

# A nonlinear stochastic finite element method for solving elastoplastic problems with uncertainties

Zhibao Zheng<sup>1</sup> | Udo Nackenhorst

Leibniz Universität Hannover, Institute of Mechanics and Computational Mechanics & International Research Training Group 2657, Hannover, Germany

## Correspondence

Zhibao Zheng, Leibniz Universität Hannover, Institute of Mechanics and Computational Mechanics, Appelstraße 9a, 30167, Hannover, Germany.  
Email: [zhibao.zheng@ibnm.uni-hannover.de](mailto:zhibao.zheng@ibnm.uni-hannover.de)

## Funding information

Alexander von Humboldt-Stiftung; Deutsche Stiftung Friedensforschung, Grant/Award Number: 433082294

## Abstract

This article presents an efficient nonlinear stochastic finite element method to solve stochastic elastoplastic problems. Similar to deterministic elastoplastic problems, we describe history-dependent stochastic elastoplastic behavior utilizing a series of (pseudo) time steps and go further to solve the corresponding stochastic solutions. For each time step, the original stochastic elastoplastic problem is considered as a time-independent nonlinear stochastic problem with initial values given by stochastic displacements, stochastic strains, and internal variables of the previous time step. To solve the stochastic solution at each time step, the corresponding nonlinear stochastic problem is transformed into a set of linearized stochastic finite element equations by means of finite element discretization and a stochastic Newton linearization, while the stochastic solution at each time step is approximated by a sum of the products of random variables and deterministic vectors. Each couple of the random variable and the deterministic vector is also used to approximate the stochastic solution of the corresponding linearized stochastic finite element equation that can be solved via a weakly intrusive method. In this method, the deterministic vector is computed by solving deterministic linear finite element equations, and corresponding random variables are solved by a non-intrusive method. Further, the proposed method avoids the curse of dimensionality successfully since its computational effort does not increase dramatically as the stochastic dimensionality increases. Four numerical cases are used to demonstrate the good performance of the proposed method.

## KEYWORDS

curse of dimensionality, nonlinear stochastic finite element method, stochastic elastoplasticity, stochastic Newton linearization, weakly intrusive approximation

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2023 The Authors. *International Journal for Numerical Methods in Engineering* published by John Wiley & Sons Ltd.

## 1 | INTRODUCTION

Since the inherent uncertainty and the epistemic uncertainty of systems are unavoidable in many practical engineering problems,<sup>1,2</sup> predicting uncertainty propagation on the physical models has become an important part of the analysis of systems. In this article, we focus on efficiently solving elastoplastic problems with uncertainties, in which the uncertainties are described by random variables or fields in the context of a probabilistic frame.

Although mathematical theories and numerical methods for solving deterministic elastoplastic problems have been well understood,<sup>3,4</sup> only a few effort is took to their extensions to stochastic cases. Over the last two decades, several methods can be used for this purpose. As a kind of powerful method for stochastic analysis, Monte Carlo simulation (MCS) and its improvements<sup>1,5-7</sup> can be used to accurately solve stochastic elastoplastic problems. This method is easy to be implemented and only requires the use of existing deterministic solvers. Its convergence does not depend on the stochastic dimensionality, thus MCS can be applied to solve high-dimensional stochastic problems without extra difficulties. However, a large number of deterministic simulations are required to achieve high-accuracy stochastic solutions, which is expensive for large-scale and nonlinear stochastic problems. Due to the simple and efficient implementation, other non-intrusive methods are also developed to solve stochastic elastoplastic problems, for example, adaptive sampling methods, surrogate model methods and so forth.<sup>8-10</sup>

As a kind of intrusive method, the spectral stochastic finite element method (SFEM) and its extensions<sup>1,11,12</sup> have been widely used to solve stochastic problems. In this method, the stochastic solution is expanded by polynomial chaos (PC) bases and the stochastic Galerkin approach is then adopted to transform the original stochastic problem into an augmented deterministic equation whose size dramatically increases as the numbers of the stochastic dimensionality, the expansion order of PC bases and the degree of freedom of the physical system increase. Several improvements<sup>13,14</sup> are developed to reduce the computational effort of the augmented deterministic equation. A direct extension of this method to stochastic elastoplastic problems can be found in Reference 15. Several modifications are studied since the classical PC expansion cannot well capture nonsmooth stochastic solutions of stochastic elastoplastic problems. References 16 and 17 introduce two fictitious bounding bodies to approximate stochastic elastoplastic constitutive equations and then PC-based expansions are adopted to solve the corresponding stochastic finite element equations. Stochastic elastoplastic problems are solved by coupling the spectral SFEM to the Fokker Planck Kolmogorov (FPK) equation approach in References 18 and 19. The randomness of the elastoplastic constitutive model is propagated through FPK equations and the unknown stochastic solution is solved via the spectral SFEM. In Reference 20, stochastic elastoplastic problems are considered as inequality-constrained stochastic boundary value problems that are investigated via stochastic variational inequalities. PC expansions and stochastic collocation methods are used to discretize the original stochastic problem and nonlinear convex programming is adopted for the numerical implementation. Also, since the multi-element PC expansion<sup>21</sup> can be used to well capture sharp and discontinuous stochastic solutions, it has been applied to stochastic elastoplastic problems in Reference 22. There are also other effective methods for solving nonsmooth stochastic solutions of nonlinear stochastic problems, for example, the stochastic perturbation method,<sup>23,24</sup> the stochastic collocation methods,<sup>25,26</sup> the polynomial/spline dimensional decomposition methods.<sup>27-29</sup>

As discussed above, developing efficient and accurate methods for solving stochastic elastoplastic problems remains an attractive topic. In this article, we propose a novel nonlinear SFEM to solve stochastic elastoplastic problems. Specifically, stochastic elastoplastic problems are considered as (pseudo) time-dependent stochastic nonlinear problems. A time discretization scheme is given to evolve the stochastic elastoplasticity and discretize the original time-dependent stochastic nonlinear problems into the time-independent stochastic nonlinear problem at each time step, and the finite element method is adopted for the spatial discretization. For each time step, the initial values of the discretized stochastic nonlinear problem are inherited from the stochastic quantities of the previous time step, for example, stochastic strains and stochastic internal variables. A stochastic Newton method is given to linearize the discretized stochastic nonlinear problem as a series of linear stochastic finite element equations. The stochastic solution at each time step thus consists of a set of stochastic increments that are also the stochastic solution of the corresponding linear stochastic finite element equation. Each stochastic increment is then decoupled into stochastic and deterministic spaces and approximated by the product of a random variable and a deterministic vector, which are calculated by solving the corresponding linear stochastic finite element equation using a weakly intrusive SFEM.<sup>30,31</sup> In this method, the deterministic vector is computed by solving deterministic linear finite element equations that are obtained by the stochastic Galerkin method,<sup>11,12</sup> and the corresponding random variable is solved by one-dimensional stochastic algebraic equations using a non-intrusive method. Furthermore, since all random sources are embedded into one-dimensional stochastic algebraic equations, the

proposed method has low computational complexity and its computational cost is weakly dependent on the stochastic dimensionality. The curse of dimensionality arising in high-dimensional stochastic elastoplastic problems is thus circumvented successfully.

The article is organized as follows: Section 2 gives the basic setting of stochastic elastoplastic problems and the time discretization scheme. Section 3 introduces a stochastic Newton-based nonlinear stochastic finite element method to solve stochastic elastoplastic problems. Computational aspects and algorithm implementations of the proposed method are elaborated in Section 4. Following that, four numerical cases are given to demonstrate the performance of the proposed method in Section 5, and conclusions and discussions are presented in Section 6.

## 2 | STOCHASTIC ELASTOPLASTIC PROBLEMS

### 2.1 | Elastoplastic equation with uncertainties

Let  $(\Theta, \Xi, \mathcal{P})$  be a suitable probability space, where  $\Theta$  denotes the space of elementary events,  $\Xi$  is a  $\sigma$ -algebra defined on  $\Theta$  and  $\mathcal{P}$  is a probability measure. In this article, we consider the following stochastic elastoplastic problem defined on a deterministic domain  $\Omega \subset \mathbb{R}^d$  with the boundary  $\partial\Omega$ ,

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

where the physical dimension may be  $d = 1, 2, 3$ ,  $\operatorname{div}(\cdot)$  represents the divergence operator,  $\boldsymbol{\sigma}(\mathbf{x}, \theta)$  is the stochastic stress tensor,  $\mathbf{f}(\mathbf{x}, \theta)$  is the stochastic external force,  $\mathbf{u}(\mathbf{x}, \theta)$  is the unknown stochastic displacement,  $\Gamma_N$  and  $\Gamma_D$  are boundary segments associated with the Neumann boundary condition  $\mathbf{g}(\mathbf{x}, \theta)$  and the Dirichlet boundary condition  $\bar{\mathbf{u}}(\mathbf{x}, \theta)$ . Without loss of generality, we let the Dirichlet boundary condition  $\bar{\mathbf{u}}(\mathbf{x}, \theta) = 0$  in this article. The following linear stochastic strain tensor is adopted to describe the stochastic displacement-strain relationship

$$\boldsymbol{\varepsilon}(\mathbf{u}(\mathbf{x}, \theta)) = \frac{1}{2} (\nabla \mathbf{u}(\mathbf{x}, \theta) + (\nabla \mathbf{u}(\mathbf{x}, \theta))^T). \quad (2)$$

Further, we consider the relationship between the stochastic stress tensor and the stochastic strain tensor given by

$$\boldsymbol{\sigma}(\mathbf{u}(\mathbf{x}, \theta), \theta) = \mathcal{G}(\boldsymbol{\varepsilon}(\mathbf{u}(\mathbf{x}, \theta)), \theta), \quad (3)$$

where  $\mathcal{G} : \mathbb{R}^{d \times d} \rightarrow \mathbb{R}^{d \times d}$  is a stochastic nonlinear operator between the stochastic strain and the stochastic stress tensors. It may be associated with stochastic yield stresses and stochastic constitutive relations, such as the Von Mises and Drucker-Prager yield criteria<sup>4</sup> involving the stochastic thermodynamical force  $\boldsymbol{\beta}(\theta)$  and the stochastic internal variable  $\rho(\theta)$ . The stochastic operator  $\mathcal{G}$  in this section is a general form and it can be applied to all kinds of elastoplastic models. We will provide detailed representations of it for some cases in the subsequent section.

The weak form of (1) is written as

$$\mathcal{W}(\mathbf{u}(\mathbf{x}, \theta), \mathbf{v}(\mathbf{x}), \theta) = \mathcal{F}(\mathbf{v}(\mathbf{x}), \theta), \quad \forall \mathbf{v}(\mathbf{x}) \in \mathcal{V}, \quad (4)$$

the left and right sides of which are given by

$$\mathcal{W}(\mathbf{u}(\mathbf{x}, \theta), \mathbf{v}(\mathbf{x}), \theta) = \int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}(\mathbf{x}, \theta), \theta) : \boldsymbol{\varepsilon}(\mathbf{v}(\mathbf{x})) \, d\mathbf{x}, \quad (5)$$

$$= \int_{\Omega} \mathcal{G}(\boldsymbol{\varepsilon}(\mathbf{u}(\mathbf{x}, \theta)), \theta) : \boldsymbol{\varepsilon}(\mathbf{v}(\mathbf{x})) \, d\mathbf{x}, \quad (6)$$

$$\mathcal{F}(\mathbf{v}(\mathbf{x}), \theta) = \int_{\Omega} \mathbf{f}(\mathbf{x}, \theta) \cdot \mathbf{v}(\mathbf{x}) \, d\mathbf{x} + \int_{\Gamma_N} \mathbf{g}(\mathbf{x}, \theta) \cdot \mathbf{v}(\mathbf{x}) \, d\mathbf{s}, \quad (7)$$

where  $\mathbf{v}(\mathbf{x})$  is the test function and  $\mathcal{V}$  is the Hilbert space with smooth displacement vectors vanishing on the boundary  $\Gamma_D$ .

## 2.2 | Elastoplastic constitutive model

In this section, we consider applying an elastoplastic constitutive model to (1). To this end, we divide the stochastic strain tensor  $\boldsymbol{\varepsilon}(\mathbf{u}(\mathbf{x}, \theta))$  into an elastic part and a plastic part, which corresponds to

$$\boldsymbol{\varepsilon}(\mathbf{u}(\mathbf{x}, \theta)) = \boldsymbol{\varepsilon}_e(\mathbf{u}(\mathbf{x}, \theta)) + \boldsymbol{\varepsilon}_p(\mathbf{u}(\mathbf{x}, \theta)), \quad (8)$$

where  $\boldsymbol{\varepsilon}_e(\mathbf{u}(\mathbf{x}, \theta))$  and  $\boldsymbol{\varepsilon}_p(\mathbf{u}(\mathbf{x}, \theta))$  denote the stochastic elastic strain tensor and the stochastic plastic strain tensor, respectively. The stochastic stress tensor  $\boldsymbol{\sigma}(\mathbf{u}, \theta)$  is represented by the stochastic elastic strain tensor  $\boldsymbol{\varepsilon}_e(\mathbf{u}(\mathbf{x}, \theta))$

$$\boldsymbol{\sigma}(\mathbf{u}(\mathbf{x}, \theta), \theta) = \mathbf{C}(\theta) \boldsymbol{\varepsilon}_e(\mathbf{u}(\mathbf{x}, \theta)) = \mathbf{C}(\theta) (\boldsymbol{\varepsilon}(\mathbf{u}(\mathbf{x}, \theta)) - \boldsymbol{\varepsilon}_p(\mathbf{u}(\mathbf{x}, \theta))), \quad (9)$$

where  $\mathbf{C}(\theta)$  is a fourth order elastic tensor and the following form is adopted in this article

$$\mathbf{C}(\theta) = \kappa(\theta) \mathbf{I} \otimes \mathbf{I} + 2\mu(\theta) \mathbf{I}_{\text{dev}}, \quad (10)$$

where the parameters  $\kappa(\theta)$  and  $\mu(\theta)$  are random variables/fields that may be related to stochastic material properties, also known as stochastic bulk modulus and stochastic shear modulus, the second order unit tensors  $\mathbf{I} \in \mathbb{R}^{d \times d}$ , the fourth order tensor  $\mathbf{I}_{\text{dev}} = \mathbf{I}_4 - \frac{1}{3} \mathbf{I} \otimes \mathbf{I}$ , where the fourth order tensor  $\mathbf{I}_4 = \delta_{ik} \delta_{jl} \mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l$ ,  $\delta_{ik}$  is the Kronecker delta.

Further, we only consider the Von Mises yield criterion and the linear kinematic hardening in this article, but other cases can also be considered in a similar way, for example, the Drucker-Prager yield criterion, the isotropic hardening and so forth.<sup>4</sup> The stochastic thermodynamical force  $\boldsymbol{\beta}(\theta)$  associated with the internal variable  $\rho(\theta)$  is given by

$$\boldsymbol{\beta}(\theta) = \hbar(\rho, \theta) = \varpi(\theta) \rho(\theta), \quad (11)$$

where the coefficient  $\varpi(\theta)$  is a positive parameter and can be a random variable/field in practice. Furthermore, the stochastic yield function is given by

$$\Psi(\boldsymbol{\sigma}, \boldsymbol{\beta}, \theta) = |\mathbf{I}_{\text{dev}} \boldsymbol{\sigma}(\theta) - \boldsymbol{\beta}(\theta)| - \sigma_Y(\theta), \quad (12)$$

where  $\sigma_Y(\theta)$  is the stochastic yield stress that can be modeled by a random variable/field,  $\mathbf{I}_{\text{dev}} \boldsymbol{\sigma}(\theta)$  is considered as the stress deviator of the stochastic stress tensor  $\boldsymbol{\sigma}(\mathbf{u}(\mathbf{x}, \theta), \theta)$ .

## 2.3 | Time discretization scheme

In this section, we extend the incremental method for solving deterministic elastoplastic problems to stochastic cases. The external force is applied incrementally in steps, which is considered as a set of pseudo time steps and described using a pseudo time variable  $t$ . The (pseudo) time variable  $t$  is discretized as  $t \in [t_0, t_1, \dots, t_{n_t}]$ . For the time step  $t_i$ ,  $i \geq 1$ , the stochastic solution  $\mathbf{u}(t_i, \theta)$  is calculated by solving a nonlinear stochastic problem that is dependent on the quantities  $\{\mathbf{u}(t_{i-1}, \theta), \boldsymbol{\varepsilon}_p(t_{i-1}, \theta), \boldsymbol{\beta}(t_{i-1}, \theta)\}$  obtained by the previous time step  $t_{i-1}$ . For simplicity, we denote  $\boldsymbol{\sigma}(t_i, \theta)$ ,  $\boldsymbol{\varepsilon}_p(t_i, \theta)$  and  $\boldsymbol{\beta}(t_i, \theta)$  as  $\boldsymbol{\sigma}_i(\theta)$ ,  $\boldsymbol{\varepsilon}_{p,i}(\theta)$  and  $\boldsymbol{\beta}_i(\theta)$  in the latter equations. According to the analytical results in Reference 32, an explicit expression of the elastoplastic constitutive relation (3) can be obtained in the stochastic case,

$$\boldsymbol{\sigma}_i(\theta) = \mathcal{G}(\boldsymbol{\varepsilon}_i(\theta), \theta; \boldsymbol{\varepsilon}_{p,i-1}(\theta), \boldsymbol{\beta}_{i-1}(\theta)) = \boldsymbol{\sigma}_{0,i}(\theta) - \frac{2\mu(\theta)}{2\mu(\theta) + \varpi(\theta)} \langle \gamma_i(\theta) \rangle \boldsymbol{\sigma}_i^*(\theta), \quad (13)$$

where  $\gamma_i(\theta) = 1 - \frac{\sigma_Y(\theta)}{|\boldsymbol{\sigma}_i^*(\theta)|}$ , the operator  $\langle \cdot \rangle = \max(\cdot, 0)$  represents the maximum value of  $\cdot$  and 0, and the stochastic stresses  $\boldsymbol{\sigma}_{0,i}(\theta)$  and  $\boldsymbol{\sigma}_i^*(\theta)$  are given by

$$\sigma_{0,i}(\theta) = \mathbf{C}(\theta) (\epsilon_i(\theta) - \epsilon_{p,i-1}(\theta)), \tag{14}$$

$$\sigma_i^*(\theta) = \mathbf{I}_{\text{dev}} \sigma_{0,i}(\theta) - \beta_{i-1}(\theta). \tag{15}$$

The  $\beta_i(\theta)$  and  $\epsilon_{p,i}(\theta)$  are then updated by using  $\sigma_i^*(\theta)$  from current time step  $t_i$  and  $\beta_{i-1}(\theta)$ ,  $\epsilon_{p,i-1}(\theta)$  from the previous time step  $t_{i-1}$ ,

$$\beta_i(\theta) = \beta_{i-1}(\theta) + \frac{\varpi(\theta)}{2\mu(\theta) + \varpi(\theta)} \langle \gamma_i(\theta) \rangle \sigma_i^*(\theta), \tag{16}$$

$$\epsilon_{p,i}(\theta) = \epsilon_{p,i-1}(\theta) + \frac{1}{2\mu(\theta) + \varpi(\theta)} \langle \gamma_i(\theta) \rangle \sigma_i^*(\theta). \tag{17}$$

It is seen that  $\langle \gamma_i(\theta) \rangle = \gamma_i(\theta) > 0$  holds and the second terms of (13), (16), and (17) are nonzero when  $|\sigma_i^*(\theta)| > \sigma_Y(\theta)$  (i.e., the true stress is greater than the yield stress), the corresponding nodes are thus in plastic states. Otherwise, these second terms are all zero and we do not need to update (13), (16), and (17). Also, the stochastic tangent tensor  $\mathcal{G}_T$  of  $\mathcal{G}$  is usually required to solve the linearized equation of the nonlinear problem at each time step. The analytical expression of  $\mathcal{G}_T$  is given by

$$\mathcal{G}_T(\epsilon_i(\theta), \theta; \epsilon_{p,i-1}(\theta), \beta_{i-1}(\theta)) = \frac{\partial \mathcal{G}(\epsilon_i(\theta), \theta; \epsilon_{p,i-1}(\theta), \beta_{i-1}(\theta))}{\partial \epsilon_i(\theta)}, \tag{18}$$

$$= \mathbf{C}(\theta) - \frac{4\mu(\theta)^2}{2\mu(\theta) + \varpi(\theta)} \left[ \mathbf{I}_{\text{dev}} + \left( \frac{1}{\gamma_i(\theta) + \delta(\gamma_i(\theta), 0)} - 1 \right) \frac{\sigma_i^*(\theta) \otimes \sigma_i^*(\theta)}{|\sigma_i^*(\theta)|^2} \right] \langle \gamma_i(\theta) \rangle, \tag{19}$$

where  $\delta(\gamma_i(\theta), 0)$  is the Kronecker delta meeting  $\delta(\gamma_i(\theta), 0) = 1$  for  $\gamma_i(\theta) = 0$  and 0 for others. Similarly,  $\mathcal{G}_T = \mathbf{C}(\theta)$  holds and we do not need to update  $\mathcal{G}_T$  when  $|\sigma_i^*(\theta)| \leq \sigma_Y(\theta)$ . In the following section, we will show that the proposed method can accurately perform the judgment  $|\sigma_i^*(\theta)| > \sigma_Y(\theta)$  (and  $|\sigma_i^*(\theta)| \leq \sigma_Y(\theta)$ ) for each sample realization of  $\theta$  for each sample realization of  $\theta$ . Therefore, the way similar to deterministic elastoplastic analysis can still be adopted although stochastic spaces are involved in stochastic elastoplastic problems.

### 3 | STOCHASTIC NEWTON-BASED NONLINEAR STOCHASTIC FINITE ELEMENT METHOD

By use of the above relations and the finite element method for spatial discretization, we can obtain a random parameter-dependent nonlinear finite element equation, also known as the nonlinear stochastic finite element equation. In this section, we give an efficient and accurate numerical algorithm to solve nonlinear stochastic finite element equations.

#### 3.1 | Stochastic Newton linearization

For simplicity, we write the nonlinear stochastic finite element equation obtained by the elastoplastic problem (1) as an abstract form,

$$\mathcal{N}(\mathbf{u}(t, \theta), \theta) = \mathbf{F}(t, \theta) \tag{20}$$

with the nonlinear operator  $\mathcal{N} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ , the stochastic solution  $\mathbf{u}(t, \theta) \in \mathbb{R}^{n \times n_t}$ , the stochastic force  $\mathbf{F}(t, \theta) \in \mathbb{R}^{n \times n_t}$  and  $n$  is the number of degrees of freedom of the spatial discretization,  $n_t$  is the number of total time steps. In practical numerical implementation, the matrix/vector implementations of  $\mathcal{N}(\mathbf{u}(t, \theta), \theta)$  and  $\mathbf{F}(t, \theta)$  are calculated by using (6) and (7), respectively. To solve the stochastic solution of (20), we extend the Newton (or called Newton–Raphson)

method for solving deterministic nonlinear equations to stochastic cases. A stochastic Newton method is used to linearize (20) as

$$\mathbf{K}_T(\mathbf{u}_{k-1}(t_i, \theta), \theta) \Delta \mathbf{u}_k(t_i, \theta) = \mathbf{F}_k(\mathbf{u}_{k-1}(t_i, \theta), \theta), \quad \forall \theta \in \Theta \quad (21)$$

for  $i = 1, \dots, n_t$ , where the stochastic tangent stiffness matrix  $\mathbf{K}_T(\mathbf{u}_{k-1}(t_i, \theta), \theta) \in \mathbb{R}^{n \times n}$  and the stochastic force vector  $\mathbf{F}_k(\mathbf{u}_{k-1}(t_i, \theta), \theta) \in \mathbb{R}^n$  are given by

$$\mathbf{K}_T(\mathbf{u}_{k-1}(t_i, \theta), \theta) = \left[ \partial \mathcal{N}(\mathbf{u}) / \partial \mathbf{u} \right] \Big|_{\mathbf{u}=\mathbf{u}_{k-1}(t_i, \theta)}, \quad (22)$$

$$\mathbf{F}_k(\mathbf{u}_{k-1}(t_i, \theta), \theta) = \mathbf{F}(t_i, \theta) - \mathcal{N}(\mathbf{u}_{k-1}(t_i, \theta), \theta), \quad (23)$$

and the matrix assembly of  $\mathbf{K}_T(\mathbf{u}_{k-1}(t_i, \theta), \theta)$  (or  $\partial \mathcal{N}(\mathbf{u}) / \partial \mathbf{u}$ ) is implemented by using the tangent tensor  $\mathcal{G}_T$  in (19). In this way, the stochastic solution  $\mathbf{u}_k(t_i, \theta)$  of the  $k$ th iteration at the time step  $t_i$  in (21) is approximated as

$$\mathbf{u}_k(t_i, \theta) = \mathbf{u}_{k-1}(t_i, \theta) + \Delta \mathbf{u}_k(t_i, \theta), \quad (24)$$

where  $\mathbf{u}_{k-1}(t_i, \theta) \in \mathbb{R}^n$  is the stochastic solution of the previous iteration that has been determined and  $\Delta \mathbf{u}_k(t_i, \theta) \in \mathbb{R}^n$  is the  $k$ th stochastic increment that needs to be solved.

### 3.2 | Approximation of stochastic solutions

In practice, (24) is only of theoretical significance since  $\Delta \mathbf{u}_k(t_i, \theta)$  is not as easy to be solved as deterministic cases. On the basis of (24), we develop a more detailed approximation for the stochastic solution  $\mathbf{u}(t_i, \theta)$  of each time step  $t_i$ , which is given by

$$\mathbf{u}_k(t_i, \theta) = \tilde{\mathbf{u}}(t_{i-1}, \theta) + \sum_{j=1}^k \lambda_j(t_i, \theta) \mathbf{d}_j(t_i), \quad (25)$$

$$= \tilde{\mathbf{u}}(t_{i-1}, \theta) + \mathbf{D}(t_i) \boldsymbol{\Lambda}(t_i, \theta), \quad (26)$$

where  $\tilde{\mathbf{u}}(t_{i-1}, \theta)$  is the stochastic solution of the time step  $t_{i-1}$  and has been determined,  $\lambda_j(t_i, \theta) \in \mathbb{R}$  are random variables,  $\boldsymbol{\Lambda}(t_i, \theta) = [\lambda_1(t_i, \theta), \dots, \lambda_k(t_i, \theta)]^T \in \mathbb{R}^k$  is the corresponding random vector,  $\mathbf{d}_j(t_i) \in \mathbb{R}^n$  are deterministic vectors,  $\mathbf{D}(t_i) = [\mathbf{d}_1(t_i), \dots, \mathbf{d}_k(t_i)] \in \mathbb{R}^{n \times k}$  is the corresponding matrix,  $k$  is the number of approximate terms. Equation (25) is a kind of intrusive approximation, but we approximate the random variables  $\lambda_j(t_i, \theta)$  using random samples in practice, which is a non-intrusive way. Combining these two points, (25) is considered as a weakly intrusive approach and we will explain this point in detail in the subsequent section. In fact, the solution  $\mathbf{u}_k(t_i, \theta)$  can be considered as a random field. Equation (25) is very similar to the classical series expansions of the random field  $\mathbf{u}_k(t_i, \theta)$ , for example, the Karhunen–Loève expansion and the Polynomial Chaos expansion<sup>11</sup> that decompose the random fields into stochastic and deterministic spaces. Different from the known probability properties such as mean values and covariance functions of the classical random fields, the constraints of  $\mathbf{u}_k(t_i, \theta)$  are only from the stochastic governing Equation (20). In this way, all quantities  $\{\lambda_j(t_i, \theta), \mathbf{d}_j(t_i)\}_{j=1}^k$  cannot be known a priori. The random variables  $\{\lambda_j(t_i, \theta)\}_{j=1}^k$  may no longer be uncorrelated as they are in the Karhunen–Loève expansion. We need to address this issue and develop dedicated numerical methods for the purpose. The convergence and accuracy of (25) may benefit from these classical expansions, but are not well understood currently. Further investigation is still required. Several explorations for solving linear static stochastic problems, linear dynamic stochastic problems and the problems defined on random domains using (25)-like expansions can be found in our previous work.<sup>30,31,33</sup> We also remark that the second part-like expansion in (25) (i.e.,  $\sum_{j=1}^k \lambda_j(t_i, \theta) \mathbf{d}_j(t_i)$ ) has been widely used in the context of proper generalized decomposition methods.<sup>34–36</sup> However, these classical expansions are not applicable to the stochastic elastoplastic problem. In this article, we develop an efficient and accurate numerical algorithm to solve stochastic elastoplastic problems by using the weakly intrusive approximation (25) and a dedicated iteration.



Combining (24) and (25) we rewrite (25) as an incremental format similar to (24). To this end, the previous approximation  $\mathbf{u}_{k-1}(t_i, \theta)$  and the stochastic increment  $\Delta \mathbf{u}_k(t_i, \theta)$  in (24) are given by

$$\mathbf{u}_{k-1}(t_i, \theta) = \tilde{\mathbf{u}}(t_{i-1}, \theta) + \sum_{j=1}^{k-1} \lambda_j(t_i, \theta) \mathbf{d}_j(t_i), \tag{27}$$

$$\Delta \mathbf{u}_k(t_i, \theta) = \lambda_k(t_i, \theta) \mathbf{d}_k(t_i), \tag{28}$$

where the first  $k - 1$  couples  $\{\lambda_j(t_i, \theta), \mathbf{d}_j(t_i)\}_{j=1}^{k-1}$  are assumed to be known. In this way, we can still solve the unknown stochastic increment by taking advantage of the stochastic Newton method mentioned in the previous section and the stochastic increment is given by  $\lambda_k(t_i, \theta) \mathbf{d}_k(t_i)$ . Substituting (27) and (28) into (21) we have

$$\mathbf{K}_T(\mathbf{u}_{k-1}(t_i, \theta), \theta) \lambda_k(t_i, \theta) \mathbf{d}_k(t_i) = \mathbf{F}_k(\mathbf{u}_{k-1}(t_i, \theta), \theta). \tag{29}$$

An iterative algorithm for solving the couple  $\{\lambda_k(t_i, \theta), \mathbf{d}_k(t_i)\}$  will be discussed in the next section.

### 3.3 | Iterative algorithm for stochastic solutions

It is not easy to calculate  $\lambda_k(t_i, \theta)$  and  $\mathbf{d}_k(t_i)$  in (29) at once. We adopt an alternating iterative method to solve  $\lambda_k(t_i, \theta)$  and  $\mathbf{d}_k(t_i)$  one by one. Specifically, we are to fix one of them and solve the other, and then the fixed one is updated according to the solution of the other. In detail, if the random variable  $\lambda_k(t_i, \theta)$  has been known (or given an initial value), the stochastic Galerkin method<sup>11,12</sup> is used to transform the original stochastic problem (29) into a deterministic finite element equation

$$\mathbf{K}_{T,k}^*(t_i) \mathbf{d}_k(t_i) = \mathbf{F}_k^*(t_i), \tag{30}$$

where the deterministic matrix  $\mathbf{K}_{T,k}^*(t_i) \in \mathbb{R}^{n \times n}$  and the deterministic vector  $\mathbf{F}_k^*(t_i) \in \mathbb{R}^n$  are calculated by

$$\mathbf{K}_{T,k}^*(t_i) = \mathbb{E} \left\{ \lambda_k^2(t_i, \theta) \mathbf{K}_T(\mathbf{u}_{k-1}(t_i, \theta), \theta) \right\}, \tag{31}$$

$$\mathbf{F}_k^*(t_i) = \mathbb{E} \left\{ \lambda_k(t_i, \theta) \mathbf{F}_k(\mathbf{u}_{k-1}(t_i, \theta), \theta) \right\}, \tag{32}$$

where  $\mathbb{E} \{ \cdot \}$  represents the expectation operator. It is noted that the above equations are the approximations of the probability integrals  $\int_{\Theta} \lambda_k^2(t_i, \theta) \mathbf{K}_T(\mathbf{u}_{k-1}(t_i, \theta), \theta) d\mathcal{P}(\theta)$  and  $\int_{\Theta} \lambda_k(t_i, \theta) \mathbf{F}_k(\mathbf{u}_{k-1}(t_i, \theta), \theta) d\mathcal{P}(\theta)$ , where  $\mathcal{P}(\theta)$  is the probability measurement of the random input. In this way, the probability integrals can be achieved cheaply, but theoretically, high-accuracy approximation requires a large number of random samples. In numerical examples, we will show that the sample size requirement can be greatly relaxed, that is, only a small number of random samples are used to perform (31) and (32), but high-accuracy stochastic solutions are still obtained. Equation (30) is a linear finite element equation and existing linear solvers<sup>4</sup> can be adopted to efficiently solve the vector  $\mathbf{d}_k(t_i)$ . To speed up the convergence in practical computations, we orthogonalize the vector  $\mathbf{d}_k(t_i)$  by Gram–Schmidt orthogonalization, which corresponds to

$$\mathbf{d}_k(t_i) = \mathbf{d}_k(t_i) - \sum_{j=1}^{k-1} \frac{\mathbf{d}_k^T(t_i) \mathbf{d}_j(t_i)}{\mathbf{d}_j^T(t_i) \mathbf{d}_j(t_i)} \mathbf{d}_j(t_i), \tag{33}$$

where  $\{\mathbf{d}_j(t_i)\}_{j=1}^{k-1}$  are orthogonal vectors that have been determined via the first  $k - 1$  iterations and they meet  $\mathbf{d}_m^T(t_i) \mathbf{d}_n(t_i) = \delta_{mn}$ ,  $m, n = 1, \dots, k - 1$ .

According to the deterministic vector  $\mathbf{d}_k(t_i)$  obtained by (30), the random variable  $\lambda_k(t_i, \theta)$  is updated by a classical Galerkin procedure

$$\mathbf{a}_k(t_i, \theta) \lambda_k(t_i, \theta) = \mathbf{b}_k(t_i, \theta), \tag{34}$$

where the random variables  $a_k(t_i, \theta) \in \mathbb{R}$  and  $b_k(t_i, \theta) \in \mathbb{R}$  are given by

$$a_k(t_i, \theta) = \mathbf{d}_k^T(t_i) \mathbf{K}_T(\mathbf{u}_{k-1}(t_i, \theta), \theta) \mathbf{d}_k(t_i), \quad (35)$$

$$b_k(t_i, \theta) = \mathbf{d}_k^T(t_i) \mathbf{F}_k(\mathbf{u}_{k-1}(t_i, \theta), \theta). \quad (36)$$

A common method for solving (34) is known as the PC-based methods.<sup>11,37</sup> In this method, the random variable  $\lambda_k(t_i, \theta)$  is expanded by PC bases and the expanded coefficients are calculated via an augmented deterministic equation. It alleviates the computational burden as the augmented deterministic equation is not coupled with degrees of freedom discretized by the finite element method, but it does not overcome the curse of dimensionality arising in high-dimensional stochastic problems. To avoid this issue, a non-intrusive method is adopted to calculate random samples of the random variable  $\lambda_k(t_i, \theta)$ ,<sup>30,31,33</sup> which corresponds to

$$\lambda_k(t_i, \hat{\theta}) = b_k(t_i, \hat{\theta}) \oslash a_k(t_i, \hat{\theta}) \in \mathbb{R}^{n_s}, \quad (37)$$

where  $\hat{\theta} = \{\theta^{(i)}\}_{i=1}^{n_s}$  is the sample realizations of random sources, the random sample vectors  $\lambda_k(t_i, \hat{\theta})$ ,  $a_k(t_i, \hat{\theta})$ ,  $b_k(t_i, \hat{\theta}) \in \mathbb{R}^{n_s}$  are the sample realizations of the random variables  $\lambda_k(t_i, \theta)$ ,  $a_k(t_i, \theta)$ , and  $b_k(t_i, \theta)$ , respectively,  $\oslash$  is the Hadamard division operator representing the element-wise division of the vectors  $b_k(t_i, \hat{\theta})$  and  $a_k(t_i, \hat{\theta})$ . Based on the random sample vector  $\lambda_k(t_i, \hat{\theta})$ , statistical methods are adopted to provide the probability characteristics of the random variable  $\lambda_k(t_i, \theta)$ . We can calculate the  $k$ th couple  $\{\lambda_k(t_i, \theta), \mathbf{d}_k(t_i)\}$  by iteratively solving (30) and (37) until a specified convergence criterion is satisfied, the details of which will be discussed in Section 4. It is noted that both (30) and (37) are insensitive to stochastic dimensionalities since all random sources are embedded into the random sample vectors  $a_k(t_i, \hat{\theta})$  and  $b_k(t_i, \hat{\theta})$  and the computational effort of (37) is very low, which avoids the curse of dimensionality successfully and will be verified in the numerical example in Section 5.4.

### 3.4 | Improved scheme of stochastic solutions

We recall the approximation (28) and the stochastic increment is approximated by a stochastic-deterministic decoupling decomposition  $\Delta \mathbf{u}_k(t_i, \theta) \approx \lambda_k(t_i, \theta) \mathbf{d}_k(t_i)$ , which has low accuracy in many cases. For the explanation of this point, we consider the sample descriptions  $\Delta \mathbf{u}_k(t_i, \theta) \approx \Delta \mathbf{u}_k(t_i, \hat{\theta}) \in \mathbb{R}^{n \times n_s}$  and  $\lambda_k(t_i, \theta) \approx \lambda_k(t_i, \hat{\theta}) \in \mathbb{R}^{n_s}$ . The approximation (28) is then written as

$$\Delta \mathbf{u}_k(t_i, \hat{\theta}) \approx \mathbf{d}_k(t_i) \lambda_k^T(t_i, \hat{\theta}), \quad (38)$$

which is considered as a rank-1 approximation of the matrix  $\Delta \mathbf{u}_k(t_i, \hat{\theta})$ , thus it is less accurate for many problems. To avoid this problem, we let the known matrix  $\mathbf{D}(t_i)$  in (26) be a set of reduced bases and recalculate the random vector  $\Lambda(t_i, \theta)$ . To solve the random vector  $\Lambda(t_i, \theta)$ , we still adopt the stochastic Newton method for the linearization and the Galerkin method for building a reduced-order linearized stochastic finite element equation,

$$[\mathbf{D}^T(t_i) \mathbf{K}_T(\tilde{\mathbf{u}}(t_{i-1}, \theta) + \mathbf{D}(t_i) \Lambda^{(j)}(t_i, \theta), \theta) \mathbf{D}(t_i)] \Delta \Lambda^{(j)}(t_i, \theta) = \mathbf{D}^T(t_i) \mathbf{F}^{(j)}(\tilde{\mathbf{u}}(t_{i-1}, \theta) + \mathbf{D}(t_i) \Lambda^{(j)}(t_i, \theta), \theta), \quad (39)$$

where the vector  $\Lambda^{(j)}(t_i, \theta) \in \mathbb{R}^k$  is the stochastic solution of the  $j$ th iteration and the vector  $\Delta \Lambda^{(j)}(t_i, \theta) \in \mathbb{R}^k$  is the  $j$ th stochastic increment. To solve the increment  $\Delta \Lambda^{(j)}(t_i, \theta)$ , we use a sampling method to solve (39) for all sample realizations, which is achieved by solving the following equation

$$\tilde{\mathbf{K}}_T(\Lambda^{(j)}(t_i, \theta^{(q)})) \Delta \Lambda^{(j)}(t_i, \theta^{(q)}) = \tilde{\mathbf{f}}^{(j)}(\Lambda^{(j)}(t_i, \theta^{(q)})) \quad (40)$$



for  $q = 1, \dots, n_s$ , where the reduced-order matrix  $\tilde{\mathbf{k}}_{\mathbf{T}}(\Lambda^{(j)}(t_i, \theta^{(q)})) \in \mathbb{R}^{k \times k}$  and the reduced-order vector  $\tilde{\mathbf{f}}^{(j)}(\Lambda^{(j)}(t_i, \theta^{(q)})) \in \mathbb{R}^k$  are given by

$$\tilde{\mathbf{k}}_{\mathbf{T}}(\Lambda^{(j)}(t_i, \theta^{(q)})) = \mathbf{D}^T(t_i) \mathbf{K}_{\mathbf{T}}(\tilde{\mathbf{u}}(t_{i-1}, \theta^{(q)}) + \mathbf{D}(t_i) \Lambda^{(j)}(t_i, \theta^{(q)}), \theta^{(q)}) \mathbf{D}(t_i), \tag{41}$$

$$\tilde{\mathbf{f}}^{(j)}(\Lambda^{(j)}(t_i, \theta^{(q)})) = \mathbf{D}^T(t_i) \mathbf{F}^{(j)}(\tilde{\mathbf{u}}(t_{i-1}, \theta^{(q)}) + \mathbf{D}(t_i) \Lambda^{(j)}(t_i, \theta^{(q)}), \theta^{(q)}). \tag{42}$$

Further, the random vector is updated via

$$\Lambda^{(j+1)}(t_i, \theta^{(q)}) = \Lambda^{(j)}(t_i, \theta^{(q)}) + \Delta \Lambda^{(j)}(t_i, \theta^{(q)}), \quad q = 1, \dots, n_s. \tag{43}$$

The computational effort of (40) is very low since its size  $k$  is usually not too large. For all sample realizations  $\{\Delta \Lambda^{(j)}(t_i, \theta^{(q)})\}_{q=1}^{n_s}$ , total computational costs of  $n_s$  solutions of (40) are still low. It is noted that compared to intrusive methods, the non-intrusive approximation of the random vector  $\Lambda(t_i, \theta)$  more easily and accurately captures the nodes in elastic/plastic states. Also, another advantage of the proposed method is that the initial random samples of random sources are propagated from beginning to the end, that is, the sample realization  $\theta^{(q)}$  is propagated to the final stochastic solution via  $\theta^{(q)} \rightarrow \Lambda(t_i, \theta^{(q)}) \rightarrow \mathbf{u}(t_i, \theta^{(q)})$ , which provides an easy-to-use and highly accurate way for uncertainty quantification.

## 4 | ALGORITHM IMPLEMENTATION

### 4.1 | Computational aspects

In this section, we propose efficient numerical formats to accelerate the numerical computation of the proposed method. Computational costs of the proposed method are concentrated in the assembly of the stochastic tangent stiffness matrix  $\mathbf{K}_{\mathbf{T}}(\mathbf{u}_{k-1}(t_i, \theta), \theta)$  in (29), the expectation calculations in (31) and (32), the vector-stochastic-matrix multiplications in (35) and (36), and the matrix-stochastic-matrix multiplications in (41) and (42). To speed up these computations, we follow the idea for numerical implementation in Reference 38 and write the stochastic tangent stiffness matrix  $\mathbf{K}_{\mathbf{T}}(\mathbf{u}_{k-1}(t_i, \theta), \theta)$  as

$$\mathbf{K}_{\mathbf{T}}(\mathbf{u}_{k-1}(t_i, \theta), \theta) = \mathbf{B}^T \mathbf{M}_{\mathbf{T}}(\mathbf{u}_{k-1}(t_i, \theta), \theta) \mathbf{B}, \tag{44}$$

where  $\mathbf{B} \in \mathbb{R}^{n_1 \times n}$  is a deterministic sparse matrix related to the strain-displacement operator,  $n_1$  is the total number of integration points,  $\mathbf{M}_{\mathbf{T}}(\theta) \in \mathbb{R}^{n_1 \times n_1}$  is a block diagonal sparse matrix related to the stochastic elastoplastic constitutive model. We decompose the stochastic matrix  $\mathbf{M}_{\mathbf{T}}(\theta)$  into two parts

$$\mathbf{M}_{\mathbf{T}}(\mathbf{u}_{k-1}(t_i, \theta), \theta) = \mathbf{M}_{\mathbf{e}}(\theta) + \mathbf{M}_{\mathbf{T,p}}(\mathbf{u}_{k-1}(t_i, \theta), \theta), \tag{45}$$

where the stochastic matrix  $\mathbf{M}_{\mathbf{e}}(\theta) \in \mathbb{R}^{n_1 \times n_1}$  is related to the elastic constitutive part (i.e., linear problem) of the material and the stochastic matrix  $\mathbf{M}_{\mathbf{T,p}}(\mathbf{u}_{k-1}(t_i, \theta), \theta) \in \mathbb{R}^{n_1 \times n_1}$  is related to the elastoplastic constitutive part (i.e., nonlinear problem). The stochastic matrix  $\mathbf{M}_{\mathbf{T,p}}(\mathbf{u}_{k-1}(t_i, \theta), \theta)$  is much sparser than the stochastic matrix  $\mathbf{M}_{\mathbf{T}}(\mathbf{u}_{k-1}(t_i, \theta), \theta)$  since the nodes in plastic states are only a part of the whole domain, or even a small part in most cases. Thus, the stochastic matrix  $\mathbf{K}_{\mathbf{T}}(\mathbf{u}_{k-1}(t_i, \theta), \theta)$  can be rewritten as

$$\mathbf{K}_{\mathbf{T}}(\mathbf{u}_{k-1}(t_i, \theta), \theta) = \mathbf{K}_{\mathbf{e}}(\theta) + \mathbf{K}_{\mathbf{T,p}}(\mathbf{u}_{k-1}(t_i, \theta), \theta), \tag{46}$$

where the stochastic elastic stiffness matrix  $\mathbf{K}_{\mathbf{e}}(\theta) \in \mathbb{R}^{n \times n}$  corresponding to  $\mathbf{M}_{\mathbf{e}}(\theta)$  and the stochastic plastic tangent stiffness matrix  $\mathbf{K}_{\mathbf{T,p}}(\mathbf{u}_{k-1}(t_i, \theta), \theta) \in \mathbb{R}^{n \times n}$  corresponding to  $\mathbf{M}_{\mathbf{T,p}}(\mathbf{u}_{k-1}(t_i, \theta), \theta)$  are

$$\mathbf{K}_{\mathbf{e}}(\theta) = \mathbf{B}^T \mathbf{M}_{\mathbf{e}}(\theta) \mathbf{B}, \quad \mathbf{K}_{\mathbf{T,p}}(\mathbf{u}_{k-1}(t_i, \theta), \theta) = \mathbf{B}^T \mathbf{M}_{\mathbf{T,p}}(\mathbf{u}_{k-1}(t_i, \theta), \theta) \mathbf{B}. \tag{47}$$

Similarly, the stochastic force vector can be written as

$$\mathbf{F}_k(\mathbf{u}_{k-1}(t_i, \theta), \theta) = \mathbf{F}(t_i, \theta) - \mathbf{B}^T \mathbf{F}_1(\mathbf{u}_{k-1}(t_i, \theta), \theta). \quad (48)$$

It is noted that the matrix  $\mathbf{K}_e(\theta)$  can be precomputed via  $\mathbf{M}_e(\theta)$  and does not need to be updated during the iterations, while the matrix  $\mathbf{K}_{T,p}(\mathbf{u}_{k-1}(t_i, \theta), \theta)$  is updated using  $\mathbf{M}_{T,p}(\mathbf{u}_{k-1}(t_i, \theta), \theta)$  on each iteration. Let  $\mathbf{M}_{T,p}^{\text{nonzero}}(\mathbf{u}_{k-1}(t_i, \theta), \theta) \in \mathbb{R}^{n_{ip}(\theta) \times n_{ip}(\theta)}$  be the nonzero submatrix of  $\mathbf{M}_{T,p}(\mathbf{u}_{k-1}(t_i, \theta), \theta)$  (i.e., the stochastic matrix corresponding to plastic points), where  $n_{ip}(\theta)$  is a random variable representing the number of integration points of elements in plastic states. In the numerical implementation, we compute and store  $\mathbf{M}_{T,p}^{\text{nonzero}}(\mathbf{u}_{k-1}(t_i, \theta), \theta)$  via a third-order tensor  $\mathbf{M}_{T,p}^{\text{nonzero}}(\mathbf{u}_{k-1}(t_i, \hat{\theta}), \hat{\theta}) \in \mathbb{R}^{n_{ip}(\theta) \times n_{ip}(\theta) \times n_s}$  and all operations on  $\mathbf{M}_{T,p}(\mathbf{u}_{k-1}(t_i, \theta), \theta)$  can be performed on  $\mathbf{M}_{T,p}(\mathbf{u}_{k-1}(t_i, \hat{\theta}), \hat{\theta})$  (equivalent to  $\mathbf{M}_{T,p}^{\text{nonzero}}(\mathbf{u}_{k-1}(t_i, \hat{\theta}), \hat{\theta})$ ), which only requires little computational effort and storage memory. In this way, we can efficiently implement the proposed method. Specifically, (31) and (32) are calculated via

$$\mathbf{K}_{T,k}^*(t_i) = \mathbf{B}^T \widehat{\mathbb{E}} \left\{ \lambda_k^2(t_i, \hat{\theta}) \mathbf{M}_e(\hat{\theta}) \right\} \mathbf{B} + \mathbf{B}^T \widehat{\mathbb{E}} \left\{ \lambda_k^2(t_i, \hat{\theta}) \mathbf{M}_{T,p}(\mathbf{u}_{k-1}(t_i, \hat{\theta}), \hat{\theta}) \right\} \mathbf{B}, \quad (49)$$

$$\mathbf{F}_k^*(t_i) = \widehat{\mathbb{E}} \left\{ \lambda_k(t_i, \hat{\theta}) \mathbf{F}(t_i, \hat{\theta}) \right\} - \mathbf{B}^T \widehat{\mathbb{E}} \left\{ \lambda_k(t_i, \hat{\theta}) \mathbf{F}_1(\mathbf{u}_{k-1}(t_i, \hat{\theta}), \hat{\theta}) \right\}, \quad (50)$$

where  $\widehat{\mathbb{E}}\{\cdot\}$  represents the operator for estimating expectations using random samples, for instance, we have

$$\widehat{\mathbb{E}} \left\{ \lambda_k^2(t_i, \hat{\theta}) \mathbf{M}_e(\hat{\theta}) \right\} = \frac{1}{n_s} \sum_{q=1}^{n_s} \lambda_k^2(t_i, \theta^{(q)}) \mathbf{M}_e(\theta^{(q)}). \quad (51)$$

Further, the random sample vectors of the random variables  $a_k(t_i, \theta)$  and  $b_k(t_i, \theta)$  in (35) and (36) are calculated by

$$a_k(t_i, \hat{\theta}) = (\mathbf{B} \mathbf{d}_k(t_i))^T \mathbf{M}_e(\hat{\theta}) (\mathbf{B} \mathbf{d}_k(t_i)) + (\mathbf{B} \mathbf{d}_k(t_i))^T \mathbf{M}_{T,p}(\mathbf{u}_{k-1}(t_i, \hat{\theta}), \hat{\theta}) (\mathbf{B} \mathbf{d}_k(t_i)) \in \mathbb{R}^{n_s}, \quad (52)$$

$$b_k(t_i, \hat{\theta}) = \mathbf{d}_k^T(t_i) \mathbf{F}(t_i, \hat{\theta}) - (\mathbf{B} \mathbf{d}_k(t_i))^T \mathbf{F}_1(\mathbf{u}_{k-1}(t_i, \hat{\theta}), \hat{\theta}) \in \mathbb{R}^{n_s}, \quad (53)$$

where the deterministic vector  $\mathbf{B} \mathbf{d}_k(t_i) \in \mathbb{R}^{n_i}$  is only calculated at once. Also, introducing a deterministic matrix  $\mathbf{B}_D(t_i) = \mathbf{B} \mathbf{D}(t_i) \in \mathbb{R}^{n_i \times k}$  we calculate (41) and (42) by

$$\tilde{\mathbf{k}}_T(\Lambda^{(j)}(t_i, \theta^{(q)})) = \mathbf{B}_D^T(t_i) \mathbf{M}_e(\theta^{(q)}) \mathbf{B}_D(t_i) + \mathbf{B}_D^T(t_i) \mathbf{M}_{T,p}(\Lambda^{(j)}(t_i, \theta^{(q)}), \theta^{(q)}) \mathbf{B}_D(t_i), \quad (54)$$

$$\tilde{\mathbf{f}}^{(j)}(\Lambda^{(j)}(t_i, \theta^{(q)})) = \mathbf{D}^T(t_i) \mathbf{F}(t_i, \theta^{(q)}) - \mathbf{B}_D^T(t_i) \mathbf{F}_1^{(j)}(\Lambda^{(j)}(t_i, \theta^{(q)}), \theta^{(q)}). \quad (55)$$

## 4.2 | Algorithm scheme

The proposed nonlinear stochastic finite element method for solving stochastic elastoplastic problems is summarized in Algorithm 1, which consists of three-loop procedures. The outer loop is from step 3 to step 26, which is used to compute the stochastic solution  $\mathbf{u}(t_i, \theta)$  at each time step  $t_i$  and the looping number of which is the same as total time steps  $n_t$ . The middle loop from step 5 to step 20 is used to solve the couples  $\{\lambda_j(t_i, \theta), \mathbf{d}_j(t_i)\}_{j=1}^k$  in (27), and the inner loop from step 8 to step 16 is used to solve each couple  $\{\lambda_k(t_i, \theta), \mathbf{d}_k(t_i)\}$ . Before the outer loop, we precompute the stochastic matrices  $\mathbf{M}_e(\hat{\theta}) \in \mathbb{R}^{n_i \times n_i \times n_{s,1}}$ ,  $\mathbf{M}_e(\hat{\theta}^*) \in \mathbb{R}^{n_i \times n_i \times n_{s,2}}$  and the deterministic matrix  $\mathbf{B} \in \mathbb{R}^{n_i \times n_i}$  in step 1, where the sample size  $n_{s,1}$  is used to solve the deterministic vector  $\{\mathbf{d}_j(t_i)\}_{j=1}^k$  in the middle and inner loops and the sample size  $n_{s,2}$  is adopted to solve the final stochastic solution in the outer loop. Significant computational effort is saved by setting  $n_{s,1} \ll n_{s,2}$ , but good accuracy is still achieved. The performance of this strategy will be illustrated in numerical examples. In step 2, stochastic solutions  $\tilde{\mathbf{u}}^{(1)}(t_0, \hat{\theta})$  and  $\tilde{\mathbf{u}}^{(2)}(t_0, \hat{\theta}^*)$  are initialized for the middle and inner loops and the final solution in the outer loop. To execute the inner loop, a random sample vector  $\lambda_k^{(0)}(t_i, \hat{\theta}) \in \mathbb{R}^{n_{s,1}}$  is initialized in step 7. It is noted

**Algorithm 1.** Algorithm for solving stochastic elastoplastic problems

- 1: Precompute the stochastic matrices  $\mathbf{M}_e(\hat{\theta})$ ,  $\mathbf{M}_e(\hat{\theta}^*)$  and the deterministic matrix  $\mathbf{B}$
- 2: Set stochastic solutions  $\tilde{\mathbf{u}}^{(1)}(t_0, \hat{\theta}) = \mathbf{0} \in \mathbb{R}^{n \times n_{s,1}}$  and  $\tilde{\mathbf{u}}^{(2)}(t_0, \hat{\theta}^*) = \mathbf{0} \in \mathbb{R}^{n \times n_{s,2}}$  of the step  $t_0$
- 3: **for**  $t_i = t_1, \dots, t_{n_t}$  **do**
- 4:     Initialize the stochastic solution  $\mathbf{u}_0(t_i, \hat{\theta}) = \tilde{\mathbf{u}}^{(1)}(t_{i-1}, \hat{\theta})$  of time step  $t_i$
- 5:     **while**  $\epsilon_{u,k} \geq \epsilon_u$  **do**
- 6:         Calculate the sparse stochastic matrix  $\mathbf{M}_{T,p}(\mathbf{u}_{k-1}(t_i, \hat{\theta}), \hat{\theta})$  ( $\Rightarrow \mathbf{K}_{T,p}(\mathbf{u}_{k-1}(t_i, \hat{\theta}), \hat{\theta})$ )
- 7:         Initialize the random samples  $\lambda_k^{(0)}(t_i, \hat{\theta}) = \left\{ \lambda_k^{(0)}(t_i, \theta^{(q)}) \right\}_{q=1}^{n_{s,1}} \in \mathbb{R}^{n_{s,1}}$
- 8:         **while**  $\epsilon_{d,j} \geq \epsilon_d$  **do**
- 9:             Assemble the matrix  $\mathbf{K}_{T,k}^*(t_i)$  and the vector  $\mathbf{F}_k^*(t_i)$  by (49) and (50)
- 10:             Solve the deterministic vector  $\mathbf{d}_k^{(j)}(t_i) \in \mathbb{R}^n$  by (30)
- 11:             Orthogonalize  $\mathbf{d}_k^{(j)}(t_i) \perp \{\mathbf{d}_l(t_i)\}_{l=1}^{k-1}$  and normalize  $\|\mathbf{d}_k^{(j)}(t_i)\| = 1$
- 12:             Calculate the random sample vectors  $a_k(t_i, \hat{\theta}), b_k(t_i, \hat{\theta}) \in \mathbb{R}^{n_{s,1}}$  by (52) and (53)
- 13:             Solve the random sample vector  $\lambda_k^{(j)}(t_i, \hat{\theta}) \in \mathbb{R}^{n_{s,1}}$  by (37)
- 14:             Compute the iterative error  $\epsilon_{d,j}$
- 15:              $j \leftarrow j + 1$
- 16:         **end while**
- 17:         Update the stochastic solution  $\mathbf{u}_k(t_i, \hat{\theta}) = \mathbf{u}_{k-1}(t_i, \hat{\theta}) + \lambda_k(t_i, \hat{\theta}) \mathbf{d}_k(t_i)$
- 18:         Compute the iterative error  $\epsilon_{u,k}$
- 19:          $k \leftarrow k + 1$
- 20:     **end while**
- 21:     Update the random vector  $\Lambda^{(1)}(t_i, \hat{\theta}) \in \mathbb{R}^{k \times n_{s,1}}$  by Algorithm 2
- 22:     Update the stochastic solution  $\tilde{\mathbf{u}}^{(1)}(t_i, \hat{\theta}) = \tilde{\mathbf{u}}^{(1)}(t_{i-1}, \hat{\theta}) + \mathbf{D}(t_i) \Lambda^{(1)}(t_i, \hat{\theta})$
- 23:     Solve the random vector  $\Lambda^{(2)}(t_i, \hat{\theta}^*) \in \mathbb{R}^{k \times n_{s,2}}$  by Algorithm 2
- 24:     Compute the stochastic solution  $\tilde{\mathbf{u}}^{(2)}(t_i, \hat{\theta}^*) = \tilde{\mathbf{u}}^{(2)}(t_{i-1}, \hat{\theta}^*) + \mathbf{D}(t_i) \Lambda^{(2)}(t_i, \hat{\theta}^*)$
- 25:      $t_i \leftarrow t_{i+1}$
- 26: **end for**

that any nonzero vectors of size  $n_{s,1}$  can be used as the initialization and it almost has no influence on the computational accuracy and efficiency of the proposed method. With the known random sample vector, the deterministic vector  $\mathbf{d}_k^{(j)}(t_i)$  is solved in step 10 via linearized finite element equations. By using the Gram–Schmidt orthogonalization (33) in step 11,  $\mathbf{d}_k^{(j)}(t_i)$  is orthogonalized and normalized along the whole iteration. With the obtained vector  $\mathbf{d}_k^{(j)}(t_i)$ , the random sample vector  $\lambda_k^{(j)}(t_i, \hat{\theta})$  of the random variable  $\lambda_k^{(j)}(t_i, \theta)$  is calculated in step 12 by the non-intrusive approach. After the inner loop, the stochastic solution  $\mathbf{u}_k(t_i, \hat{\theta})$  is recursively updated in step 17 in the middle loop. Then we can calculate the sparse stochastic matrix  $\mathbf{M}_{T,p}(\mathbf{u}_{k-1}(t_i, \hat{\theta}), \hat{\theta})$  in step 17 based on the new stochastic solution and start the next inner loop. Based on the known matrix  $\mathbf{D}(t_i)$  obtained by the middle and inner loops, we can generate reduced-order stochastic problems in steps 21 to 24 in the outer loop. The random vectors  $\Lambda^{(1)}(t_i, \hat{\theta})$  and  $\Lambda^{(2)}(t_i, \hat{\theta}^*)$  are recalculated by solving  $n_{s,1}$  and  $n_{s,2}$   $k$ -dimensional nonlinear stochastic finite element equations using Algorithm 2, which simply extends the classical Newton iteration to the reduced-order nonlinear stochastic finite element equations but has better convergence and lower computational effort. Corresponding stochastic solutions  $\tilde{\mathbf{u}}^{(1)}(t_i, \hat{\theta})$  and  $\tilde{\mathbf{u}}^{(2)}(t_i, \hat{\theta}^*)$  are thus updated. The stochastic solution  $\tilde{\mathbf{u}}^{(1)}(t_i, \hat{\theta})$  is only used to the next middle and inner loops and  $\tilde{\mathbf{u}}^{(2)}(t_i, \hat{\theta}^*)$  is considered as the final stochastic solution of the time step  $t_i$ .

**Algorithm 2.** Algorithm of the reduced-order Newton iteration

---

```

1: for  $q = 1, \dots, n_s$  do
2:   Initialize the solution  $\Lambda(\theta^{(q)}) = \mathbf{0} \in \mathbb{R}^k$ 
3:   while  $\epsilon_{\text{NR},k} \geq \epsilon_{\text{NR}}$  do
4:     Assemble the reduced-order tangent matrix  $\tilde{\mathbf{k}}_T(\Lambda^{(j)}(t_i, \theta)) \in \mathbb{R}^{k \times k}$  and the vector  $\tilde{\mathbf{f}}^{(k)}(\Lambda^{(j)}(t_i, \theta)) \in \mathbb{R}^k$ 
       by (54) and (55)
5:     Solve the increment  $\Delta\Lambda(\theta^{(q)}) \in \mathbb{R}^k$  by (40)
6:     Update the solution  $\Lambda^{(k+1)}(\theta^{(q)}) = \Lambda^{(k)}(\theta^{(q)}) + \Delta\Lambda^{(k)}(\theta^{(q)})$ 
7:     Compute the iterative error  $\epsilon_{\text{NR},k}$ 
8:      $k \leftarrow k + 1$ 
9:   end while
10:   $q \leftarrow q + 1$ 
11: end for

```

---

Two iterative criteria in Algorithm 1 are used to check the convergence, that is,  $\epsilon_{\mathbf{d},j}$  in step 14 and  $\epsilon_{\mathbf{u},k}$  in step 18. The locally iterative error  $\epsilon_{\mathbf{d},j}$  is defined as

$$\epsilon_{\mathbf{d},j} = \frac{\|\mathbf{d}_k^{(j)}(t_i) - \mathbf{d}_k^{(j-1)}(t_i)\|}{\|\mathbf{d}_k^{(j)}(t_i)\|} = 2 - 2\mathbf{d}_k^{(j)\text{T}}(t_i)\mathbf{d}_k^{(j-1)}(t_i), \quad (56)$$

which measures the difference between the vectors  $\mathbf{d}_k^{(j)}(t_i)$  and  $\mathbf{d}_k^{(j-1)}(t_i)$  and the calculation is stopped when  $\mathbf{d}_k^{(j)}(t_i)$  is almost the same as  $\mathbf{d}_k^{(j-1)}(t_i)$ . Similarly, the globally iterative error  $\epsilon_{\mathbf{u},k}$  is defined as

$$\epsilon_{\mathbf{u},k} = \frac{\|\mathbf{u}_k(t_i, \hat{\theta}) - \mathbf{u}_{k-1}(t_i, \hat{\theta})\|}{\|\mathbf{u}_k(t_i, \hat{\theta})\|} = \frac{\hat{\mathbb{E}}\left\{\lambda_k^2(t_i, \hat{\theta})\right\}\mathbf{d}_k^{\text{T}}(t_i)\mathbf{d}_k(t_i)}{\sum_{i,j=1}^k \hat{\mathbb{E}}\left\{\lambda_i(t_i, \hat{\theta})\lambda_j(t_i, \hat{\theta})\right\}\mathbf{d}_i^{\text{T}}(t_i)\mathbf{d}_j(t_i)} = \frac{\hat{\mathbb{E}}\left\{\lambda_k^2(t_i, \hat{\theta})\right\}}{\sum_{i=1}^k \hat{\mathbb{E}}\left\{\lambda_i^2(t_i, \hat{\theta})\right\}}, \quad (57)$$

which measures the contribution of the  $k$ th couple  $\{\lambda_k(t_i, \theta), \mathbf{d}_k(t_i)\}$  to the stochastic solution  $\mathbf{u}_k(t_i, \theta)$ . However, (57) is not a good criterion in many cases since the random variables  $\{\lambda_j(t_i, \theta)\}_{j=1}^k$  are calculated in a sequential way and  $\hat{\mathbb{E}}\left\{\lambda_k^2(t_i, \hat{\theta})\right\}$  does not keep decreasing. We avoid this issue by replacing  $\{\lambda_j(t_i, \theta)\}_{j=1}^k$  in (57) with new random variables  $\{\tilde{\lambda}_j(t_i, \theta)\}_{j=1}^k$  that are calculated as following.<sup>31</sup> To this end, we calculate the autocorrelation function of the random vector  $\Lambda(t_i, \theta)$  by

$$\mathbf{C}_{\Lambda\Lambda}(t_i) = \hat{\mathbb{E}}\left\{\Lambda(t_i, \hat{\theta})\Lambda^{\text{T}}(t_i, \hat{\theta})\right\} \in \mathbb{R}^{k \times k}, \quad (58)$$

which is decomposed into

$$\mathbf{C}_{\Lambda\Lambda}(t_i) = \mathbf{Q}(t_i)\mathbf{Z}(t_i)\mathbf{Q}^{\text{T}}(t_i) \quad (59)$$

by the eigen decomposition, where  $\mathbf{Q}(t_i) \in \mathbb{R}^{k \times k}$  is an orthonormal matrix and  $\mathbf{Z}(t_i)$  is a diagonal matrix with elements on the diagonal in descending order. A new random vector is given by  $\tilde{\Lambda}(t_i, \theta) = \mathbf{Q}^{\text{T}}(t_i)\Lambda(t_i, \theta) = [\tilde{\lambda}_1(t_i, \theta), \dots, \tilde{\lambda}_k(t_i, \theta)]^{\text{T}} \in \mathbb{R}^k$  and its autocorrelation function is  $\mathbf{C}_{\tilde{\Lambda}\tilde{\Lambda}}(t_i) = \mathbf{Z}(t_i)$ . Substituting new random variables  $\{\tilde{\lambda}_j(t_i, \theta)\}_{j=1}^k$  into (57) the iterative error  $\epsilon_{\mathbf{u},k}$  becomes as

$$\epsilon_{\mathbf{u},k} = \frac{\hat{\mathbb{E}}\left\{\tilde{\lambda}_k^2(t_i, \theta)\right\}}{\sum_{i=1}^k \hat{\mathbb{E}}\left\{\tilde{\lambda}_i^2(t_i, \theta)\right\}} = \frac{\mathbf{Z}_k(t_i)}{\text{Tr}(\mathbf{Z}(t_i))}, \quad (60)$$

where  $\text{Tr}(\cdot)$  is the trace operator and  $\mathbf{Z}_k(t_i)$  is the  $k$ th diagonal element of the matrix  $\mathbf{Z}(t_i)$ . In this way, the iterative error  $\epsilon_{\mathbf{u},k}$  keeps decreasing as the retained item  $k$  increases. More details of this improved error estimation can also be found in Reference 31. Also, the iterative error  $\epsilon_{\text{NR},k}$  in step 3 in Algorithm 2 is defined as

$$\epsilon_{\mathbf{u},k} = \frac{\|\Delta\mathbf{\Lambda}^{(k)}(\theta^{(q)})\|}{\|\mathbf{\Lambda}^{(k)}(\theta^{(q)})\|}, \tag{61}$$

which measures the contribution of the  $k$ th increment  $\Delta\mathbf{\Lambda}^{(k)}(\theta^{(q)})$  to the solution  $\mathbf{\Lambda}^{(k)}(\theta^{(q)})$  and the iteration stops when the increment contributes little to the final solution.

Let us revisit the weak intrusiveness of the proposed method. On one hand, the matrix  $\mathbf{K}_{T,k}^*(t_i)$  and the vector  $\mathbf{F}_k^*(t_i)$  assembled in step 9 in Algorithm 1 only involve the sample matrices  $\mathbf{M}_c(\hat{\theta})$ ,  $\mathbf{M}_{T,p}(\mathbf{u}_{k-1}(t_i, \hat{\theta}), \hat{\theta})$  and the sample vectors  $\mathbf{F}(t_i, \hat{\theta})$ ,  $\mathbf{F}_I(\mathbf{u}_{k-1}(t_i, \hat{\theta}), \hat{\theta})$  in (49) and (50), which can be assembled using existing codes in non-intrusive ways (a slight modification taking advantage of sparsity to save memory and accelerate computation). In step 10, existing numerical solvers are used to solve (30), where the matrix  $\mathbf{K}_{T,k}^*(t_i)$  keeps the same size and matrix properties as the original stochastic matrix  $\mathbf{K}_T(\mathbf{u}_{k-1}(t_i, \theta), \theta)$ . On the other hand, the recalculated Equation (40) only requires few effort for assembling the reduced-order matrix and vector in (41) and (42) and its solution by Algorithm 2 is fully non-intrusive. In these senses, we implement the intrusive approximation (25) of the stochastic solution only in weakly intrusive ways and most existing codes can be reused as-is or slightly modified.

## 5 | NUMERICAL EXAMPLES

In this section, we test the proposed nonlinear stochastic finite element algorithm by a two-dimensional stochastic elastoplastic problem. The convergence errors are set as  $\epsilon_d = 1 \times 10^{-3}$ ,  $\epsilon_{\mathbf{u}} = 1 \times 10^{-6}$  in Algorithm 1 and  $\epsilon_{\text{NR}} = 1 \times 10^{-10}$  in Algorithm 2. For all examples, the sample sizes in step 2 in Algorithm 1 are set as  $n_{s,1} = 100$  and  $n_{s,2} = 1 \times 10^4$ , respectively. Reference solutions are obtained by using  $1 \times 10^4$  MC simulations. To eliminate the influence caused by sampling processes, the same  $1 \times 10^4$  random samples are used for both the proposed SFEM and MC simulations. All examples are performed on a laptop (dual-core, Intel Core i7, 2.40 GHz).

### 5.1 | Problem setting

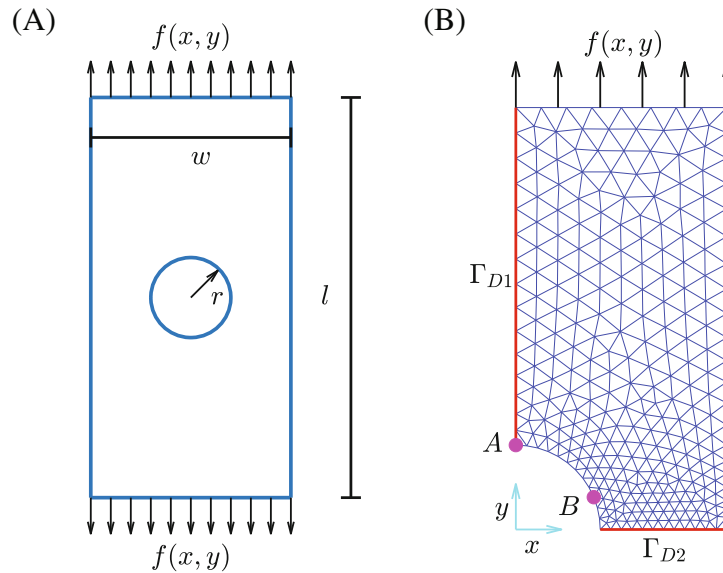
We consider a plate with a hole shown in Figure 1A, where the length  $l = 2$  m and the width  $w = 1$  m. The radius of the circle is  $r = 0.2$  m. Vertical forces  $f(x, y)$  are applied on two sides. We only consider a quarter plate shown in Figure 1B for the analysis due to the symmetry. It is noted that we need to assume that the random material properties have the same symmetry as the geometry. Otherwise, we can only perform the analysis on the whole domain instead of the quarter. By means of finite element discretization,  $n_p = 292$  nodes and  $n_e = 514$  linear triangular elements are generated, and  $n = 584$  degrees of freedom are involved in total. The stochastic displacement in the  $x$  direction  $\mathbf{u}_x(t, \theta) = 0$  holds on the boundary  $\Gamma_{D1}$  and the stochastic displacement in the  $y$  direction  $\mathbf{u}_y(t, \theta) = 0$  holds on the boundary  $\Gamma_{D2}$ .

We only consider a deterministic hardening parameter  $\varpi(\theta) = 1 \times 10^4$  Pa in (11), but it is not a limitation of the proposed method. A random variable/field  $\varpi(\theta)$  is easy to be embedded into the proposed computational framework. The stochastic bulk modulus  $\kappa(\theta)$  and the stochastic shear modulus  $\mu(\theta)$  in (10) are given by

$$\kappa(\theta) = \frac{E(x, y, \theta)}{3(1 - 2\nu)}, \quad \mu(\theta) = \frac{E(x, y, \theta)}{2(1 + \nu)}, \tag{62}$$

where the Poisson rate  $\nu = 0.29$ , the Young's modulus  $E(x, y, \theta)$  is a Gaussian random field with the mean value  $E_0(x, y) = 2.07 \times 10^5$  Pa and the covariance function

$$\text{Cov}(x_1, y_1; x_2, y_2) = \sigma_{\text{Cov}}^2 \exp\left(-\frac{|x_2 - x_1|}{l_x} - \frac{|y_2 - y_1|}{l_y}\right), \tag{63}$$



**FIGURE 1** Model setting: (A) Model of the plate with a hole (B) finite element mesh of the quarter plate and the two reference point A and B. (A) Physical model. (B) Finite element mesh.

where the standard deviation  $\sigma_{\text{Cov}} = 0.1E_0(x, y)$  and the correlation lengths  $l_x = l_y = 1$  m. By using the Karhunen–Loève (KL) expansion,<sup>39,40</sup> the random field  $E(x, y, \theta)$  is approximated as

$$E(x, y, \theta) = E_0(x, y) + \sum_{j=1}^{r_E} \xi_j(\theta) \sqrt{\chi_{E,j}} E_j(x, y), \quad (64)$$

where  $r_E$  is the number of the truncation,  $\{\xi_j(\theta)\}_{j=1}^{r_E}$  are mutually independent standard Gaussian random variables, and  $\{\chi_{E,j}, E_j(x, y)\}_{j=1}^{r_E}$  are eigenvalues and eigenfunctions of the covariance function  $\text{Cov}(x_1, y_1; x_2, y_2)$ . They are solved by the following homogeneous Fredholm integral equation of the second kind,

$$\int_{\Omega} \text{Cov}(x_1, y_1; x_2, y_2) E_j(x_1, y_1) dx_1 dy_1 = \chi_{E,j} E_j(x_2, y_2). \quad (65)$$

Similarly, the yield stress  $\sigma_Y(x, y, \theta)$  is a Gaussian random field with the mean value  $\sigma_{Y,0}(x, y) = 250$  Pa and the covariance function given in (63). It is also approximated as

$$\sigma_Y(x, y, \theta) = \sigma_{Y,0}(x, y) + \sum_{i=1}^{r_{\sigma_Y}} \eta_i(\theta) \sqrt{\chi_{\sigma_Y,i}} \sigma_{Y,i}(x, y), \quad (66)$$

where the standard deviation  $\sigma_{\text{Cov}} = 0.1\sigma_{Y,0}(x, y)$  and  $\{\eta_i(\theta)\}_{i=1}^{r_{\sigma_Y}}$  are mutually independent standard Gaussian random variables and they are independent of the random variables  $\{\xi_i(\theta)\}_{i=1}^{r_E}$ . We let the number of the truncation  $r_{\sigma_Y} = r_E = 10$  and the stochastic dimensionalities are 20 in total. Studies on the truncation error can be found in Reference 25, which is beyond the scope of this article. According to (65),  $\chi_{\sigma_Y,i} = \chi_{E,i}$  and  $\sigma_{Y,i}(x, y) = E_i(x, y)$ ,  $i = 1, \dots, r_E$  hold. The samples  $\theta^{(i)}$  such that  $\min_{x,y \in \Omega} E(x, y, \theta^{(i)}) \leq 1 \times 10^{-3}$  and  $\min_{x,y \in \Omega} \sigma_Y(x, y, \theta^{(i)}) \leq 1 \times 10^{-3}$  are dropped out in practical implementation. Furthermore, in many applications, the modified exponential covariance kernel is suggested to simulate random fields due to its good properties.<sup>41,42</sup> We only consider the classical exponential covariance kernel in this article. It is easy to consider the modified exponential covariance kernel in the proposed SFEM by a slight modification of (63).

According to the KL expansion (64), the stochastic matrix  $\mathbf{M}_e(\theta)$  in (45) has a separated form

$$\mathbf{M}_e(\theta) = \sum_{i=0}^{r_E} \xi_i(\theta) \mathbf{M}_{e,i}. \quad (67)$$



The elastic stochastic stiffness matrix  $\mathbf{K}_e(\theta)$  is thus calculated by

$$\mathbf{K}_e(\theta) = \mathbf{B}^T \mathbf{M}_e(\theta) \mathbf{B} = \sum_{i=0}^{r_E} \xi_i(\theta) \mathbf{K}_{e,i}, \tag{68}$$

where the deterministic matrices  $\mathbf{K}_{e,i} = \mathbf{B}^T \mathbf{M}_{e,i} \mathbf{B} \in \mathbb{R}^{n \times n}$ ,  $i = 0, 1, \dots, r_E$ . The matrices in (49), (52), and (54) are then computed via

$$\mathbf{B}^T \mathbb{E} \left\{ \lambda_k^2(t_j, \hat{\theta}) \mathbf{M}_e(\hat{\theta}) \right\} \mathbf{B} = \sum_{i=0}^{r_E} \mathbb{E} \left\{ \lambda_k^2(t_j, \hat{\theta}) \xi_i(\hat{\theta}) \right\} \mathbf{K}_{e,i}, \tag{69}$$

$$(\mathbf{B} \mathbf{d}_k(t_i))^T \mathbf{M}_e(\hat{\theta}) (\mathbf{B} \mathbf{d}_k(t_i)) = \sum_{i=0}^{r_E} \xi_i(\hat{\theta}) (\mathbf{d}_k^T(t_i) \mathbf{K}_{e,i} \mathbf{d}_k(t_i)), \tag{70}$$

$$\mathbf{B}_D^T(t_i) \mathbf{M}_e(\theta) \mathbf{B}_D(t_i) = \sum_{i=0}^{r_E} \xi_i(\theta) (\mathbf{D}^T(t_i) \mathbf{K}_{e,i} \mathbf{D}(t_i)). \tag{71}$$

In this way, we can deal with the stochastic elastic part of the nonlinear problem by precomputing and storing the matrices  $\mathbf{B}$  and  $\{\mathbf{K}_{e,i}\}_{i=0}^{r_E}$ . Only the stochastic matrix  $\mathbf{M}_p(\theta)$  needs to be assembled in each iteration, which saves a lot of computational effort and storage memory.

### 5.2 | Example 1: Subjected to a monotonic load

In this case, we consider the force  $f(x, y) = 200\text{N}$ , which is loaded monotonically in  $n_t = 11$  time steps  $\{t_0, t_1, \dots, t_{10}\}$ . For each time step, the iterative errors  $\epsilon_{\mathbf{u},k}$  are shown in Figure 2, which demonstrates that the proposed method has good convergence. For all time steps,  $k = 6$  to 12 retained terms  $\{\lambda_j(t_i, \theta), \mathbf{d}_j(t_i)\}_{j=1}^k$  can achieve high-accuracy approximations of the stochastic solutions  $\mathbf{u}(t_i, \theta)$ . For the final stochastic solutions and the post-processing, 90 deterministic vectors  $\mathbf{d}_j(t_i) \in \mathbb{R}^{584}$  and 90 sample vectors  $\lambda_j(t_i, \hat{\theta}) \in \mathbb{R}^{10^4}$  are required to store. The first six deterministic vectors  $\{\mathbf{d}_j(t_i)\}_{j=1}^6$  in the  $y$  direction at the time steps  $t_1$  and  $t_{10}$  are depicted in the first and second lines of Figure 3, respectively. For the time step  $t_1$ , all elements are in elastic states. The vectors  $\{\mathbf{d}_j(t_1)\}_j$  only involve global behavior and they are not used to capture local nonlinear properties. Further, the vectors  $\{\mathbf{d}_j(t_1)\}_j$  can be used to approximate the stochastic solution as long as all elements are in elastic states. In other words, only one set of vectors  $\{\mathbf{d}_j(t_1)\}_j$  is sufficient to approximate all stochastic solutions of stochastic elastic problems, which is independent of the time step. As a comparison, the vectors  $\{\mathbf{d}_j(t_{10})\}_j$

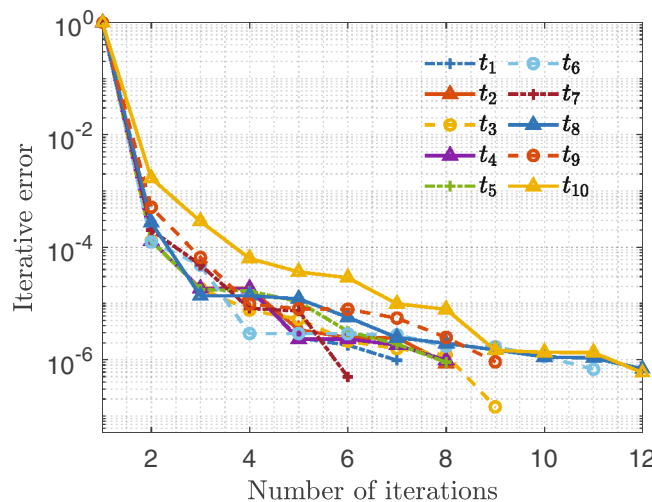


FIGURE 2 Iterative errors of different time steps.

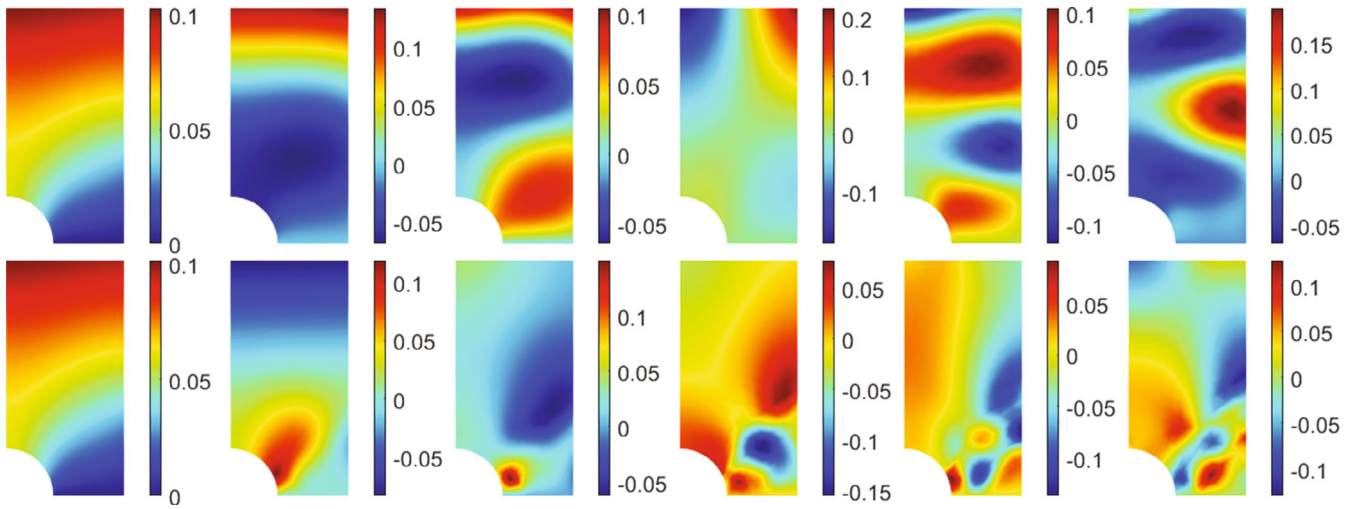


FIGURE 3 The first six vectors  $\{\mathbf{d}_j(t_i)\}_{j=1}^6$  in the  $y$  direction at the time step  $t_1$  (the first line) and the time step  $t_{10}$  (the second line).

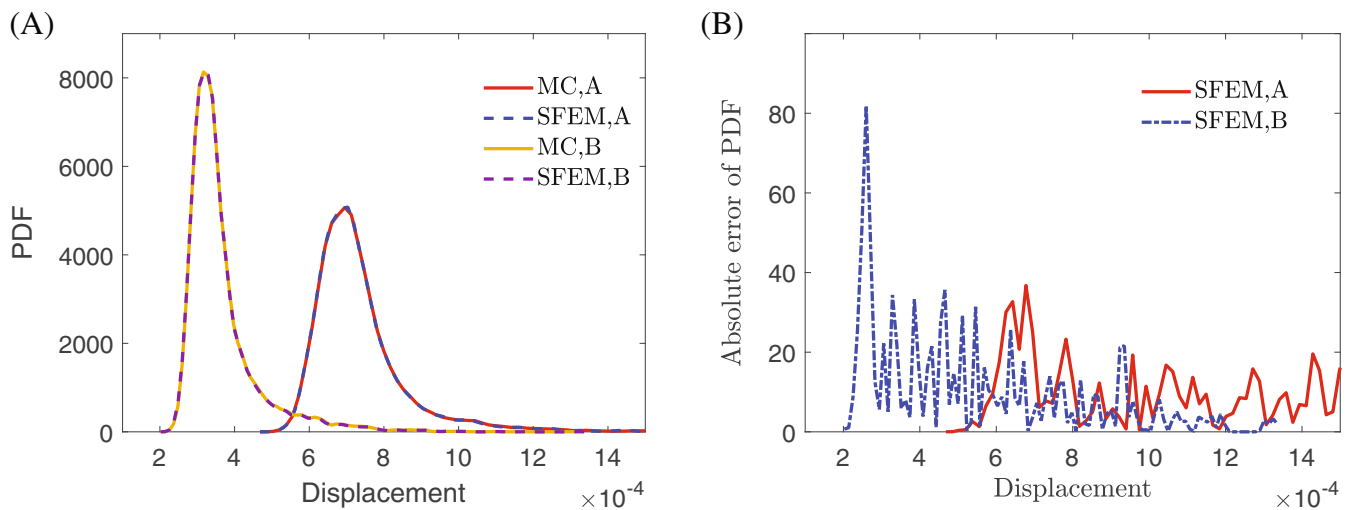
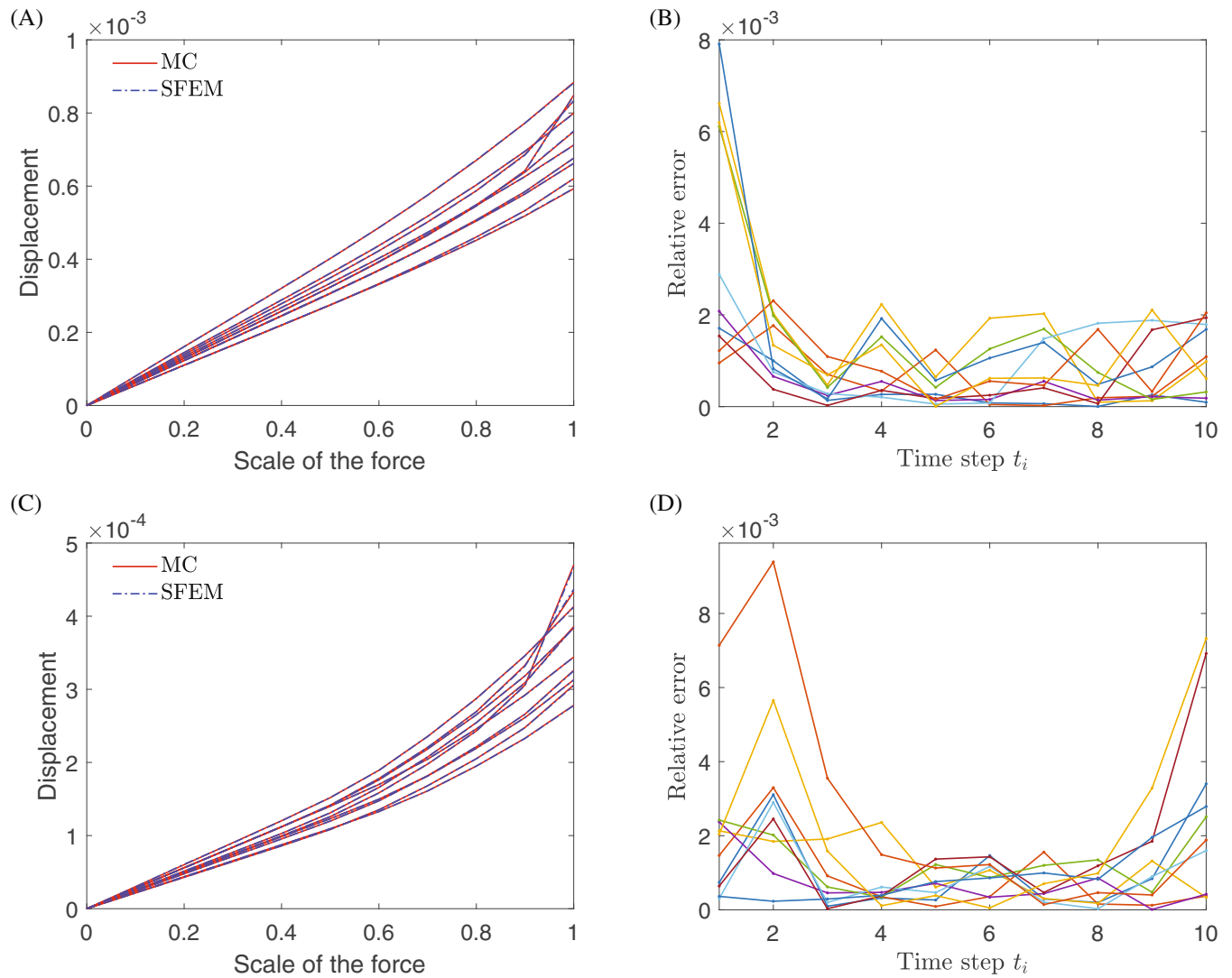


FIGURE 4 PDFs of displacements of the points  $A$  and  $B$  in the  $y$  direction at the time step  $t_{10}$  obtained by the proposed SFEM and MCS. (A) PDFs obtained by the proposed SFEM and MCS. (B) Absolute errors between the proposed SFEM and MCS.

involve local nonlinear properties to capture the stochastic solutions in plastic states. They are different for different time steps since more elements are in plastic states as the time step increases.

To verify the computational accuracy of the proposed method, probability density functions (PDFs) of the points  $A$  and  $B$  (shown in Figure 1B) obtained by the proposed nonlinear SFEM and MCS are compared in Figure 4A, which indicates that the PDFs obtained by the proposed method are very consistent with that of MCS. To quantify the approximate errors of PDFs, the absolute errors between the proposed SFEM and MCS are given in Figure 4B. Since the values of tails of the PDFs are very close to zero, relative errors are not suitable to describe the approximate errors for whole PDFs. The relative error around the maximum PDF (the corresponding displacement is about  $7.0 \times 10^{-4}$ ) of the point  $A$  can be roughly estimated to be 1%, and that of the point  $B$  (the corresponding displacement is about  $3.2 \times 10^{-4}$ ) is less than 1%, which indicates the high accuracy of the proposed method. We highlight that the proposed method can also be used as a stochastic reduced-order method. To show this point, ten sample realizations of displacement-load curves of the points  $A$  and  $B$  obtained by the reduced basis  $\{\mathbf{d}_j(t_i)\}_{j=1}^6$  and MC simulations (i.e., full-order equations) are depicted in Figure 5A,C, respectively, and the corresponding errors relative to MCS are shown in Figure 5B,D. The maximum relative errors of both point  $A$  and point  $B$  are less than 0.01%, which demonstrates the high-accuracy approximation of the reduced-order equations.



**FIGURE 5** Ten sample realizations of displacements of the points *A* and *B* in the *y* direction. (A) Sample realizations of the displacement of the point *A*. (B) Relative errors relative to the displacements from MCS. (C) Sample realizations of the displacement of the point *B*. (D) Relative errors relative to the displacements from MCS.

A fine mesh including  $n_p = 722$  nodes and  $n_e = 1331$  elements is further solved by using the proposed nonlinear SFEM. PDFs and sample realizations of similar accuracy in Figures 4 and 5 are achieved. For the two meshes, the numbers of retained terms at different time steps are compared in Figure 6. It is seen that the numbers of retained terms have no big difference in the two cases. For the coarse mesh, the computational time of the proposed SFEM is 126.83 s, including the solving time 20.74 s for calculating the matrices  $\{\mathbf{D}(t_i)\}_i$  (i.e., the total time from the step 4 to the step 22 in Algorithm 1) and the updating time 106.09 s for solving the final stochastic solutions (i.e., the total time of the steps 23 and 24 in Algorithm 1), and the computational times of MCS is 1689.69 s, which indicates that the proposed SFEM saves a lot of computational time compared to MCS. As a comparison, the solving time and the updating time for the fine mesh are 95.13 and 265.89 s, respectively. Since the sizes of the stochastic matrix  $\mathbf{M}_T(\mathbf{u}_{k-1}(t_i, \theta), \theta)$  and the finite element Equation (30) of the fine mesh are larger than that of the coarse mesh, higher computational effort is required for the fine mesh.

### 5.3 | Example 2: Subjected to a cyclic load

In this case we consider a cyclic load shown in Figure 7 and  $n_t = 41$  time steps  $\{t_0, t_1, \dots, t_{40}\}$  are adopted for the time discretization. The numbers of retained terms at different time steps  $\{t_i\}_{i=1}^{40}$  are listed in Figure 8. It is seen that more retained

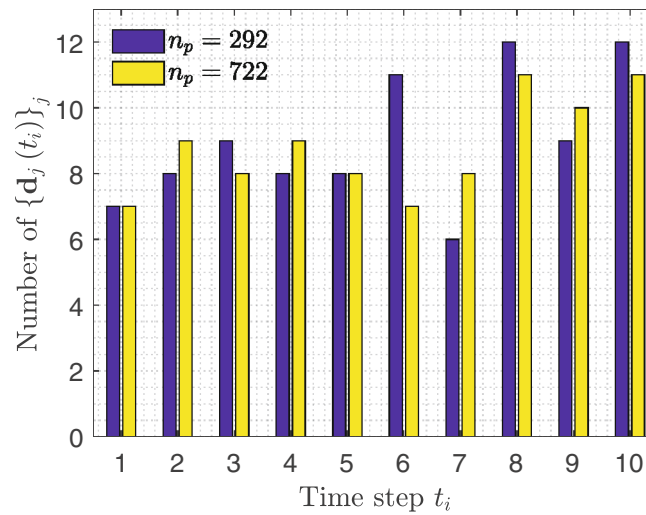


FIGURE 6 Comparison of the numbers of retained terms at different time steps for models with 292 and 722 nodes.

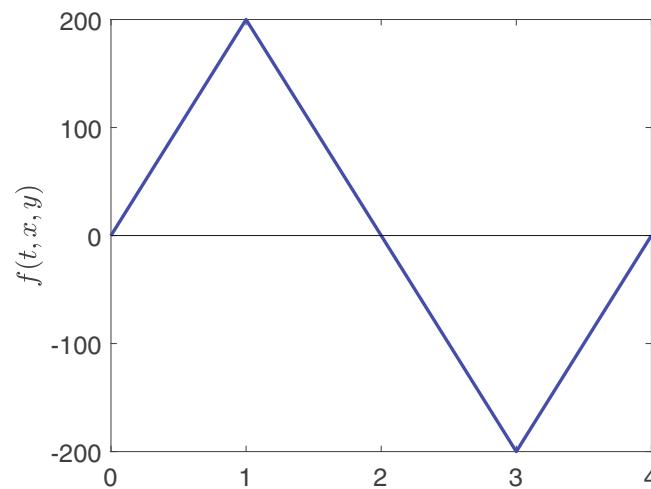


FIGURE 7 History of the cyclic load.

terms are required for the time steps close to  $t_{10}$  and  $t_{30}$ . Since the time steps  $t_{10}$  and  $t_{30}$  correspond to the maximum and the minimum loads, more terms are needed to approximate the strongly nonlinear stochastic solutions. As comparisons, we need less retained terms to approximate the stochastic solution of the step times close to  $t_{20}$  and  $t_{40}$  since the corresponding loads are very small (even zeros when  $t_{20}$  and  $t_{40}$ ). The first six vectors  $\{\mathbf{d}_j(t_i)\}_{j=1}^6$  at the time steps  $t_{30}$  and  $t_{40}$  are depicted in the first and second lines of Figure 9. Comparing the time steps  $t_{10}$  and  $t_{30}$ , although the two cases are subjected to loads with the same absolute value, the vectors  $\{\mathbf{d}_j(t_{10})\}_j$  shown in the second line of Figure 3 and  $\{\mathbf{d}_j(t_{30})\}_j$  shown in the first line of Figure 9 have different modes due to the effects of the plastic strains. For the time step  $t_{40}$ , the stochastic solution is linear in most regions and nonlinearities only occur in the regions with stochastic residual plastic strains. In this case, the vectors  $\{\mathbf{d}_j(t_{40})\}_j$  shown in the second line of Figure 9 are only used to capture the stochastic displacements caused by the stochastic residual plastic strains, and its number is only slightly more than the number of vectors for approximating stochastic solutions of elastic states.

The PDFs of the displacements of the points A and B in the y direction at the time step  $t_{40}$  are shown in Figure 10A and the corresponding absolute errors are shown in Figure 10B. The relative errors around the magnitudes of PDFs (the corresponding displacements are about  $-4.5 \times 10^{-5}$  for the point A and  $-6.3 \times 10^{-5}$  for the point B) are less than 2.5% and 2.0%, respectively, which indicates that the proposed SFEM still has high accuracy for the case under cyclic loads. But the errors are slightly increased than that of the case under monotonic loads illustrated in Example 5.2. Furthermore, ten

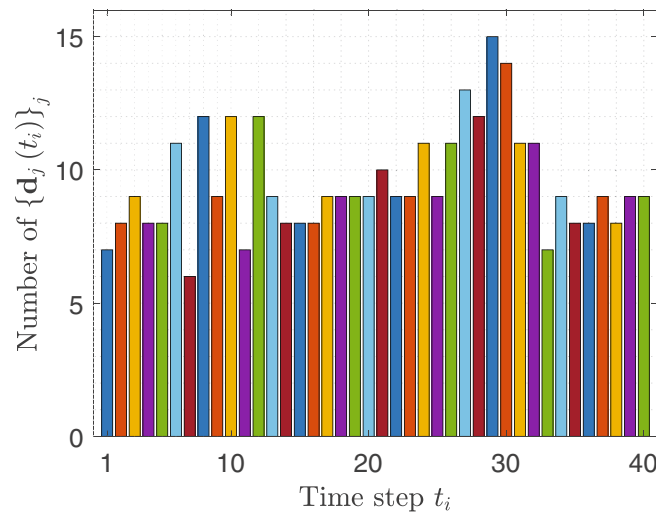


FIGURE 8 The numbers of retained terms at different time steps.

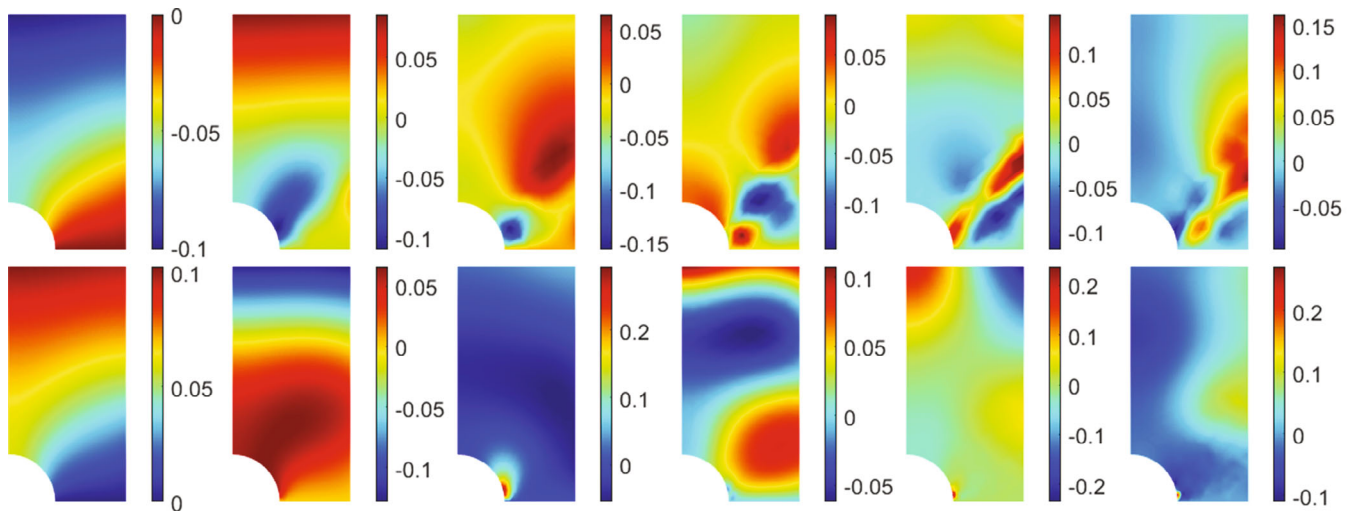


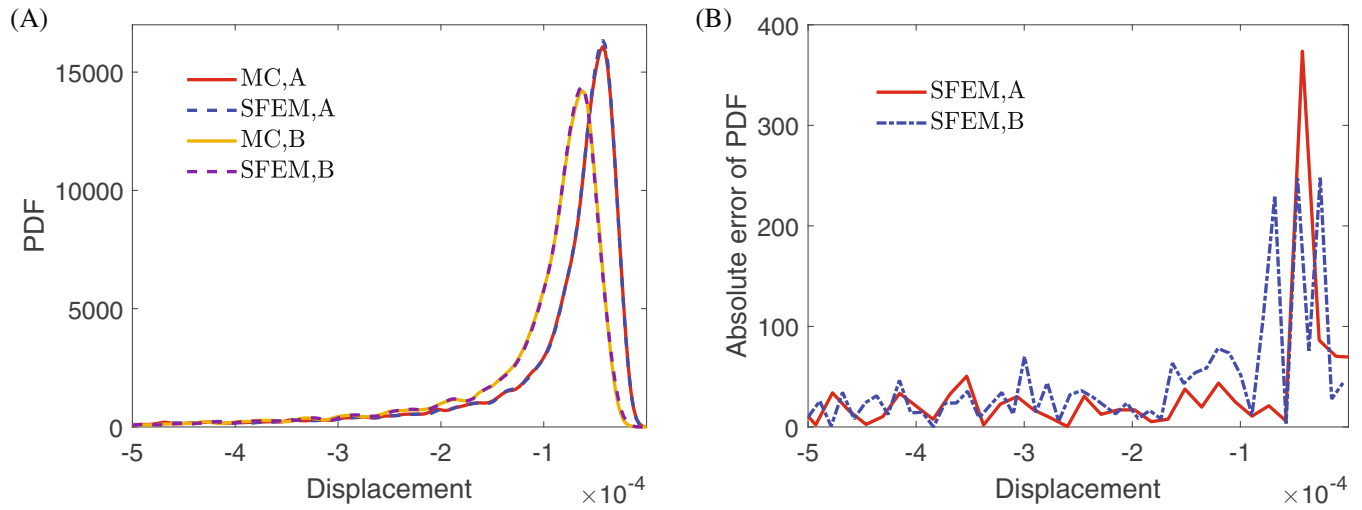
FIGURE 9 The first six vectors  $\{\mathbf{d}_j(t_i)\}_{j=1}^6$  in y direction at time step  $t_{30}$  (the first line) and time step  $t_{40}$  (the second line).

sample realizations of displacement-load curves of the points *A* and *B* obtained by the proposed nonlinear SFEM and MC simulations are shown in Figure 11A,C, respectively. The corresponding errors relative to MCS are shown in Figure 11B,D, which indicates that relative errors of stochastic displacements of the points *A* and *B* are very small and the proposed SFEM can still be used as a stochastic reduced-order approach under complex loading. It is also seen that most relative errors are still less than 0.01%. The relative errors are a bit larger for some steps corresponding to the time steps 20 to 30 due to the stochastic displacements close to zero. The computational times of the proposed nonlinear SFEM and MC simulations are 427.88 s (including the solving time 88.16 s and the updating time 339.72 s) and 6414.12 s, respectively, which demonstrates that the proposed method is still cheap enough. Also, compared to the computational cost for this example and that for Example 5.2, it is found that the computational cost almost proportionally increases with the number of total time steps. Thus, reducing the number of total time steps with an adaptive time-stepping method will greatly reduce the computational cost, which is beyond the scope of this article and will be investigated in future research.

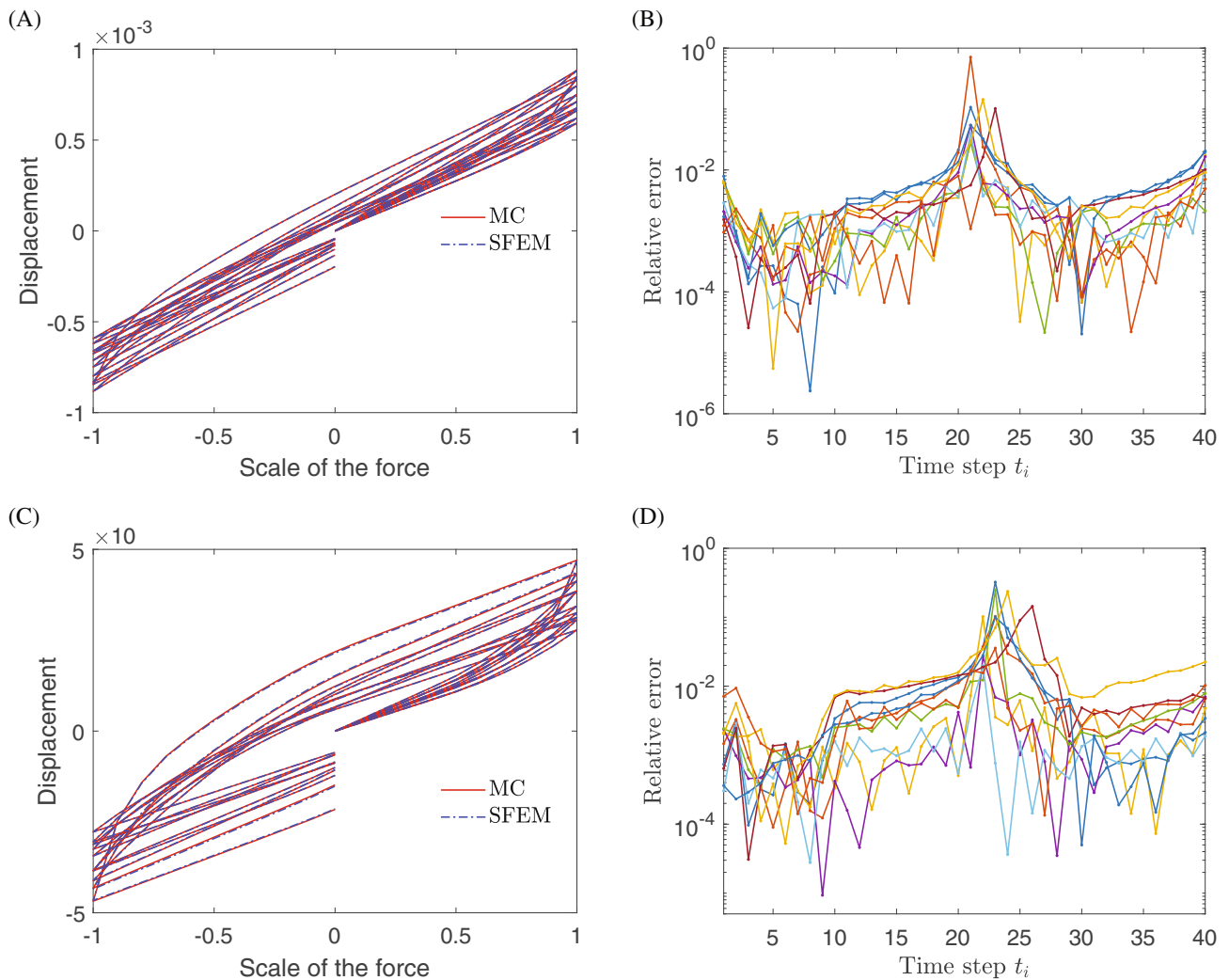
### 5.4 | Example 3: High-dimensional stochastic case

In this example, we consider a high-dimensional stochastic case by letting the number of the truncation  $r_E = r_{\sigma_Y} = 50$  in (64) and the total stochastic dimensionalities are 100 in this case. The sample size in step 2 in Algorithm 1 are reset





**FIGURE 10** PDFs of displacements of the points *A* and *B* in the *y* direction at the time step  $t_{40}$  obtained by the proposed SFEM and MCS. (A) PDFs obtained by the proposed SFEM and MCS. (B) Absolute errors relative to the PDFs by MCS.



**FIGURE 11** Ten sample realizations of the stochastic displacements of the points *A* and *B* in the *y* direction. (A) Sample realizations of the displacement of the point *A*. (B) Relative errors relative to the displacements from MCS. (C) Sample realizations of the displacement of the point *B*. (D) Relative errors relative to the displacements from MCS.



as  $n_{s,1} = 1000$ . The monotonic load in Example 5.2 and 11 time steps are adopted. The iterative errors  $\epsilon_{\mathbf{u},k}$  of each time step  $\{t_i\}_{i=1}^{10}$  are shown in Figure 12 and  $k = 8$  to 14 retained terms  $\{\lambda_j(t_i, \theta), \mathbf{d}_j(t_i)\}_{j=1}^k$  are required for the specified precision. Compared to the iterative errors shown in Figure 2, the convergence rate is lower in this case due to the large uncertainties caused by high-dimensional random variables. The comparison between the number of retained terms of each time step for the stochastic dimensionalities 20 (i.e., the case in Example 5.2) and 100 is depicted in Figure 13. The number of retained terms of each time step in this case is greater than that of the stochastic dimensionalities 20 and 109 deterministic vectors  $\mathbf{d}_j(t_i) \in \mathbb{R}^{584}$  and 109 random sample vectors  $\lambda_j(t_i, \hat{\theta}) \in \mathbb{R}^{10^4}$  are involved in total. The solving and updating times of the proposed method and the MCS time are 49.92, 137.62, and 1726.35 s, respectively, which is a little bit time-consuming since more retained terms  $\{\lambda_j(t_i, \theta), \mathbf{d}_j(t_i)\}$  are required to be resolved.

The PDFs of the stochastic displacements of the points *A* and *B* in the *y* direction at the time step  $t_{10}$  are shown in Figure 14A and the corresponding absolute errors are seen from Figure 14B. Comparing Figures 4B and 14B, larger absolute errors are obtained in high-dimensional stochastic cases. The PDFs shown in Figure 14A are very close to that of the low-dimensional case considered in Figure 4A since the last few dozens of random terms expanded by KL expansion (64) maybe have less contributions to the quantities of interest. It is noted that the proposed SFEM solves both low- and high-dimensional stochastic problems in a unified computational frame and no additional modifications for

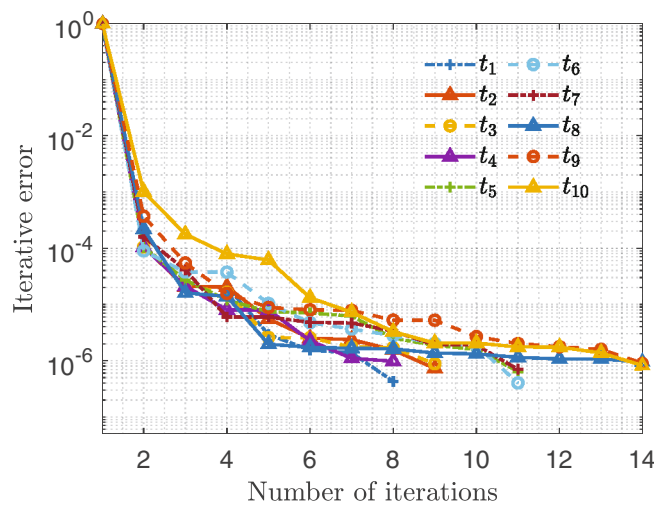


FIGURE 12 Iterative errors of different time steps when the stochastic dimensionality 100.

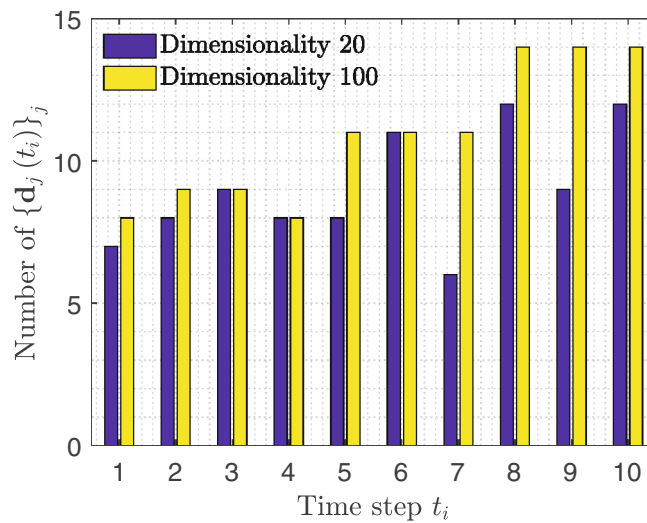
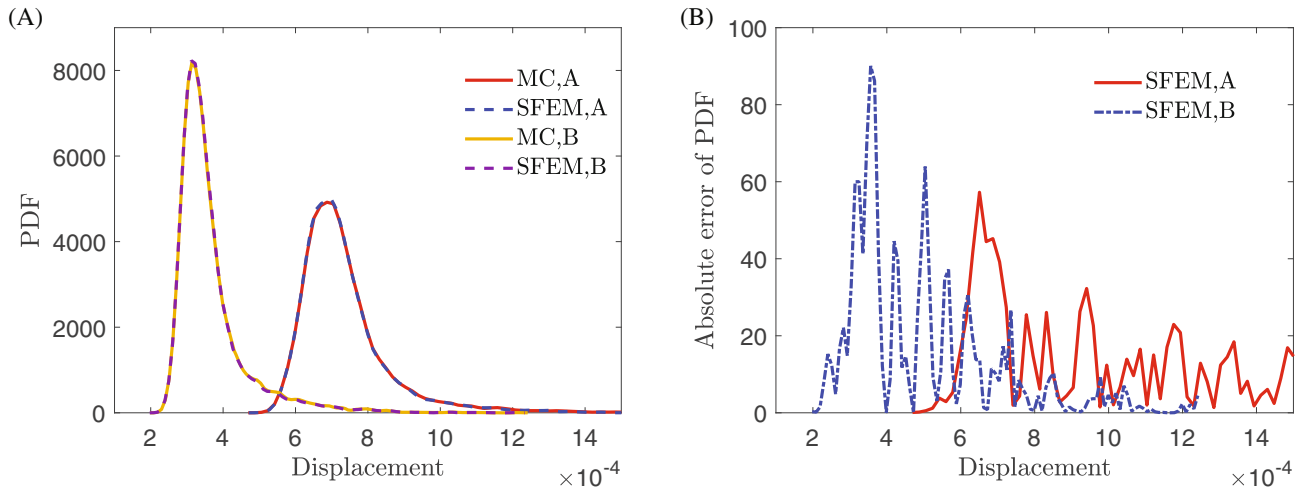
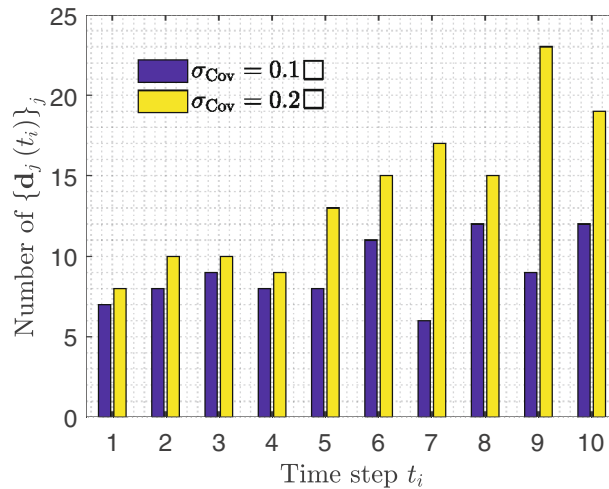


FIGURE 13 Comparison of the numbers of retained terms at different time steps with the stochastic dimensionalities 20 and 100.



**FIGURE 14** PDFs of displacements of the points A and B in the y direction at the time step  $t_{10}$  obtained by the proposed SFEM and MCS when the stochastic dimensionality 100. (A) PDFs obtained by the proposed SFEM and MCS. (B) Absolute errors relative to the PDFs by MCS.

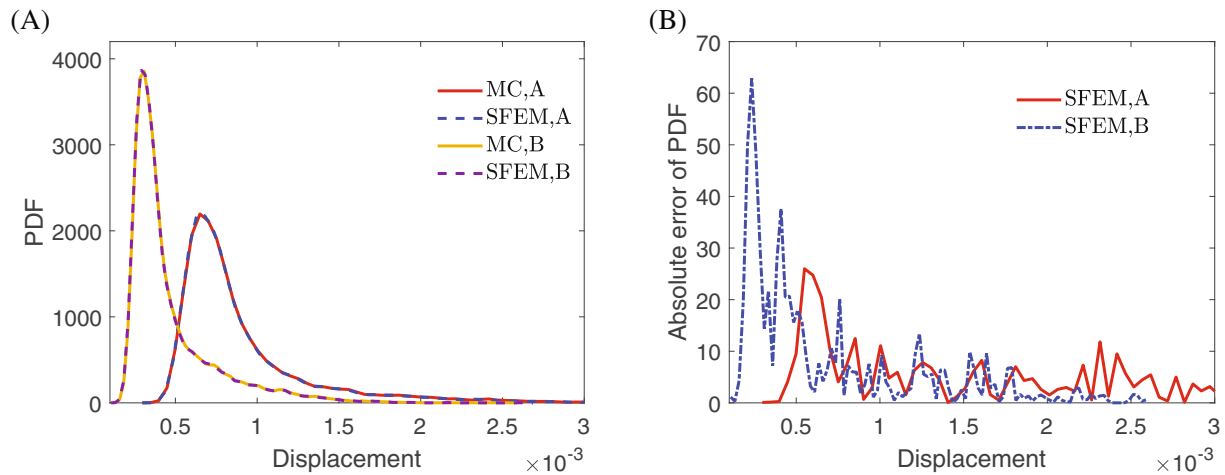


**FIGURE 15** Comparison of the numbers of retained terms at different time steps with the standard deviations  $\sigma_{Cov} = 0.1$  and  $0.2$ .

high-dimensional stochastic problems are performed. Thus, we only need to execute the stochastic analysis under 100 stochastic dimensionalities directly in this case. If there are larger uncertainties caused by high-dimensional expansions, only more additional retained terms are required to capture the large variability of the stochastic solutions.

## 5.5 | Example 4: Random sources with large variability

In this example, we consider that the stochastic Young's modulus  $E(x, y, \theta)$  and the stochastic yield stress  $\sigma_Y(x, y, \theta)$  have larger variability by letting the standard deviations as  $0.2E_0(x, y)$  and  $0.2\sigma_{Y,0}(x, y)$ , that is, 0.2 times the mean values, hereinafter which is simply notated as  $\sigma_{Cov} = 0.2$ . The number of the truncation  $r_E = r_{\sigma_Y} = 10$  in (64) and the total stochastic dimensionalities are 20. The monotonic load in Example 5.2 and 11 time steps  $\{t_0, t_1, \dots, t_{10}\}$  are still adopted in this case. The comparison between the number of retained terms of each time step for the cases  $\sigma_{Cov} = 0.1$  (i.e., the case in Example 5.2) and  $\sigma_{Cov} = 0.2$  is depicted in Figure 15. It is seen that the number of retained terms of the case  $\sigma_{Cov} = 0.2$  are greater than that of the case  $\sigma_{Cov} = 0.1$  for all time steps, and 139 terms  $\{\lambda_j(t_i, \theta), \mathbf{d}_j(t_i)\}$  are retained to approximate the stochastic solution with large variability. The computational time is 178.06 s, including the solving time 35.75 s and the updating time 142.31 s, which is much less than the computational time 1746.46 s for MC simulations.



**FIGURE 16** PDFs of displacements of the points *A* and *B* in the *y* direction at the time step  $t_{10}$  obtained by the proposed SFEM and MCS. (A) PDFs obtained by the proposed SFEM and MCS. (B) Absolute errors relative to the PDFs by MCS.

Since more terms  $\{\lambda_j(t_i, \theta), \mathbf{d}_j(t_i)\}$  are solved in this case, the computational cost is little higher than that of the case  $\sigma_{\text{Cov}} = 0.1$ . The PDFs of the displacements of the points *A* and *B* in the *y* direction at the time step  $t_{10}$  are shown in Figure 16A and the corresponding absolute errors are shown in Figure 16B. Magnitudes of absolute errors are very small compared to the magnitudes of their PDFs and the proposed SFEM still has good accuracy in this case. Also, the proposed method can well capture the long-tailed characteristics of the PDFs, which is of great help for uncertainty quantification in many stochastic problems. For instance, a high-accuracy approximation of PDF tails is critical in structural reliability analysis. Applying the proposed SFEM to uncertainty quantification for stochastic elastoplastic problems is beyond the scope of this article and will be carried out in a follow-up study.

## 6 | CONCLUSIONS AND DISCUSSIONS

This article develops an accurate and efficient nonlinear stochastic finite element method to solve stochastic elastoplastic problems. A stochastic Newton method is adopted to linearize the nonlinear stochastic finite element equation at each time step. A weakly intrusive approximation of the stochastic solution and a dedicated iteration are used to solve the corresponding linearized stochastic finite element equations. In this way, the original stochastic problem is transformed into a series of deterministic linear finite element equations and the corresponding one-dimensional stochastic algebraic equations. Existing FEM solvers can be used to solve the deterministic linear finite element equations without any modification. All stochastic analyses are performed by the one-dimensional stochastic algebraic equations that are efficiently solved by a proposed non-intrusive method. The curse of dimensionality of high-dimensional stochastic elastoplastic problems is thus circumvented with great success, which has been illustrated via a numerical case of up to 100 stochastic dimensionalities. However, there are still several intriguing issues that require to be investigated further. Only stochastic Newton iteration is used to deal with the stochastic nonlinearity in this article. Other nonlinear iterations, for example, stochastic quasi-Newton method, can also be developed for this purpose. Also, it is attractive to introduce adaptive time-stepping strategies to reduce the number of total time steps, which will save a lot of computational effort of the proposed method.

### ACKNOWLEDGMENTS

The authors are grateful to the Alexander von Humboldt Foundation and the International Research Training Group 2657 (IRTG 2657) funded by the German Research Foundation (DFG) (Grant number 433082294). Open Access funding enabled and organized by Projekt DEAL.

### DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

## ORCID

Zhibao Zheng  <https://orcid.org/0000-0003-4189-8817>

## REFERENCES

1. Stefanou G. The stochastic finite element method: past, present and future. *Comput Methods Appl Mech Eng.* 2009;198(9-12):1031-1051.
2. Dai H, Zhang R, Beer M. A new method for stochastic analysis of structures under limited observations. *Mech Syst Signal Process.* 2023;185:109730.
3. Simo JC, Hughes TJ. *Computational Inelasticity.* Springer Science & Business Media; 2006.
4. De Borst R, Crisfield MA, Remmers JJ, Verhoosel CV. *Nonlinear Finite Element Analysis of Solids and Structures.* John Wiley & Sons; 2012.
5. Papadrakakis M, Papadopoulos V. Robust and efficient methods for stochastic finite element analysis using Monte Carlo simulation. *Comput Methods Appl Mech Eng.* 1996;134(3-4):325-340.
6. Graham IG, Kuo FY, Nuyens D, Scheichl R, Sloan IH. Quasi-Monte Carlo methods for elliptic PDEs with random coefficients and applications. *J Comput Phys.* 2011;230(10):3668-3694.
7. Badia S, Hampton J, Principe J. Embedded multilevel Monte Carlo for uncertainty quantification in random domains. *Int J Uncertain Quantif.* 2021;11(1):119-142.
8. Doostan A, Iaccarino G. A least-squares approximation of partial differential equations with high-dimensional random inputs. *J Comput Phys.* 2009;228(12):4332-4345.
9. Khuri AI, Mukhopadhyay S. Response surface methodology. *Wiley Interdiscip Rev Comput Stat.* 2010;2(2):128-149.
10. Fuhg JN, Fau A, Nackenhorst U. State-of-the-art and comparative review of adaptive sampling methods for kriging. *Arch Comput Methods Eng.* 2021;28(4):2689-2747.
11. Ghanem RG, Spanos PD. *Stochastic Finite Elements: a Spectral Approach.* Courier Corporation; 2003.
12. Xiu D, Karniadakis GE. The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J Sci Comput.* 2002;24(2):619-644.
13. Keese A, Matthies HG. Hierarchical parallelisation for the solution of stochastic finite element equations. *Comput Struct.* 2005;83(14):1033-1047.
14. Blatman G, Sudret B. Adaptive sparse polynomial chaos expansion based on least angle regression. *J Comput Phys.* 2011;230(6):2345-2367.
15. Rosic BV, Matthies HG. Stochastic Galerkin method for the elastoplasticity problem with uncertain parameters. In: Mueller-Hoeppe D, Loehnert S, Reese S, eds. *Recent Developments and Innovative Applications in Computational Mechanics.* Springer; 2011:303-310.
16. Anders M, Hori M. Stochastic finite element method for elasto-plastic body. *Int J Numer Methods Eng.* 1999;46(11):1897-1916.
17. Anders M, Hori M. Three-dimensional stochastic finite element method for elasto-plastic bodies. *Int J Numer Methods Eng.* 2001;51(4):449-478.
18. Sett K, Jeremić B, Kavvas ML. Stochastic elastic-plastic finite elements. *Comput Methods Appl Mech Eng.* 2011;200(9-12):997-1007.
19. Karapiperis K, Sett K, Kavvas ML, Jeremić B. Fokker-Planck linearization for non-Gaussian stochastic elastoplastic finite elements. *Comput Methods Appl Mech Eng.* 2016;307:451-469.
20. Arnst M, Ghanem R. A variational-inequality approach to stochastic boundary value problems with inequality constraints and its application to contact and elastoplasticity. *Int J Numer Methods Eng.* 2012;89(13):1665-1690.
21. Wan X, Karniadakis GE. An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *J Comput Phys.* 2005;209(2):617-642.
22. Basmaji A, Fau A, Urrea-Quintero J, Voelsen E, Nackenhorst U. Anisotropic multi-element polynomial chaos expansion for high-dimensional non-linear structural problems. *Probabilistic Eng Mech.* 2022;70:103366.
23. Liu WK, Mani A, Belytschko T. Finite element methods in probabilistic mechanics. *Probabilistic Eng Mech.* 1987;2(4):201-213.
24. Strakowski M, Kamiński M. Stochastic finite element method elasto-plastic analysis of the necking bar with material microdefects. *ASCE-ASME J Risk Uncertain Eng Syst B Mech Eng.* 2019;5(3):030908.
25. Dannert MM, Bense F, Fau A, Fleury RM, Nackenhorst U. Investigations on the restrictions of stochastic collocation methods for high dimensional and nonlinear engineering applications. *Probabilistic Eng Mech.* 2022;69:103299.
26. Ajith G, Ghosh D. A novel method for solving nonlinear stochastic mechanics problems using FETI-DP. *Int J Numer Methods Eng.* 2022;123(10):2290-2308.
27. Rahman S. A polynomial dimensional decomposition for stochastic computing. *Int J Numer Methods Eng.* 2008;76(13):2091-2116.
28. Rahman S. A new computational method for probabilistic elastic-plastic fracture analysis. *J Press Vessel Technol.* 2009;131(6):061402.
29. Rahman S, Jahanbin R. A spline dimensional decomposition for uncertainty quantification in high dimensions. *SIAM/ASA J Uncertain Quantif.* 2022;10(1):404-438.
30. Zheng Z, Dai H. Structural stochastic responses determination via a sample-based stochastic finite element method. *Comput Methods Appl Mech Eng.* 2021;381:113824.
31. Zheng Z, Beer M, Dai H, Nackenhorst U. A weak-intrusive stochastic finite element method for stochastic structural dynamics analysis. *Comput Methods Appl Mech Eng.* 2022;399:115360.
32. Carstensen C, Klose R. Elastoviscoplastic finite element analysis in 100 lines of Matlab. *J Numer Math.* 2002;10(3):157-192.
33. Zheng Z, Valdebenito M, Beer M, Nackenhorst U. A stochastic finite element scheme for solving partial differential equations defined on random domains. *Comput Methods Appl Mech Eng.* 2023;405:115860.
34. Nouy A, Le Maître OP. Generalized spectral decomposition for stochastic nonlinear problems. *J Comput Phys.* 2009;228(1):202-235.

35. Chinesta F, Keunings R, Leygue A. *The Proper Generalized Decomposition for Advanced Numerical Simulations: A Primer*. Springer Science & Business Media; 2013.
36. Bergheau JM, Zuchiatti S, Roux JC, Feulvarch É, Tissot S, Perrin G. The proper generalized decomposition as a space–time integrator for elastoplastic problems. *C R Méc*. 2016;344(11-12):759-768.
37. Xiu D. *Numerical Methods for Stochastic Computations*. Princeton University Press; 2010.
38. Čermák M, Sysala S, Valdman J. Efficient and flexible MATLAB implementation of 2D and 3D elastoplastic problems. *Appl Math Comput*. 2019;355:595-614.
39. Zheng Z, Dai H. Simulation of multi-dimensional random fields by Karhunen–Loève expansion. *Comput Methods Appl Mech Eng*. 2017;324:221-247.
40. Zheng Z, Dai H, Wang Y, Wang W. A sample-based iterative scheme for simulating non-stationary non-Gaussian stochastic processes. *Mech Syst Signal Process*. 2021;151:107420.
41. Faes MG, Broggi M, Spanos PD, Beer M. Elucidating appealing features of differentiable auto-correlation functions: a study on the modified exponential kernel. *Probabilistic Eng Mech*. 2022;69:103269.
42. Spanos PD, Beer M, Red-Horse J. Karhunen–Loève expansion of stochastic processes with a modified exponential covariance kernel. *J Eng Mech*. 2007;133(7):773-779.

**How to cite this article:** Zheng Z, Nackenhorst U. A nonlinear stochastic finite element method for solving elastoplastic problems with uncertainties. *Int J Numer Methods Eng*. 2023;124(16):3411-3435. doi: 10.1002/nme.7253