf{x}, \theta), \dots, u_d(\mathbf{x}, \theta)]^T \in \mathbb{R}^d$ is the unknown stochastic solution that needs to be solved, the vector-valued field $\mathbf{f}(\mathbf{x}, \theta) = [f_1(\mathbf{x}, \theta), \dots, f_d(\mathbf{x}, \theta)]^T \in \mathbb{R}^d$ is associated with stochastic external forces, and Γ_N and Γ_D are boundary segments associated with the Neumann boundary condition $\mathbf{g}(\mathbf{x}, \theta) = [g_1(\mathbf{x}, \theta), \dots, g_d(\mathbf{x}, \theta)]^T \in \mathbb{R}^d$ and the Dirichlet boundary condition $\mathbf{u}_D(\mathbf{x}, \theta) = [u_{D,1}(\mathbf{x}, \theta), \dots, u_{D,d}(\mathbf{x}, \theta)]^T \in \mathbb{R}^d$. In this paper, we only consider linear elastic stochastic problems. The following linear stochastic strain tensor and linear elastic constitutive relation are adopted

$$\boldsymbol{\varepsilon}(\mathbf{u}(\mathbf{x}, \theta)) = \frac{1}{2} [\nabla \mathbf{u}(\mathbf{x}, \theta) + (\nabla \mathbf{u}(\mathbf{x}, \theta))^T] \in \mathbb{R}^{d \times d}, \quad (2)$$

$$\boldsymbol{\sigma}(\mathbf{u}(\mathbf{x}, \theta), \theta) = \mathbf{C}(\mathbf{x}, \theta) \boldsymbol{\varepsilon}(\mathbf{u}(\mathbf{x}, \theta)) \in \mathbb{R}^{d \times d},$$

where $\mathbf{C}(\mathbf{x}, \theta)$ is a fourth order elastic tensor that may be related to stochastic material properties, e.g. stochastic Young's modulus and stochastic Poisson ratio.

To solve (1), let us consider its weak form written as follows: find a stochastic function $\mathbf{u}(\mathbf{x}, \theta) \in \mathcal{V} : \Omega \times \Theta \rightarrow \mathbb{R}^d$ such that the following equation holds for \mathcal{P} -almost surely $\theta \in \Theta$,

$$\mathcal{W}(\mathbf{u}(\mathbf{x}, \theta), \mathbf{v}(\mathbf{x}), \theta) = \mathcal{F}(\mathbf{v}(\mathbf{x}), \theta), \quad \forall \mathbf{v}(\mathbf{x}) \in \mathcal{V}, \quad (3)$$

where the functional space is defined as $\mathcal{V} = \{ \mathbf{v} \in [\mathcal{H}^1(\Omega)]^d : \mathbf{v} = \mathbf{0} \text{ on } \Gamma_D \}$, and $\mathcal{H}^1(\Omega)$ is the subspace of the space of square integrable scalar functions on Ω that contains both the function and its weak derivatives [37]. The left-side term $\mathcal{W}(\mathbf{u}(\mathbf{x}, \theta), \mathbf{v}(\mathbf{x}), \theta)$ and the right-side term $\mathcal{F}(\mathbf{v}(\mathbf{x}), \theta)$ are given by

$$\begin{aligned} \mathcal{W}(\mathbf{u}(\mathbf{x}, \theta), \mathbf{v}(\mathbf{x}), \theta) &= \int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}(\mathbf{x}, \theta), \theta) : \boldsymbol{\varepsilon}(\mathbf{v}(\mathbf{x})) \, d\mathbf{x} \\ &= \int_{\Omega} [\mathbf{C}(\mathbf{x}, \theta) \boldsymbol{\varepsilon}(\mathbf{u}(\mathbf{x}, \theta))] : \boldsymbol{\varepsilon}(\mathbf{v}(\mathbf{x})) \, d\mathbf{x}, \end{aligned} \quad (4)$$

$$\begin{aligned} \mathcal{F}(\mathbf{v}(\mathbf{x}), \theta) &= \int_{\Omega} \mathbf{f}(\mathbf{x}, \theta) \cdot \mathbf{v}(\mathbf{x}) \, d\mathbf{x} \\ &+ \int_{\Gamma_N} \mathbf{g}(\mathbf{x}, \theta) \cdot \mathbf{v}(\mathbf{x}) \, ds. \end{aligned} \quad (5)$$

2.2 Stochastic virtual element discretization

We adopt the stochastic virtual element discretization for the weak form (3). Specifically, the domain Ω is partitioned into n_e non-overlapping polygonal elements $\Omega = \bigcup_{e=1}^{n_e} \Omega^{(e)}$, and

each element $\Omega^{(e)}$, $e = 1, \dots, n_e$ includes $n^{(e)}$ vertices and $m^{(e)}$ edges. In this paper, we only consider the lowest-order virtual element, but the proposed method can be extended to higher-order virtual elements [12]. We give the following approximate discretized virtual space $\mathcal{V}_h(\Omega^{(e)}) \subset \mathcal{V}$ of the element $\Omega^{(e)}$

$$\begin{aligned} \mathcal{V}_h(\Omega^{(e)}) &= \left\{ \mathbf{v}_h \in \left[\mathcal{H}^1(\Omega^{(e)}) \cap \mathcal{C}^0(\Omega^{(e)}) \right]^d : \right. \\ &\mathbf{v}_{h, \partial\Omega_i^{(e)}} \in \left[\mathcal{P}_1(\partial\Omega_i^{(e)}) \right]^d, \forall \partial\Omega_i^{(e)} \in \partial\Omega^{(e)}, \\ &\left. \nabla \cdot [\mathbf{C}(\theta) \boldsymbol{\varepsilon}(\mathbf{v}_h)] = \mathbf{0}, \forall \theta \in \Theta \text{ on } \Omega^{(e)} \right\}, \end{aligned} \quad (6)$$

where \mathcal{P}_1 represents the space of polynomials of degree up to 1, \mathbf{v}_h is a polynomial on each edge $\partial\Omega_i^{(e)}$, $i = 1, \dots, m^{(e)}$ of $\Omega^{(e)}$ and \mathcal{C}^0 -continuity on the element $\Omega^{(e)}$, and $\nabla \cdot [\mathbf{C}(\theta) \boldsymbol{\varepsilon}(\mathbf{v}_h)]$ vanishes on the element $\Omega^{(e)}$ for all $\theta \in \Theta$. In this way, we couple the classical virtual element space and the random input θ . The function \mathbf{v}_h is not known on the element $\Omega^{(e)}$ but explicitly known on the edge $\partial\Omega^{(e)}$. We now consider the weak form (3) on the discretized space: find $\mathbf{u}_h(\mathbf{x}, \theta) \in \mathcal{V}_h : \Omega \times \Theta \rightarrow \mathbb{R}^d$ such that the following equation holds for \mathcal{P} -almost surely $\theta \in \Theta$ and $\forall \mathbf{v}_h(\mathbf{x}) \in \mathcal{V}_h$

$$\mathcal{W}_h(\mathbf{u}_h(\mathbf{x}, \theta), \mathbf{v}_h(\mathbf{x}), \theta) = \mathcal{F}_h(\mathbf{v}_h(\mathbf{x}), \theta), \quad (7)$$

where \mathcal{W}_h and \mathcal{F}_h are assembled by looping over all elements $\{\Omega^{(e)}\}_{e=1}^{n_e}$

$$\begin{aligned} \mathcal{W}_h(\mathbf{u}_h(\mathbf{x}, \theta), \mathbf{v}_h(\mathbf{x}), \theta) &= \sum_{e=1}^{n_e} \mathcal{W}_h^{(e)}(\mathbf{u}_h(\mathbf{x}, \theta), \mathbf{v}_h(\mathbf{x}), \theta), \end{aligned} \quad (8)$$

$$\mathcal{F}_h(\mathbf{v}_h(\mathbf{x}), \theta) = \sum_{e=1}^{n_e} \mathcal{F}_h^{(e)}(\mathbf{v}_h(\mathbf{x}), \theta). \quad (9)$$

It is noted that $\mathcal{W}_h^{(e)}(\mathbf{u}_h(\mathbf{x}, \theta), \mathbf{v}_h(\mathbf{x}), \theta)$ and $\mathcal{F}_h^{(e)}(\mathbf{v}_h(\mathbf{x}), \theta)$ cannot be evaluated in a similar way to the classical finite element method since the function \mathbf{v}_h is unknown on the element $\Omega^{(e)}$. To this end, a projection operator $\boldsymbol{\Pi}^{(e)} : \mathcal{V}_h(\Omega^{(e)}) \rightarrow [\mathcal{P}_1(\Omega^{(e)})]^d$ is defined similar to the deterministic VEM [11, 30] such that for $\forall \mathbf{v}_h \in \mathcal{V}_h(\Omega^{(e)})$, $\forall \mathbf{p}_1 \in [\mathcal{P}_1(\Omega^{(e)})]^d$ and $\forall \theta \in \Theta$,

$$\mathcal{W}_h^{(e)}(\mathbf{v}_h(\mathbf{x}) - \boldsymbol{\Pi}^{(e)} \mathbf{v}_h(\mathbf{x}), \mathbf{p}_1(\mathbf{x}), \theta) = 0 \quad (10)$$

holds. Further, we can reformulate the stochastic term $\mathcal{W}_h^{(e)}(\mathbf{u}_h(\mathbf{x}, \theta), \mathbf{v}_h(\mathbf{x}), \theta)$ as

$$\begin{aligned}
 & \mathcal{W}_h^{(e)}(\mathbf{u}_h(\mathbf{x}, \theta), \mathbf{v}_h(\mathbf{x}), \theta) \\
 &= \mathcal{W}_h^{(e)}\left(\Pi^{(e)}\mathbf{u}_h(\mathbf{x}, \theta) + \mathbf{u}_h(\mathbf{x}, \theta) - \Pi^{(e)}\mathbf{u}_h(\mathbf{x}, \theta), \right. \\
 & \quad \left. \Pi^{(e)}\mathbf{v}_h(\mathbf{x}) + \mathbf{v}_h(\mathbf{x}) - \Pi^{(e)}\mathbf{v}_h(\mathbf{x}), \theta\right), \tag{11} \\
 &= \underbrace{\mathcal{W}_h^{(e)}\left(\Pi^{(e)}\mathbf{u}_h(\mathbf{x}, \theta), \Pi^{(e)}\mathbf{v}_h(\mathbf{x}), \theta\right)}_{=\mathcal{W}_C^{(e)}(\theta)} \\
 & \quad + \underbrace{\mathcal{W}_h^{(e)}\left(\mathbf{u}_h(\mathbf{x}, \theta) - \Pi^{(e)}\mathbf{u}_h(\mathbf{x}, \theta), \mathbf{v}_h(\mathbf{x}) - \Pi^{(e)}\mathbf{v}_h(\mathbf{x}), \theta\right)}_{=\mathcal{W}_S^{(e)}(\theta)} \\
 & \quad + \underbrace{\mathcal{W}_h^{(e)}\left(\Pi^{(e)}\mathbf{u}_h(\mathbf{x}, \theta), \mathbf{v}_h(\mathbf{x}) - \Pi^{(e)}\mathbf{v}_h(\mathbf{x}), \theta\right)}_{=0} \\
 & \quad + \underbrace{\mathcal{W}_h^{(e)}\left(\mathbf{u}_h(\mathbf{x}, \theta) - \Pi^{(e)}\mathbf{u}_h(\mathbf{x}, \theta), \Pi^{(e)}\mathbf{v}_h(\mathbf{x}), \theta\right)}_{=0}, \tag{12}
 \end{aligned}$$

where $\mathcal{W}_C^{(e)}(\theta)$ and $\mathcal{W}_S^{(e)}(\theta)$ are called the stochastic consistent term and the stochastic stabilization term, respectively. The last two terms are obtained according to (10) due to $\Pi^{(e)}\mathbf{v}_h(\mathbf{x}), \Pi^{(e)}\mathbf{u}_h(\mathbf{x}, \theta) \in [\mathcal{P}_1(\Omega^{(e)})]^d$. Furthermore, $\boldsymbol{\varepsilon}^{(e)}(\Pi^{(e)}\mathbf{u}_h(\mathbf{x}, \theta))$ and $\boldsymbol{\varepsilon}^{(e)}(\Pi^{(e)}\mathbf{v}_h(\mathbf{x}))$ are (stochastic) constant strain tensors. Therefore, the stochastic consistent term $\mathcal{W}_C^{(e)}(\theta)$ is evaluated via

$$\begin{aligned}
 \mathcal{W}_C^{(e)}(\theta) &= \int_{\Omega^{(e)}} \left[\mathbf{C}^{(e)}(\mathbf{x}, \theta) \boldsymbol{\varepsilon}^{(e)}(\Pi^{(e)}\mathbf{u}_h(\mathbf{x}, \theta)) \right] : \\
 & \quad \boldsymbol{\varepsilon}^{(e)}(\Pi^{(e)}\mathbf{v}_h(\mathbf{x})) \, d\mathbf{x} \\
 &= a^{(e)} \left[\mathbf{C}^{(e)}(\mathbf{x}, \theta) \boldsymbol{\varepsilon}^{(e)}(\Pi^{(e)}\mathbf{u}_h(\mathbf{x}, \theta)) \right] : \\
 & \quad \boldsymbol{\varepsilon}^{(e)}(\Pi^{(e)}\mathbf{v}_h(\mathbf{x})), \tag{13}
 \end{aligned}$$

where $a^{(e)}$ is the area (for 2D polygonal element) or the volume (for 3D polygonal element) of the element $\Omega^{(e)}$.

Further, we let the stochastic solution $\mathbf{u}_h(\mathbf{x}, \theta)$ and the function $v_h(\mathbf{x})$ on the element $\Omega^{(e)}$ be approximated using a set of virtual basis functions $\{\varphi_i(\mathbf{x}) \in \mathcal{V}_h(\Omega^{(e)})\}_{i=1}^{n^{(e)}}$

$$\begin{aligned}
 \mathbf{u}_h(\mathbf{x}, \theta) &= \sum_{i=1}^{n^{(e)}} \varphi_i(\mathbf{x}) \mathbf{u}_i^{(e)}(\theta), \\
 v_h(\mathbf{x}) &= \sum_{i=1}^{n^{(e)}} \varphi_i(\mathbf{x}) \mathbf{v}_i^{(e)}, \tag{14}
 \end{aligned}$$

where the stochastic solution vector $\mathbf{u}_i^{(e)}(\theta)$ of the element $\Omega^{(e)}$ is given by $\mathbf{u}^{(e)}(\theta) = [\mathbf{u}_1^{(e)T}(\theta), \dots, \mathbf{u}_{n^{(e)}}^{(e)T}(\theta)]^T \in \mathbb{R}^{n^{(e)}d}$, and the component $\mathbf{u}_i^{(e)}(\theta) = [u_{i,1}^{(e)}(\theta), \dots, u_{i,d}^{(e)}(\theta)]^T \in \mathbb{R}^d$ is the stochastic solution vector of the i -th vertex. The

vector $\mathbf{v}_i^{(e)} \in \mathbb{R}^{n^{(e)}d}$ has a similar expression but does not involve the random input θ . Applying the projection operator $\Pi^{(e)}$ to $\mathbf{u}_h(\mathbf{x}, \theta)$ and $\mathbf{v}_h(\mathbf{x})$ we have

$$\begin{aligned}
 \Pi^{(e)}\mathbf{u}_h(\mathbf{x}, \theta) &= \sum_{i=1}^{n^{(e)}} \Pi^{(e)}\varphi_i(\mathbf{x}) \mathbf{u}_i^{(e)}(\theta), \\
 \Pi^{(e)}\mathbf{v}_h(\mathbf{x}) &= \sum_{i=1}^{n^{(e)}} \Pi^{(e)}\varphi_i(\mathbf{x}) \mathbf{v}_i^{(e)}. \tag{15}
 \end{aligned}$$

On the basis of this we rewrite the strain tensors $\boldsymbol{\varepsilon}^{(e)}(\Pi^{(e)}\mathbf{u}_h(\mathbf{x}, \theta))$ and $\boldsymbol{\varepsilon}^{(e)}(\Pi^{(e)}\mathbf{v}_h(\mathbf{x}))$ as the following vector forms

$$\begin{aligned}
 \boldsymbol{\varepsilon}_{\Pi, \text{vec}}^{(e)}(\mathbf{u}^{(e)}(\theta)) &= \mathbf{B}^{(e)}\mathbf{u}^{(e)}(\theta) \\
 &= [\mathbf{B}_1^{(e)}, \dots, \mathbf{B}_{n^{(e)}}^{(e)}] \mathbf{u}^{(e)}(\theta) \in \mathbb{R}^{\frac{d(d+1)}{2}}, \\
 \boldsymbol{\varepsilon}_{\Pi, \text{vec}}^{(e)}(\mathbf{v}^{(e)}) &= \mathbf{B}^{(e)}\mathbf{v}^{(e)} \in \mathbb{R}^{\frac{d(d+1)}{2}}, \tag{16}
 \end{aligned}$$

where the component matrices $\mathbf{B}_i^{(e)} \in \mathbb{R}^{\frac{d(d+1)}{2} \times d}$, $i = 1, \dots, n^{(e)}$ are given by

$$\begin{aligned}
 \mathbf{B}_i^{(e)} &= \begin{bmatrix} \frac{\partial \Pi^{(e)}\varphi_i(\mathbf{x})}{\partial x_1} & 0 \\ 0 & \frac{\partial \Pi^{(e)}\varphi_i(\mathbf{x})}{\partial x_2} \\ \frac{\partial \Pi^{(e)}\varphi_i(\mathbf{x})}{\partial x_2} & \frac{\partial \Pi^{(e)}\varphi_i(\mathbf{x})}{\partial x_1} \end{bmatrix} \quad (d = 2) \\
 \text{or} &= \begin{bmatrix} \frac{\partial \Pi^{(e)}\varphi_i(\mathbf{x})}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial \Pi^{(e)}\varphi_i(\mathbf{x})}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial \Pi^{(e)}\varphi_i(\mathbf{x})}{\partial x_3} \\ \frac{\partial \Pi^{(e)}\varphi_i(\mathbf{x})}{\partial x_2} & \frac{\partial \Pi^{(e)}\varphi_i(\mathbf{x})}{\partial x_1} & 0 \\ 0 & \frac{\partial \Pi^{(e)}\varphi_i(\mathbf{x})}{\partial x_3} & \frac{\partial \Pi^{(e)}\varphi_i(\mathbf{x})}{\partial x_2} \\ \frac{\partial \Pi^{(e)}\varphi_i(\mathbf{x})}{\partial x_3} & 0 & \frac{\partial \Pi^{(e)}\varphi_i(\mathbf{x})}{\partial x_1} \end{bmatrix} \quad (d = 3), \tag{17}
 \end{aligned}$$

where the components $\frac{\partial \Pi^{(e)}\varphi_i(\mathbf{x})}{\partial x_j}$, $i = 1, \dots, n^{(e)}$, $j = 1, \dots, d$ are evaluated via transferring the calculations to edges and faces of the element $\Omega^{(e)}$ based on (10), which is the same as the deterministic virtual element method, see [5, 14, 30, 31] for details. In this way, it does not require knowing explicit representations of the functions $\{\varphi_i(\mathbf{x})\}_{i=1}^{n^{(e)}}$ and only needs to know their traces on edges. The Lagrangian linear basis functions similar to those used in the classical finite element method can be adopted for the purpose. Substituting (16) into (13) we have

$$\mathcal{W}_C^{(e)}(\theta) = a^{(e)} \mathbf{v}^{(e)T} \mathbf{B}^{(e)T} \mathbf{G}^{(e)}(\theta) \mathbf{B}^{(e)} \mathbf{u}^{(e)}(\theta), \tag{18}$$

where $\mathbf{G}^{(e)}(\theta) \in \mathbb{R}^{\frac{d(d+1)}{2} \times \frac{d(d+1)}{2}}$ is the matrix form of the tensor $\mathbf{C}(\mathbf{x}, \theta)$ of the element $\Omega^{(e)}$. Two detailed representations of $\mathbf{G}(\theta)$ for 2D and 3D problems can be found in the

numerical example section. Hence, the stochastic element stiffness matrix corresponding to the stochastic consistent term $\mathcal{W}_C^{(e)}(\theta)$ is given by

$$\mathbf{k}_C^{(e)}(\theta) = a^{(e)} \mathbf{B}^{(e)T} \mathbf{G}^{(e)}(\theta) \mathbf{B}^{(e)} \in \mathbb{R}^{n^{(e)d} \times n^{(e)d}}. \tag{19}$$

Further, let us consider the calculation of the stochastic stabilization term $\mathcal{W}_S^{(e)}(\theta)$ in (12), which can be achieved taking advantage of several numerical strategies [5, 14, 30]. Here we adopt the approach presented in [10, 31], which corresponds to

$$\mathcal{W}_S^{(e)}(\theta) = \gamma_S^{(e)}(\theta) \sum_{i=1}^{n^{(e)}} \left[\mathbf{u}_h(\mathbf{x}_i, \theta) - \mathbf{\Pi}^{(e)} \mathbf{u}_h(\mathbf{x}_i, \theta) \right] \cdot \left[\mathbf{v}_h(\mathbf{x}_i) - \mathbf{\Pi}^{(e)} \mathbf{v}_h(\mathbf{x}_i) \right], \tag{20}$$

where $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,d})$, $x_{i,j}$ denotes the i -th coordinate value of the j -th vertex of the element $\Omega^{(e)}$. In the practical implementation, it is calculated as [10]

$$\mathcal{W}_S^{(e)}(\theta) = \gamma_S^{(e)}(\theta) \mathbf{v}^{(e)T} \left(\mathbf{I}_{n^{(e)d}} - \mathbf{S}^{(e)} \right)^T \cdot \left(\mathbf{I}_{n^{(e)d}} - \mathbf{S}^{(e)} \right) \mathbf{u}^{(e)}(\theta), \tag{21}$$

where the coefficient $\gamma_S^{(e)}(\theta) = \frac{1}{2} \text{Tr}(\mathbf{G}^{(e)}(\theta))$, $\text{Tr}(\cdot)$ is the trace operator of matrices, $\mathbf{I}_{n^{(e)d}} \in \mathbb{R}^{n^{(e)d} \times n^{(e)d}}$ is the identity matrix, and the deterministic matrix $\mathbf{S}^{(e)} \in \mathbb{R}^{n^{(e)d} \times n^{(e)d}}$ is given by

$$\mathbf{S}^{(e)} = \begin{bmatrix} h_{11}^{(e)} \mathbf{I}_d & \cdots & h_{1n^{(e)}}^{(e)} \mathbf{I}_d \\ \vdots & \ddots & \vdots \\ h_{n^{(e)}1}^{(e)} \mathbf{I}_d & \cdots & h_{n^{(e)}n^{(e)}}^{(e)} \mathbf{I}_d \end{bmatrix}, \tag{22}$$

$$h_{ij}^{(e)} = \mathbf{X}_i^{(e)T} \mathbf{A}_j^{(e)} + \frac{1}{n^{(e)}}, \tag{23}$$

where the vectors $\mathbf{X}_i^{(e)} \in \mathbb{R}^d$ and $\mathbf{A}_i^{(e)} \in \mathbb{R}^d$ are given by

$$\mathbf{X}_i^{(e)} = \left[x_{1,i} - \frac{1}{n^{(e)}} \sum_{j=1}^{n^{(e)}} x_{1,j}, \dots, x_{d,i} - \frac{1}{n^{(e)}} \sum_{j=1}^{n^{(e)}} x_{d,j} \right]^T$$

$$\mathbf{A}_i^{(e)} = \left[\frac{\partial \mathbf{\Pi} \varphi_i(\mathbf{x})}{\partial x_1}, \dots, \frac{\partial \mathbf{\Pi} \varphi_i(\mathbf{x})}{\partial x_d} \right]^T \tag{24}$$

Hence, the stochastic element stiffness matrix $\mathbf{k}_S^{(e)}(\theta) \in \mathbb{R}^{n^{(e)d} \times n^{(e)d}}$ corresponding to the stochastic stabilization term $\mathcal{W}_S^{(e)}(\theta)$ is given by

$$\mathbf{k}_S^{(e)}(\theta) = \gamma_S^{(e)}(\theta) \left(\mathbf{I}_{n^{(e)d}} - \mathbf{S}^{(e)} \right)^T \left(\mathbf{I}_{n^{(e)d}} - \mathbf{S}^{(e)} \right), \tag{25}$$

which is very close to that in the deterministic VEM, but the coefficient $\gamma_S^{(e)}(\theta)$ involves the random input θ . The calculation procedure of deterministic virtual element matrices can thus be inherited for $\mathbf{k}_S^{(e)}(\theta)$. We only need to pay a little attention to the calculation of the random coefficient $\gamma_S^{(e)}(\theta)$.

In the last step, let us consider the stochastic term $\mathcal{F}_h^{(e)}(\mathbf{v}_h(\mathbf{x}), \theta)$ calculated exactly using the one-point integration rule on the edges and the face (for 2D case) or the faces and the element (for 3D case)

$$\begin{aligned} \mathcal{F}_h^{(e)}(\mathbf{v}_h(\mathbf{x}), \theta) &= \int_{\Omega^{(e)}} f^{(e)}(\mathbf{x}, \theta) \cdot \mathbf{v}_h(\mathbf{x}) \, d\mathbf{x} \\ &\quad + \int_{\Gamma_N^{(e)}} \mathbf{g}^{(e)}(\mathbf{x}, \theta) \cdot \mathbf{v}_h(\mathbf{x}) \, d\mathbf{s} \\ &= \frac{a^{(e)}}{(n^{(e)})^2} \mathbf{v}^{(e)T} \mathbf{Z}_1^T \mathbf{Z}_1 \mathbf{f}^{(e)}(\theta) \\ &\quad + \mathbf{v}^{(e)T} \sum_{j=1}^{n_{\Gamma_N}^{(e)}} \frac{b_j^{(e)}}{(n_{\Gamma_N,j}^{(e)})^2} \mathbf{Z}_{2,j}^T \mathbf{Z}_{2,j} \mathbf{g}^{(e)}(\theta), \end{aligned} \tag{26}$$

where $n_{\Gamma_N}^{(e)}$ is the number of edges (for 2D case) or faces (for 3D case) of the element $\Omega^{(e)}$, $\Gamma_{N,j}^{(e)}$ is the j -th edge or face and includes $n_{\Gamma_{N,j}^{(e)}}^{(e)}$ vertices, and $b_j^{(e)}$ is the length (for 2D element) or the area (for 3D element) of $\Gamma_{N,j}^{(e)}$. The vectors consisting of the values on each vertex are $\mathbf{f}^{(e)}(\theta) = \left[f_1^{(e)T}(\theta), \dots, f_{n^{(e)}}^{(e)T}(\theta) \right]^T \in \mathbb{R}^{n^{(e)d}}$, $\mathbf{f}_i^{(e)}(\theta) = \left[f_{i,1}^{(e)}(\theta), \dots, f_{i,d}^{(e)}(\theta) \right]^T \in \mathbb{R}^d$ and $\mathbf{g}^{(e)}(\theta) = \left[\mathbf{g}_1^{(e)T}(\theta), \dots, \mathbf{g}_{n^{(e)}}^{(e)T}(\theta) \right]^T \in \mathbb{R}^{n^{(e)d}}$, $\mathbf{g}_i^{(e)}(\theta) = \left[g_{i,1}^{(e)}(\theta), \dots, g_{i,d}^{(e)}(\theta) \right]^T \in \mathbb{R}^d$. The deterministic matrices

are given by $\mathbf{Z}_1 = \left[\begin{matrix} \mathbf{I}_d, \dots, \mathbf{I}_d \\ n^{(e)} \end{matrix} \right] \in \mathbb{R}^{d \times n^{(e)d}}$ and $\mathbf{Z}_{2,j} = \left[\delta_{1,\Gamma_{N,j}^{(e)}}^* \mathbf{I}_d, \dots, \delta_{n^{(e)},\Gamma_{N,j}^{(e)}}^* \mathbf{I}_d \right] \in \mathbb{R}^{d \times n^{(e)d}}$, where $\delta_{i,\Gamma_{N,j}^{(e)}}^* = 1$ if the vertex $i \in \Gamma_{N,j}^{(e)}$ and 0 otherwise. Thus, the stochastic element force vector is calculated as

$$\begin{aligned} \mathbf{f}^{(e)}(\theta) &= \frac{a^{(e)}}{(n^{(e)})^2} \mathbf{Z}_1^T \mathbf{Z}_1 \mathbf{f}^{(e)}(\theta) \\ &\quad + \sum_{j=1}^{n_{\Gamma_N}^{(e)}} \frac{b_j^{(e)}}{(n_{\Gamma_{N,j}^{(e)}}^{(e)})^2} \mathbf{Z}_{2,j}^T \mathbf{Z}_{2,j} \mathbf{g}^{(e)}(\theta) \in \mathbb{R}^{n^{(e)d}}. \end{aligned} \tag{27}$$

Assembling the above stochastic element stiffness matrices and stochastic element force vector we obtain the following SVEE

$$\mathbf{K}(\theta) \mathbf{u}(\theta) = \mathbf{F}(\theta), \quad (28)$$

where the global stochastic stiffness matrix $\mathbf{K}(\theta)$ and the the global stochastic force vector $\mathbf{F}(\theta)$ are assembled by

$$\begin{aligned} \mathbf{K}(\theta) &= \bigcup_{e=1}^{n_e} \left(\mathbf{k}_C^{(e)}(\theta) + \mathbf{k}_S^{(e)}(\theta) \right) \in \mathbb{R}^{n \times n}, \\ \mathbf{F}(\theta) &= \bigcup_{e=1}^{n_e} \mathbf{f}^{(e)}(\theta) \in \mathbb{R}^n, \end{aligned} \quad (29)$$

where $\bigcup(\cdot)$ represents the assembly operator for all stochastic element matrices and vectors, and the total degree of freedom (DoF) is given by $n = n_e d$.

Further, let us consider that the stochastic matrix $\mathbf{G}(\mathbf{x}, \theta)$ (or the tensor $\mathbf{C}(\mathbf{x}, \theta)$) can be approximated using the following separated form in many cases

$$\mathbf{G}(\mathbf{x}, \theta) = \sum_{i=0}^m \xi_i(\theta) \mathbf{G}_i, \quad (30)$$

where $\xi_0(\theta) \equiv 1$, $\{\xi_i(\theta)\}_{i=1}^m$ are scalar random variables, $\{\mathbf{G}_i \in \mathbb{R}^{\frac{d(d+1)}{2} \times \frac{d(d+1)}{2}}\}_{i=0}^m$ are deterministic matrices. For the non-separated stochastic matrix $\mathbf{G}(\mathbf{x}, \theta)$, the approaches for simulating random fields can be adopted to achieve (30)-like approximations for both Gaussian and non-Gaussian random inputs, e.g. Karhunen–Loève expansion and Polynomial Chaos expansion [25, 39, 40]. In this way, we can reformulate the SVEE (28) as

$$\left(\sum_{i=0}^m \xi_i(\theta) \mathbf{K}_i \right) \mathbf{u}(\theta) = \mathbf{F}(\theta), \quad (31)$$

where the deterministic matrices $\{\mathbf{K}_i\}_{i=0}^m$ are assembled via

$$\mathbf{K}_i = \bigcup_{e=1}^{n_e} \left(\mathbf{k}_C^{(e)}(\mathbf{G}_i) + \mathbf{k}_S^{(e)}(\mathbf{G}_i) \right) \in \mathbb{R}^{n \times n} \quad (32)$$

since the stochastic element matrices $\mathbf{k}_C^{(e)}(\theta)$ in (19) and $\mathbf{k}_S^{(e)}(\theta)$ in (25) depend linearly on the matrix components $\{\mathbf{G}_i\}_{i=0}^m$.

3 PC-SVEM: Polynomial Chaos based spectral stochastic virtual element method

The PC-based methods have been well developed and widely applied to solve a variety of stochastic problems. In this section, we present a PC-SVEM to solve the SVEE (28) (or the separated form (31)). In this method, the stochastic solution $\mathbf{u}(\theta)$ is expanded using (generalized) PC basis as follows

$$\mathbf{u}_{\text{PC},k}(\theta) = \sum_{i=1}^k \Gamma_i(\theta) \mathbf{d}_{\text{PC},i}, \quad (33)$$

where $\{\mathbf{d}_{\text{PC},i}\}_{i=1}^k$ are the corresponding deterministic vectors to be solved, $\{\Gamma_i(\theta)\}_{i=1}^k$ are the PC basis. In practice, we can choose different PC basis for different types of random inputs, such as the Hermite PC basis for Gaussian random variables and the Legendre PC basis for uniform random variables [15, 36]. The stochastic Galerkin approach is then used to transform SVEE (28) into the following deterministic equation [15]

$$\int_{\Theta} \left[\mathbf{K}(\theta) \sum_{i=1}^k \Gamma_i(\theta) \mathbf{d}_{\text{PC},i} - \mathbf{F}(\theta) \right] \Gamma_j(\theta) d\mathcal{P}(\theta) = 0 \quad (34)$$

for $j = 1, \dots, k$, where $\mathcal{P}(\theta)$ is the probability measurement of the random input θ . Further, the above equation can be rewritten as a compact form

$$\mathbf{K}_{\text{PC}} \mathbf{d}_{\text{PC}} = \mathbf{F}_{\text{PC}}, \quad (35)$$

where the augmented deterministic matrix $\mathbf{K}_{\text{PC}} \in \mathbb{R}^{nk \times nk}$ and the augmented deterministic vectors $\mathbf{d}_{\text{PC}}, \mathbf{F}_{\text{PC}} \in \mathbb{R}^{nk}$ are assembled by

$$\begin{aligned} \mathbf{K}_{\text{PC}} &= \begin{bmatrix} \mathbf{K}_{\text{PC},11} & \cdots & \mathbf{K}_{\text{PC},1k} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{\text{PC},k1} & \cdots & \mathbf{K}_{\text{PC},kk} \end{bmatrix}, \\ \mathbf{d}_{\text{PC}} &= \begin{bmatrix} \mathbf{d}_{\text{PC},1} \\ \vdots \\ \mathbf{d}_{\text{PC},k} \end{bmatrix}, \quad \mathbf{F}_{\text{PC}} = \begin{bmatrix} \mathbf{F}_{\text{PC},1} \\ \vdots \\ \mathbf{F}_{\text{PC},k} \end{bmatrix}, \end{aligned} \quad (36)$$

where the matrix and vector components $\mathbf{K}_{\text{PC},ji} \in \mathbb{R}^{n \times n}$ and $\mathbf{F}_{\text{PC},j} \in \mathbb{R}^n$, $i, j = 1, \dots, k$ are

$$\begin{aligned} \mathbf{K}_{\text{PC},ji} &= \int_{\Theta} \mathbf{K}(\theta) \Gamma_i(\theta) \Gamma_j(\theta) d\mathcal{P}(\theta), \\ \mathbf{F}_{\text{PC},j} &= \int_{\Theta} \mathbf{F}(\theta) \Gamma_j(\theta) d\mathcal{P}(\theta). \end{aligned} \quad (37)$$

Further, if the separated form (31) is considered, the above calculation of the matrices $\mathbf{K}_{PC,ji}$ is simplified as

$$\mathbf{K}_{PC,ji} = \sum_{l=0}^m \left[\int_{\Theta} \xi_l(\theta) \Gamma_i(\theta) \Gamma_j(\theta) d\mathcal{P}(\theta) \right] \mathbf{K}_l, \quad (38)$$

which only involves the numerical integration $\int_{\Theta} \xi_l(\theta) \Gamma_i(\theta) \Gamma_j(\theta) d\mathcal{P}(\theta)$. For low-dimensional stochastic problems, the calculation is cheap enough benefiting from efficient numerical integration strategies in stochastic spaces [35].

It is noted that the total number of PC basis is $k = \frac{(m+r)!}{m!r!}$, where $(\cdot)!$ is the factorial operator, r is the expansion order of the PC basis. Thus, similar to classical PC-based methods, the proposed PC-SVEM still suffers from the curse of dimensionality since the matrix/vector size nk in (35) increases sharply as the spatial DoF n of the physical model, the stochastic dimension m of the random input and the expansion order r of the PC basis increase. For instance, the size is about $nk = 1 \times 10^6$ when $n = 1 \times 10^3$, $m = 10$ and $r = 4$, which requires extremely expensive computational effort. Although several methods are developed to alleviate the computational burden, e.g. dedicated iterative algorithms and sparse PC approximations [8, 18, 22]. It is still challenging to solve very high-dimensional stochastic problems using the PC-SVEM.

4 WIN-SVEM: weakly intrusive stochastic virtual element method

4.1 A weakly intrusive stochastic virtual element method

To avoid the curse of dimensionality arising in the above PC-SVEM, we present a WIN-SVEM in this section, which can be considered as an extension of our previous work [41, 42] on stochastic finite element methods to SVEM. To this end, we consider the stochastic solution $\mathbf{u}(\theta)$ approximated by the following series expansion

$$\mathbf{u}_{WIN,k}(\theta) = \sum_{i=1}^k \lambda_i(\theta) \mathbf{d}_{WIN,i}, \quad (39)$$

where $\{\lambda_i(\theta) \in \mathbb{R}\}_{i=1}^k$ are scalar random variables, $\{\mathbf{d}_{WIN,i} \in \mathbb{R}^n\}_{i=1}^k$ are deterministic vectors, and k is the number of retained terms. It is noted that the number k and all pairs $\{\lambda_i(\theta), \mathbf{d}_{WIN,i}\}_{i=1}^k$ are not known a priori. An iterative algorithm is presented to solve the pairs $\{\lambda_i(\theta), \mathbf{d}_{WIN,i}\}$ one by one. Specifically, we assume that the $(k-1)$ -th approximation $\mathbf{u}_{WIN,k-1}(\theta) = \sum_{i=1}^{k-1} \lambda_i(\theta) \mathbf{d}_{WIN,i}$ has been known and the goal is to solve the k -th pair $\{\lambda_k(\theta), \mathbf{d}_{WIN,k}\}$. The original SVEE (28) can be rewritten as

$$\mathbf{K}(\theta) \lambda_k(\theta) \mathbf{d}_{WIN,k} = \mathbf{F}_k(\theta), \quad (40)$$

where the stochastic vector $\mathbf{F}_k(\theta) = \mathbf{F}(\theta) - \mathbf{K}(\theta) \sum_{i=1}^{k-1} \lambda_i(\theta) \mathbf{d}_{WIN,i}$. In this way, (40) only involves one unknown pair $\{\lambda_k(\theta), \mathbf{d}_{WIN,k}\}$. However, different from that the random basis (i.e. PC basis) has been known in PC-SVEM, both the random variable $\lambda_k(\theta)$ and the deterministic vector $\mathbf{d}_{WIN,k}$ are unknown in this case. To avoid this issue, an alternating iteration is adopted to solve them. Specifically, if the random variable $\lambda_k(\theta)$ has been known (or given an initial value), (40) is transformed into the following deterministic virtual element equation by taking advantage of the stochastic Galerkin procedure [15] similar to that in (34)

$$\int_{\Theta} [\mathbf{K}(\theta) \lambda_k(\theta) \mathbf{d}_{WIN,k} - \mathbf{F}_k(\theta)] \lambda_k(\theta) d\mathcal{P}(\theta) = 0, \quad (41)$$

which is equivalent to

$$\mathbf{K}_{WIN,k} \mathbf{d}_{WIN,k} = \mathbf{F}_{WIN,k}, \quad (42)$$

where the deterministic matrix $\mathbf{K}_{WIN,k} = \int_{\Theta} \mathbf{K}(\theta) \lambda_k^2(\theta) d\mathcal{P}(\theta) \in \mathbb{R}^{n \times n}$ and the deterministic vector $\mathbf{F}_{WIN,k} = \int_{\Theta} \mathbf{F}_k(\theta) \lambda_k(\theta) d\mathcal{P}(\theta) \in \mathbb{R}^n$. Existing numerical solvers can be adopted to solve it efficiently and accurately [38]. Note that the size of (42) is the same as the original SVEE (28), which is different from the augmented size of PC-based derived equation (35) and can thus save a lot of computational effort. In practical implementations, we let the vector $\mathbf{d}_{WIN,k}$ orthogonal to the obtained vectors $\{\mathbf{d}_{WIN,i}\}_{i=1}^{k-1}$ to speed up the convergence, which is achieved by using the following Gram-Schmidt orthogonalization

$$\mathbf{d}_{WIN,k} = \mathbf{d}_{WIN,k} - \sum_{i=1}^{k-1} \left(\mathbf{d}_{WIN,k}^T \mathbf{d}_{WIN,i} \right) \mathbf{d}_{WIN,i} \quad (43)$$

and the normalization $\mathbf{d}_{WIN,k}^T \mathbf{d}_{WIN,k} = 1$, where $\{\mathbf{d}_{WIN,i}\}_{i=1}^{k-1}$ are normalized orthogonal vectors that meet $\mathbf{d}_{WIN,i}^T \mathbf{d}_{WIN,j} = \delta_{ij}$, where δ_{ij} is the Kronecker delta.

With the deterministic vector $\mathbf{d}_{WIN,k}$ solved using (42), the random variable $\lambda_k(\theta)$ is then recalculated taking advantage of the following classical Galerkin procedure

$$\mathbf{d}_{WIN,k}^T [\mathbf{K}(\theta) \lambda_k(\theta) \mathbf{d}_{WIN,k} - \mathbf{F}_k(\theta)] = 0. \quad (44)$$

Since the stochastic matrix $\mathbf{K}(\theta)$ is positive definite and $\mathbf{z}^T \mathbf{K}(\theta) \mathbf{z} > 0, \forall \mathbf{z} \neq \mathbf{0} \in \mathbb{R}^n, \forall \theta \in \Theta$ holds, the above equation can be rewritten as

$$\lambda_k(\theta) = \frac{\mathbf{d}_{WIN,k}^T \mathbf{F}_k(\theta)}{\mathbf{d}_{WIN,k}^T \mathbf{K}(\theta) \mathbf{d}_{WIN,k}}. \quad (45)$$

To avoid the curse of dimensionality arising in the high-dimensional problems, we adopt a non-intrusive sampling approach [42] to solve (45) instead of the PC-based approximation, which is to calculate the sample vector $\lambda_k(\hat{\theta}) \in \mathbb{R}^{n_s}$ by

$$\lambda_k(\hat{\theta}) = \left[\mathbf{d}_{\text{WIN},k}^T \mathbf{F}_k(\hat{\theta}) \right] \oslash \left[\mathbf{d}_{\text{WIN},k}^T \mathbf{K}(\hat{\theta}) \mathbf{d}_{\text{WIN},k} \right], \quad (46)$$

where $\square(\hat{\theta})$ represents n_s random sample realizations of $\square(\theta)$, $\mathbf{d}_{\text{WIN},k}^T \mathbf{F}_k(\hat{\theta}) \in \mathbb{R}^{n_s}$ and $\mathbf{d}_{\text{WIN},k}^T \mathbf{K}(\hat{\theta}) \mathbf{d}_{\text{WIN},k} \in \mathbb{R}^{n_s}$ are the sample vectors of the random variables $\mathbf{d}_{\text{WIN},k}^T \mathbf{F}_k(\theta)$ and $\mathbf{d}_{\text{WIN},k}^T \mathbf{K}(\theta) \mathbf{d}_{\text{WIN},k}$, respectively, and \oslash represents the element-wise division of two sample vectors, also known as Hadamard division operator. In this way, all random inputs are embedded into the sample realization vectors $\mathbf{d}_{\text{WIN},k}^T \mathbf{F}_k(\hat{\theta})$ and $\mathbf{d}_{\text{WIN},k}^T \mathbf{K}(\hat{\theta}) \mathbf{d}_{\text{WIN},k}$, which is insensitive to the stochastic dimension of random inputs. The curse of dimensionality can thus be avoided successfully, which will be discussed in detail in the next section.

We can solve the k -th pair $\{\lambda_k(\theta), \mathbf{d}_{\text{WIN},k}\}$ by performing the iterative process of (42) and (46) until reaching a specified precision. A similar iteration is also adopted to calculate other pairs $\{\lambda_{k+1}(\theta), \mathbf{d}_{\text{WIN},k+1}\}, \dots$ until a good approximation of the stochastic solution $\mathbf{u}_{\text{WIN},k}(\theta)$ in (39) is approximated in a sequential way and it does not exactly fulfill the original SVEE (28). The approximation has low accuracy for some cases [41]. We introduce a recalculation process to avoid this problem. To this end, $\mathbf{D}_{\text{WIN}} = [\mathbf{d}_{\text{WIN},1}, \dots, \mathbf{d}_{\text{WIN},k}] \in \mathbb{R}^{n \times k}$ is considered as a set of reduced basis functions and the random variable vector $\mathbf{\Lambda}(\theta) = [\lambda_1(\theta), \dots, \lambda_k(\theta)]^T \in \mathbb{R}^k$ is recalculated via the following reduced-order stochastic equation

$$\left[\mathbf{D}_{\text{WIN}}^T \mathbf{K}(\theta) \mathbf{D}_{\text{WIN}} \right] \mathbf{\Lambda}(\theta) = \mathbf{D}_{\text{WIN}}^T \mathbf{F}(\theta), \quad (47)$$

which requires to be solved repeatedly for n_s sample realizations to get the final solution $\mathbf{\Lambda}(\theta^{(i)})$, $i = 1, \dots, n_s$, but only very low computational effort is involved since the sizes of the reduced-order stochastic matrix $\mathbf{D}_{\text{WIN}}^T \mathbf{K}(\theta) \mathbf{D}_{\text{WIN}} \in \mathbb{R}^{k \times k}$ and the reduced-order stochastic vector $\mathbf{D}_{\text{WIN}}^T \mathbf{F}(\theta) \in \mathbb{R}^k$ are greatly reduced compared to the original SVEE (28) in most cases.

Let us highlight the weak intrusiveness of the proposed method. On one hand, (39) is considered as a kind of intrusive approximation of the stochastic solution, which is very similar to the PC-based intrusive approximation (33). However, on the other hand, the implementation for solving $\mathbf{d}_{\text{WIN},k}$ in (42) only involves deterministic calculations and the matrix $\mathbf{K}_{\text{WIN},k}$ keeps the same size and matrix properties as the original stochastic matrix $\mathbf{K}(\theta)$, which is weakly intrusive. Also, (46) for calculating the random variable $\lambda_k(\theta)$ is fully

non-intrusive. In these senses, we implement the intrusive stochastic solution approximation only in weakly intrusive and fully non-intrusive ways. The method combines the high efficiency of intrusive methods and the weak dimensionality dependence of non-intrusive methods. It can solve high-dimensional stochastic problems efficiently and accurately.

4.2 High-dimensional stochastic problems

In this section, we will show that the proposed WIN-SVEM can be applied to high-dimensional stochastic problems without any modification. We explain this point from the perspective of the influence of high stochastic dimensions on solving (42) and (46). We only consider (31) in this section and a large number m is truncated in (31) to generate a high-dimensional stochastic problem. In this way, the deterministic matrix $\mathbf{K}_{\text{WIN},k}$ and the deterministic vector $\mathbf{F}_{\text{WIN},k}$ in (42) are calculated via

$$\mathbf{K}_{\text{WIN},k} = \sum_{i=0}^m \left[\int_{\Theta} \xi_i(\theta) \lambda_k^2(\theta) d\mathcal{P}(\theta) \right] \mathbf{K}_i, \quad (48)$$

$$\mathbf{F}_{\text{WIN},k} = \int_{\Theta} \mathbf{F}(\theta) \lambda_k(\theta) d\mathcal{P}(\theta) - \sum_{i=0}^m \sum_{j=1}^{k-1} \left[\int_{\Theta} \xi_i(\theta) \lambda_j(\theta) \lambda_k(\theta) d\mathcal{P}(\theta) \right] \mathbf{K}_i \mathbf{d}_{\text{WIN},j}, \quad (49)$$

where the probability integrals are approximated using the following non-intrusive sampling approach

$$\int_{\Theta} \xi_i(\theta) \lambda_j(\theta) \lambda_k(\theta) d\mathcal{P}(\theta) = \widehat{\mathbb{E}} \{ \xi_i(\hat{\theta}) \odot \lambda_j(\hat{\theta}) \odot \lambda_k(\hat{\theta}) \}, \quad (50)$$

for $i = 0, \dots, m$, $j = 1, \dots, k$, where $\{\xi_i(\hat{\theta}) \in \mathbb{R}^{n_s}\}_{i=0}^m$ are sample vectors of the random variables $\{\xi_i(\theta)\}_{i=0}^m$, the operator \odot represents the element-wise multiplication of sample vectors, $\widehat{\mathbb{E}}\{\cdot\}$ is the expectation operator of the sample vector. (50) takes a total of $k(m+1)$ expectation operations, which is not sensitive to the stochastic dimension and has low computational effort even for very high stochastic dimensions. Note that although we only illustrate the high-dimensional input in the stochastic matrix $\mathbf{K}(\theta)$, the above calculation also works efficiently for $\int_{\Theta} \mathbf{F}(\theta) \lambda_k(\theta) d\mathcal{P}(\theta)$ if the stochastic vector $\mathbf{F}(\theta)$ involves high-dimensional random inputs.

Further, the sample vectors in right side of (46) are calculated via

$$\mathbf{d}_{\text{WIN},k}^T \mathbf{K}(\hat{\theta}) \mathbf{d}_{\text{WIN},k} = \sum_{i=0}^m \xi_i(\hat{\theta}) \left(\mathbf{d}_{\text{WIN},k}^T \mathbf{K}_i \mathbf{d}_{\text{WIN},k} \right) \in \mathbb{R}^{n_s}, \quad (51)$$

$$\mathbf{d}_{\text{WIN},k}^T \mathbf{F}_k(\hat{\boldsymbol{\theta}}) = \mathbf{d}_{\text{WIN},k}^T \mathbf{F}(\theta) - \sum_{i=0}^m \sum_{j=1}^{k-1} [\xi_i(\hat{\boldsymbol{\theta}}) \odot \lambda_j(\hat{\boldsymbol{\theta}})] (\mathbf{d}_{\text{WIN},k}^T \mathbf{K}_i \mathbf{d}_{\text{WIN},j}) \in \mathbb{R}^{n_s}, \tag{52}$$

which requires a total of $k(m+1)$ operations for $\mathbf{d}_{\text{WIN},k}^T \mathbf{K}_i \mathbf{d}_{\text{WIN},j}$, $i = 0, \dots, m$, $j = 1, \dots, k$ and is also computationally cheap for high-dimensional stochastic problems. Therefore, both (42) and (46) are insensitive to the stochastic dimension. The proposed method can avoid the curse of dimensionality successfully.

4.3 Algorithm implementation

The above proposed WIN-SVEM for solving SVEEs is summarized in Algorithm 1, which includes two loop processes. The inner loop is from step 4 to step 9 and used to solve the k -th pair $\{\lambda_k(\theta), \mathbf{d}_{\text{WIN},k}\}$. The outer loop from step 2 to step 13 is to approximate the stochastic solution using a set of pairs $\{\lambda_i(\theta), \mathbf{d}_{\text{WIN},i}\}_{i=1}^k$. To execute the inner loop, a random sample vector $\lambda_k^{(0)}(\hat{\boldsymbol{\theta}}) \in \mathbb{R}^{n_s}$ is initialized in step 3. In the numerical implementation, any nonzero vectors of size n_s can be chosen as the initialization since the initial samples have little influence on the computational accuracy and efficiency of the proposed method. Following each inner loop, we only need to update the stochastic force vector $\mathbf{F}_{k+1}(\theta)$ in step 10 and store the reduced-order matrix \mathbf{D}_{WIN} in step 11 in the outer loop.

Algorithm 1 WIN-SVEM for solving SVEEs

- 1: $k \leftarrow 1$
 - 2: **while** $\epsilon_{\mathbf{u},k} > \epsilon_{\mathbf{u}}$ **do**
 - 3: Initialize random samples $\lambda_k^{(0)}(\hat{\boldsymbol{\theta}}) = \{\lambda_k^{(0)}(\theta^{(i)})\}_{i=1}^{n_s} \in \mathbb{R}^{n_s}$
 - 4: **while** $\epsilon_{\mathbf{d},j} > \epsilon_{\mathbf{d}}$ **do**
 - 5: Calculate the deterministic vector $\mathbf{d}_{\text{WIN},k}^{(j)}$ by solving (42)
 - 6: Orthonormalize $\mathbf{d}_{\text{WIN},k}^{(j)}$ using (43)
 - 7: Update the random sample vector $\lambda_k^{(j)}(\hat{\boldsymbol{\theta}}) \in \mathbb{R}^{n_s}$ via (46)
 - 8: Compute the locally iterative error $\epsilon_{\mathbf{d},j}$, $j \leftarrow j + 1$
 - 9: **end while**
 - 10: Update the stochastic force vector $\mathbf{F}_{k+1}(\theta) = \mathbf{F}_k(\theta) - \mathbf{K}(\theta) \lambda_k(\theta) \mathbf{d}_{\text{WIN},k}$
 - 11: Update the deterministic matrix $\mathbf{D}_{\text{WIN}} = [\mathbf{D}_{\text{WIN}}, \mathbf{d}_{\text{WIN},k}] \in \mathbb{R}^{n \times k}$
 - 12: Compute the locally iterative error $\epsilon_{\mathbf{u},k}$, $k \leftarrow k + 1$
 - 13: **end while**
 - 14: Recalculate the random variable vector $\boldsymbol{\Lambda}(\theta) \in \mathbb{R}^k$ via (47)
-

Two iterative criteria are involved in the above algorithm to check the convergence, i.e. $\epsilon_{\mathbf{d},j}$ in step 8 for the inner loop and $\epsilon_{\mathbf{u},k}$ in step 12 for the outer loop. The iterative error $\epsilon_{\mathbf{d},j}$ is defined as

$$\epsilon_{\mathbf{d},j} = \frac{(\mathbf{d}_{\text{WIN},k}^{(j)} - \mathbf{d}_{\text{WIN},k}^{(j-1)})^T (\mathbf{d}_{\text{WIN},k}^{(j)} - \mathbf{d}_{\text{WIN},k}^{(j-1)})}{\mathbf{d}_{\text{WIN},k}^{(j)T} \mathbf{d}_{\text{WIN},k}^{(j)}} = 2 - 2\mathbf{d}_{\text{WIN},k}^{(j)T} \mathbf{d}_{\text{WIN},k}^{(j-1)}, \tag{53}$$

which measures the difference between the vectors $\mathbf{d}_{\text{WIN},k}^{(j)}$ and $\mathbf{d}_{\text{WIN},k}^{(j-1)}$ and the calculation is stopped when $\epsilon_{\mathbf{d},j} < \epsilon_{\mathbf{d}}$ is met. Similarly, the iterative error $\epsilon_{\mathbf{u},k}$ is defined as

$$\begin{aligned} \epsilon_{\mathbf{u},k} &= \frac{1}{\int_{\Theta} \mathbf{u}_{\text{WIN},k}^T(\theta) \mathbf{u}_{\text{WIN},k}(\theta) d\mathcal{P}(\theta)} \cdot \left[\int_{\Theta} [\mathbf{u}_{\text{WIN},k}(\theta) - \mathbf{u}_{\text{WIN},k-1}(\theta)]^T \cdot [\mathbf{u}_{\text{WIN},k}(\theta) - \mathbf{u}_{\text{WIN},k-1}(\theta)] d\mathcal{P}(\theta) \right] \\ &= \frac{\int_{\Theta} \lambda_k^2(\theta) d\mathcal{P}(\theta) \mathbf{d}_{\text{WIN},k}^T \mathbf{d}_{\text{WIN},k}}{\sum_{i,j=1}^k \int_{\Theta} \lambda_i(\theta) \lambda_j(\theta) d\mathcal{P}(\theta) \mathbf{d}_{\text{WIN},i}^T \mathbf{d}_{\text{WIN},j}} \\ &= \frac{\int_{\Theta} \lambda_k^2(\theta) d\mathcal{P}(\theta)}{\sum_{i=1}^k \int_{\Theta} \lambda_i^2(\theta) d\mathcal{P}(\theta)}, \end{aligned} \tag{54}$$

which measures the contribution of the k -th pair $\{\lambda_k(\theta), \mathbf{d}_{\text{WIN},k}\}$ to the stochastic solution $\mathbf{u}_{\text{WIN},k}(\theta)$. However, (54) may be not a good error indicator in many cases [41] since the random variables $\{\lambda_i(\theta)\}_{i=1}^k$ are calculated in a sequential way and $\int_{\Theta} \lambda_k^2(\theta) d\mathcal{P}(\theta)$ may not keep decreasing. We avoid this problem by replacing $\{\lambda_i(\theta)\}_{i=1}^k$ in (54) with equivalent random variables $\{\tilde{\lambda}_i(\theta)\}_{i=1}^k$. To calculate equivalent random variable, we first calculate the autocorrelation function of the random variable vector $\boldsymbol{\Lambda}(\theta) = [\lambda_1(\theta), \dots, \lambda_k(\theta)]^T$ by

$$\mathbf{C}_{\boldsymbol{\Lambda}\boldsymbol{\Lambda}} = \widehat{\mathbb{E}} \left\{ \boldsymbol{\Lambda}(\hat{\boldsymbol{\theta}}) \boldsymbol{\Lambda}^T(\hat{\boldsymbol{\theta}}) \right\} \in \mathbb{R}^{k \times k}, \tag{55}$$

which is decomposed by the eigendecomposition into

$$\mathbf{C}_{\boldsymbol{\Lambda}\boldsymbol{\Lambda}} = \mathbf{Q}\mathbf{Z}\mathbf{Q}^T, \tag{56}$$

where $\mathbf{Q} \in \mathbb{R}^{k \times k}$ is an orthonormal matrix and \mathbf{Z} is a diagonal matrix consisting of descending eigenvalues of the matrix $\mathbf{C}_{\boldsymbol{\Lambda}\boldsymbol{\Lambda}}$. We construct an equivalent random variable vector $\tilde{\boldsymbol{\Lambda}}(\theta) = \mathbf{Q}^T \boldsymbol{\Lambda}(\theta) = [\tilde{\lambda}_1(\theta), \dots, \tilde{\lambda}_k(\theta)]^T \in \mathbb{R}^k$ whose autocorrelation function happens to be $\mathbf{C}_{\tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{\Lambda}}} = \mathbf{Z}$. Substituting the equivalent random variables $\{\tilde{\lambda}_i(\theta)\}_{i=1}^k$ into (54) we recalculate the iterative error $\epsilon_{\mathbf{u},k}$ as

$$\epsilon_{\mathbf{u},k} = \frac{\int_{\Theta} \tilde{\lambda}_k^2(\theta) d\mathcal{P}(\theta)}{\sum_{i=1}^k \int_{\Theta} \tilde{\lambda}_i^2(\theta) d\mathcal{P}(\theta)} = \frac{\mathbf{Z}_k}{\text{Tr}(\mathbf{Z})}, \tag{57}$$

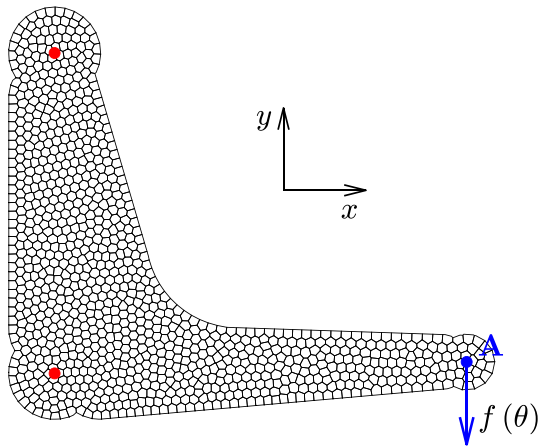


Fig. 1 Geometry and Voronoi mesh of the 2D model

where \mathbf{Z}_k is the k -th diagonal element of the matrix \mathbf{Z} . It is noted that the above reformulation does not improve the approximation accuracy of the stochastic solution and only provides an equivalent representation. In this way, the iterative error $\epsilon_{\mathbf{u},k}$ keeps decreasing as the retained item k increases. More details of the basic implementation and the comparison between (54) and (57) can be found in [41].

5 Numerical examples

In this section, we test the two proposed methods with the aid of 2D and 3D numerical examples. For Algorithm 1, the convergence errors $\epsilon_{\mathbf{d}}$ in step 2 for the inner loop and $\epsilon_{\mathbf{u}}$ in step 4 for the outer loop are set as 1×10^{-3} and 1×10^{-6} , respectively. For both examples, $n_s = 1 \times 10^4$ random samples are used for performing MC simulations and generating reference solutions. In our cases, 1×10^4 samples are enough to achieve convergent probabilistic solutions. The same 1×10^4 random samples are also used in the proposed WIN-SFEM to eliminate the influence caused by sampling processes. Further, the examples are performed on one core of a desktop computer (sixteen cores, Intel Core i7, 2.50GHz).

5.1 Example 1: SVEM analysis of a 2D stochastic problem

5.1.1 Simulation of random inputs

In this example, we consider the SVEM-based plane stress analysis of a 2D model shown in Fig. 1 [29]. The Voronoi mesh is adopted for the spatial discretization, including a total of 2018 vertices, 1000 Voronoi elements and 4036 DoFs. The model is fixed at the two red points as shown in Fig. 1. A stochastic force $f(\theta) = -1000 - 100\xi_f(\theta)$ (unit: N) is applied to the blue point along the y direction, where $\xi_f(\theta)$

is a standard Gaussian random variable. Further, the material property matrix $\mathbf{G}(x, y, \theta) \in \mathbb{R}^{3 \times 3}$ is given by

$$\mathbf{G}(x, y, \theta) = \frac{E(x, y, \theta)}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1}{2}(1 - \nu) \end{bmatrix}, \quad (58)$$

where the Poisson ratio $\nu = 0.3$ and the stochastic Young's modulus $E(x, y, \theta)$ is modeled as a two-dimensional random field with the mean value $E_0(x, y) = 100$ MPa and the covariance function [27]

$$\text{Cov}_{EE}(x_1, y_1; x_2, y_2) = \sigma_E^2 \left(1 + \frac{|x_1 - x_2|}{l_x} \right) \left(1 + \frac{|y_1 - y_2|}{l_y} \right) \exp \left(-\frac{|x_1 - x_2|}{l_x} - \frac{|y_1 - y_2|}{l_y} \right), \quad (59)$$

where the standard deviation $\sigma_E = 10$ MPa, and l_x and l_y are the correlation lengths in the x and y directions, respectively. The random field $E(x, y, \theta)$ is approximated by the following Karhunen–Loève expansion [15, 39]

$$E(x, y, \theta) = E_0(x, y) + \sum_{i=1}^m \xi_i(\theta) \sqrt{\kappa_i} E_i(x, y), \quad (60)$$

where $\{\xi_i(\theta)\}_{i=1}^m$ are mutually independent standard Gaussian random variables and they are also independent of the random variable $\xi_f(\theta)$. $\{\kappa_i, E_i(x, y)\}_{i=1}^m$ are eigenvalues and eigenvectors of the covariance function $\text{Cov}_{EE}(x_1, y_1; x_2, y_2)$. They are solved by the following Fredholm integral equation of the second kind

$$\int_{\Omega} \text{Cov}_{EE}(x_1, y_1; x_2, y_2) E_i(x_1, y_1) dx_1 dy_1 = \kappa_i E_i(x_2, y_2), \quad (61)$$

which can be solved efficiently by taking advantage of existing eigenvalue solvers [24]. To ensure $E(x, y, \theta) > 0$, the samples $\theta^{(i)}$ such that $\min_{x, y \in \Omega} E(x, y, \theta^{(i)}) \leq 1 \times 10^{-3}$ are dropped out in numerical implementations. In this way, the deterministic matrices $\{\mathbf{G}_i(x, y) \in \mathbb{R}^{3 \times 3}\}_{i=0}^m$ in (30) are given by

$$\mathbf{G}_i(x, y) = \frac{\sqrt{\kappa_i} E_i(x, y, \theta)}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1}{2}(1 - \nu) \end{bmatrix}, \quad (62)$$

where $\kappa_0 = 1$.

In this example, we consider a low-dimensional case and a high-dimensional case via introducing different truncated items m in (60). For the low-dimensional case, we let the correlation lengths be $l_x = \max(x) - \min(x)$ and $l_y = \max(y) - \min(y)$. To achieve the truncated error $\kappa_m / \sum_{i=1}^m \kappa_i < 1 \times$

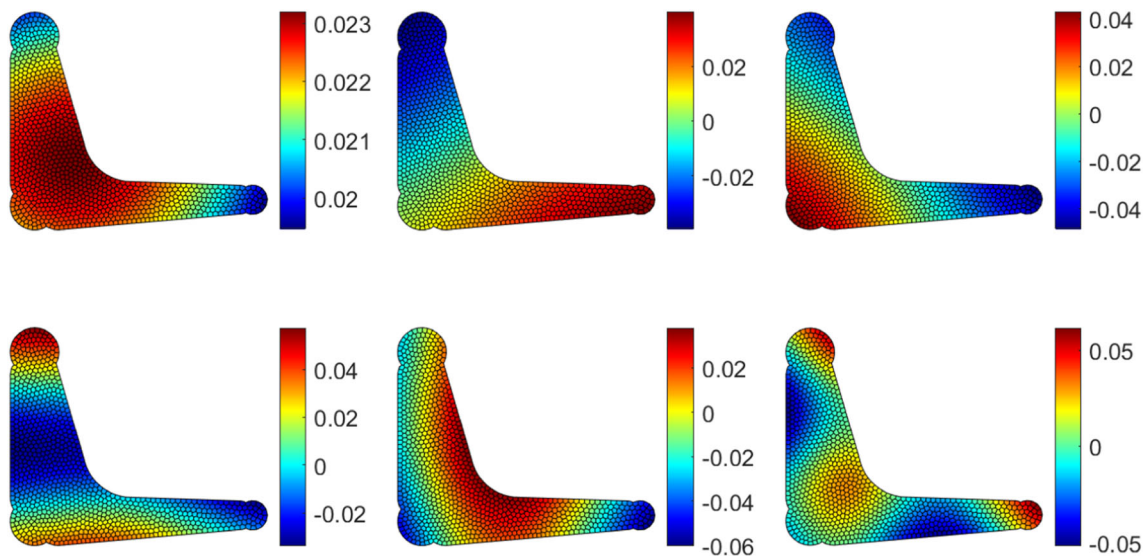


Fig. 2 Eigenvectors $\{E_i(x, y)\}_{i=1}^6$ of the covariance function $\text{Cov}_{EE}(x_1, y_1; x_2, y_2)$ of the low-dimensional case

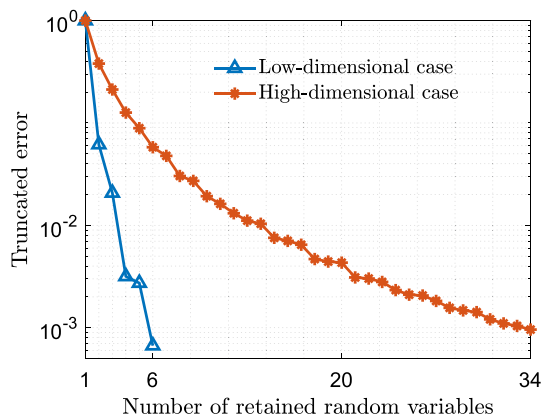


Fig. 3 Truncated errors of different numbers of retained random variables

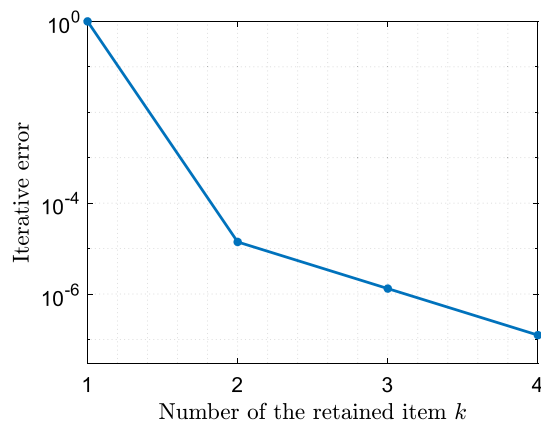


Fig. 4 Iterative errors of different retained terms in the low-dimensional case

10^{-3} , the number of truncated items is $m = 6$. Corresponding eigenvectors $\{E_i(x, y)\}_{i=1}^6$ of the covariance function $\text{Cov}_{EE}(x_1, y_1; x_2, y_2)$ are shown in Fig. 2. It is noted that we can solve (61) using the discretized covariance matrix $\mathbf{Cov}_{EE} \in \mathbb{R}^{2018 \times 2018}$ of $\text{Cov}_{EE}(x_1, y_1; x_2, y_2)$, which is only dependent of the vertices and independent of elements. In this way, the eigenvectors in Fig. 2 are plotted on each vertex of the mesh. For the high-dimensional case, the correlation lengths are set as $l_x = \frac{1}{8} [\max(x) - \min(x)]$ and $l_y = \frac{1}{8} [\max(y) - \min(y)]$, and $m = 34$ truncated items are retained to achieve $\kappa_{34} / \sum_{i=1}^{34} \kappa_i < 1 \times 10^{-3}$. The truncated errors of low- and high-dimensional cases are seen from Fig. 3. Due to smaller correlation lengths in the high-dimensional case, it is slower to converge to the specified truncated error and more truncated items are required to capture the local correlation property.

5.1.2 Low-dimensional case

In this section, we solve the low-dimensional case using the proposed PC-SVEM and WIN-SVEM. For the PC-SVEM, the second order Hermite PC basis of seven standard Gaussian random variables ($\{\xi_i(\theta)\}_{i=1}^6$ and $\xi_f(\theta)$) is adopted for $\{\Gamma_i(\theta)\}_{i=1}^{36}$. The size of the augmented deterministic equation (35) is 145,296, which is much larger than the 4036 DoFs of the original stochastic problem. We do not perform the numerical implementations for higher order PC basis in this example. On one hand, the second order PC basis is enough to achieve a good approximation of the stochastic solution. On the other hand, the size of the augmented equation (35) is 484,320 if the third order PC basis is adopted, which leads to too high computational burden in terms of storage and solution for a problem of such a small spatial scale.

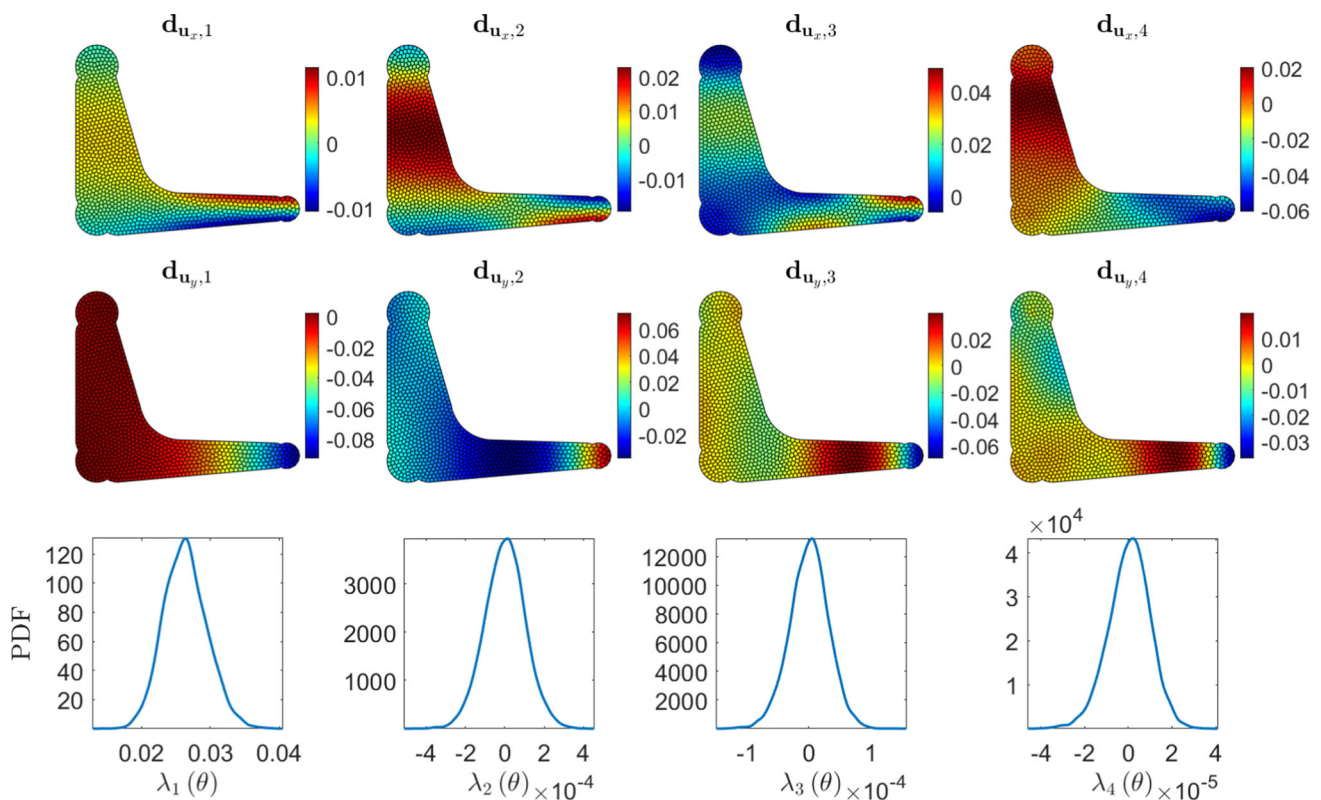


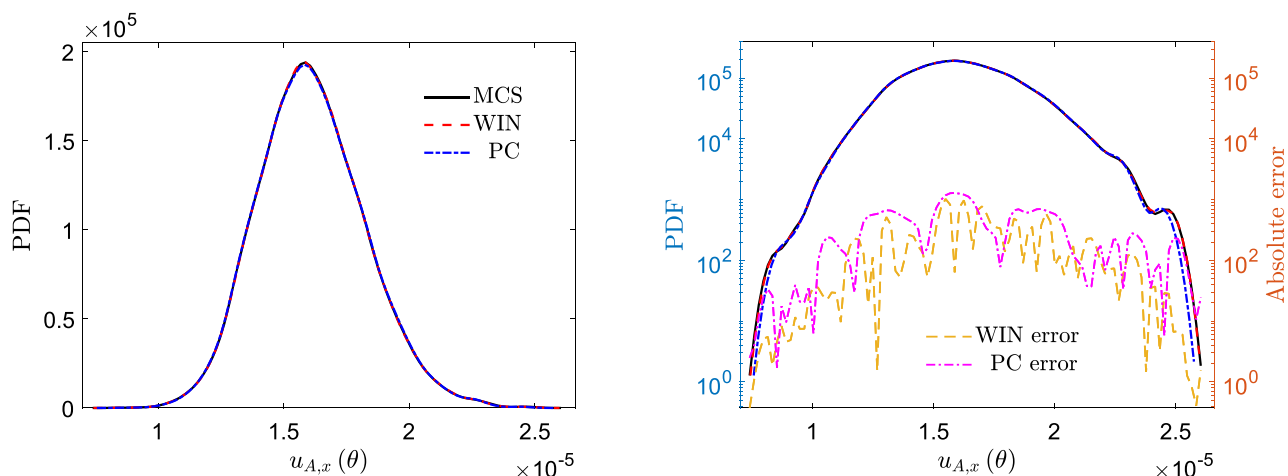
Fig. 5 Components of the stochastic solution: the deterministic vectors $\{\mathbf{d}_{\mathbf{u}_x,i}\}_{i=1}^4$ in the x direction (the first line), the deterministic vectors $\{\mathbf{d}_{\mathbf{u}_y,i}\}_{i=1}^4$ in the y direction (the second line) and PDFs of the random variables $\{\lambda_i(\theta)\}_{i=1}^4$ (the third line), respectively

For the WIN-SVEM, Algorithm 1 is performed and the iterative errors $\epsilon_{\mathbf{u},k}$ of different retained terms are shown in Fig. 4. Only four terms $k = 4$ are retained for the stochastic solution approximation (39), which demonstrates the good convergence of the proposed WIN-SVEM. Further, the components $\{\mathbf{d}_{\mathbf{u}_x,i} \in \mathbb{R}^{2018}\}_{i=1}^4$ and $\{\mathbf{d}_{\mathbf{u}_y,i} \in \mathbb{R}^{2018}\}_{i=1}^4$ of the deterministic vectors $\{\mathbf{d}_{\text{WIN},i}\}_{i=1}^4$ in the x and y directions are depicted in the first and second lines of Fig. 5, respectively, and probability density functions (PDFs) of corresponding random variables $\{\lambda_i(\theta)\}_{i=1}^4$ solved by (47) are seen from the third line of Fig. 5. As the retained term k increases, the ranges of random variables $\{\lambda_i(\theta)\}_{i=1}^4$ become smaller and more concentrated around zero, which indicates that the subsequent retained terms contribute less and less to the stochastic solution.

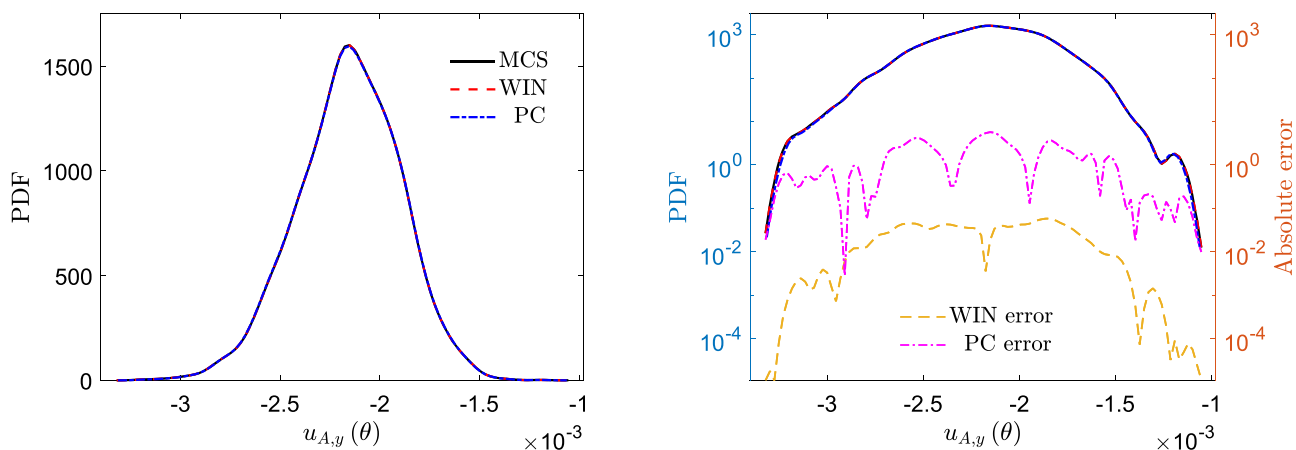
To show the computational accuracy of the proposed PC-SVEM and WIN-SVEM, we compare PDFs of the stochastic displacements $u_{A,x}(\theta)$ and $u_{A,y}(\theta)$ in the x and y directions of the point A (i.e. the blue point where the force $f(\theta)$ acts as shown in Fig. 1) obtained by PC-, WIN- and MCS-based SVEMs and their absolute errors in Fig. 6. It is seen from Fig. 6a, c that the PDFs of both stochastic displacements $u_{A,x}(\theta)$ and $u_{A,y}(\theta)$ obtained by PC- and WIN-SVEMs are in good accordance with those of MCS, which illustrates the good accuracy of the two proposed methods. The

comparison in logarithmic scales shown in Fig. 6b, d demonstrates that WIN-SVEM can achieve smaller absolute errors than PC-SVEM, especially for the stochastic displacement $u_{A,y}(\theta)$ in the y direction. Further, WIN-SVEM can capture tails of the PDFs more accurately than PC-SVEM, which is very useful for many uncertainty quantification problems, such as the simulation of physical phenomena with long tailed probability distributions and the estimation of small failure probability in structural reliability analysis. Therefore, WIN-SVEM is recommended for problems with such requirements.

Further, let us focus on the computational efficiency of the proposed methods. Computational times (unit: second) of the numerical execution of PC-SVEM, WIN-SVEM and MCS are listed in the second to fourth columns of Table 1, where the solving and recalculating times of WIN-SVEM are the computational times of step 2 to step 13 and the recalculation step 14 of Algorithm 1, respectively. The cost of the recalculation process of WIN-SVEM is almost negligible since only deterministic equations with size 4 are solved for different sample realizations. It is found that both PC-SVEM and WIN-SVEM are much cheaper than MCS. However, since a very large deterministic equation needs to be solved in PC-SVEM, it is more computationally intensive than WIN-SVEM.



(a) Comparison of PDFs of the stochastic displacement $u_{A,x}(\theta)$ in the x direction of the point A obtained by PC-, WIN-SVEMs and MCS, respectively. (b) Absolute errors of PDFs of the stochastic displacement $u_{A,x}(\theta)$ in the x direction of the point A obtained by PC- and WIN-SVEMs to the MCS reference PDF.



(c) Comparison of PDFs of the stochastic displacement $u_{A,y}(\theta)$ in the y direction of the point A obtained by PC-, WIN-SVEMs and MCS, respectively. (d) Absolute errors of PDFs of the stochastic displacement $u_{A,y}(\theta)$ in the y direction of the point A obtained by PC- and WIN-SVEMs to the MCS reference PDF.

Fig. 6 PDFs of the stochastic displacements $u_{A,x}(\theta)$ and $u_{A,y}(\theta)$ in the x and y directions of the point A obtained by PC-, WIN-SVEMs and MCS and their absolute errors

Table 1 Computational costs of the stochastic dimensions 7 and 35

Method	WIN	PC	MCS	WIN	MCS
Stochastic dimension	7			35	
Solving time	0.51			12.33	
Recalculating time	0.04			0.35	
Total time (s)	0.55	26.09	159.13	12.68	221.52

5.1.3 High-dimensional case

In this section, we solve the high-dimensional case with a total of 35 stochastic dimensions ($\{\xi_i(\theta)\}_{i=1}^{34}$ and $\xi_f(\theta)$). Only WIN-SVEM is adopted in this section. For PC-SVEM, even if we only use the second order PC basis, the size of the derived deterministic equation (35) is about 2.69×10^6 ,

which suffers from the curse of dimensionality. By using Algorithm 1, the iterative errors $\epsilon_{u,k}$ of different retained terms for the high-dimensional case are shown in Fig. 7. The proposed WIN-SVEM still has good convergence for high-dimensional stochastic problems. $k = 6$ terms are retained in this case, which is slightly increased compared to the low-dimensional case. Computational times for this case are seen from the fifth and sixth columns of Table 1. WIN-SVEM has very low cost even for high-dimensional cases and is much cheaper than MCS. Compared to the low-dimensional case, the computational cost of the high-dimensional case does not increase dramatically as the stochastic dimension increases. In these senses, the proposed WIN-SVEM avoids the curse of dimensionality successfully. Further, PDFs of the stochastic displacements $u_{A,x}(\theta)$ and $u_{A,y}(\theta)$ in the x and y directions of the point A obtained by WIN-SVEM

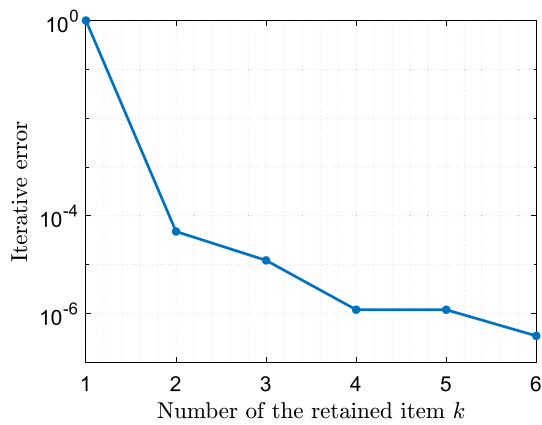


Fig. 7 Iterative errors of different retained terms in the high-dimensional case

and MCS are compared in Fig. 8. For both two stochastic displacements, the computational accuracy of WIN-SVEM is still comparable to MCS.

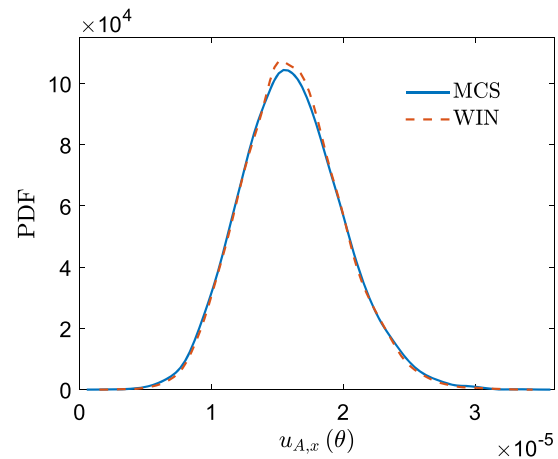
5.2 Example 2: SVEM analysis of a 3D mechanical part

In this case, we consider the SVEM analysis of a 3D mechanical part as shown in Fig. 9a, where Dirichlet boundary conditions $\mathbf{u}_x(\theta) = \mathbf{u}_y(\theta) = \mathbf{u}_z(\theta) = \mathbf{0}$ are imposed on the red surface and an external force $f(x, y, z) = -500 \text{ N/mm}^2$ is applied along the x direction on the blue surface. The model is discretized by use of the Voronoi mesh depicted in Fig. 9b, including a total of 28,232 vertices, 4389 elements and 84,696 DoFs. In this example, the stochastic material matrix $\mathbf{G}(x, y, z, \theta) \in \mathbb{R}^{6 \times 6}$ is given by

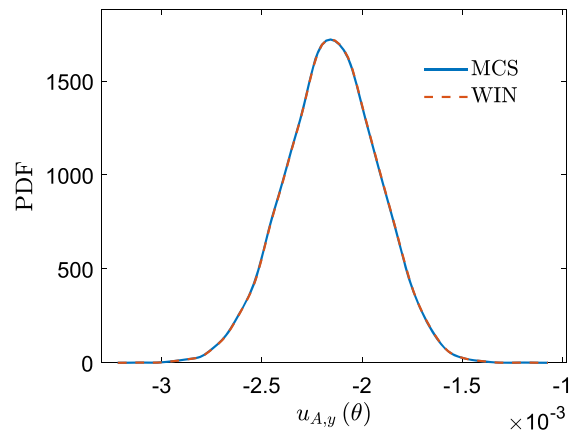
$$\mathbf{G}(x, y, z, \theta) = \frac{E(x, y, z, \theta)}{(1 + \nu)(1 - 2\nu)} \begin{bmatrix} 1 - \nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1 - \nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1 - \nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} - \nu & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} - \nu & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} - \nu \end{bmatrix}, \quad (63)$$

where the Poisson ratio $\nu = 0.3$ and the stochastic Young's modulus $E(x, y, z, \theta)$ is a three-dimensional random field with the covariance function

$$\text{Cov}_{EE}(x_1, y_1, z_1; x_2, y_2, z_2) = \sigma_E^2 \exp\left(-\frac{|x_1 - x_2|}{l_x} - \frac{|y_1 - y_2|}{l_y} - \frac{|z_1 - z_2|}{l_z}\right), \quad (64)$$



(a) Comparison of PDFs of the stochastic displacement $u_{A,x}(\theta)$ in the x direction of the point A obtained by WIN-SVEM and MCS, respectively.



(b) Comparison of PDFs of the stochastic displacement $u_{A,y}(\theta)$ in the y direction of the point A obtained by WIN-SVEM and MCS, respectively.

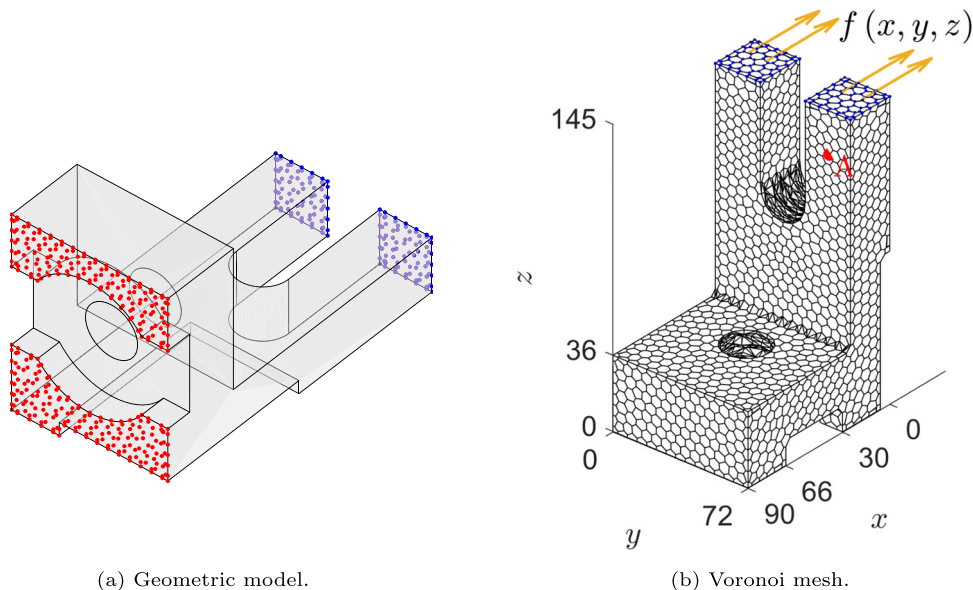
Fig. 8 PDFs of the stochastic displacements $u_{A,x}(\theta)$ and $u_{A,y}(\theta)$ in the x and y directions of the point A obtained by WIN-SVEMs and MCS, respectively

where the standard deviation $\sigma_E = 41.8 \text{ GPa}$, and the correlation lengths are given by $l_x = \max(x) - \min(x)$, $l_y = \max(y) - \min(y)$ and $l_z = \max(z) - \min(z)$. The random field $E(x, y, z, \theta)$ has a (60)-like series expansion

$$E(x, y, z, \theta) = E_0(x, y, z) + \sum_{i=1}^m \xi_i(\theta) \sqrt{\kappa_i} E_i(x, y, z), \quad (65)$$

where the function $E_0(x, y, z) = 208 \text{ GPa}$, and $\{\xi_i(\theta)\}_{i=1}^m$ are mutually independent uniform random variables on $[0, 1]$. It is noted that the mean value of the random field $E(x, y, z, \theta)$ is $E_0(x, y, z) + 0.5 \sum_{i=1}^m \sqrt{\kappa_i} E_i(x, y, z)$ instead of $E_0(x, y, z)$. Similarly, $\{\kappa_i, E_i(x, y)\}_{i=1}^m$ are eigenvalues and eigenvectors of the covariance function

Fig. 9 Geometry of the 3D mechanical part (left) and its Voronoi mesh (right)



$Cov_{EE}(x_1, y_1, z_1; x_2, y_2, z_2)$ and can be solved by the (61)-like integral equation. The truncated number is set as $m = 13$ in this case to achieve the truncated error $\kappa_{13} / \sum_{i=1}^{13} \kappa_i < 1 \times 10^{-3}$.

Similar to the example in the previous section, PC-SVEM suffers from the curse of dimensionality in this example since the size of the derived deterministic equation (35) is about 8.89×10^6 , even with only the second order PC basis. Thus, only WIN-SVEM is used to solve this problem. Iterative errors of different retained terms are seen from Fig. 10. Five terms are retained to meet the specified convergence error, which verifies the good convergence of WIN-SVEM for 3D stochastic problems. Regarding the computational accuracy, PDFs of the stochastic displacements $u_{A,x}(\theta)$, $u_{A,y}(\theta)$ and $u_{A,z}(\theta)$ in the x , y and z directions of the point A (shown in Fig. 9b) obtained by WIN-SVEM and MCS are compared in Fig. 11. For the PDFs of stochastic displacements $u_{A,x}(\theta)$ and $u_{A,z}(\theta)$, WIN-SVEM is in very good agreement with MCS. The PDF of the stochastic displacement $u_{A,y}(\theta)$ is slightly less accurate than those of $u_{A,x}(\theta)$ and $u_{A,z}(\theta)$, but acceptable accuracy is still achieved. We can simply retain more terms in the stochastic solution approximation to improve the accuracy if higher accuracy is required in practice. Further, the computational times of WIN-SVEM and MCS in this example are 213.45 s (including the solving time 212.89 s and the recalculating time 0.56 s) and 2.74×10^4 s, respectively. The proposed WIN-SVEM is still

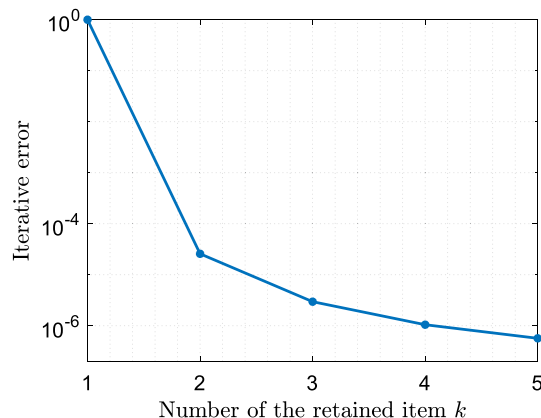
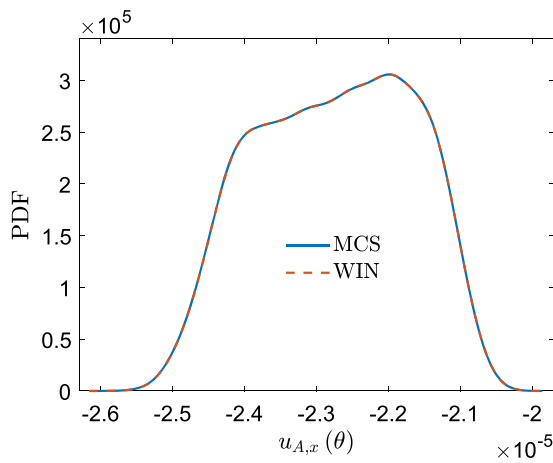


Fig. 10 Iterative errors of different retained terms

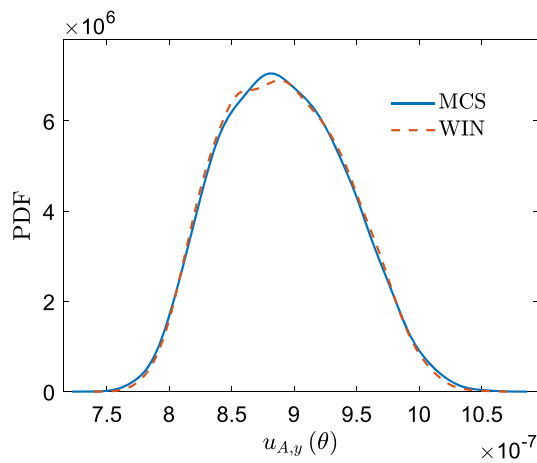
much cheaper than MCS and a speedup of more than one hundred times is achieved.

We also highlight that statistical properties of the stochastic solution are easily computed based on WIN-SVEM. Here we focus on the first and second order global statistical moments, that is, the mean value and the standard deviation dependent on spatial positions. According to (39), the mean value vector $\bar{\mathbf{u}}$ is computed as

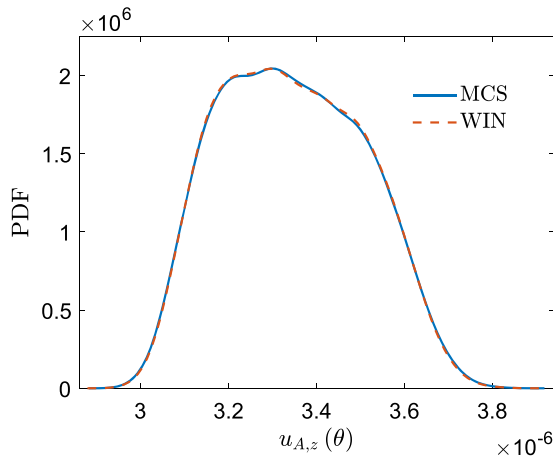
$$\bar{\mathbf{u}} = \sum_{i=1}^k \widehat{\mathbb{E}} \{ \lambda_i(\hat{\boldsymbol{\theta}}) \} \mathbf{d}_{\text{WIN},i}, \tag{66}$$



(a) Comparison of PDFs of the stochastic displacement $u_{A,x}(\theta)$ in the x direction of the point A obtained by WIN-SVEM and MCS.



(b) Comparison of PDFs of the stochastic displacement $u_{A,y}(\theta)$ in the y direction of the point A obtained by WIN-SVEM and MCS.



(c) Comparison of PDFs of the stochastic displacement $u_{A,z}(\theta)$ in the z direction of the point A obtained by WIN-SVEM and MCS.

Fig. 11 PDFs of the stochastic displacements $u_{A,x}(\theta)$ (top left), $u_{A,y}(\theta)$ (top right) and $u_{A,z}(\theta)$ (bottom) in the x , y and z directions of the point A obtained by WIN-SVEM and MCS, respectively

which only involves the expectations of sample vectors $\{\lambda_i(\hat{\theta}) \in \mathbb{R}^{10^4}\}_{i=1}^k$ and has very low computational effort. The mean value components \bar{u}_x , \bar{u}_y and \bar{u}_z of $\bar{\mathbf{u}}$ in the x , y and z directions are seen from the first line of Fig. 12. Further, the standard deviation vector $\sigma_{\mathbf{u}}$ is calculated via

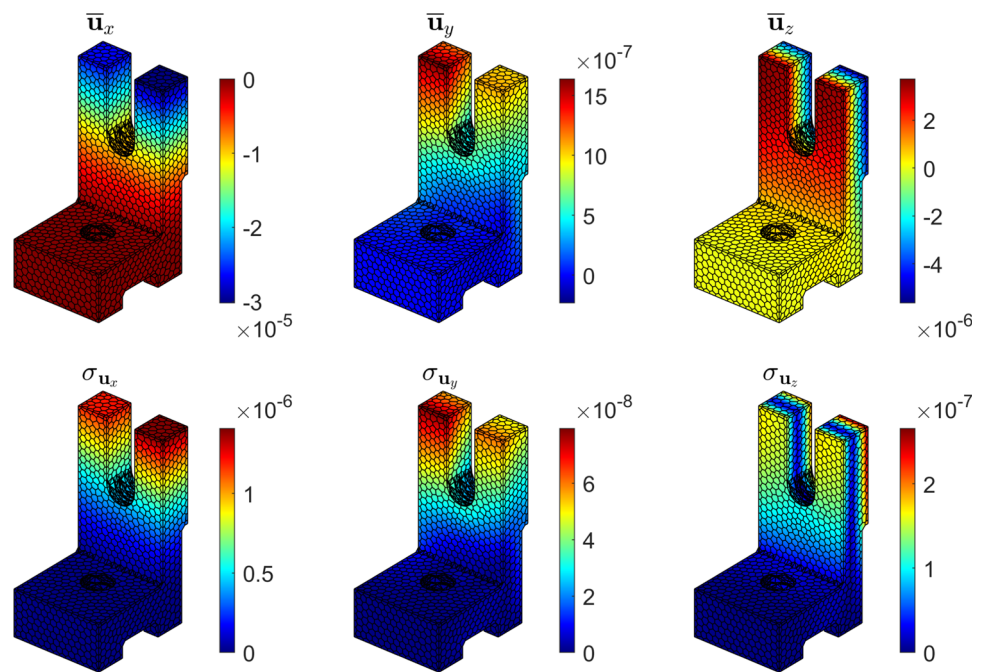
$$\sigma_{\mathbf{u}} = \sqrt{\sum_{i,j=1}^k \widehat{\mathbb{E}} \left\{ \lambda_i^{\square}(\hat{\theta}) \odot \lambda_j^{\square}(\hat{\theta}) \right\} \mathbf{d}_{\text{WIN},i} \odot \mathbf{d}_{\text{WIN},j}}, \quad (67)$$

where the random sample vectors are given by $\lambda_i^{\square}(\hat{\theta}) = \lambda_i(\hat{\theta}) - \widehat{\mathbb{E}} \left\{ \lambda_i(\hat{\theta}) \right\} \in \mathbb{R}^{10^4}$, $i = 1, \dots, k$. The components σ_{u_x} , σ_{u_y} and σ_{u_z} of $\sigma_{\mathbf{u}}$ in the x , y and z directions can be found in the second line of Fig. 12.

6 Conclusions

We presented two numerical approaches, PC-SVEM and WIN-SVEM, for solving stochastic systems derived from the stochastic virtual element discretization of 2D and 3D linear elastic stochastic problems. The deterministic virtual element method is first extended to SVEM to solve problems involving stochastic material properties and stochastic external forces. Several key calculations of SVEM can be inherited from the deterministic virtual element method, e.g. the gradient computations of virtual basis functions in (17) and the stabilizing element stiffness matrix in (25) (except for the random coefficient $\gamma_S^{(e)}(\theta)$). Numerical results demonstrate that both PC-SVEM and WIN-SVEM have comparable accuracy to MCS. However, PC-SVEM suffers from the curse of dimensionality and cannot be applied to high-dimensional stochastic problems. As a comparison, WIN-SVEM can efficiently solve both low- and high-dimensional stochastic problems without any modification, which has been verified by a numerical example of up to 35 stochastic dimensions. Further, although only linear elastic stochastic problems are concerned in this paper, both the proposed PC-SVEM and WIN-SVEM can be applied to more general cases, which will be further investigated in subsequent research.

Fig. 12 The mean functions \bar{u}_x , \bar{u}_y and \bar{u}_z (the first line) in the x , y and z directions, and the standard derivation functions σ_{u_x} , σ_{u_y} and σ_{u_z} (the second line) in the x , y and z directions



Acknowledgements 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 number 433082294).

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aldakheel F, Hudobivnik B, Hussein A et al (2018) Phase-field modeling of brittle fracture using an efficient virtual element scheme. *Comput Methods Appl Mech Eng* 341:443–466. <https://doi.org/10.1016/j.cma.2018.07.008>
- Aldakheel F, Hudobivnik B, Artioli E et al (2020) Curvilinear virtual elements for contact mechanics. *Comput Methods Appl Mech Eng* 372(113):394. <https://doi.org/10.1016/j.cma.2020.113394>
- Antonietti PF, Bruggi M, Scacchi S et al (2017) On the virtual element method for topology optimization on polygonal meshes: a numerical study. *Comput Math Appl* 74(5):1091–1109. <https://doi.org/10.1016/j.camwa.2017.05.025>
- Antonietti PF, da Veiga LB, Manzini G (2022) The virtual element method and its applications. Springer, Berlin. <https://doi.org/10.1007/978-3-030-95319-5>
- Artioli E, Beirão da Veiga L, Lovadina C et al (2017) Arbitrary order 2D virtual elements for polygonal meshes: part I, elastic problem. *Comput Mech* 60:355–377. <https://doi.org/10.1007/s00466-017-1404-5>
- Babuška I, Nobile F, Tempone R (2007) A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J Numer Anal* 45(3):1005–1034. <https://doi.org/10.1137/050645142>
- Benedetto MF, Berrone S, Pieraccini S et al (2014) The virtual element method for discrete fracture network simulations. *Comput Methods Appl Mech Eng* 280:135–156. <https://doi.org/10.1016/j.cma.2014.07.016>
- Blatman G, Sudret B (2010) An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Probab Eng Mech* 25(2):183–197. <https://doi.org/10.1016/j.probangmech.2009.10.003>
- Chi H, Da Veiga LB, Paulino G (2017) Some basic formulations of the virtual element method (VEM) for finite deformations. *Comput Methods Appl Mech Eng* 318:148–192. <https://doi.org/10.1016/j.cma.2016.12.020>
- Chi H, Pereira A, Menezes IF et al (2020) Virtual element method (VEM)-based topology optimization: an integrated framework. *Struct Multidiscip Optim* 62:1089–1114. <https://doi.org/10.1007/s00158-019-02268-w>
- Da Veiga LB, Brezzi F, Marini LD (2013) Virtual elements for linear elasticity problems. *SIAM J Numer Anal* 51(2):794–812. <https://doi.org/10.1137/120874746>

12. Da Veiga LB, Dassi F, Russo A (2017) High-order virtual element method on polyhedral meshes. *Comput Math Appl* 74(5):1110–1122. <https://doi.org/10.1016/j.camwa.2017.03.021>
13. Fuhg JN, Fau A, Nackenhorst U (2021) State-of-the-art and comparative review of adaptive sampling methods for kriging. *Arch Comput Methods Eng* 28(4):2689–2747. <https://doi.org/10.1007/s11831-020-09474-6>
14. Gain AL, Talischi C, Paulino GH (2014) On the virtual element method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes. *Comput Methods Appl Mech Eng* 282:132–160. <https://doi.org/10.1016/j.cma.2014.05.005>
15. Ghanem RG, Spanos PD (2003) Stochastic finite elements: a spectral approach. Courier Corporation, North Chelmsford. <https://doi.org/10.1007/978-1-4612-3094-6>
16. Graham IG, Kuo FY, Nuyens D et al (2011) Quasi-Monte Carlo methods for elliptic PDEs with random coefficients and applications. *J Comput Phys* 230(10):3668–3694. <https://doi.org/10.1016/j.jcp.2011.01.023>
17. Hussein A, Hudobivnik B, Wriggers P (2020) A combined adaptive phase field and discrete cutting method for the prediction of crack paths. *Comput Methods Appl Mech Eng* 372(113):329. <https://doi.org/10.1016/j.cma.2020.113329>
18. Keese A, Matthies HG (2005) Hierarchical parallelisation for the solution of stochastic finite element equations. *Comput Struct* 83(14):1033–1047. <https://doi.org/10.1016/j.compstruc.2004.11.014>
19. Khuri AI, Mukhopadhyay S (2010) Response surface methodology. *Wiley Interdiscip Rev Comput Stat* 2(2):128–149. <https://doi.org/10.1002/wics.73>
20. Mengolini M, Benedetto MF, Aragón AM (2019) An engineering perspective to the virtual element method and its interplay with the standard finite element method. *Comput Methods Appl Mech Eng* 350:995–1023. <https://doi.org/10.1016/j.cma.2019.02.043>
21. Papadrakakis M, Papadopoulos V (1996) Robust and efficient methods for stochastic finite element analysis using Monte Carlo simulation. *Comput Methods Appl Mech Eng* 134(3–4):325–340. [https://doi.org/10.1016/0045-7825\(95\)00978-7](https://doi.org/10.1016/0045-7825(95)00978-7)
22. Pellissetti MF, Ghanem RG (2000) Iterative solution of systems of linear equations arising in the context of stochastic finite elements. *Adv Eng Softw* 31(8–9):607–616. [https://doi.org/10.1016/S0965-9978\(00\)00034-X](https://doi.org/10.1016/S0965-9978(00)00034-X)
23. Quarteroni A, Valli A (2008) Numerical approximation of partial differential equations. Springer, Berlin. <https://doi.org/10.1007/978-3-540-85268-1>
24. Saad Y (2011) Numerical methods for large eigenvalue problems. SIAM, Philadelphia. <https://doi.org/10.1137/1.9781611970739>
25. Sakamoto S, Ghanem R (2002) Polynomial chaos decomposition for the simulation of non-Gaussian nonstationary stochastic processes. *J Eng Mech* 128(2):190–201. [https://doi.org/10.1061/\(ASCE\)0733-9399\(2002\)128:2\(190\)](https://doi.org/10.1061/(ASCE)0733-9399(2002)128:2(190))
26. Smith RC (2013) Uncertainty quantification: theory, implementation, and applications, vol 12. SIAM, Philadelphia
27. Spanos PD, Beer M, Red-Horse J (2007) Karhunen–Loève expansion of stochastic processes with a modified exponential covariance kernel. *J Eng Mech* 133(7):773–779. [https://doi.org/10.1061/\(ASCE\)0733-9399\(2007\)133:7\(773\)](https://doi.org/10.1061/(ASCE)0733-9399(2007)133:7(773))
28. Stefanou G (2009) The stochastic finite element method: past, present and future. *Comput Methods Appl Mech Eng* 198(9–12):1031–1051. <https://doi.org/10.1016/j.cma.2008.11.007>
29. Talischi C, Paulino GH, Pereira A et al (2012) Polymesher: a general-purpose mesh generator for polygonal elements written in matlab. *Struct Multidiscip Optim* 45:309–328. <https://doi.org/10.1007/s00158-011-0706-z>
30. Beirão da Veiga L, Brezzi F, Cangiani A et al (2013) Basic principles of virtual element methods. *Math Models Methods Appl Sci* 23(01):199–214. <https://doi.org/10.1142/S0218202512500492>
31. Beirão da Veiga L, Brezzi F, Marini LD et al (2014) The Hitchhiker’s guide to the virtual element method. *Math Models Methods Appl Sci* 24(08):1541–1573. <https://doi.org/10.1142/S021820251440003X>
32. Wriggers P, Rust WT (2019) A virtual element method for frictional contact including large deformations. *Eng Comput* 36(7):2133–2161. <https://doi.org/10.1108/EC-02-2019-0043>
33. Wriggers P, Rust WT, Reddy B (2016) A virtual element method for contact. *Comput Mech* 58:1039–1050. <https://doi.org/10.1007/s00466-016-1331-x>
34. Wriggers P, De Bellis M, Hudobivnik B (2021) A Taylor-hood type virtual element formulations for large incompressible strains. *Comput Methods Appl Mech Eng* 385(114):021. <https://doi.org/10.1016/j.cma.2021.114021>
35. Xiu D (2010) Numerical methods for stochastic computations: a spectral method approach. Princeton University Press, Princeton. <https://doi.org/10.1515/9781400835348>
36. Xiu D, Karniadakis GE (2002) The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM J Sci Comput* 24(2):619–644. <https://doi.org/10.1137/S1064827501387826>
37. Yosida K (2012) Functional analysis. Springer, Berlin. <https://doi.org/10.1007/978-3-642-61859-8>
38. Young DM (2014) Iterative solution of large linear systems. Elsevier, Amsterdam. <https://doi.org/10.1016/C2013-0-11733-3>
39. Zheng Z, Dai H (2017) Simulation of multi-dimensional random fields by Karhunen–Loève expansion. *Comput Methods Appl Mech Eng* 324:221–247. <https://doi.org/10.1016/j.cma.2017.05.022>
40. Zheng Z, Dai H, Wang Y et al (2021) A sample-based iterative scheme for simulating non-stationary non-Gaussian stochastic processes. *Mech Syst Signal Process* 151(107):420. <https://doi.org/10.1016/j.ymsp.2020.107420>
41. Zheng Z, Beer M, Dai H et al (2022) A weak-intrusive stochastic finite element method for stochastic structural dynamics analysis. *Comput Methods Appl Mech Eng* 399(115):360. <https://doi.org/10.1016/j.cma.2022.115360>
42. Zheng Z, Valdebenito M, Beer M et al (2023) A stochastic finite element scheme for solving partial differential equations defined on random domains. *Comput Methods Appl Mech Eng* 405(115):860. <https://doi.org/10.1016/j.cma.2022.115860>

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.