

A stochastic finite element scheme for solving partial differential equations defined on random domains

Zhibao Zheng^{a,d,*}, Marcos Valdebenito^c, Michael Beer^{b,e,f}, 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*Chair for Reliability Engineering, TU Dortmund University, Leonhard-Euler-Straße 5, 44227 Dortmund, Germany*

^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 proposes a novel stochastic finite element scheme to solve partial differential equations defined on random domains. A geometric mapping algorithm first transforms the random domain into a reference domain. By combining the mesh topology (i.e. the node numbering and the element numbering) of the reference domain and random nodal coordinates of the random domain, random meshes of the original problem are obtained by only one mesh of the reference domain. In this way, the original problem is still discretized and solved on the random domain instead of the reference domain. A random isoparametric mapping of random meshes is then proposed to generate the stochastic finite element equation of the original problem. We adopt a weak-intrusive method to solve the obtained stochastic finite element equation. In this method, the unknown stochastic solution is decoupled into a sum of the products of random variables and deterministic vectors. Deterministic vectors are computed by solving deterministic finite element equations, and corresponding random variables are solved by a proposed sampling method. The computational effort of the proposed method does not increase dramatically as the stochastic dimension increases and it can solve high-dimensional stochastic problems with low computational effort, thus the proposed method avoids the curse of dimensionality successfully. Four numerical examples are given to demonstrate the good performance of the proposed method.

Keywords: Random domains; Mesh transformation; Random isoparametric mapping; Stochastic finite element method; Random interfaces;

1. Introduction

As predicting uncertainty propagation of physical models has become an important part of the analysis of many engineering problems, it is necessary to develop efficient computational schemes for dealing with a variety of uncertainties, e.g. random material properties, random external forces, random geometries and their couplings [1, 2]. The focus of this work is on the geometric uncertainty, which may arise from many kinds of problems, e.g. topology optimization, random multi-phased materials, etc [3, 4, 5, 6, 7].

In this paper, we pay attention on solving deterministic/stochastic partial differential equations (PDEs) defined on random/parameter-dependent domains. In the last decades, several methods have been proposed for this purpose. Following the approaches to deal with random geometries, they can be divided into the remeshing method [8], the geometric mapping method [9], the fictitious method [10, 11] and the extended stochastic finite element method [5, 6], etc. A straightforward method to deal with random domains is the Monte Carlo simulation (MCS) [12] and other non-intrusive techniques, e.g. multilevel MCS, response surface method, stochastic collocation method, etc [8, 13, 14]. This kind of method remeshes the geometry associated with each random sample and classical finite element solvers are then used to solve the solution on each sampled mesh. It is easy to implement and existing codes can be adopted without difficulties. The method is independent or weakly dependent on stochastic dimensions, thus it can be applied to the geometries related to a large number of random variables. However, numerous deterministic simulations are necessary in order to obtain a high-precision stochastic solution, which leads to expensive computational costs. Also, since the samples of the stochastic solution are obtained on different domains and meshes, extra attention is required for the postprocessing of the stochastic solution. In order to avoid remeshing the domain for each sample realization, the geometric mapping

*Corresponding author

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

24 method is proposed in [9, 15, 16]. The method maps the random domain into a reference domain
25 and then transforms the PDE defined on the random domain into a stochastic PDE defined on the
26 reference domain. In this method, a boundary-conforming coordinate system is used to map the
27 random geometry, which can be considered as a stochastic extension of deterministic mesh map-
28 ping methods [17]. For the obtained stochastic PDE, classical stochastic finite element solvers
29 are used to solve the stochastic finite element equation (SFEE), e.g. MCS, stochastic collocation
30 method, spectral stochastic finite element method, etc [9, 18, 19, 20]. However, extra complexities
31 are introduced when tackling nonlinear PDEs since the differential operator is coupled with the
32 geometric mapping.

33 The fictitious domain method, as a stochastic extension of the deterministic fictitious domain
34 method [21], is proposed to solve PDEs defined on random domains [10, 11]. The fictitious do-
35 main is a deterministic domain including all possible sample realizations of the random domain
36 and it usually has simple geometry. This kind of method can be applied to complex geometries,
37 but effective methods are required to tackle the random boundaries of the random domain in the
38 fictitious domain. In [10], the original PDE is transformed into a saddle-point problem defined on
39 the fictitious domain and the original boundary conditions are enforced by Lagrange multipliers.
40 In [11], the original PDE is reformulated on the fictitious domain. The proper generalized decom-
41 position (PGD) method is used to approximate the stochastic solution and a stochastic indicator
42 function is proposed to capture the random domain in the fictitious domain. Furthermore, the
43 extended stochastic finite element method is proposed in [5, 6, 22, 23]. This method still reformu-
44 lates the PDE on a fictitious domain and extends the deterministic extended finite element method
45 [24] to handle the discontinuities of the stochastic solution on the fictitious domain. The spectral
46 stochastic finite element method [18, 25] is then used to solve the solution on the mesh of the
47 fictitious domain. Other methods have also been developed, e.g. the computational frameworks
48 and error estimations for Neumann and Dirichlet boundary value problems defined on random
49 domains [26, 27], the perturbation-based methods for problems with small geometric variability
50 [28, 29, 30] and the PGD-based descriptions for geometric parameters [31, 32].

51 In this paper, we develop an efficient numerical scheme for solving linear deterministic/stochastic
52 partial differential equations (PDEs) defined on random/parameter-dependent domains. To avoid

53 remeshing the random domain, a geometric mapping algorithm is adopted to transform the random
54 domain into a reference domain. As the unknown stochastic solution of the transformation equa-
55 tion, random nodal coordinates of the random domain are obtained by solving a Laplace equation
56 with random boundary conditions. The random mesh of the random domain can be generated by
57 combining the mesh topology (i.e. the node numbering and the element numbering) of the ref-
58 erence domain and random nodal coordinates. A random isoparametric mapping of the random
59 mesh is developed to assemble the stochastic stiffness matrix and the stochastic force vector. The
60 assembly procedure is the same as the classical FEM assembly and is completely non-intrusive.
61 Existing FEM codes can be embedded without any modification.

62 We mention that a similar random mesh approach is also used in [16] as a discretized version
63 of the method proposed in [9, 15] and conditions are given to ensure the well-posedness of the
64 geometric transformation. However, the method relies on the polynomial chaos approximation
65 of the random coordinates and still assembles stochastic matrices and vectors in the reference
66 domain, which leads to the coupling of the differential operator of the PDE and the geometric
67 uncertainty. Much attention is needed to tackle the coupling when nonlinear PDEs defined on
68 random domains are solved. Our proposed method still discretizes and solves the original PDE on
69 the random domain instead of the reference domain. Hence it decouples the differential operator
70 of the PDE and the geometric uncertainty, which is consistent with the fact that the differential
71 operators of PDEs are not sensitive to geometric domains when using the finite element method.
72 We also mention that a random isoparametric mapping is used in [28], which embeds the random
73 isoparametric mapping of the random boundary into the spectral SFEM frame. [This method relies](#)
74 [on the polynomial chaos-based approximations of random boundaries, mapping of differential op-](#)
75 [erators and corresponding Jacobian matrices, which is considered as a kind of intrusive approach.](#)
76 [Additional computational complexity is thus required. Also, high-dimensional stochastic prob-](#)
77 [lems remain challenging for this method.](#) The proposed random isoparametric mapping in this
78 paper has the same computational complexity as the deterministic isoparametric mapping and it is
79 easy to implement by existing FEM assembly codes.

80 After assembling stochastic matrices and vectors, we adopt a weak-intrusive method to solve
81 the obtained stochastic finite element equation. In this method, the unknown stochastic solution

82 is decoupled into a sum of the products of a set of random variables and deterministic vectors.
83 Deterministic vectors are obtained by solving deterministic finite element equations that are gen-
84 erated by applying the stochastic Galerkin method to the original SFEE. Corresponding random
85 variables are the solutions of one-dimensional stochastic algebraic equations that are solved ef-
86 ficiently by use of a proposed sampling method. Since the computational effort of the proposed
87 method does not increase dramatically as the stochastic dimension increases and it can solve high-
88 dimensional stochastic problems with low computational effort, the proposed method avoids the
89 curse of dimensionality successfully.

90 The paper is organized as follows: Section 2 gives the basic setting of deterministic/stochastic
91 PDEs on random domains and a geometric transformation from the random domain to a reference
92 domain is presented. Section 3 introduces a random isoparametric mapping method to assemble
93 SFEE. A weak-intrusive SFEM is then used to solve the obtained SFEE in Section 4. The al-
94 gorithm implementation of the proposed method is elaborated in Section 5. Following that, four
95 numerical examples, including an elastic equation defined on a random domain, a stochastic ellip-
96 tic PDE with a random interface and a case from orthodontics with random material properties and
97 random geometry, are given to demonstrate the performance of the proposed method in Section 6,
98 and conclusions and discussions follow in Section 7.

99 2. Geometric transformation of the random domain

100 2.1. PDEs defined on random domains

101 Let $(\Theta, \Xi, \mathcal{P})$ be a suitable probability space, where Θ denotes the space of elementary events,
102 Ξ is a σ -algebra defined on Θ and \mathcal{P} is a probability measure. In this paper, we consider the
103 deterministic/stochastic linear PDE defined on a random domain $\mathcal{D}(\theta) \subset \mathbb{R}^d$ with the physical di-
104 mension $d = 1, 2, 3$ and the random boundary $\partial\mathcal{D}(\theta)$. We solve the following stochastic problem:
105 find a stochastic solution $\mathbf{u}(\mathbf{x}, \theta) : \mathcal{D}(\theta) \rightarrow \mathbb{R}$ such that the following PDE holds for all $\theta \in \Theta$,

$$\mathcal{L}(\mathbf{u}(\mathbf{x}, \theta), \mathbf{x}) = 0 \quad \text{in } \mathcal{D}(\theta), \quad (1)$$

106 subjected to boundary conditions on $\partial\mathcal{D}(\theta)$, where $\mathcal{L}(\cdot)$ is a linear differential operator, $\mathbf{x} =$
107 $(x_1, \dots, x_d) \in \mathbb{R}^d$ is the Cartesian coordinate. We assume that the random boundary $\partial\mathcal{D}(\theta)$ is

108 sufficiently regular and that Eq. (1) is well-posed. The randomness of the domain $\mathcal{D}(\theta)$ is only
 109 described by the random boundary $\partial\mathcal{D}(\theta)$ that is related to random variables/fields (the parameters
 110 for describing unfixed domains can also be considered as special random variables). Although we
 111 only consider linear PDEs in this paper, nonlinear PDEs defined on random domains are readily
 112 solved in the computational frame proposed in this paper.

113 2.2. Geometric transformation from random domain to reference domain

114 To avoid remeshing the random domain $\mathcal{D}(\theta^{(i)})$ for each sample $\theta^{(i)}$, $i = 1, \dots, n_s$, where
 115 n_s is the number of random samples, we transform the random domain into a reference domain
 116 and represent the random domain by the mesh topology of the reference domain. To transform
 117 the random domain $\mathcal{D}(\theta) \subset \mathbb{R}^d$ into a reference domain $\bar{\mathcal{D}} \subset \mathbb{R}^d$, we consider that the random
 118 coordinate $\mathbf{x}(\theta) = (x_1(\theta), \dots, x_d(\theta)) \in \mathcal{D}(\theta)$ is mapped into the deterministic coordinate $\bar{\mathbf{x}} =$
 119 $(\bar{x}_1, \dots, \bar{x}_d) \in \bar{\mathcal{D}}$ by the following transformation,

$$\bar{\mathbf{x}} = \mathcal{M}^{-1}(\mathbf{x}(\theta), \theta), \quad (2)$$

120 and the random coordinate $\mathbf{x}(\theta)$ is represented by the deterministic coordinate $\bar{\mathbf{x}}$ and the inverse
 121 mapping of Eq. (2),

$$\mathbf{x}(\theta) = \mathcal{M}(\bar{\mathbf{x}}, \theta), \quad (3)$$

122 where $\mathcal{M}(\cdot, \theta)$ represents the mapping operator and $\mathcal{M}^{-1}(\cdot, \theta)$ is its inverse operator. In this way,
 123 we can use the deterministic coordinate of the reference domain to represent the random coordinate
 124 of the random domain. In other words, the random domain is fully described by the reference
 125 domain and the realization θ . The selection of $\mathcal{M}(\cdot, \theta)$ is not unique and it is dependent on the
 126 selection of the reference domain and the uncertainty involved in the problem under consideration.
 127 We can adopt an arbitrary reference domain as long as the ease of implementation and well-
 128 posedness of the transformation can be ensured [9, 17]. In practice, the [mean](#) of the random
 129 domain is usually chosen as the reference domain.

130 2.3. Geometric mapping based on the Laplace equation

131 We execute the transformation Eq. (3) on the mesh of the reference domain. Correspondingly,
 132 the random mesh of the random domain is obtained by the mesh transformation of the reference

133 domain, which can be considered a mesh-based random geometry description. Several differen-
 134 tial system-based mapping methods are initially developed for the mesh transformation, e.g. the
 135 elliptic equations, hyperbolic equations, etc [17], which are then extended to deal with random
 136 geometries in [9, 15]. In this paper we adopt the Laplace equation for the transformation $\mathcal{M}(\cdot, \theta)$
 137 in Eq. (3), which corresponds to

$$\Delta x_i(\theta) = 0 \quad \text{in } \overline{\mathcal{D}}, \quad i = 1, \dots, d \quad (4)$$

138 with the geometric boundary constraints

$$x_i(\theta)|_{\Gamma_j} = \mathcal{M}_{i,j}(\bar{\mathbf{x}}|_{\Gamma_j}, \theta), \quad j = 1, \dots, b, \quad (5)$$

139 where $\Delta = \sum_{j=1}^d \frac{\partial^2}{\partial \bar{x}_j^2}$ represents the Laplace operator, $\bar{\mathbf{x}}|_{\Gamma_j} = [\bar{x}_1|_{\Gamma_j}, \dots, \bar{x}_d|_{\Gamma_j}] \in \mathbb{R}^d$ and $\mathbf{x}(\theta)|_{\Gamma_j} =$
 140 $[x_1(\theta)|_{\Gamma_j}, \dots, x_d(\theta)|_{\Gamma_j}] \in \mathbb{R}^d$ represent the deterministic and random coordinates on the boundary
 141 Γ_j , $\mathcal{M}_{i,j}(\cdot, \theta)$ represents the mapping of the i -th coordinate on the boundary Γ_j , b is the total number
 142 of geometry boundaries under consideration. In this paper, we assume that the obtained random
 143 mesh has a good quality and can be used to solve the PDE, e.g., there should be no element flips
 144 in the random mesh. Conditions to ensure the well-posedness of the transformation can be found
 145 in [16, 17]. Further, several advanced mesh deformation methods [33, 34] can be used for more
 146 general cases that the Laplace equation-based mesh transformation does not work well, e.g., the
 147 case of too large mesh deformations caused by large uncertainties, which is beyond the scope of
 148 this paper and will be presented in follow-up studies.

149 2.4. Finite element solution of the transformation equation

150 To solve Eq. (4), we discretize the reference domain by use of the finite element method [35].
 151 Let $\bar{\mathbf{X}}_i \in \mathbb{R}^n$, $i = 1, \dots, d$ be the discretized nodal coordinates of the coordinate \bar{x}_i obtained
 152 by the finite element discretization, where n is the number of nodes, and $\mathbf{X}_i(\theta) \in \mathbb{R}^n$ be the
 153 corresponding stochastic solution on the discretized nodes, which is also the discretization of the
 154 random coordinate $x_i(\theta)$. The finite element equation of Eq. (4) is thus given by

$$\begin{cases} \mathbf{K}_G \mathbf{X}_i(\theta) = 0 \\ \mathbf{X}_i(\theta)|_{\Gamma_j} = \mathcal{M}_{i,j}(\bar{\mathbf{X}}|_{\Gamma_j}, \theta), \quad j = 1, \dots, b \end{cases}, \quad i = 1, \dots, d, \quad (6)$$

155 which can be rewritten as

$$\begin{bmatrix} \widetilde{\mathbf{K}}_{\mathcal{G},0,0} & \widetilde{\mathbf{K}}_{\mathcal{G},0,1} & \cdots & \widetilde{\mathbf{K}}_{\mathcal{G},0,b} \\ \widetilde{\mathbf{K}}_{\mathcal{G},1,0} & \widetilde{\mathbf{K}}_{\mathcal{G},1,1} & \cdots & \widetilde{\mathbf{K}}_{\mathcal{G},1,b} \\ \vdots & \vdots & \ddots & \vdots \\ \widetilde{\mathbf{K}}_{\mathcal{G},b,0} & \widetilde{\mathbf{K}}_{\mathcal{G},b,1} & \cdots & \widetilde{\mathbf{K}}_{\mathcal{G},b,b} \end{bmatrix} \begin{bmatrix} \mathbf{X}_i(\theta)|_{\mathcal{D}(\theta)} \\ \mathcal{M}_{i,1}(\overline{\mathbf{X}}|_{\Gamma_1}, \theta) \\ \vdots \\ \mathcal{M}_{i,b}(\overline{\mathbf{X}}|_{\Gamma_b}, \theta) \end{bmatrix} = 0, \quad i = 1, \dots, d, \quad (7)$$

156 where $\mathbf{X}_i(\theta)|_{\mathcal{D}(\theta)} \in \mathbb{R}^{n_0}$ represents the unknown stochastic solution of the nodes not on the con-
 157 straint boundaries, n_0 is the number of the nodes not on the constraint boundaries, $\mathbf{X}_i(\theta)|_{\Gamma_j(\theta)} =$
 158 $\mathcal{M}_{i,j}(\overline{\mathbf{X}}|_{\Gamma_j}, \theta) \in \mathbb{R}^{n_j}$, $j = 1, \dots, b$ are the stochastic solutions of the nodes on the boundary Γ_j and
 159 they can be calculated via Eq. (5), $\widetilde{\mathbf{K}}_{\mathcal{G},i,j} \in \mathbb{R}^{n_i \times n_j}$ are corresponding submatrices, n_1, \dots, n_b are the
 160 number of nodes on the boundary Γ_j , $j = 1, \dots, b$. The matrix $\mathbf{K}_{\mathcal{G}}$ is also not unique and depends
 161 on the selection of the reference domain. For a given reference domain and its mesh discretization,
 162 the matrix $\mathbf{K}_{\mathcal{G}}$ is unique and assembled by using the classical finite element method [35]. Eq. (7)
 163 is considered as a deterministic finite element equation with stochastic Dirichlet boundary condi-
 164 tions. To calculate the submatrix $\widetilde{\mathbf{K}}_{\mathcal{G},i,j}$, $i, j = 0, 1, \dots, b$, we let $\{\mathcal{I}_{i,1}, \dots, \mathcal{I}_{i,n_i}\}$ be the numbering
 165 of nodes on the i -th mapped boundary (the case $i = 0$ corresponds to the nodes not on mapped
 166 boundaries). The index matrix $\mathcal{I}\mathcal{I}_{ij}$ is given by

$$\mathcal{I}\mathcal{I}_{ij} = \begin{bmatrix} \mathcal{I}_{i,1} \\ \vdots \\ \mathcal{I}_{i,n_i} \end{bmatrix} \otimes [\mathcal{I}_{j,1}, \dots, \mathcal{I}_{j,n_j}] = \begin{bmatrix} (\mathcal{I}_{i,1}, \mathcal{I}_{j,1}) & \cdots & (\mathcal{I}_{i,1}, \mathcal{I}_{j,n_j}) \\ \vdots & \ddots & \vdots \\ (\mathcal{I}_{i,n_i}, \mathcal{I}_{j,1}) & \cdots & (\mathcal{I}_{i,n_i}, \mathcal{I}_{j,n_j}) \end{bmatrix} \in \mathbb{R}^{n_i \times n_j}, \quad (8)$$

167 where \otimes represents the outer product operator. The submatrix $\widetilde{\mathbf{K}}_{\mathcal{G},i,j}$ is thus calculated by

$$\widetilde{\mathbf{K}}_{\mathcal{G},i,j} = \mathbf{K}_{\mathcal{G}} [\mathcal{I}\mathcal{I}_{ij}]. \quad (9)$$

168 Based on the first row of Eq. (7), the stochastic solution $\mathbf{X}_i(\theta)|_{\mathcal{D}(\theta)}$ is solved by an explicit form

$$\mathbf{X}_i(\theta)|_{\mathcal{D}(\theta)} = \sum_{j=1}^b \underline{\mathbf{K}}_{\mathcal{G},j} \mathcal{M}_{i,j}(\overline{\mathbf{X}}|_{\Gamma_j}, \theta), \quad i = 1, \dots, d, \quad (10)$$

169 where the matrices $\underline{\mathbf{K}}_{\mathcal{G},j} = -\widetilde{\mathbf{K}}_{\mathcal{G},0,0}^{-1} \widetilde{\mathbf{K}}_{\mathcal{G},0,j} \in \mathbb{R}^{n_0 \times n_j}$.

170 The above transformation is summarized in Algorithm 1. In step 1, we choose a reference
 171 domain and generate its mesh topology $[\mathcal{K}\{\text{Node, Ele}\}, \{\overline{\mathbf{X}}_i\}_{i=1}^d]$, where $\mathcal{K}\{\text{Node, Ele}\}$ represents
 172 the topology of the node numbering and the element numbering. The matrix $\mathbf{K}_{\mathcal{G}}$ in Eq. (6) (or

Algorithm 1 Algorithm for mapping random domains to reference domains

- 1: Choose a reference domain and generate its mesh topology $\left[\mathcal{K} \{ \text{Node, Ele} \}, \{ \bar{\mathbf{X}}_i \}_{i=1}^d \right]$
 - 2: Assemble the deterministic matrix $\mathbf{K}_{\mathcal{G}}$ in Eq. (6) (or Eq. (7))
 - 3: **for** the geometric boundary $j = 1, \dots, b$ **do**
 - 4: Calculate the matrices $\underline{\mathbf{K}}_{\mathcal{G},j} = -\widetilde{\mathbf{K}}_{\mathcal{G},0,0}^{-1} \widetilde{\mathbf{K}}_{\mathcal{G},0,j} \in \mathbb{R}^{n_0 \times n_j}$
 - 5: **end**
 - 6: **for** the geometric boundary $j = 1, \dots, b$ **do**
 - 7: **for** the coordinate dimension $i = 1, \dots, d$ **do**
 - 8: The coordinate transformation $\mathcal{M}_{i,j}(\bar{\mathbf{X}}|_{\Gamma_j}, \theta)$ on the j -th boundary
 - 9: **end**
 - 10: **end**
 - 11: **for** the coordinate dimension $i = 1, \dots, d$ **do**
 - 12: Solve the random coordinate $\mathbf{X}_i(\theta)|_{\mathcal{D}(\theta)}$ via Eq. (10)
 - 13: **end**
-

Eq. (7)) is assembled in step 2, the computation of which is fully deterministic for a given reference domain. From step 3 to 5, the matrices $\widetilde{\mathbf{K}}_{\mathcal{G},j}$ are solved by a set of systems of linear equations,

$$\widetilde{\mathbf{K}}_{\mathcal{G},0,0} \underline{\mathbf{K}}_{\mathcal{G},j} = \widetilde{\mathbf{K}}_{\mathcal{G},0,j}, \quad j = 1, \dots, b, \quad (11)$$

which can be solved efficiently by use of existing FEM solvers [35] and can be executed in parallel for each boundary j . After that, the random boundaries are computed by step 6 to 10, which only depends on the choice of the reference domain and is almost independent of the finite element discretization (weakly depends on the number of the discretized nodes on the boundaries). We adopt a non-intrusive way to execute the transformation $\mathcal{M}_{i,j}(\bar{\mathbf{X}}|_{\Gamma_j}, \theta) \in \mathbb{R}^{n_j \times n_s}$ of the i -th coordinate on the j -th boundary, where $\theta = \{\theta^{(i)}\}_{i=1}^{n_s}$ are n_s sample realizations. In this way, changes in the geometric shape can be accurately captured compared to the approximation method [9, 16]. Also, the procedure from step 6 to 10 is independent of the matrix assembly from step 3 to 5 and they can be executed in parallel. Taking into account $\widetilde{\mathbf{K}}_{\mathcal{G},j}$ and $\mathcal{M}_{i,j}(\bar{\mathbf{X}}|_{\Gamma_j}, \theta)$ as calculated in the previous steps, the stochastic solutions $\mathbf{X}_i(\theta)|_{\mathcal{D}(\theta)}$ are computed in step 12.

185 3. Discretization and assembly on random domains

186 Based on the above geometric mapping, we transform the random domain into a reference
 187 domain. A popular method is to transform the PDE defined on the random domain into a stochastic
 188 PDE defined on the reference domain [9, 15]. In this kind of method, the dependency of the
 189 PDE on the geometric randomness is transformed into the dependence on stochastic coefficients
 190 over the reference domain by means of a random Jacobian matrix. The differential operator of
 191 the PDE and the randomness are coupled in this way. In this section, we propose a random
 192 isoparametric mapping to deal with the PDEs defined on the random domain, which decouples the
 193 differential operator of the PDE and the randomness and has the same computational complexity
 194 as the classical isoparametric mapping.

195 3.1. Random isoparametric mapping

196 We recall the mesh topology $[\mathcal{K}\{\text{Node, Ele}\}, \bar{\mathbf{X}}]$ of the reference domain generated in step 1
 197 in Algorithm 1, where $\bar{\mathbf{X}} = [\bar{\mathbf{X}}_1, \dots, \bar{\mathbf{X}}_d] \in \mathbb{R}^{n \times d}$ is the set of node coordinates of the reference
 198 domain. We construct a random mesh of the random domain in the form $[\mathcal{K}\{\text{Node, Ele}\}, \mathbf{X}(\theta)]$,
 199 where $\mathbf{X}(\theta) = [\mathbf{X}_1(\theta), \dots, \mathbf{X}_d(\theta)] \in \mathbb{R}^{n \times d}$ is the stochastic solutions of Eq. (6). In other words,
 200 the random mesh is obtained by combining the topology $\mathcal{K}\{\text{Node, Ele}\}$ of the reference domain
 201 and the random coordinates $\mathbf{X}(\theta)$ of the random domain. To illustrate the construction of the
 202 random mesh, we consider two sample realizations $\mathcal{D}(\theta^{(1)})$ and $\mathcal{D}(\theta^{(2)})$ of the random domain
 203 shown in Fig. 1a and both of them are transformed into the reference domain $\bar{\mathcal{D}}$. The mesh
 204 topology $\mathcal{K}\{\text{Node, Ele}\}$ is generated on the reference domain by one meshing and the random
 205 coordinates $\mathbf{X}(\theta)$ of the random domain is solved by Eq. (6). The realizations of two random
 206 meshes $[\mathcal{K}\{\text{Node, Ele}\}, \mathbf{X}(\theta^{(1)})]$ and $[\mathcal{K}\{\text{Node, Ele}\}, \mathbf{X}(\theta^{(2)})]$ are thus obtained. They have the
 207 same mesh topology $\mathcal{K}\{\text{Node, Ele}\}$ and only one meshing is involved. Further, we consider the
 208 element analysis of the random elements $\mathcal{E}_i(\theta^{(1)})$ and $\mathcal{E}_i(\theta^{(2)})$ in random domains $\mathcal{D}(\theta^{(1)})$ and
 209 $\mathcal{D}(\theta^{(2)})$, which are mapped from the element \mathcal{E}_i in the reference domain $\bar{\mathcal{D}}$. As shown in Fig. 1b,
 210 the elements \mathcal{E}_i , $\mathcal{E}_i(\theta^{(1)})$ and $\mathcal{E}_i(\theta^{(2)})$ have the same element number (i.e. i) and consist of the
 211 same node group (i.e. $\{n_i^{(1)}, n_i^{(2)}, n_i^{(3)}, n_i^{(4)}\}$). The only difference between them is the coordinates of

212 the node group $\{n_i^{(1)}, n_i^{(2)}, n_i^{(3)}, n_i^{(4)}\}$. We extend the deterministic isoparametric mapping to random
 213 cases and map elements $\mathcal{E}_i(\theta^{(1)})$ and $\mathcal{E}_i(\theta^{(2)})$ into the isoparametric element \mathcal{E}_{iso} .

214 Let us simply recall the deterministic isoparametric mapping [35, 36]. For complex geometric
 215 domains, irregular elements are used to fit irregular curves (or surfaces), which increases com-
 216 putational complexity of the element analysis. Isoparametric mapping enables meshing geomet-
 217 ric domains with irregular elements but performing the element analysis using regular elements.
 218 Isoparametric formulations are used to map the irregular elements into regular elements and the
 219 mapping functions are usually the same as shape functions used for the solutions. Based on the
 220 mapping, the numerical integration on the irregular elements is transformed into that on the reg-
 221 ular isoparametric elements. For the explanation of this point, we consider a two-dimensional
 222 irregular rectangular element \mathcal{E} with four nodes (x_i, y_i) , $i = 1, \dots, 4$ as the physical element, and
 223 a two-dimensional regular rectangular element \mathcal{E}_{iso} with four nodes (η_i, ζ_i) , $i = 1, \dots, 4$ as the
 224 reference element. The basic idea of the deterministic isoparametric mapping is to use the shape
 225 functions $N_i(\eta, \zeta)$, $i = 1, \dots, 4$ on the reference element \mathcal{E}_{iso} to describe the shape of \mathcal{E} through
 226 the following equations

$$x(\eta, \zeta) = \sum_{i=1}^4 x_i N_i(\eta, \zeta), \quad y(\eta, \zeta) = \sum_{i=1}^4 y_i N_i(\eta, \zeta). \quad (12)$$

227 The solution $u(\eta, \zeta)$ is approximated in a similar way

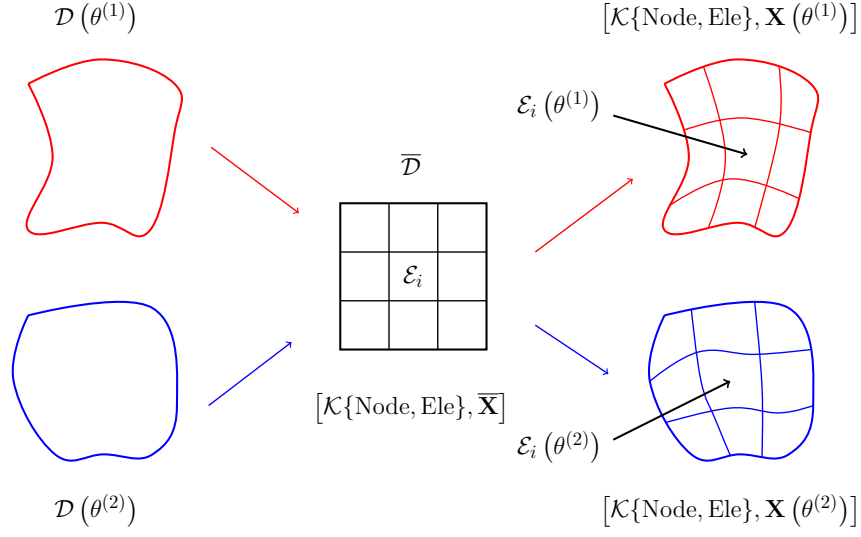
$$u(\eta, \zeta) = \sum_{i=1}^4 u_i N_i(\eta, \zeta). \quad (13)$$

228 After the above mapping, we can transform the irregular elements and the solution into those on
 229 the reference elements. The element analysis is then performed on the reference elements, such as
 230 calculating numerical integration and assembling element matrices.

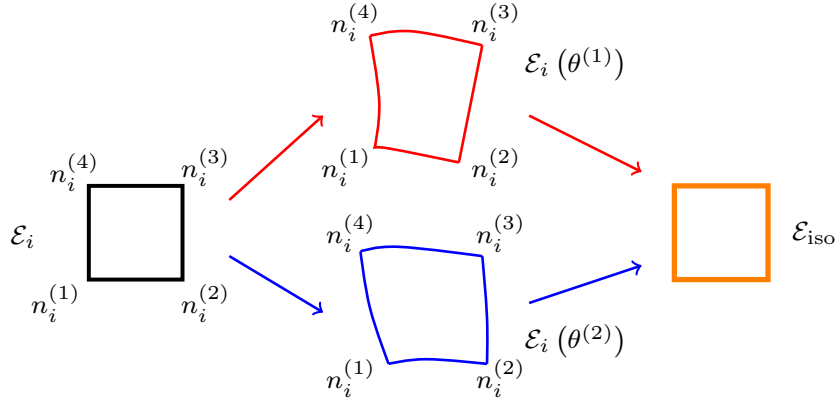
231 We extend the above idea of deterministic isoparametric mapping to random cases. Consider-
 232 ing the weak form of Eq. (1) discretized by finite element method, the stochastic solution $\mathbf{u}^{(i)}(\theta)$
 233 of the i -th element is thus represented by shape functions of the isoparametric element in a way
 234 similar to the deterministic isoparametric mapping Eq. (13)

$$\mathbf{u}^{(i)}(\theta) = \sum_{j=1}^{n_I} N_j(\boldsymbol{\eta}) \mathbf{u}^{(i,j)}(\theta) \quad (14)$$

235 and the random coordinates are approximated in a similar way,



(a) Random domain transformation and the mesh topology reuse.



(b) Random isoparametric mapping.

Figure 1: Domain transformation and random isoparametric mapping: (a) domain transformation and random mesh obtained from the reference domain, (b) random isoparametric mapping.

$$\mathbf{x}_k^{(i)}(\theta) = \sum_{j=1}^{n_I} N_j(\boldsymbol{\eta}) \mathbf{x}_k^{(i,j)}(\theta), \quad k = 1, \dots, d, \quad (15)$$

236 where $N_j(\boldsymbol{\eta})$ is the shape function of the isoparametric element, $\boldsymbol{\eta} \in \mathbb{R}^d$ represents the local
 237 coordinate of the isoparametric element, $\mathbf{u}^{(i,j)}(\theta)$ is the value of the stochastic solution $\mathbf{u}(\theta)$ on
 238 the j -th node of the i -th element, $\mathbf{x}_k^{(i,j)}(\theta)$ is the value of the random coordinate $\mathbf{x}_k(\theta)$ on the j -
 239 th node of the i -th element, n_I is the number of nodes of the i -th element. It is noted that the

240 random isoparametric mapping in this section is applicable to any type of element although only
 241 the rectangular element is shown in Fig. 1. Also, different $N_j(\boldsymbol{\eta})$ and n_I should be adopted if
 242 elements of different types and orders are used for the finite element discretization. Thus Eq. (14)
 243 and Eq. (15) can represent the isoparametric mapping analyses of general cases. Also, from now
 244 on we discard the reference domain $\overline{\mathcal{D}}$ and all analyses are performed on the random domain and
 245 the corresponding random mesh.

246 Inspired by the deterministic isoparametric analysis, the random transformation between the
 247 global coordinate $\mathbf{x}^{(i)}(\theta)$ and the local coordinate $\boldsymbol{\eta}$ is given by

$$\frac{\partial N_j(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}} = \mathbf{J}(\mathbf{x}^{(i)}(\theta)) \frac{\partial N_j(\boldsymbol{\eta})}{\partial \mathbf{x}^{(i)}(\theta)}, \quad j = 1, \dots, n_I, \quad (16)$$

248 where the vectors $\frac{\partial N_j(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}} \in \mathbb{R}^d$ and $\frac{\partial N_j(\boldsymbol{\eta})}{\partial \mathbf{x}^{(i)}(\theta)} \in \mathbb{R}^d$ are defined as

$$\frac{\partial N_j(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}} = \left[\frac{\partial N_j(\boldsymbol{\eta})}{\partial \eta_1}, \dots, \frac{\partial N_j(\boldsymbol{\eta})}{\partial \eta_d} \right]^T \in \mathbb{R}^d, \quad \frac{\partial N_j(\boldsymbol{\eta})}{\partial \mathbf{x}^{(i)}(\theta)} = \left[\frac{\partial N_j(\boldsymbol{\eta})}{\partial x_1^{(i)}(\theta)}, \dots, \frac{\partial N_j(\boldsymbol{\eta})}{\partial x_d^{(i)}(\theta)} \right]^T \in \mathbb{R}^d \quad (17)$$

249 and $\mathbf{J}(\mathbf{x}^{(i)}(\theta))$ is the random Jacobian matrix given by

$$\mathbf{J}(\mathbf{x}^{(i)}(\theta)) = \begin{bmatrix} \frac{\partial x_1^{(i)}(\theta)}{\partial \eta_1} & \dots & \frac{\partial x_d^{(i)}(\theta)}{\partial \eta_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_1^{(i)}(\theta)}{\partial \eta_d} & \dots & \frac{\partial x_d^{(i)}(\theta)}{\partial \eta_d} \end{bmatrix} \in \mathbb{R}^{d \times d}. \quad (18)$$

250 The inverse transformation of Eq. (16) is

$$\frac{\partial N_j(\boldsymbol{\eta})}{\partial \mathbf{x}^{(i)}(\theta)} = \mathbf{J}^{-1}(\mathbf{x}^{(i)}(\theta)) \frac{\partial N_j(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}}, \quad (19)$$

251 where $\mathbf{J}^{-1}(\mathbf{x}^{(i)}(\theta)) \in \mathbb{R}^{d \times d}$ is the inverse matrix of the Jacobian matrix $\mathbf{J}(\mathbf{x}^{(i)}(\theta))$. In this way,
 252 we transform the random global coordinates into the deterministic local coordinates. The above
 253 random isoparametric mapping is very similar to the classical isoparametric analysis but involves
 254 the random coordinates instead of the deterministic coordinates. We will extend the classical
 255 isoparametric mapping for solving PDEs to a random case in the next section.

256 3.2. Elastic equation defined on the random domain

257 We illustrate the proposed random isoparametric analysis using an elastic equation defined on
 258 a random domain, but the proposed method can be applied to more general PDEs, as will be shown
 259 in the example 6.2. Consider a two-dimensional linear elastic equation written as

$$\operatorname{div}(\sigma(\mathbf{x}(\theta))) + f(\mathbf{x}(\theta)) = 0 \quad \text{in } \mathcal{D}(\theta), \quad (20)$$

260 where $\operatorname{div}(\cdot)$ represents the divergence operator, $\sigma(\mathbf{x}(\theta))$ is the stochastic stress tensor and $f(\mathbf{x}(\theta))$
 261 is the stochastic external force. Based on the finite element discretization of Eq. (20), the stochastic
 262 strain $\varepsilon^{(i)}(\theta)$ of the i -th element is given by

$$\varepsilon^{(i)}(\theta) = \left[\frac{\partial u_1^{(i)}(\mathbf{x}^{(i)}(\theta))}{\partial x_1^{(i)}(\theta)}, \frac{\partial u_2^{(i)}(\mathbf{x}^{(i)}(\theta))}{\partial x_2^{(i)}(\theta)}, \frac{\partial u_1^{(i)}(\mathbf{x}^{(i)}(\theta))}{\partial x_2^{(i)}(\theta)} + \frac{\partial u_2^{(i)}(\mathbf{x}^{(i)}(\theta))}{\partial x_1^{(i)}(\theta)} \right] = [\mathbf{B}_1(\theta), \dots, \mathbf{B}_{n_I}(\theta)] \mathbf{u}^{(i)}(\theta), \quad (21)$$

263 where $\mathbf{u}^{(i)}(\theta) = [\mathbf{u}^{(i,1)}(\theta)^T, \dots, \mathbf{u}^{(i,n_I)}(\theta)^T]^T$ is the stochastic solution of all nodes of the i -th ele-
 264 ment, $\mathbf{u}^{(i,j)}(\theta) = [u_1^{(i,j)}(\theta), u_2^{(i,j)}(\theta)]^T$ is the stochastic solution of the j -th node of the i -th element.

265 The submatrix $\mathbf{B}_j(\theta)$ of the strain matrix $\mathbf{B}(\theta) = [\mathbf{B}_1(\theta), \dots, \mathbf{B}_{n_I}(\theta)]$ is defined as

$$\mathbf{B}_j(\theta) = \begin{bmatrix} \frac{\partial N_j(\boldsymbol{\eta})}{\partial x_1^{(i)}(\theta)} & 0 \\ 0 & \frac{\partial N_j(\boldsymbol{\eta})}{\partial x_2^{(i)}(\theta)} \\ \frac{\partial N_j(\boldsymbol{\eta})}{\partial x_2^{(i)}(\theta)} & \frac{\partial N_j(\boldsymbol{\eta})}{\partial x_1^{(i)}(\theta)} \end{bmatrix} \in \mathbb{R}^{3 \times 2}, \quad j = 1, \dots, n_I. \quad (22)$$

266 Further, the element stochastic stiffness matrix $\mathbf{k}^{(i)}(\theta)$ can be calculated via the isoparametric ele-
 267 ment [35, 36]

$$\mathbf{k}^{(i)}(\theta) = \int_{\mathcal{D}^{(i)}(\theta)} \mathbf{B}^T(\theta) \mathbf{C} \mathbf{B}(\theta) d\mathbf{x}(\theta) = \int_{\mathcal{E}_{\text{iso}}} \mathbf{B}^T(\theta) \mathbf{C} \mathbf{B}(\theta) |\mathbf{J}(\mathbf{x}(\theta))| d\boldsymbol{\eta} \in \mathbb{R}^{2n_I \times 2n_I}, \quad (23)$$

268 where $i = 1, \dots, n_e$ is the element numbering, $\mathcal{D}^{(i)}(\theta)$ is the i -th random element, $|\mathbf{J}(\mathbf{x}(\theta))|$ repre-
 269 sents determinant of the matrix $\mathbf{J}(\mathbf{x}(\theta))$ and the material matrix \mathbf{C} is given by

$$\mathbf{C} = \frac{E}{1 - \mu^2} \begin{bmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & \frac{1-\mu}{2} \end{bmatrix}. \quad (24)$$

270 where E and μ are the Young's modulus and the Poisson's ratio, respectively. We can compute the
 271 global stochastic stiffness matrix by assembling the element stochastic stiffness matrix,

$$\mathbf{K}(\theta) = \bigcup_{i=1}^{n_e} (\mathbf{k}^{(i)}(\theta)) \in \mathbb{R}^{n \times n}, \quad (25)$$

272 where $\bigcup_{i=1}^{n_e}$ is the assembly operation. The global stochastic force vector $\mathbf{F}(\theta) = \bigcup_{e=1}^{n_e} (\mathbf{f}^{(e)}(\theta)) \in \mathbb{R}^n$
 273 can be obtained in a similar way, where $\mathbf{f}^{(e)}(\theta) \in \mathbb{R}^{2n_I}$ is the element stochastic force vector.

274 The inverse Jacobian matrix $\mathbf{J}^{-1}(\mathbf{x}^{(i)}(\theta))$ in Eq. (19) has the following form for 2D finite ele-
 275 ments,

$$\mathbf{J}^{-1}(\mathbf{x}^{(i)}(\theta)) = \frac{1}{|\mathbf{J}(\mathbf{x}^{(i)}(\theta))|} \begin{bmatrix} \frac{\partial x_2^{(i)}(\theta)}{\partial \eta_2} & -\frac{\partial x_2^{(i)}(\theta)}{\partial \eta_1} \\ -\frac{\partial x_1^{(i)}(\theta)}{\partial \eta_2} & \frac{\partial x_1^{(i)}(\theta)}{\partial \eta_1} \end{bmatrix} \in \mathbb{R}^{2 \times 2}, \quad (26)$$

276 where the component is given by

$$\frac{\partial x_k^{(i)}(\theta)}{\partial \eta_j} = \sum_{l=1}^{n_l} \frac{\partial N_l(\boldsymbol{\eta})}{\partial \eta_j} x_k^{(i,l)}(\theta), \quad k, j = 1, 2 \quad (27)$$

277 and its determinant is

$$|\mathbf{J}(\mathbf{x}^{(i)}(\theta))| = \frac{\partial x_1^{(i)}(\theta)}{\partial \eta_1} \frac{\partial x_2^{(i)}(\theta)}{\partial \eta_2} - \frac{\partial x_1^{(i)}(\theta)}{\partial \eta_2} \frac{\partial x_2^{(i)}(\theta)}{\partial \eta_1}. \quad (28)$$

278 Combining Eq. (19) and Eq. (26) we rewrite Eq. (22) as

$$\mathbf{B}_j(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{J}^{-1}(\mathbf{x}^{(i)}(\theta)) \left[\frac{\partial N_j(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}}, \mathbf{0}_{2 \times 1} \right] + \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \mathbf{J}^{-1}(\mathbf{x}^{(i)}(\theta)) \left[\mathbf{0}_{2 \times 1}, \frac{\partial N_j(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}} \right], \quad (29)$$

279 which indicates that the random coordinate $\mathbf{x}(\theta)$ is only related to the random inverse Jacobian ma-
 280 trix $\mathbf{J}^{-1}(\mathbf{x}^{(i)}(\theta))$. We compute the random coordinate $\mathbf{x}(\theta)$ by the non-intrusive solution of Eq. (6)
 281 in practice. Following that, we still adopt the non-intrusive way to compute the matrix $\mathbf{B}_j(\theta)$ in
 282 Eq. (29) and assemble the element stochastic stiffness matrix in Eq. (23), which can be performed
 283 by using the deterministic FEM procedure to loop random samples $\mathbf{x}(\theta^{(i)})$, $i = 1, \dots, n_s$. It is noted
 284 that Eq. (23) and Eq. (29) are only applicable to the elastic equation considered in this section, and
 285 other assembly formats should be adopted for more general problems.

286 It is seen from Eq. (23) that if the material parameters are not spatially dependent, the material
 287 matrix \mathbf{C} is independent of the random coordinate $\mathbf{x}(\theta)$. **If the random material matrix $\mathbf{C}(\theta)$ is**
 288 **involved in the problem**, the same computational framework as described above can be employed
 289 to assemble the stiffness matrix. In the numerical example 6.2, we will show that the proposed
 290 method can be applied to the stochastic PDE defined on the random geometry, the random co-
 291 efficients of which are simulated as random variables. However, since simulating random fields
 292 defined on random domains is still an open problem, PDEs that couple random geometries and
 293 random coefficient fields require further study. Also, although we only consider linear PDEs in

294 this paper, the proposed method is readily extended to nonlinear PDEs. For instance, we can
 295 describe inelastic behavior by adopting a nonlinear material matrix \mathbf{C} in Eq. (23), which is still in-
 296 dependent of the random coordinate and can be assembled in the same way as deterministic FEM.
 297 [The applicability of the proposed method to nonlinear PDEs on random domains will be verified](#)
 298 [with the aid of a nonlinear heat equation on a random domain in Section 6.4.](#)

299 4. Solution algorithm of stochastic finite element equations

300 Based on the above assembly of the stochastic matrix and the stochastic vector, we can obtain
 301 a stochastic finite element equation (SFEE)

$$\mathbf{K}(\theta) \mathbf{u}(\theta) = \mathbf{F}(\theta). \quad (30)$$

302 It is noted that the stochastic matrix $\mathbf{K}(\theta)$ and the stochastic vector $\mathbf{F}(\theta)$ are assembled in a non-
 303 intrusive way. By repeating the realizations $\mathbf{K}(\theta^{(i)}) \in \mathbb{R}^{n \times n}$ and $\mathbf{F}(\theta^{(i)}) \in \mathbb{R}^n$ for n_s different
 304 samples, they usually have the forms $\mathbf{K}(\theta) \in \mathbb{R}^{n \times n \times n_s}$ and $\mathbf{F}(\theta) \in \mathbb{R}^{n \times n_s}$. Much memory is needed
 305 to store $\mathbf{K}(\theta)$ and $\mathbf{F}(\theta)$ if the sample size n_s is large. However, if a small size n_s is adopted, the
 306 accuracy of the stochastic solution will be low. To avoid the difficulties, we adopt a weak-intrusive
 307 SFEM [37, 38] to solve Eq. (30) in this paper.

308 4.1. A weak-intrusive SFEM

309 To solve Eq. (30), we approximate the stochastic solution $\mathbf{u}(\theta)$ in the form,

$$\mathbf{u}(\theta) \approx \sum_{i=1}^k \lambda_i(\theta) \mathbf{d}_i = \mathbf{D}\boldsymbol{\Lambda}(\theta), \quad (31)$$

310 where $\lambda_i(\theta) \in \mathbb{R}$ denotes a scalar random variables, $\boldsymbol{\Lambda}(\theta) = [\lambda_1(\theta), \dots, \lambda_k(\theta)]^T \in \mathbb{R}^k$ is a random
 311 variable vector, $\mathbf{d}_i \in \mathbb{R}^n$ denotes a deterministic vector and $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_k] \in \mathbb{R}^{n \times k}$ is a matrix.
 312 All of these terms are not known a priori and need to be solved. As it is not easy to compute them
 313 at once, we adopt a sequential way to solve the couple $\{\lambda_i(\theta), \mathbf{d}_i\}_i$ one by one. For this purpose,
 314 we assume that the first $k - 1$ couples $\{\lambda_i(\theta), \mathbf{d}_i\}_{i=1}^{k-1}$ have been determined and Eq. (30) is thus
 315 transformed into

$$\mathbf{K}(\theta) \lambda_k(\theta) \mathbf{d}_k = \mathbf{F}_k(\theta), \quad (32)$$

316 where $\mathbf{F}_k(\theta) = \mathbf{F}(\theta) - \mathbf{K}(\theta) \sum_{i=1}^{k-1} \lambda_i(\theta) \mathbf{d}_i$. In this way, only $\lambda_k(\theta)$ and \mathbf{d}_k are unknown. They are
 317 solved via an alternating iterative algorithm as follows.

318 For a known random variable $\lambda_k(\theta)$ (or a given initial value, the details of which can be found
 319 in Section 5), a deterministic finite element equation is obtained by use of the stochastic Galerkin
 320 method [18, 25],

$$\mathbb{E} \left\{ \lambda_k^2(\theta) \mathbf{K}(\theta) \right\} \mathbf{d}_k = \mathbb{E} \left\{ \lambda_k(\theta) \mathbf{F}_k(\theta) \right\}, \quad (33)$$

321 which can be solved efficiently via existing FEM solvers [35]. To speed up the convergence, we
 322 let the vector \mathbf{d}_k orthogonal to the obtained vectors $\{\mathbf{d}_i\}_{i=1}^{k-1}$. The Gram-Schmidt orthogonalization
 323 is adopted,

$$\mathbf{d}_k = \mathbf{d}_k - \sum_{i=1}^{k-1} \frac{\mathbf{d}_k^T \mathbf{d}_i}{\mathbf{d}_i^T \mathbf{d}_i} \mathbf{d}_i, \quad (34)$$

324 where $\{\mathbf{d}_i\}_{i=1}^{k-1}$ are normalized orthogonal vectors that meet $\mathbf{d}_i^T \mathbf{d}_j = \delta_{ij}$, where δ_{ij} is the Kronecker
 325 delta. Based on the solution \mathbf{d}_k of Eq. (33), the random variable $\lambda_k(\theta)$ is updated via the Galerkin
 326 approach,

$$\left[\mathbf{d}_k^T \mathbf{K}(\theta) \mathbf{d}_k \right] \lambda_k(\theta) = \mathbf{d}_k^T \mathbf{F}_k(\theta). \quad (35)$$

327 In this paper we assume that the matrix $\mathbf{K}(\theta)$ is positive (or negative) definite, i.e. $\mathbf{x}^T \mathbf{K}(\theta) \mathbf{x} >$
 328 $0 (< 0)$ holds for the nonzero vector $\mathbf{x} \in \mathbb{R}^n$, which holds true in many problems. For the indefinite
 329 matrix $\mathbf{K}(\theta)$, Eq. (35) is insolvable when the realization $\mathbf{d}_k^T \mathbf{K}(\theta^{(i)}) \mathbf{d}_k = 0$. The residual minimiza-
 330 tion can be used to build a numerically stable equation $\left[\mathbf{d}_k^T \mathbf{K}(\theta)^T \mathbf{K}(\theta) \mathbf{d}_k \right] \lambda_k(\theta) = \mathbf{d}_k^T \mathbf{K}(\theta)^T \mathbf{F}_k(\theta)$,
 331 the details of which will not be discussed in this paper. In order to solve the stochastic algebraic
 332 equation (35) efficiently, we adopt a non-intrusive way [37, 38], which is easily implemented and
 333 can be applied to high-dimensional stochastic problems. Specifically, if the matrix $\mathbf{K}(\theta)$ is not an
 334 indefinite matrix, Eq. (35) is solved by

$$\lambda_k(\theta) = \frac{\mathbf{d}_k^T \mathbf{F}_k(\theta)}{\mathbf{d}_k^T \mathbf{K}(\theta) \mathbf{d}_k} \in \mathbb{R}^{n_s}, \quad (36)$$

335 where $\theta = \left\{ \theta^{(i)} \right\}_{i=1}^{n_s} \in \mathbb{R}^{n_s}$ represents n_s sample realizations of the random variables, $\lambda_k(\theta) \in \mathbb{R}^{n_s}$
 336 is the sample vector of the random variable $\lambda_k(\theta)$ and statistical methods are used to provide the
 337 probability characteristics of $\lambda_k(\theta)$. Eq. (36) is insensitive to stochastic dimensions and has low

338 computational costs even for very high-dimensional stochastic problems. In this way, Eq. (31)
 339 is considered a weak-intrusive approximation, which combines the high efficiency of intrusive
 340 methods and the weakly dimensional dependence of non-intrusive methods.

341 A set of couples $\{\lambda_i(\theta), \mathbf{d}_i\}_{i=1}^k$ can be obtained by the above iteration and a specific criterion
 342 for the selection of the size k will be discussed in Section 5. It is noted that these couples are
 343 solved in a sequential way, thus they do not exactly fulfill Eq. (30). The stochastic solution $\mathbf{u}(\theta)$
 344 approximated by the couples $\{\lambda_i(\theta), \mathbf{d}_i\}_{i=1}^k$ may have poor accuracy in some cases. To improve the
 345 accuracy, we consider $\{\mathbf{d}_i\}_{i=1}^k$ as reduced bases and recompute the random vector $\Lambda(\theta)$ in Eq. (31)
 346 by

$$[\mathbf{D}^T \mathbf{K}(\theta) \mathbf{D}] \Lambda(\theta) = \mathbf{D}^T \mathbf{F}(\theta), \quad (37)$$

347 where the reduced-order matrix $\mathbf{D}^T \mathbf{K}(\theta) \mathbf{D} \in \mathbb{R}^{k \times k}$ and the reduced-order vector $\mathbf{D}^T \mathbf{F}(\theta) \in \mathbb{R}^k$. The
 348 computational effort is very cheap due to the small size of Eq. (37). The final stochastic solution
 349 is obtained by n_s solutions $\Lambda(\theta^{(i)})$, $i = 1, \dots, n_s$ of Eq. (37).

350 We mention that although similar expansions have been widely used in the computational
 351 framework of PGD methods [39, 40], the proposed approximation Eq. (31) is more powerful
 352 than the classical PGD methods and is more suitable for solving high-dimensional and nonlinear
 353 stochastic problems due to its weak intrusiveness. With the aid of numerical examples, we will
 354 show that it is also applicable to solve Eq. (30) with a small sample size n_s .

355 4.2. Computational aspects

356 In this section we will discuss the numerical details of the implementation of the proposed
 357 method, including the implementations at the global and element levels.

358 4.2.1. Implementation at the global level

359 We recall that the stochastic matrix $\mathbf{K}(\theta) \in \mathbb{R}^{n \times n \times n_s}$ and the stochastic vector $\mathbf{F}(\theta) \in \mathbb{R}^{n \times n_s}$ are
 360 obtained in a non-intrusive way and the sample vector of the random variable $\lambda_k(\theta)$ are given by
 361 $\lambda_k(\theta) \in \mathbb{R}^{n_s}$. The computations of Eq. (33) and Eq. (36) can be executed at the global level via

$$\mathbb{E} \left\{ \lambda_k^2(\boldsymbol{\theta}) \mathbf{K}(\boldsymbol{\theta}) \right\} = \frac{1}{n_s} \sum_{j=1}^{n_s} \lambda_k^2(\boldsymbol{\theta}^{(j)}) \mathbf{K}(\boldsymbol{\theta}^{(j)}) \in \mathbb{R}^{n \times n}, \quad (38)$$

$$\mathbb{E} \left\{ \lambda_k(\boldsymbol{\theta}) \mathbf{F}(\boldsymbol{\theta}) \right\} = \frac{1}{n_s} \mathbf{F}(\boldsymbol{\theta}) \lambda_k(\boldsymbol{\theta}) \in \mathbb{R}^n, \quad (39)$$

$$\mathbf{d}_k^T \mathbf{K}(\boldsymbol{\theta}) \mathbf{d}_k = \left[\mathbf{d}_k^T \mathbf{K}(\boldsymbol{\theta}^{(1)}) \mathbf{d}_k, \dots, \mathbf{d}_k^T \mathbf{K}(\boldsymbol{\theta}^{(n_s)}) \mathbf{d}_k \right]^T \in \mathbb{R}^{n_s}, \quad (40)$$

$$\mathbf{d}_k^T \mathbf{F}(\boldsymbol{\theta}) = \mathbf{F}(\boldsymbol{\theta})^T \mathbf{d}_k \in \mathbb{R}^{n_s}, \quad (41)$$

362 which provides a direct way to perform the stochastic computations. It is seen that Eq. (39) and
 363 Eq. (41) can be computed efficiently. However, Eq. (38) and Eq. (40) may need much more
 364 computational effort and storage memory. In practice, we store $\mathbf{K}(\boldsymbol{\theta}) \in \mathbb{R}^{n \times n \times n_s}$ in a sparse tensor
 365 structure and adopt efficient tensor multiplications to execute the computations.

366 4.2.2. Implementation at the element level

367 To avoid the large storage memory required for the above implementation, we implement the
 368 stochastic computations at the element level and deterministic assemblies are then executed, which
 369 corresponds

$$\mathbb{E} \left\{ \lambda_k^2(\boldsymbol{\theta}) \mathbf{K}(\boldsymbol{\theta}) \right\} = \bigcup_{i=1}^{n_e} \left(\mathbb{E} \left\{ \lambda_k^2(\boldsymbol{\theta}) \mathbf{k}^{(i)}(\boldsymbol{\theta}) \right\} \right) = \bigcup_{i=1}^{n_e} \left(\frac{1}{n_s} \sum_{j=1}^{n_s} \lambda_k^2(\boldsymbol{\theta}^{(j)}) \mathbf{k}^{(i)}(\boldsymbol{\theta}^{(j)}) \right) \in \mathbb{R}^{n \times n}, \quad (42)$$

$$\mathbb{E} \left\{ \lambda_k(\boldsymbol{\theta}) \mathbf{F}(\boldsymbol{\theta}) \right\} = \bigcup_{i=1}^{n_e} \left(\mathbb{E} \left\{ \lambda_k(\boldsymbol{\theta}) \mathbf{f}^{(i)}(\boldsymbol{\theta}) \right\} \right) = \bigcup_{i=1}^{n_e} \left(\frac{1}{n_s} \mathbf{f}^{(i)}(\boldsymbol{\theta}) \lambda_k(\boldsymbol{\theta}) \right) \in \mathbb{R}^n, \quad (43)$$

$$\mathbf{d}_k^T \mathbf{K}(\boldsymbol{\theta}) \mathbf{d}_k = \left[\sum_{i=1}^{n_e} \mathbf{d}_k^{(i)T} \mathbf{k}^{(i)}(\boldsymbol{\theta}^{(1)}) \mathbf{d}_k^{(i)}, \dots, \sum_{i=1}^{n_e} \mathbf{d}_k^{(i)T} \mathbf{k}^{(i)}(\boldsymbol{\theta}^{(n_s)}) \mathbf{d}_k^{(i)} \right]^T \in \mathbb{R}^{n_s}, \quad (44)$$

$$\mathbf{d}_k^T \mathbf{F}(\boldsymbol{\theta}) = \sum_{i=1}^{n_e} \mathbf{f}^{(i)}(\boldsymbol{\theta})^T \mathbf{d}_k^{(i)} \in \mathbb{R}^{n_s}, \quad (45)$$

370 where the element matrix and vector are $\mathbf{k}^{(i)}(\boldsymbol{\theta}) \in \mathbb{R}^{n_1 \times n_1 \times n_s}$ and $\mathbf{f}^{(i)}(\boldsymbol{\theta}) \in \mathbb{R}^{n_1 \times n_s}$. In this way,
 371 only the element matrix and vector are stored for each iteration, which reduces the storage mem-
 372 ory. However, we need to reassemble the element matrix and vector for each iteration and more
 373 computational effort is needed for the assemblies compared to the implementation at the global
 374 level.

375 5. Algorithm implementation

376 The above iterative algorithm for solving PDEs defined on random domains is summarized in
 377 Algorithm 2, which consists of two-loop procedures. The inner loop is from step 7 to 24 and is
 378 used to compute the couple $\{\lambda_k(\theta), \mathbf{d}_k\}$. To execute the inner loop, random samples $\lambda_k^{(0)}(\theta) \in \mathbb{R}^{n_s}$ are
 379 initialized in step 6. All nonzero vectors of size n_s can be used as the initial random samples and it
 380 has little influence on the computational accuracy and efficiency of the proposed method. With the
 381 initial random samples, the deterministic vector $\mathbf{d}_k^{(j)}$ is computed in step 14 by solving linear finite
 382 element equations. By using the Gram-Schmidt orthogonalization in step 15, $\mathbf{d}_k^{(j)}$ is orthogonalized
 383 and normalized along the whole iteration. With the obtained vector $\mathbf{d}_k^{(j)}$, the random variable $\lambda_k^{(j)}(\theta)$
 384 is calculated in the form of random samples $\lambda_k^{(j)}(\theta) \in \mathbb{R}^{n_s}$ in step 22. Two numerical strategies can
 385 be adopted to implement the above iteration, i.e. the implementation at the global level via steps
 386 3, 9, 17 and the implementation at the local level via steps 11, 12, 19, 20. After the inner loop, the
 387 stochastic solution $\mathbf{u}_k(\theta)$ in step 25 of the outer loop is approximated recursively to meet Eq. (30).
 388 Following that, based on the known matrix \mathbf{D} , the random vector $\mathbf{\Lambda}(\theta)$ is recalculated by solving
 389 n_s k -dimensional linear stochastic finite element equations. It is noted that we can adopt different
 390 sample sizes in step 6 and step 28. In practice, a small size n_s is first used in step 6 and a large
 391 sample size is adopted to recompute the random vector $\mathbf{\Lambda}(\theta)$, which saves a lot of computational
 392 costs but still has good accuracy. The performance of this strategy will be illustrated in numerical
 393 examples.

394 Also, two iterative criteria in Algorithm 2 are used to check the convergence, i.e. $\varepsilon_{\mathbf{d},j}$ in step
 395 23 and $\varepsilon_{\mathbf{u},k}$ in step 26. The locally iterative error $\varepsilon_{\mathbf{d},j}$ is defined as

$$\varepsilon_{\mathbf{d},j} = \frac{\|\mathbf{d}_k^{(j)} - \mathbf{d}_k^{(j-1)}\|}{\|\mathbf{d}_k^{(j)}\|} = 2 - 2\mathbf{d}_k^{(j)\top} \mathbf{d}_k^{(j-1)}, \quad (46)$$

396 where the operator $\|\square\| = \mathbb{E}\{\square^\top \square\}$. It measures the difference between the vectors $\mathbf{d}_k^{(j)}$ and $\mathbf{d}_k^{(j-1)}$
 397 and the calculation is stopped when $\mathbf{d}_k^{(j)}$ is almost the same as $\mathbf{d}_k^{(j-1)}$. Similarly, the globally iterative
 398 error $\varepsilon_{\mathbf{u},k}$ is defined as

$$\varepsilon_{\mathbf{u},k} = \frac{\|\mathbf{u}_k(\theta) - \mathbf{u}_{k-1}(\theta)\|}{\|\mathbf{u}_k(\theta)\|} = \frac{\mathbb{E}\{\lambda_k^2(\theta)\} \mathbf{d}_k^\top \mathbf{d}_k}{\sum_{i,j=1}^k \mathbb{E}\{\lambda_i(\theta) \lambda_j(\theta)\} \mathbf{d}_i^\top \mathbf{d}_j} = \frac{\mathbb{E}\{\lambda_k^2(\theta)\}}{\sum_{i=1}^k \mathbb{E}\{\lambda_i^2(\theta)\}}, \quad (47)$$

Algorithm 2 Algorithm for solving PDEs defined on random domains

- 1: Generate the mesh topology \mathcal{K} {Node, Ele} of the reference domain and solve the random coordinate $\mathbf{x}(\theta)$ via Algorithm 1
 - 2: **if** *Implementation at the global level* **then**
 - 3: Assemble the tensor $\mathbf{K}(\theta) \in \mathbb{R}^{n \times n \times n_s}$ and the matrix $\mathbf{F}(\theta) \in \mathbb{R}^{n \times n_s}$
 - 4: **end**
 - 5: **while** $\varepsilon_{\mathbf{u},k} > \varepsilon_{\mathbf{u}}$ **do**
 - 6: Initialize random samples $\lambda_k^{(0)}(\theta) = \{\lambda_k^{(0)}(\theta^{(i)})\}_{i=1}^{n_s} \in \mathbb{R}^{n_s}$
 - 7: **while** $\varepsilon_{\mathbf{d},j} > \varepsilon_{\mathbf{d}}$ **do**
 - 8: **if** *Implementation at the global level* **then**
 - 9: Compute $\mathbb{E}\{\lambda_k^2(\theta) \mathbf{K}(\theta)\}$ and $\mathbb{E}\{\lambda_k(\theta) \mathbf{F}_k(\theta)\}$ by Eq. (38) and Eq. (39)
 - 10: **else if** *Implementation at the local level* **then**
 - 11: Assemble the tensor $\mathbf{k}^{(i)}(\theta) \in \mathbb{R}^{n_l \times n_l \times n_s}$ and the matrix $\mathbf{f}^{(i)}(\theta) \in \mathbb{R}^{n_l \times n_s}$
 - 12: Compute $\mathbb{E}\{\lambda_k^2(\theta) \mathbf{K}(\theta)\}$ and $\mathbb{E}\{\lambda_k(\theta) \mathbf{F}_k(\theta)\}$ by Eq. (42) and Eq. (43)
 - 13: **end**
 - 14: Compute the deterministic vector $\mathbf{d}_k^{(j)}$ via Eq. (33)
 - 15: Orthogonalize $\mathbf{d}_k^{(j)} \perp \{\mathbf{d}_i\}_{i=1}^{k-1}$ and normalize $\|\mathbf{d}_k^{(j)}\| = 1$
 - 16: **if** *Implementation at the global level* **then**
 - 17: Compute $\mathbf{d}_k^T \mathbf{K}(\theta) \mathbf{d}_k$ and $\mathbf{d}_k^T \mathbf{F}_k(\theta)$ by Eq. (40) and Eq. (41)
 - 18: **else if** *Implementation at the local level* **then**
 - 19: Assemble the tensor $\mathbf{k}^{(i)}(\theta) \in \mathbb{R}^{n_l \times n_l \times n_s}$ and the matrix $\mathbf{f}^{(i)}(\theta) \in \mathbb{R}^{n_l \times n_s}$
 - 20: Compute $\mathbf{d}_k^T \mathbf{K}(\theta) \mathbf{d}_k$ and $\mathbf{d}_k^T \mathbf{F}_k(\theta)$ by Eq. (44) and Eq. (45)
 - 21: **end**
 - 22: Update $\lambda_k^{(j)}(\theta) \in \mathbb{R}^{n_s}$ via Eq. (36)
 - 23: Compute the locally iterative error $\varepsilon_{\mathbf{d},j}$
 - 24: **end**
 - 25: Update the stochastic solution $\mathbf{u}_k(\theta) = \mathbf{u}_{k-1}(\theta) + \lambda_k(\theta) \mathbf{d}_k$
 - 26: Compute the globally iterative error $\varepsilon_{\mathbf{u},k}$
 - 27: **end**
 - 28: Recompute the random vector $\mathbf{\Lambda}(\theta) \in \mathbb{R}^k$ via Eq. (37)
-

399 which measures the contribution of the k -th couple $\{\lambda_k(\theta), \mathbf{d}_k\}$ to the stochastic solution $\mathbf{u}_k(\theta)$.
400 However, since the random variables $\{\lambda_i(\theta)\}$ are calculated in a sequential way and the value of
401 $\mathbb{E}\{\lambda_k^2(\theta)\}$ does not keep decreasing in some cases [38], Eq. (47) may be not a good convergence
402 checker in practice. To avoid this issue, we adopt a new error indicator by modifying the random
403 variables $\{\lambda_i(\theta)\}$ in Eq. (47). To this end, we consider the autocorrelation function of the random
404 vector $\mathbf{\Lambda}(\theta)$

$$\mathbf{C}_{\mathbf{\Lambda}\mathbf{\Lambda}} = \mathbb{E}\{\mathbf{\Lambda}(\theta)\mathbf{\Lambda}(\theta)^T\}, \quad (48)$$

405 which is decomposed into

$$\mathbf{C}_{\mathbf{\Lambda}\mathbf{\Lambda}} = \mathbf{Q}\mathbf{Z}\mathbf{Q}^T \quad (49)$$

406 by the eigendecomposition, where $\mathbf{Q} \in \mathbb{R}^{k \times k}$ is an orthonormal matrix and \mathbf{Z} is a diagonal matrix.

407 We rewrite the stochastic solution $\mathbf{u}(\theta)$ in Eq. (31) as

$$\mathbf{u}(\theta) = \mathbf{D}\mathbf{Q}\mathbf{Q}^T\mathbf{\Lambda}(\theta) \quad (50)$$

408 and let a new random vector $\tilde{\mathbf{\Lambda}}(\theta) = \mathbf{Q}^T\mathbf{\Lambda}(\theta) = [\tilde{\lambda}_1(\theta), \dots, \tilde{\lambda}_k(\theta)]^T \in \mathbb{R}^k$, the autocorrelation
409 function of which is given by

$$\tilde{\mathbf{C}}_{\tilde{\mathbf{\Lambda}}\tilde{\mathbf{\Lambda}}} = \mathbb{E}\{\tilde{\mathbf{\Lambda}}(\theta)\tilde{\mathbf{\Lambda}}(\theta)^T\} = \mathbf{Q}^T\mathbb{E}\{\mathbf{\Lambda}(\theta)\mathbf{\Lambda}(\theta)^T\}\mathbf{Q} = \mathbf{Z}. \quad (51)$$

410 To improve Eq. (47), we replace the random variables $\{\lambda_i(\theta)\}$ with the new random variables
411 $\{\tilde{\lambda}_i(\theta)\}$ and the iterative error $\varepsilon_{\mathbf{u},k}$ thus becomes

$$\varepsilon_{\mathbf{u},k} = \frac{\mathbb{E}\{\tilde{\lambda}_k^2(\theta)\}}{\sum_{i=1}^k \mathbb{E}\{\tilde{\lambda}_i^2(\theta)\}} = \frac{\mathbf{Z}_k}{\text{Tr}(\mathbf{Z})}, \quad (52)$$

412 where $\text{Tr}(\cdot)$ is the trace operator and \mathbf{Z}_k is the element at position (k, k) of the matrix \mathbf{Z} . In this way,
413 the iterative error $\varepsilon_{\mathbf{u},k}$ keeps decreasing as the retained item k increases. It is noted that Eq. (50)
414 does not improve the accuracy of the stochastic solution and just provides a new representation.

415 6. Numerical examples

416 We test the proposed method with the aid of four numerical examples. The convergence errors
417 are set as $\varepsilon_{\mathbf{d}} = 1 \times 10^{-3}$ and $\varepsilon_{\mathbf{u}} = 1 \times 10^{-8}$ in Algorithm 2. All tests are performed on a laptop

418 (dual-core, Intel Core i7, 2.40GHz). Without loss of generality, we ignore the dimension of the
 419 physical quantities and execute the dimensionless analysis in the first two examples.

420 6.1. Example 1: elastic equation defined on a random domain

421 6.1.1. Problem setting

422 In this example, we consider a two-dimensional elastic problem as discussed in Eq. (20), which
 423 is defined on the random domain shown in Fig. 2. The outer bound of the domain is a deterministic
 424 square of length 2. The inner boundary $\Gamma_r(\theta)$ of the domain is a hole described by a random
 425 elliptic curve that is controlled by four mutually independent random variables $x_c(\theta)$, $y_c(\theta)$, $l_x(\theta)$,
 426 $l_y(\theta)$. As depicted in Fig. 2, the location $(x_c(\theta), y_c(\theta))$ of the center point of the ellipse are two
 427 uniformly distributed random variables on $[-0.2, 0.2]$ and the major and minor (or minor and
 428 major) axes $l_x(\theta)$ and $l_y(\theta)$ are two uniformly distributed random variables on $[0.9, 1.1]$. The
 429 boundary conditions are given by the vertical force $f(\bar{x}, \bar{y}) = -1$ on the upper boundary Γ_N and
 430 the Dirichlet condition $u(\bar{x}, \bar{y}) = 0$ on the lower boundary Γ_D . The Young's modulus and the
 431 Poisson's ratio are 2.10×10^8 and 0.3, respectively.

432 We choose the mean value of the random boundary $\Gamma_r(\theta)$ as the inner boundary of the reference
 433 domain. As shown in Fig. 3a, the reference domain has the same outer boundary as the random
 434 domain and its inner boundary is a circle with the center $(0, 0)$ and the diameter 1. The finite ele-

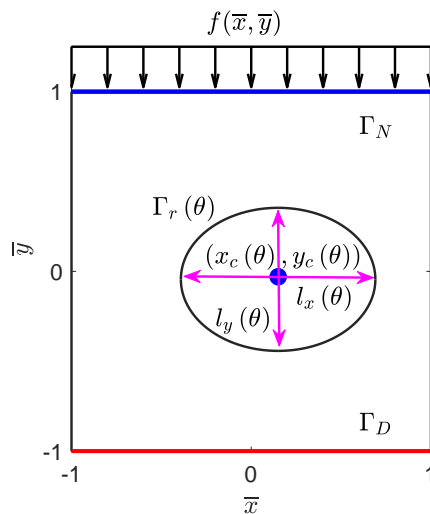
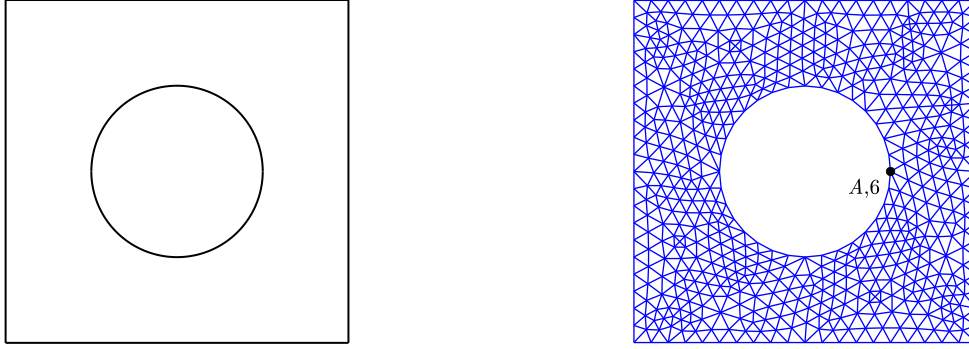


Figure 2: The domain with a random inner boundary.



(a) The reference domain.

(b) Finite element mesh of the reference domain.

Figure 3: The reference domain and its finite element mesh.

435 ment mesh of the reference domain is shown in Fig. 3b, and 624 nodes and 1136 linear triangular
 436 elements are included. Following that, we construct the mapping between the reference and the
 437 random domains based on the mesh of the reference domain. The random boundary $\Gamma_r(\theta)$ can be
 438 represented by random scaling and shift transformations of the circle in the reference domain

$$\begin{bmatrix} x(\theta) \\ y(\theta) \end{bmatrix} = \mathbf{D}(\theta) \mathbf{S}(\theta) \begin{bmatrix} \bar{x} \\ \bar{y} \\ 1 \end{bmatrix} \quad \text{on } \Gamma_r(\theta), \quad (53)$$

439 where the random shift transformation matrix $\mathbf{D}(\theta)$ and the random scaling transformation matrix
 440 $\mathbf{S}(\theta)$ are given by

$$\mathbf{D}(\theta) = \begin{bmatrix} 1 & 0 & x_c(\theta) \\ 0 & 1 & y_c(\theta) \end{bmatrix} \in \mathbb{R}^{2 \times 3}, \quad \mathbf{S}(\theta) = \begin{bmatrix} l_x(\theta) & 0 & 0 \\ 0 & l_y(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3 \times 3}. \quad (54)$$

441 6.1.2. Numerical results

442 By using Algorithm 1, we can solve the random nodal coordinates of the random mesh of the
 443 random domain. To solve the displacement $\mathbf{u}(\theta)$ in Eq. (20), we set the sample size as $n_{s,1} = 40$
 444 in step 6 in Algorithm 2. After obtaining the reduced-order matrix \mathbf{D} , we reset the sample size as
 445 $n_{s,2} = 1 \times 10^4$ in step 28 in Algorithm 2. It is noted that the choice of the sample size in step 6 is
 446 still an open problem. In this paper we adopt the sample size $n_{s,1} = 10r$, where r is the number of

447 random variables. A large sample size is suggested if storage requirements are ignored.

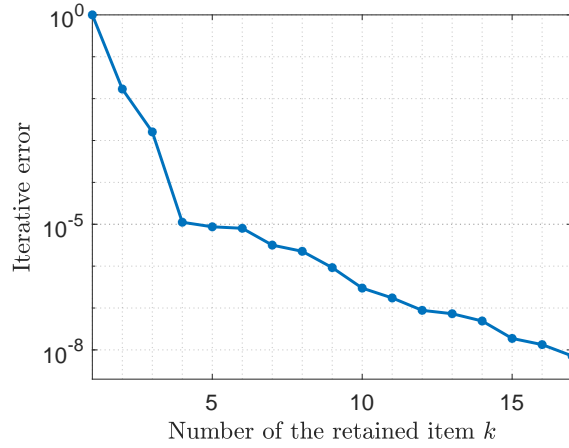


Figure 4: Iterative errors of different numbers of the retained item k calculated by Eq. (52).

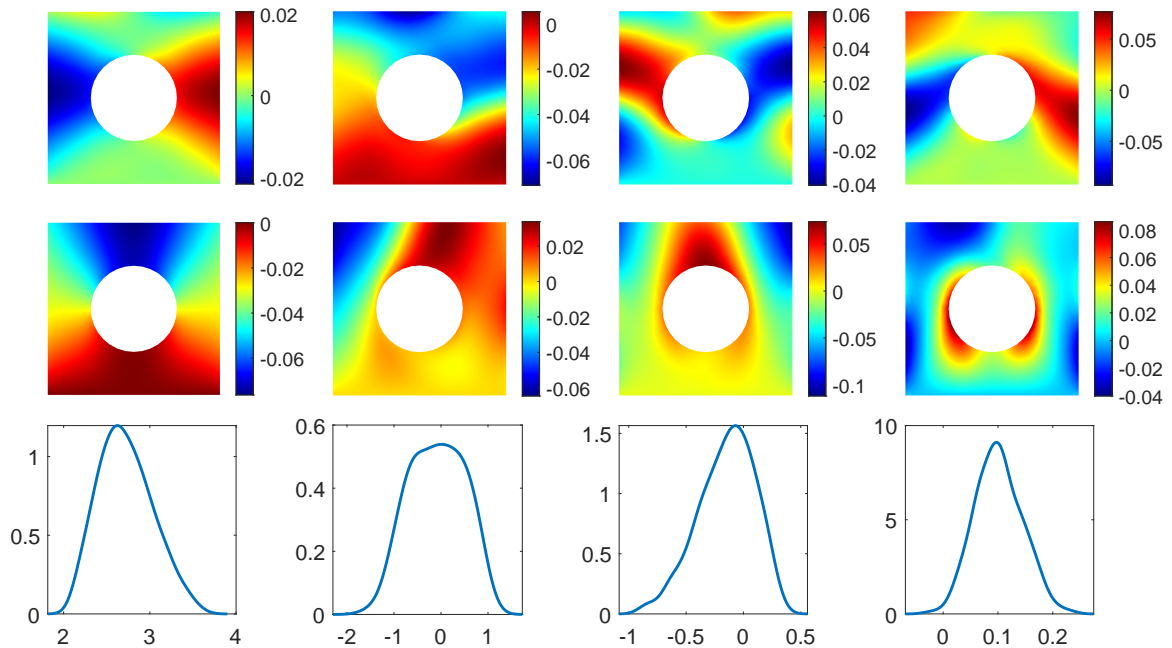


Figure 5: Solutions of the components: the first four solutions $\{\mathbf{d}_i\}_{i=1}^4$ in the \bar{x} direction (the first row), the first four solutions $\{\mathbf{d}_i\}_{i=1}^4$ in the \bar{y} direction (the second row) and PDFs of the first four random variables $\{\lambda_i(\theta)\}_{i=1}^4$ (the third row).

448 Iterative errors $\varepsilon_{\mathbf{u},k}$ in step 26 in Algorithm 2 calculated using Eq. (52) are shown in Fig. 4
 449 and retaining 17 terms achieves the specified accuracy, which demonstrates the good convergence

450 of the proposed method. It is seen that the iterative errors keep decreasing as the retained items
451 increase, which verifies the effectiveness of the proposed error indicator Eq. (52). First four com-
452 ponents of the deterministic vectors $\{\mathbf{d}_i\}_{i=1}^{17}$ are depicted in Fig. 5, where $\{\mathbf{d}_i\}_{i=1}^4$ in the \bar{x} direction
453 and $\{\mathbf{d}_i\}_{i=1}^4$ in the \bar{y} direction are seen from the first and second rows of Fig. 5, respectively. PDFs
454 of the first four components of the recomputed random variables $\{\lambda_i(\theta)\}_{i=1}^{17}$ are shown in the third
455 row of Fig. 5, which are obtained by 1×10^4 random samples. It is noted that the j -th value $\mathbf{d}_{i,j}$
456 of the vector \mathbf{d}_i is the solution of the j -th node whose coordinate is given by $(x_j(\theta), y_j(\theta))$. It is
457 not easy to show a vector on random coordinates, thus the vector \mathbf{d}_i is described in the node-index
458 coordinate system instead of the Cartesian coordinate system in this paper. Although we show the
459 vector \mathbf{d}_i on the reference domain, they can be depicted on all possible domains.

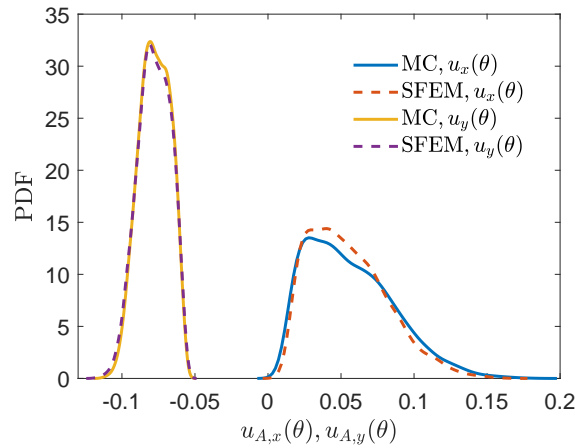


Figure 6: PDFs of stochastic displacements of the point A in the \bar{x} and \bar{y} directions obtained by the proposed method and 1×10^4 MCS.

460 To test the accuracy of the proposed method, we compare PDFs of stochastic displacements of
461 the point A (its node number is 6 as shown in Fig. 3b) obtained by the proposed method and 1×10^4
462 MCS. As shown in Fig. 6, PDFs of the stochastic displacements in both \bar{x} and \bar{y} directions have
463 good agreements with MCS, which indicates that the proposed method has comparable accuracy
464 to MCS. It is seen from Fig. 6 that compared to that in the \bar{y} direction, the PDF of the stochastic
465 displacement in the \bar{x} direction is slightly less accurate, but its accuracy is still acceptable in most
466 problems. If a more accurate stochastic solution is required in some cases, we can retain more

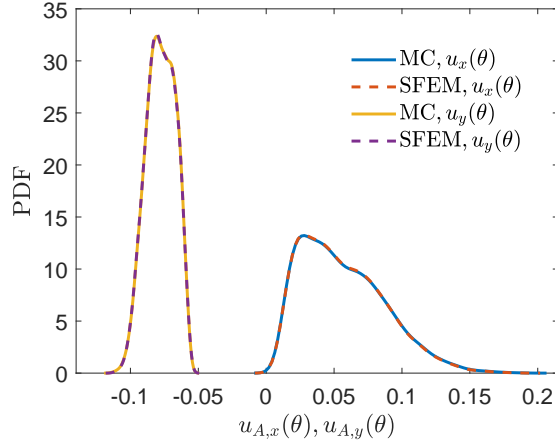


Figure 7: PDFs of stochastic displacements of the point A in the \bar{x} and \bar{y} directions obtained by the proposed method retaining 50 terms and 1×10^4 MCS.

467 terms for the purpose. PDFs of the stochastic displacements of the point A in \bar{x} and \bar{y} directions
468 obtained by the proposed method retaining 50 terms are depicted in Fig. 7. They have much higher
469 accuracy than that in Fig. 6 and reach very good agreements with MCS. Furthermore, we test the
470 computational efficiency of the proposed method. Computational costs of different implementa-
471 tions are listed in Tab. 1, where GI $(n_{s,1}, n_{s,2})$ and LI $(n_{s,1}, n_{s,2})$ represent the implementation at
472 global and local levels with sample sizes $n_{s,1}, n_{s,2}$ in steps 6 and 28 in Algorithm 2. The solving
473 cost and the recomputing cost are the computational cost from step 5 to 27 and the computational
474 cost of step 28 in Algorithm 2, respectively. Total computational times of three implementations
475 are much less than the cost of 1×10^4 MCS, which demonstrates the high efficiency of the proposed
476 method. It is noted that stochastic solutions obtained by implementations at global and local levels
477 have the same accuracy and only the matrix and the vector are formed differently. Compared to GI
478 $(40, 1 \times 10^4)$, LI $(40, 1 \times 10^4)$ needs a bit more solving costs since more assemblies are performed
479 in steps 9 and 19 in Algorithm 2. As a comparison, we test the case LI $(1 \times 10^3, 1 \times 10^4)$, i.e.
480 $n_{s,1} = 1 \times 10^3$ samples are adopted in step 9. A solution of similar accuracy to the case LI $(40,$
481 $1 \times 10^4)$ is obtained, which indicates that $n_{s,1} = 40$ can reach good accuracy in this example. But
482 the case LI $(1 \times 10^3, 1 \times 10^4)$ requires more costs for sample assemblies. The recomputing costs
483 of the three implementations are close since they have the same number of retained items and the

484 size of the reduced-order stochastic finite element equation (37) is fixed.

Table 1: Computational costs of different implementations.

Methods	GI (40, 1×10^4)	LI (40, 1×10^4)	LI (1×10^3 , 1×10^4)	MCS (1×10^4)
Solving costs	324.49	361.74	427.31	
Recomputing costs	28.02	27.53	30.67	
Total costs (second)	352.51	389.27	457.98	2519.66

485 6.1.3. Postprocessing of the stochastic solution

486 As discussed above, the vector \mathbf{d}_i is depicted on the node-index coordinates, which is different
 487 from the classical FEM. Thus we need to pay extra attention to the postprocessing of stochastic
 488 solutions. In practice, to perform the postprocessing of stochastic solutions, we need to combine
 489 each realization of stochastic solutions and the corresponding realization of random meshes. For
 490 an explanation of this point, let us consider the postprocessing of the stochastic solution under the
 491 sample realization $[x_c(\theta^*), y_c(\theta^*), l_x(\theta^*), l_y(\theta^*)] = [0.1215, -0.1564, 1.0876, 0.9390]$. As shown
 492 in Fig. 8 (left), the random mesh is obtained via combining the mesh topology of the reference
 493 domain and the random coordinates, where the green part is the reference domain. The realization
 494 of the stochastic solution in \bar{x} and \bar{y} directions are then depicted on the random mesh shown in

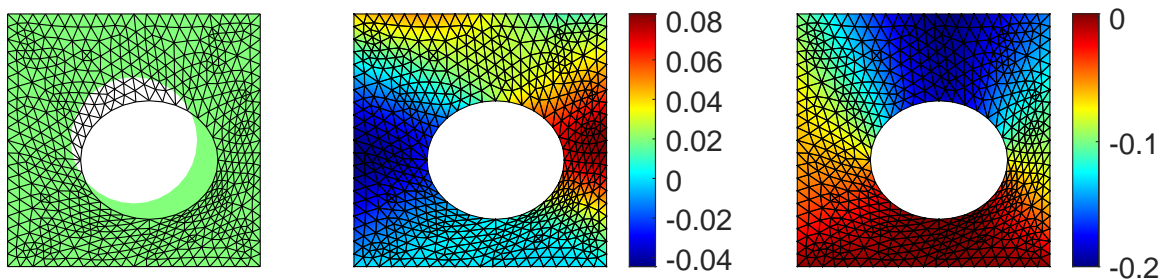


Figure 8: Postprocessing of the stochastic solution of the sample realization $[x_c(\theta^*), y_c(\theta^*), l_x(\theta^*), l_y(\theta^*)] = [0.1215, -0.1564, 1.0876, 0.9390]$: The reference domain (green part) and the random mesh (left), the solution in the \bar{x} direction (mid) and the solution in the \bar{y} direction (right).

495 Fig. 8 (mid and right). In this way, the stochastic solution is shown in the Cartesian coordinate
 496 system, the same way as the classical FEM postprocessing.

497 6.2. Example 2: stochastic elliptic PDE with a random interface

498 6.2.1. Problem setting

499 In this example, we consider a stochastic elliptic PDE

$$-\nabla \cdot (c(x(\theta), y(\theta), \theta) \nabla u(x(\theta), y(\theta), \theta)) = f(x(\theta), y(\theta)) \quad (55)$$

500 defined on the random domain $\mathcal{D}(\theta)$ shown in Fig. 9, which has been widely used in many
 501 problems with random interfaces [5, 6, 15, 41]. Boundary conditions are given by the Dirich-
 502 let condition $u(x(\theta), y(\theta)) = 0$ on Γ_D , the Neumann conditions $\frac{\partial u}{\partial \mathbf{n}} \Big|_{\Gamma_{N,1}} = 1$, $\frac{\partial u}{\partial \mathbf{n}} \Big|_{\Gamma_{N,2}} = 2$ and the
 503 $f(x(\theta), y(\theta)) = 1$ in $\mathcal{D}(\theta)$. We consider the discontinuously random coefficients

$$c_1(x(\theta), y(\theta), \theta) = \xi_{c,1}(\theta) + 1, \quad c_2(x(\theta), y(\theta), \theta) = 2\xi_{c,2}(\theta) + 2, \quad (56)$$

504 where $\xi_{c,1}(\theta)$ and $\xi_{c,2}(\theta)$ are independently uniform random variables on $[0, 1]$.

505 The random interface $\Gamma_r(\theta)$ is considered as a Gaussian random field $\Gamma(\bar{x}, \theta)$ with the mean
 506 function $\bar{\Gamma}(\bar{x}) = 0$ and the covariance function

$$C_{\Gamma}(\bar{x}_1, \bar{x}_2) = \sigma_{\Gamma}^2 (\min(\bar{x}_1, \bar{x}_2) - \bar{x}_1 \bar{x}_2), \quad (57)$$

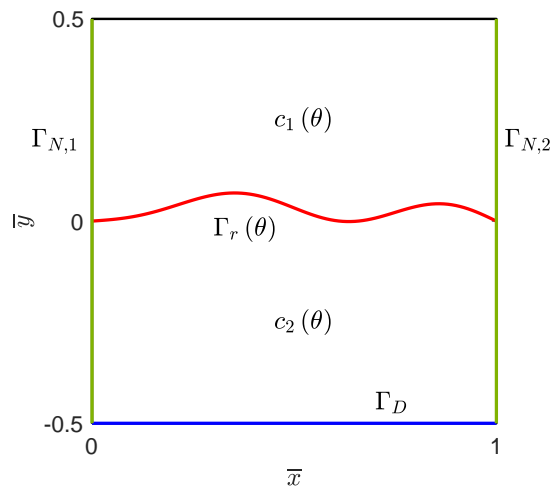
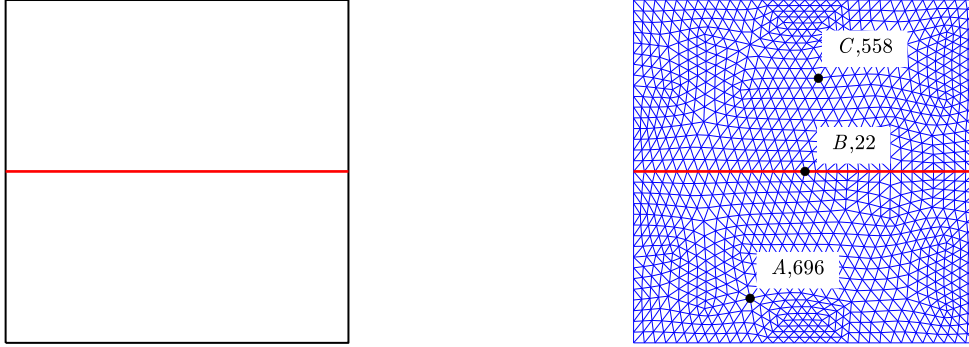


Figure 9: The domain with a random interface.



(a) The reference domain.

(b) Finite element mesh of the reference domain.

Figure 10: The reference domain and its finite element mesh.

507 where the standard deviation $\sigma_\Gamma = 0.1$. By use of KL expansion [18, 42, 43], the random field
 508 $\Gamma(\bar{x}, \theta)$ is approximated as

$$\Gamma(\bar{x}, \theta) = \sigma_\Gamma \sum_{i=1}^r \xi_i(\theta) \sqrt{\kappa_i} \Gamma_i(\bar{x}), \quad (58)$$

509 where $\{\kappa_i\}_{i=1}^r$ and $\{\Gamma_i(\bar{x})\}_{i=1}^r$ are eigenvalues and eigenfunctions of the covariance function $C_{\Gamma}(\bar{x}_1, \bar{x}_2)$
 510 and their analytical solutions are

$$\Gamma_i(\bar{x}) = \sqrt{2} \sin(i\pi\bar{x}), \quad \kappa_i = (i\pi)^{-2}, \quad i = 1, \dots, r. \quad (59)$$

511 In the numerical implementation, we limit the sample realization $\Gamma_i(\bar{x})$ in the interval $[-0.25, 0.25]$.
 512 To reach this point, non-Gaussian bounded distributions can also be adopted to model the random
 513 interface, such as the Beta distribution and the lognormal distribution. Further, the covariance
 514 function in Eq. (57) is non-smooth and more random variables are required to achieve a high-
 515 accuracy simulation of the random field. Smooth or differentiable covariance functions can be
 516 used to reduce the number r of the truncation in Eq. (58) [44, 45].

517 As shown in Fig. 10a, we choose the mean value $\bar{\Gamma}(\bar{x}) = 0$ of the random interface $\Gamma_r(\theta)$ as
 518 the inner interface of the reference domain. As depicted in Fig. 10b, 1217 nodes and 2304 linear
 519 triangular elements are generated for the finite element mesh of the reference domain. Based on
 520 the finite element mesh, discretized random coordinates of points on the random interface are
 521 represented as

$$x(\theta) = \bar{x}, y(\theta) = \Gamma(\bar{x}, \theta) \quad \text{on} \quad \Gamma_r(\theta). \quad (60)$$

522 6.2.2. Numerical results

523 A low-dimensional case is considered by truncating KL expansion Eq. (58) at $r = 5$ items, thus
524 7 (=5+2) random variables are involved in this example. The sample sizes $n_{s,1} = 70$ in step 6 and
525 $n_{s,2} = 1 \times 10^4$ in step 28 in Algorithm 2 are adopted. The iterative errors of different numbers of the
526 retained item k calculated by Eq. (52) are shown in Fig. 11 and 11 retained items converge to the
527 final stochastic solution, which verifies the fast convergence of the proposed method again. First
528 eight deterministic vectors $\{\mathbf{d}_i\}_{i=1}^8$ are depicted in the first and second rows of Fig. 12 and PDFs
529 of corresponding first four recomputed random variables $\{\lambda_i(\theta)\}_{i=1}^4$ are depicted in the third row.
530 It is seen from Fig. 12 that as the number of retained terms increases, the deterministic vectors
531 concentrate near the random interface, which makes that the stochastic solution near the random
532 interface can be approximated with good accuracy.

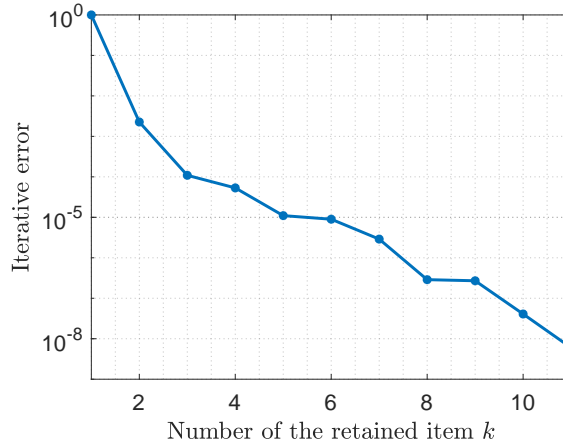


Figure 11: Iterative errors of different numbers of the retained item k calculated by Eq. (52).

533 As shown in Fig. 10b, we check the accuracy of the stochastic solution by using three char-
534 acteristic points, i.e. the point A (its node number is 696) in the lower domain, the point B (its
535 node number is 22) on the random interface and the point C (its node number is 558) in the up-
536 per domain. PDFs of the solutions at three points are calculated by the proposed method and
537 1×10^4 MCS and their comparisons are found in Fig. 13. For all three points, PDFs obtained by

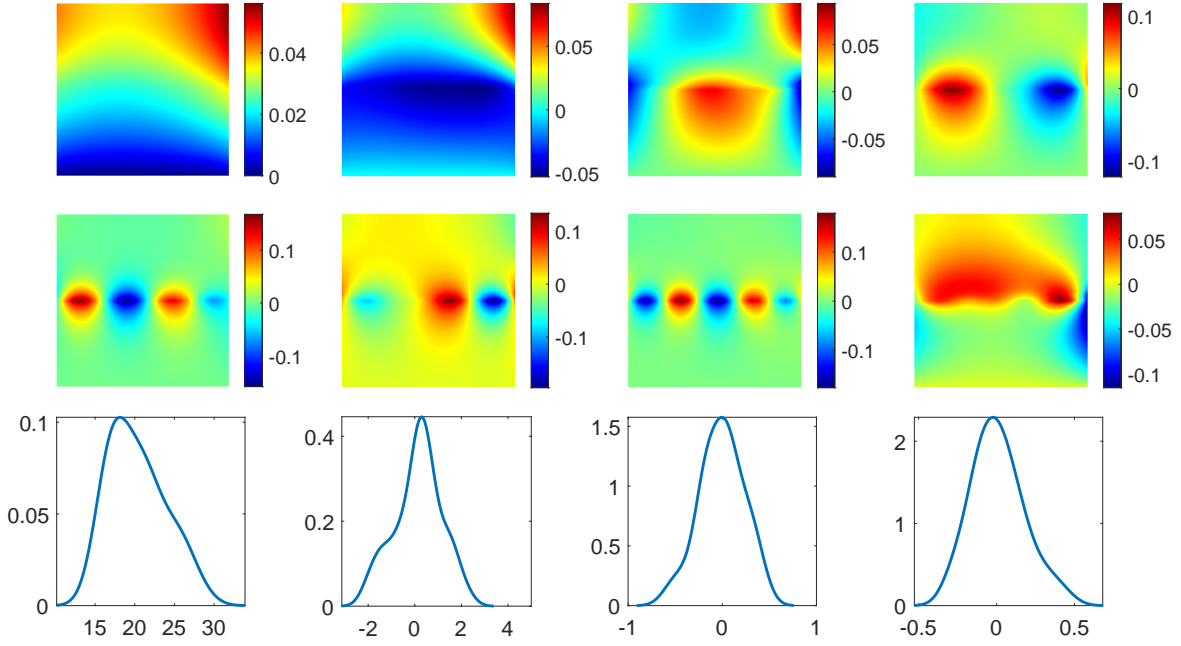


Figure 12: Solutions of the components: the first four solutions $\{\mathbf{d}_i\}_{i=1}^4$ (the first row), the fifth to eighth solutions $\{\mathbf{d}_i\}_{i=5}^8$ (the second row) and PDFs of the first four random variables $\{\lambda_i(\theta)\}_{i=1}^4$ (the third row).

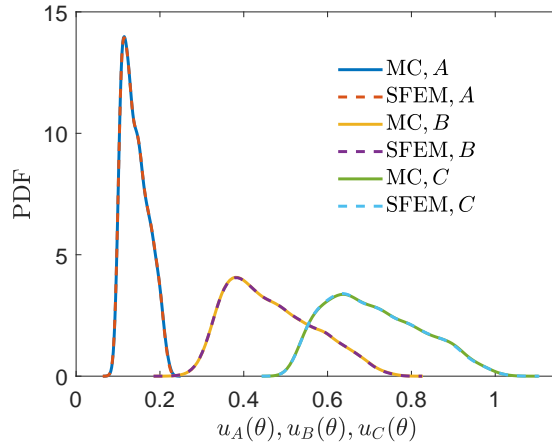


Figure 13: PDFs of the stochastic solutions of points A, B, C obtained by the proposed method and 1×10^4 MCS.

538 the proposed method have good matches with MCS, which demonstrates the high accuracy of the
 539 proposed method in this example.

540 *6.2.3. Reduced-order model*

541 Eq. (37) is considered as a reduced-order equation of the original stochastic finite element
 542 equation (30) and the matrix \mathbf{D} in Eq. (37) is considered as a set of reduced bases. Thus the pro-
 543 posed method provides a powerful way to generate a reduced-order model of the original stochastic
 544 problem. For the explanation of this point, we compare stochastic solutions of full- and reduced-
 545 order models of three sample realizations listed in Tab. 2. Three realizations of the random meshes,

Table 2: Three sample realizations of the random variables.

	$\xi_1(\theta)$	$\xi_2(\theta)$	$\xi_3(\theta)$	$\xi_4(\theta)$	$\xi_5(\theta)$	$c_1(\theta)$	$c_2(\theta)$
Sample 1	0.1194	0.0566	-0.5075	-0.8939	-0.8390	1.6139	3.2612
Sample 2	-1.4709	0.3656	1.4673	1.3410	0.8728	1.1345	2.7835
Sample 3	0.5906	-0.8849	-1.5840	0.2493	-1.0121	1.2153	2.4775

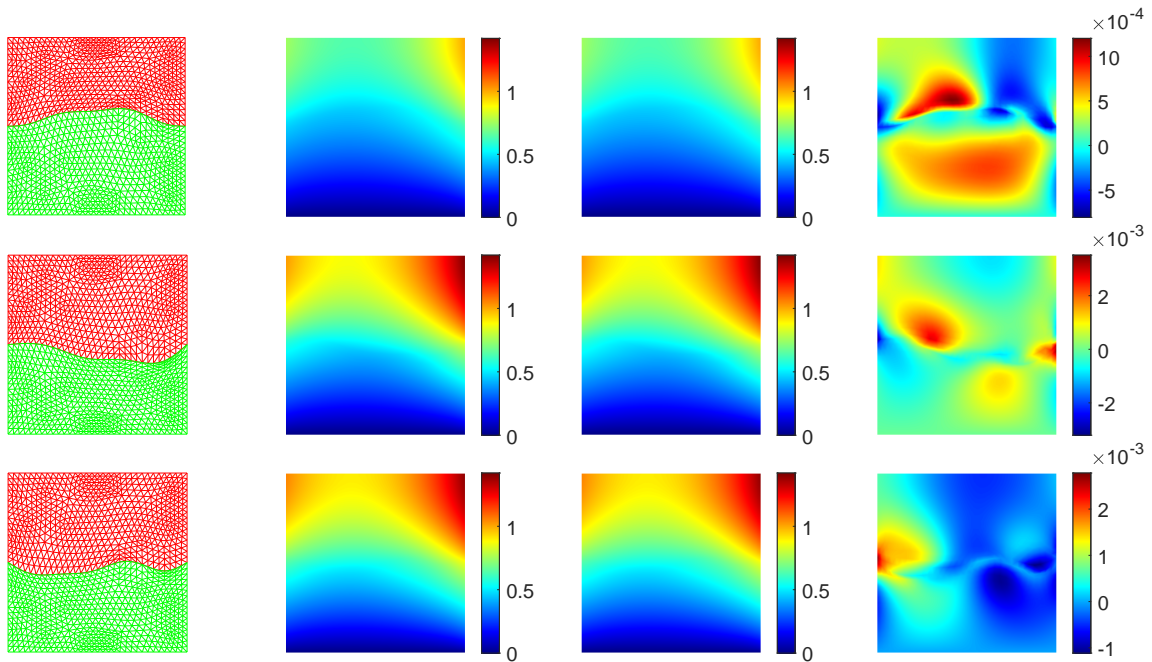


Figure 14: Comparisons between full- and reduced-order models of three sample realizations shown in Tab. 2: the random meshes (the first column), the full-order solutions (the second column), the reduced-order solutions (the third column) and their absolute errors (the fourth column).

546 the full-order stochastic solutions, the reduced-order stochastic solutions and their absolute errors
 547 are depicted in Fig. 14. The maximum of the absolute errors of three realizations is less than
 548 3×10^{-3} , which indicates that the stochastic solutions of reduced-order models have comparable
 549 accuracy to the full-order models and the proposed method can provide an accurate reduced-order
 550 model of the original problem.

551 6.2.4. High-dimensional stochastic problems

552 In this section, we show that the proposed method can be applied to high-dimensional stochas-
 553 tic problems without any modification. The truncation of KL expansion Eq. (58) is set as $r = 50$.
 554 A total of 52 random variables are considered in this example. The sample sizes $n_{s,1} = 520$ in
 555 step 6 and $n_{s,2} = 1 \times 10^4$ in step 28 in Algorithm 2 are used in this case. **Corresponding iterative**
 556 **errors $\epsilon_{u,k}$ in step 26 in Algorithm 2 calculated using Eq. (52) are depicted in Fig. 15** and 28 items
 557 are retained to achieve the convergence error. Compared to the low-dimensional case, the high-
 558 dimensional case requires more retained items to capture the high-accuracy stochastic solution.
 559 The PDFs of the stochastic solutions of the point B (shown in Fig. 10b) obtained by the proposed
 560 method and MCS are compared in Fig. 16. The PDF obtained by the proposed method is in good
 561 agreement with MCS, which indicates that the proposed method has good accuracy even for high-
 562 dimensional stochastic problems. The PDF shown in Fig. 16 is close to that of the low-dimensional

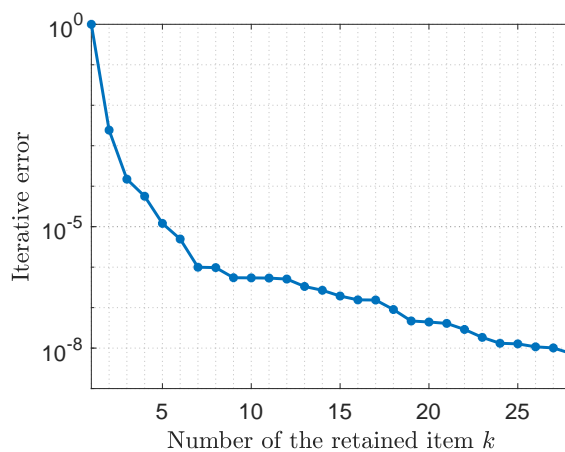


Figure 15: Iterative errors of different numbers of the retained item k calculated by Eq. (52) for the stochastic dimension 52.

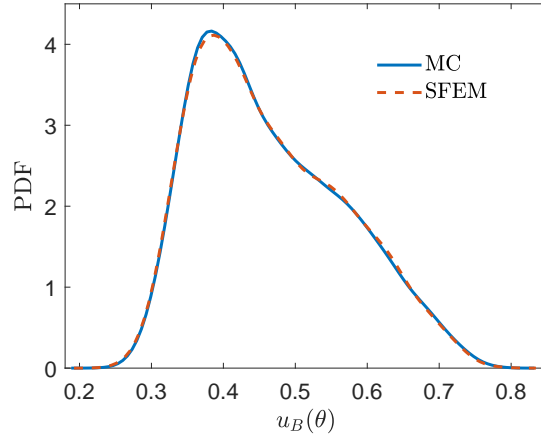


Figure 16: PDFs of the stochastic solutions of the point B when the stochastic dimension is 52.

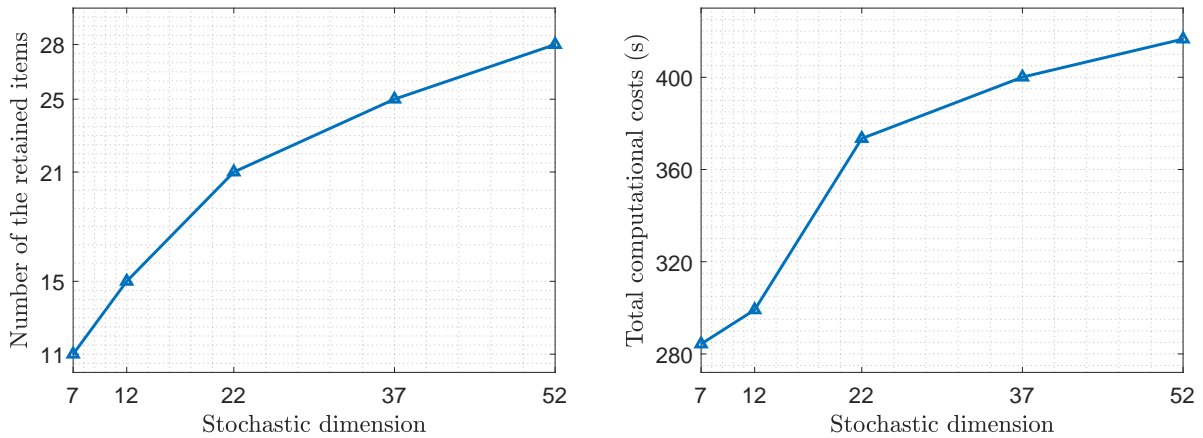
563 case shown in Fig. 13 since the last few terms expanded by KL expansion Eq. (58) may have less
 564 contributions to the quantities of interest. It is noted that the proposed method solves both low-
 565 and high-dimensional stochastic problems using a unified frame and no modification for high-
 566 dimensional cases is needed. Only more retained terms are required to capture the large variability
 567 of the stochastic solution if there are large uncertainties caused by high-dimensional expansions.

568 To show the computational efficiency of the proposed method, the computational costs of the
 569 stochastic dimensions 7 and 52 are listed in Tab. 3. As the stochastic dimension increases, the total
 570 computational cost of the proposed method increases slowly and the recomputing cost slightly
 571 increases since the number of the retained items increases. Compared to MCS, the proposed
 572 method can solve high-dimensional stochastic problems with low computational costs, thus it
 573 avoids the curse of dimensionality successfully.

Table 3: Computational costs of the stochastic dimensions 7 and 52.

Dimension	7 (=5+2)		52 (=50+2)	
Method	SFEM	MCS	SFEM	MCS
Solving costs	264.76		382.75	
Recomputing costs	19.56		33.83	
Total costs (second)	284.32	2124.76	416.58	2359.72

574 The computational efficiency of the proposed method with respect to different stochastic di-
575 mensions are further studied. The truncation r in Eq. (58) is set as 5, 10, 20, 35, 50, respectively,
576 and corresponding total dimensions are 7, 12, 22, 37, 52. We execute the iterations correspond-
577 ing to different stochastic dimensions until the specified convergence error $\varepsilon_{u,k} = 1 \times 10^{-8}$ is
578 achieved. The number of retained terms k and corresponding total computational costs are re-
579 spectively shown in Fig. 17a and Fig. 17b, which demonstrates that the number of retained items
580 increase slightly as the stochastic dimension increases. Corresponding computational costs also
581 does not increase dramatically with the stochastic dimension. For the low dimensions (not greater
582 than 12) and the high dimensions (not less than 22), the computational cost is almost proportional
583 to the stochastic dimension. The cost jumping between the dimensions 12 and 22 may be caused
584 by the large variability induced by the increased dimension.



(a) Numbers of the retained items for different dimensions. (b) Total computational costs for different dimensions.

Figure 17: Numbers of the retained items (left) and corresponding computational costs (right) for different stochastic dimensions.

585 6.3. Example 3: study case from orthodontics with random material properties and random ge- 586 ometry

587 6.3.1. Problem setting

588 In this example, we consider a typical case of orthodontics that involves the human tooth shown
589 in Fig. 18 (left), which is from [46]. It is only a simplified model of the human tooth and more

590 realistic models can be found in [47]. The Young's modulus of multi-layer material of the tooth
 591 are modeled as Gaussian random variables due to individual differences in patients. The Poisson's
 592 ratio and the mean values of Young's modulus are listed in Tab. 4. Standard deviations of all
 593 Young's modulus are 0.1 times the mean values. In the numerical implementation, to ensure that
 594 the Young's modulus is positive, random samples of the Young's modulus less than 1×10^{-3} are
 595 dropped out, thus they are considered as truncated Gaussian random variables in practice. **But it**
 596 **is noted that the truncation usually results in a loss of coercivity in the bilinear form of the finite**
 597 **element approximation [40, 48, 49]. Better stochastic modeling of the material properties should**
 598 **further considered to avoid this issue.** The force induced by the orthodontic appliance is applied
 599 to the model in the horizontal direction and its magnitude is 1N.

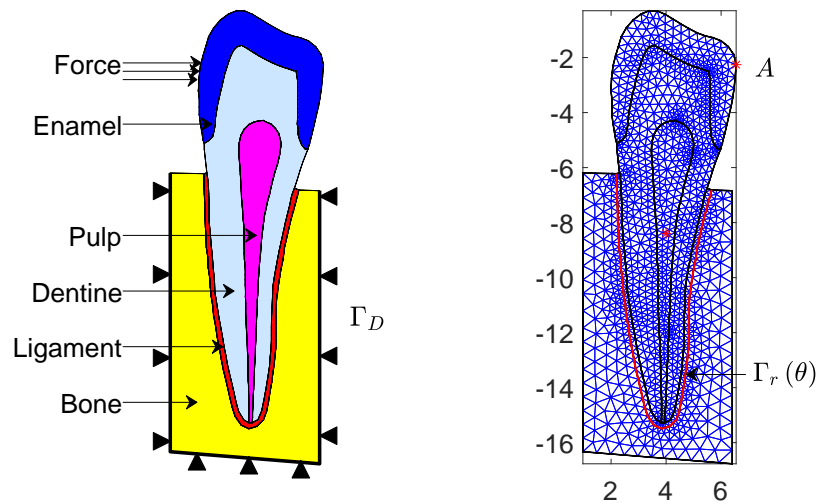


Figure 18: The geometry of tooth model (left) and its finite element mesh (right).

Table 4: Mean values of Young's modulus and Poisson's ratio.

	Bone	Ligament	Dentine	Pulp	Enamel
Young's modulus	12 Gpa	100 Mpa	18 Gpa	4 Mpa	90 Gpa
Poisson's ratio	0.2	0.3	0.2	0.35	0.2

600 In order to formulate a good orthodontics plan, we need to predict the deformation of the tooth

601 after each orthodontic appliance according to the randomness of material properties and current
 602 orthodontics outcomes (i.e. the current position of the tooth). The current position of the tooth
 603 is also uncertain due to the randomness of material properties. As shown in Fig. 18 (right), the
 604 random position is modeled by the rotation angle $\alpha(\theta)$ of the tooth around a point (the red point
 605 in Fig. 18 (right)), where $\alpha(\theta)$ is a uniform random variable on $[-2^\circ, 2^\circ]$. Thus the interface $\Gamma_r(\theta)$
 606 between the bone and the ligament is considered as a random interface depending on the position
 607 of the tooth. The quantity of interest during orthodontics is the horizontal displacement of the
 608 tooth. In this paper, we focus on the horizontal stochastic displacement $u_{A,x}(\theta)$ of the point A
 609 shown in Fig. 18 (right). We adopt the two-dimensional elastic equation discussed in Eq. (20).
 610 The Dirichlet boundary condition is given by $u(\theta) = 0$ on the boundary Γ_D (shown in Fig. 18
 611 (left)). The reference domain and its mesh are depicted in Fig. 18 (right), which has the same
 612 boundary Γ_D as the random domain and includes 1394 nodes, 2670 linear triangular elements and
 613 2788 degrees of freedom in total.

614 6.3.2. Numerical results

615 In this example, the sample sizes $n_{s,1} = 60$ in step 6 and $n_{s,2} = 1 \times 10^4$ in step 28 in Algorithm
 616 2 are used for the implementation. For different numbers of the retained item, iterative errors
 617 $\varepsilon_{\mathbf{u},k}$ in step 26 in Algorithm 2 calculated by Eq. (52) are shown in Fig. 19 and only six items are
 618 retained, which achieves a high-accuracy solution with fewer items due to the small variability of

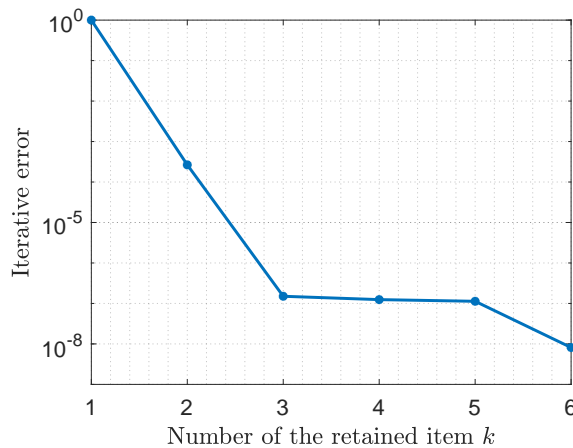


Figure 19: Iterative errors of different numbers of the retained item k calculated by Eq. (52).

619 the random geometry. The PDFs of the horizontal stochastic displacement $u_{A,x}(\theta)$ of the point A
620 obtained by the proposed method and 1×10^4 MCS are compared in Fig. 20 and they are very
621 consistent. Compared to the computational time 2948.69s for MCS, there are only 192.84s for
622 executing the proposed method, including the solving time 189.04s and the recomputing time
623 3.80s, which significantly saves computational effort.

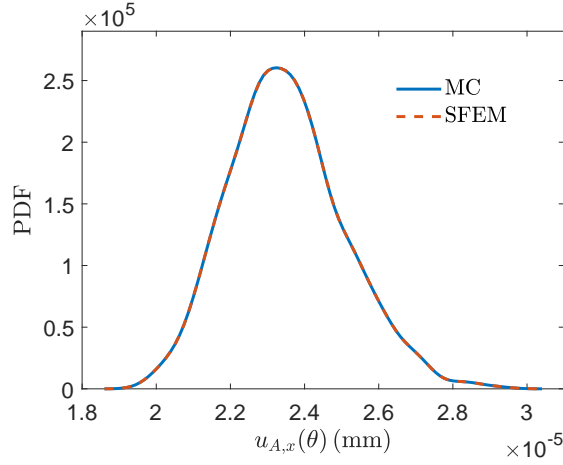


Figure 20: PDFs of the horizontal stochastic displacement $u_{A,x}(\theta)$ of the point A obtained by the proposed method and 1×10^4 MCS.

624 6.4. Example 4: nonlinear stochastic heat equation defined on a random domain

625 6.4.1. Problem setting

626 In this example, to test the applicability of the proposed method to nonlinear PDEs on random
627 domains, we consider a nonlinear stochastic heat equation

$$\begin{cases} -\nabla \cdot (c(T, \theta) \nabla T(x(\theta), y(\theta), \theta)) = 0 & \text{in } \mathcal{D}(\theta) \\ -\vec{\mathbf{n}} \cdot \nabla T(x(\theta), y(\theta), \theta) = f(x(\theta), y(\theta)) & \text{on } \Gamma_r(\theta) \end{cases} \quad (61)$$

628 defined on the random domain $\mathcal{D}(\theta)$ as shown in Fig. 21, where the nonlinear stochastic coefficient
629 $c(T, \theta)$ is given by $c(T, \theta) = \sqrt{10 + T^2(\theta)}$, the heat flux $f(x(\theta), y(\theta)) = 100\text{W/m}$ is applied to the
630 random boundary $\Gamma_r(\theta)$, the temperature $T = 0$ on upper and lower boundaries (red lines shown
631 in Fig. 21), $\vec{\mathbf{n}}$ is the outward normal. It is noted that the applied position of $f(x(\theta), y(\theta))$ is also
632 random due to the randomness of the boundary $\Gamma_r(\theta)$, and the coefficient $c(T, \theta)$ is considered

633 as a random field defined on the random domain due to the spatial dependence of the stochastic
634 solution $T(x(\theta), y(\theta), \theta)$. The geometric uncertainties, the reference domain and the mesh are
635 the same as those in Fig. 2 in Example 6.1. Further, to consider the correlations of the geometric
636 uncertainties, the Pearson correlation coefficient of any two random variables in $x_c(\theta), y_c(\theta), l_x(\theta)$
637 and $l_y(\theta)$ (see from Example 6.1) is 0.2.

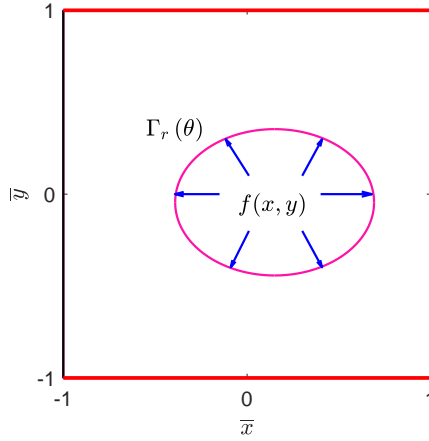


Figure 21: The domain with a random inner boundary and a moving force.

638 For the numerical implementation, the fixed-point iteration is adopted to deal with the nonlin-
639 earity in Eq. (61) [36, 50]. Similar to Eq. (32), we can get the following linearized stochastic finite
640 element equation about the unknown couple $\{\lambda_k(\theta), \mathbf{d}_k\}$ based on the previous stochastic solution
641 approximation $\mathbf{T}_{k-1}(\theta)$

$$\mathbf{K}(\mathbf{T}_{k-1}(\theta)) \lambda_k(\theta) \mathbf{d}_k = \mathbf{F}(\theta) - \mathbf{K}(\mathbf{T}_{k-1}(\theta)) \mathbf{T}_{k-1}(\theta), \quad (62)$$

642 where $\mathbf{K}(\mathbf{T}_{k-1}(\theta))$ is the linearized stochastic matrix assembled using the previous approximation
643 $\mathbf{T}_{k-1}(\theta)$, $\mathbf{F}(\theta)$ is the stochastic vector related to the heat flux $f(x(\theta), y(\theta))$. In this example,
644 $\mathbf{T}_0(\theta) = 0$ is set to initialize the above iteration.

645 6.4.2. Numerical results

646 Algorithm 2 still can be used to solve Eq. (62). But before performing each inner loop, the
647 stochastic matrix $\mathbf{K}(\mathbf{T}_{k-1}(\theta))$ is updated and reassembled based on $\mathbf{T}_{k-1}(\theta)$. The sample sizes
648 $n_{s,1} = 40$ in step 6 in Algorithm 2 and $n_{s,2} = 1 \times 10^4$ in step 28 are used. To well capture the

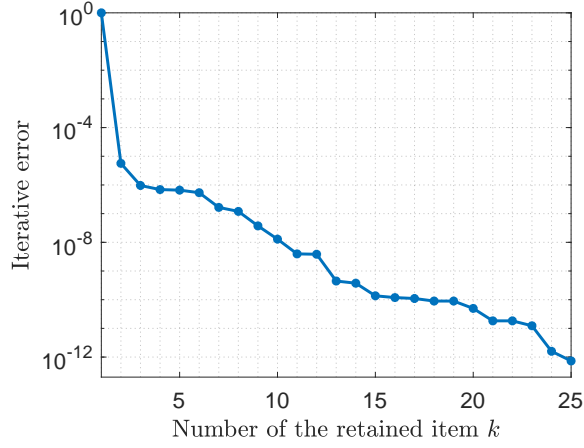


Figure 22: Iterative errors of different numbers of the retained item k calculated by Eq. (52).

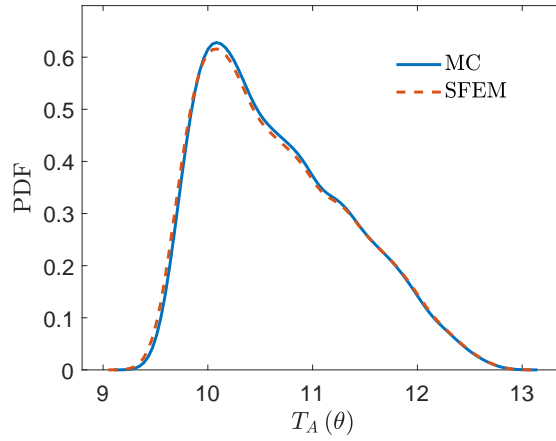


Figure 23: PDFs of the horizontal stochastic displacement $u_{A,x}(\theta)$ of the point A obtained by the proposed method and 1×10^4 MCS.

649 nonlinearity, the stopping criterion $\varepsilon_{\mathbf{u}}$ in step 5 is set as 1×10^{-12} in this case. Iterative errors $\varepsilon_{\mathbf{u},k}$
650 corresponding to different numbers of the retained item is still calculated by Eq. (52) and shown in
651 Fig. 22, which demonstrates that the convergence of the proposed method for nonlinear problems
652 is still good enough. PDFs of the stochastic solution $T_A(\theta)$ of the point A (see Fig. 3b) obtained
653 by the proposed SFEM and 1×10^4 MCS are plotted in Fig. 23. They are still in good agreement.
654 Regarding the computational efficiency, MCS costs 3324.56s, while the proposed SFEM takes
655 351.57s, including the solving time 289.64s and the recomputing time 61.93s, which indicates
656 that the proposed method is still efficient for nonlinear cases.

657 **7. Conclusions**

658 In this paper we develop an efficient stochastic finite element method for solving determinis-
659 tic/stochastic PDEs defined on random domains and illustrate its effectiveness using four numer-
660 ical examples. A reference domain is used to generate a mesh topology and to represent random
661 nodal coordinates of the random domain. Random meshes of the random domain are obtained by
662 combining the mesh topology of the reference domain and the random nodal coordinates of the
663 random domain. In this way, the proposed method still solves the PDEs on the random domain
664 instead of the reference domain, which decouples the differential operator of the PDE and the ran-
665 dom domain and can be implemented via existing FEM assembly codes. The proposed method
666 can be applied to high-dimensional stochastic problems without any modification and avoids the
667 curse of dimensionality to a great extent, which has been demonstrated by an example of up to
668 52 stochastic dimensions. [Also, a nonlinear heat equation defined on a random domain has been
669 used to verify the applicability of the proposed method to nonlinear PDEs on random domains.](#)
670 However, it is noted that the non-intrusive assembly of the stochastic stiffness matrix costs a lot of
671 storage memory compared to intrusive ways, thus it is attractive to develop an intrusive assembly
672 of stochastic stiffness matrices.

673 **Acknowledgments**

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

677 **References**

- 678 [1] O. Le Maître, O. M. Knio, Spectral methods for uncertainty quantification: with applications to computational
679 fluid dynamics, Springer Science & Business Media, 2010.
- 680 [2] H. Dai, R. Zhang, M. Beer, A new method for stochastic analysis of structures under limited observations,
681 Mechanical Systems and Signal Processing 185 (2023) 109730.
- 682 [3] S. Chen, W. Chen, A new level-set based approach to shape and topology optimization under geometric uncer-
683 tainty, Structural and Multidisciplinary Optimization 44 (2011) 1–18.

- 684 [4] W. Zhang, Z. Kang, Robust shape and topology optimization considering geometric uncertainties with stochastic
685 level set perturbation, *International Journal for Numerical Methods in Engineering* 110 (2017) 31–56.
- 686 [5] A. Nouy, A. Clement, eXtended Stochastic Finite Element Method for the numerical simulation of heteroge-
687 neous materials with random material interfaces, *International Journal for Numerical Methods in Engineering*
688 83 (2010) 1312–1344.
- 689 [6] C. Lang, A. Doostan, K. Maute, Extended stochastic FEM for diffusion problems with uncertain material
690 interfaces, *Computational Mechanics* 51 (2013) 1031–1049.
- 691 [7] M. Diez, E. F. Campana, F. Stern, Design-space dimensionality reduction in shape optimization by Karhunen-
692 Loève expansion, *Computer Methods in Applied Mechanics and Engineering* 283 (2015) 1525–1544.
- 693 [8] S. Badia, J. Hampton, J. Principe, Embedded multilevel monte carlo for uncertainty quantification in random
694 domains, *International Journal for Uncertainty Quantification* 11 (2021).
- 695 [9] D. Xiu, D. M. Tartakovsky, Numerical methods for differential equations in random domains, *SIAM Journal on*
696 *Scientific Computing* 28 (2006) 1167–1185.
- 697 [10] C. Canuto, T. Kozubek, A fictitious domain approach to the numerical solution of PDEs in stochastic domains,
698 *Numerische Mathematik* 107 (2007) 257–293.
- 699 [11] A. Nouy, M. Chevreuril, E. Safatly, Fictitious domain method and separated representations for the solution of
700 boundary value problems on uncertain parameterized domains, *Computer Methods in Applied Mechanics and*
701 *Engineering* 200 (2011) 3066–3082.
- 702 [12] M. Papadarakakis, V. Papadopoulos, Robust and efficient methods for stochastic finite element analysis using
703 Monte Carlo simulation, *Computer Methods in Applied Mechanics Engineering* 134 (1996) 325–340.
- 704 [13] J. N. Fuhg, A. Fau, U. Nackenhorst, State-of-the-art and comparative review of adaptive sampling methods for
705 kriging, *Archives of Computational Methods in Engineering* (2020) 1–59.
- 706 [14] G. Lin, A. M. Tartakovsky, D. M. Tartakovsky, Uncertainty quantification via random domain decomposition
707 and probabilistic collocation on sparse grids, *Journal of Computational Physics* 229 (2010) 6995–7012.
- 708 [15] Y. N. Lazarev, P. Petrov, D. M. Tartakovsky, Interface dynamics in randomly heterogeneous porous media,
709 *Advances in Water Resources* 28 (2005) 393–403.
- 710 [16] P. S. Mohan, P. B. Nair, A. J. Keane, Stochastic projection schemes for deterministic linear elliptic partial
711 differential equations on random domains, *International Journal for Numerical Methods in Engineering* 85
712 (2011) 874–895.
- 713 [17] V. D. Liseikin, *Grid generation methods*, Springer, 2017.
- 714 [18] R. G. Ghanem, P. D. Spanos, *Stochastic finite elements: a spectral approach*, Courier Corporation, 2003.
- 715 [19] D. Xiu, *Numerical methods for stochastic computations: a spectral method approach*, Princeton University
716 Press, 2010.
- 717 [20] J. E. Castrillón-Candás, F. Nobile, R. F. Tempone, Analytic regularity and collocation approximation for elliptic

- 718 PDEs with random domain deformations, *Computers & Mathematics with Applications* 71 (2016) 1173–1197.
- 719 [21] R. Glowinski, T.-W. Pan, J. Periaux, A fictitious domain method for Dirichlet problem and applications, *Com-*
720 *puter Methods in Applied Mechanics and Engineering* 111 (1994) 283–303.
- 721 [22] A. Nouy, F. Schoefs, N. Moës, X-SFEM, a computational technique based on X-FEM to deal with random
722 shapes, *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique* 16 (2007)
723 277–293.
- 724 [23] A. Nouy, A. Clement, F. Schoefs, N. Moës, An extended stochastic finite element method for solving stochastic
725 partial differential equations on random domains, *Computer Methods in Applied Mechanics and Engineering*
726 197 (2008) 4663–4682.
- 727 [24] A. R. Khoei, *Extended finite element method: theory and applications*, John Wiley & Sons, 2014.
- 728 [25] D. Xiu, G. E. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations, *SIAM*
729 *Journal on Scientific Computing* 24 (2002) 619–644.
- 730 [26] I. Babuška, J. Chleboun, Effects of uncertainties in the domain on the solution of Neumann boundary value
731 problems in two spatial dimensions, *Mathematics of Computation* 71 (2002) 1339–1370.
- 732 [27] I. Babuška, J. Chleboun, Effects of uncertainties in the domain on the solution of Dirichlet boundary value
733 problems, *Numerische Mathematik* 93 (2003) 583–610.
- 734 [28] A. Kundu, S. Adhikari, M. Friswell, Stochastic finite elements of discretely parameterized random systems on
735 domains with boundary uncertainty, *International Journal for Numerical Methods in Engineering* 100 (2014)
736 183–221.
- 737 [29] H. Harbrecht, M. Peters, M. Siebenmorgen, Analysis of the domain mapping method for elliptic diffusion
738 problems on random domains, *Numerische Mathematik* 134 (2016) 823–856.
- 739 [30] J. E. Castrillón-Candás, F. Nobile, R. F. Tempone, A hybrid collocation-perturbation approach for PDEs with
740 random domains, *Advances in Computational Mathematics* 47 (2021) 1–35.
- 741 [31] A. Ammar, A. Huerta, F. Chinesta, E. Cueto, A. Leygue, Parametric solutions involving geometry: a step
742 towards efficient shape optimization, *Computer Methods in Applied Mechanics and Engineering* 268 (2014)
743 178–193.
- 744 [32] A. Courard, D. Néron, P. Ladevèze, L. Ballere, Integration of PGD-virtual charts into an engineering design
745 process, *Computational Mechanics* 57 (2016) 637–651.
- 746 [33] S. M. Shontz, S. A. Vavasis, A mesh warping algorithm based on weighted Laplacian smoothing., in: *IMR,*
747 *Citeseer*, 2003, pp. 147–158.
- 748 [34] M. Selim, R. Koomullil, Mesh deformation approaches – a survey, *Journal of Physical Mathematics* 7 (2016)
749 1–9.
- 750 [35] T. J. Hughes, *The finite element method: linear static and dynamic finite element analysis*, Courier Corporation,
751 2012.

- 752 [36] K.-J. Bathe, Finite element procedures, Klaus-Jurgen Bathe, 2006.
- 753 [37] Z. Zheng, H. Dai, Structural stochastic responses determination via a sample-based stochastic finite element
754 method, *Computer Methods in Applied Mechanics and Engineering* 381 (2021) 113824.
- 755 [38] Z. Zheng, M. Beer, H. Dai, U. Nackenhorst, A weak-intrusive stochastic finite element method for stochastic
756 structural dynamics analysis, *Computer Methods in Applied Mechanics and Engineering* 399 (2022) 115360.
- 757 [39] F. Chinesta, R. Keunings, A. Leygue, The proper generalized decomposition for advanced numerical simula-
758 tions: a primer, Springer Science & Business Media, 2013.
- 759 [40] A. Nouy, A generalized spectral decomposition technique to solve a class of linear stochastic partial differential
760 equations, *Computer Methods in Applied Mechanics and Engineering* 196 (2007) 4521–4537.
- 761 [41] R. Ghanem, W. Brzakala, Stochastic finite-element analysis of soil layers with random interface, *Journal of*
762 *Engineering Mechanics* 122 (1996) 361–369.
- 763 [42] Z. Zheng, H. Dai, Simulation of multi-dimensional random fields by Karhunen–Loève expansion, *Computer*
764 *Methods in Applied Mechanics and Engineering* 324 (2017) 221–247.
- 765 [43] Z. Zheng, H. Dai, Y. Wang, W. Wang, A sample-based iterative scheme for simulating non-stationary non-
766 Gaussian stochastic processes, *Mechanical Systems and Signal Processing* 151 (2021) 107420.
- 767 [44] P. D. Spanos, M. Beer, J. Red-Horse, Karhunen–Loève expansion of stochastic processes with a modified
768 exponential covariance kernel, *Journal of Engineering Mechanics* 133 (2007) 773–779.
- 769 [45] M. G. Faes, M. Broggi, P. D. Spanos, M. Beer, Elucidating appealing features of differentiable auto-correlation
770 functions: A study on the modified exponential kernel, *Probabilistic Engineering Mechanics* (2022) 103269.
- 771 [46] I. González, M. A. Valdebenito, J. Correa, H. A. Jensen, Calculation of second order statistics of uncertain linear
772 systems applying reduced order models, *Reliability Engineering & System Safety* 190 (2019) 106514.
- 773 [47] E. Li, J. Chen, Z. Zhang, J. Fang, G. Liu, Q. Li, Smoothed finite element method for analysis of multi-layered
774 systems – Applications in biomaterials, *Computers & Structures* 168 (2016) 16–29.
- 775 [48] I. Babuška, P. Chatzipantelidis, On solving elliptic stochastic partial differential equations, *Computer Methods*
776 *in Applied Mechanics and Engineering* 191 (2002) 4093–4122.
- 777 [49] P. Frauenfelder, C. Schwab, R. A. Todor, Finite elements for elliptic problems with stochastic coefficients,
778 *Computer Methods in Applied Mechanics and Engineering* 194 (2005) 205–228.
- 779 [50] J. N. Reddy, An introduction to nonlinear finite element analysis: with applications to heat transfer, fluid me-
780 chanics, and solid mechanics, OUP Oxford, 2014.