SIAM J. Sci. Comput.
Vol. 42, No. 2, pp. A723–A750

A VARIABLE-SEPARATION METHOD FOR NONLINEAR PARTIAL DIFFERENTIAL EQUATIONS WITH RANDOM INPUTS[*]

QIUQI LI[†] AND PINGWEN ZHANG[‡]

Abstract. In this paper, we consider a variable-separation (VS) method to solve the nonlinear partial differential equations (PDEs) with random inputs. The aim of the VS method is to get a separated representation of the Galerkin solution for nonlinear PDEs with random inputs. An essential ingredient of the proposed method is the construction of the optimal stochastic basis functions. The nonlinearity can affect the computation efficiency and may bring challenges for the construction of the optimal stochastic basis functions. In order to overcome the difficulty, we develop the VS method such that the optimal stochastic basis functions are generated in an incremental constructive man- ner. At each enrichment step, a stochastic basis function is determined by the linearized equation deduced from the nonlinear problems at hand. The computation of the VS method decomposes into an offline phase and an online phase. The linearization of the construction for stochastic basis functions can significantly improve the computation efficiency in both offline and online stages. We first describe the VS method for nonlinear stochastic problems in a general framework. Then two nonlinear models with random inputs are considered to formulate the details and methodologies of the proposed method, namely, the nonlinear elliptic equations and the steady Navier–Stokes equations.

Key words. variable-separation method, random inputs, nonlinear partial differential equations, steady Navier–Stokes equation

AMS subject classifications. 65N99, 60H35, 35R60

DOI. 10.1137/19M1262486

1. Introduction.

Many complex real-world models in science and engineering usually involve nonlinear terms and often contain some uncertainties because of lacking enough knowledge about the physical properties and measurement noise. The uncertainties are often parameterized by random variables to explore the uncertainty propagation for these models. Therefore, such complex models with uncertainties can be described by the nonlinear partial differential equations (PDEs) with random inputs (e.g., model coefficients and forcing terms). In this work, we consider the numerical simulation methods for stochastic nonlinear problems and discuss the related applications.

To quantify the effects of parameters and estimate unknown parameter values from data usually require thousands to millions of realizations. This may bring challenges for numerical simulation, especially when quantifying nonlinear, multiphysics, multiscale, or coupled biological phenomena. In order to overcome the difficulty, many numerical methods have been proposed to construct the approximate solutions, i.e., surrogate models for the complex physical and engineering systems in recent decades. Spectral stochastic methods (e.g., [23, 24, 33]) have been extensively investigated to explore the uncertainty propagation in the last two decades. Most of these approaches, such as L^2 projection [26, 32], Galerkin projections [2, 21, 31], regression [3], and stochastic interpolation [1, 22, 36, 44, 45], attempt to find a functional expansion

[*]Submitted to the journal's Methods and Algorithms for Scientific Computing section May 20, 2019; accepted for publication (in revised form) January 13, 2020; published electronically March 17, 2020.
https://doi.org/10.1137/19M1262486
Funding: This work was supported by the Chinese NSF under grant 11421101 and the China Postdoctoral Science Foundation under grant 2019M650288.

[†]School of Mathematical Sciences, Peking University, Beijing, 100871, China (qiuqili@pku.edu.cn).
[‡]Corresponding author. School of Mathematical Sciences, Laboratory of Mathematics and Applied Mathematics, Peking University, Beijing 100871, China (pzhang@pku.edu.cn).

for the random solution based on a suitable set of basis functions of random variables, which are independent of the models. Model reduction methods [33] have been proposed to reduce the computation complexity especially when the original models are expensive to perform numerical simulations. These construct an approximate model with lower dimensionality but still describes important aspects of the original model. The reduced basis (RB) method is one of the model order reduction methods, arising from the field of structure mechanics, fluid dynamics. The RB method usually provides an efficient and reliable approximation of input-output relationship [4, 9, 10, 11, 12, 18, 27, 38, 39].

Another class of model reduction methods attempt to get a separated representation of the solution for stochastic problems in a systematic enrichment manner. As an example, a proper generalized decomposition (PGD) method has been proposed for solving stochastic PDEs [15, 33, 34, 35] in recent years. The PGD method is devoted to constructing optimal RB from a double orthogonality criterion. To get the separated representation of the solution requires that a few uncoupled deterministic problems are solved by classical deterministic solution techniques, and some stochastic algebraic equations are solved by classical spectral stochastic methods. In this paper, we propose a new variable-separation (VS) method for the nonlinear PDEs with random inputs to get a separated representation for the solution without iterations at each enrichment step. There have been a few works investigating the model reduction methods of parametric nonlinear models; see [9, 13, 18, 38, 40] for the RB methods, and refer to [20, 34, 42] for the PGD methodologies.

The main idea of the VS method is to construct the quasi-optimal separated representations

$$u(x,\boldsymbol{\omega}) \approx \sum_{i=1}^{N} \zeta_i(\boldsymbol{\omega}) h_i(x) \tag{1.1}$$

in a systematic enrichment manner. At each enrichment step k, we need to solve a deterministic problem induced by the original model with a fixed sample $\boldsymbol{\omega}_k$ to obtain the deterministic functions $h_k(x)$. After that, the stochastic functions $\zeta_k(\boldsymbol{\omega})$ can be determined by an algebraic equation. For the VS method, each of the stochastic functions $\zeta_k(\boldsymbol{\omega})$ can be expressed explicitly by the previous functions $\{\zeta_i(\boldsymbol{\omega})\}_{i=1}^{k-1}$, model coefficients, and forcing terms. In this context, the stochastic functions $\{\zeta_i(\boldsymbol{\omega})\}_{i=1}^{N}$ can be seen as a set of optimal basis functions of random variables deduced from the models. Compared with PGD, which requires many iterations with the arbitrary initial guess to compute $\zeta_i(\boldsymbol{\omega})$ and $h_i(x)$ at each enrichment step i, no iteration is performed at each enrichment step in the framework of the VS method. Moreover, based on the explicit constructions of the stochastic functions, it is more convenient to construct the separated representations (1.1) of the problems than with PGD. VS shares the same merits as the RB method, and it has some differences such that (i) the parameter sensitivities can be analytically computed by taking the partial derivatives based on the explicit representation of the stochastic functions $\zeta_k(\boldsymbol{\omega})$; (ii) for the online computation of the VS, we can calculate the approximation straightforwardly by the separated representation. Therefore, the VS method can provide an efficient and reliable approximation, which is crucial for the many-query context such as optimization, control design, and inverse analysis.

The VS method was first proposed in [29] for the linear stochastic problems, and the extension to stochastic saddle point problems has been developed in [28]. In this contribution, we develop the strategy of VS for the nonlinear parameterized PDEs. In the VS method, an offline-online computational decomposition is required to improve efficiency. In the offline stage, to generate the RB functions $\{h_i(x)\}_{i=1}^N$ and $\{\zeta_i(\boldsymbol{\omega})\}_{i=1}^N$, we need to compute a set of snapshots, which are the solutions of the parameterized PDEs corresponding to a set of optimal parameter samples. In the online stage, the output is computed by the quasi-optimal separated representations (1.1) for many instances of parameters, and the influence of the uncertainty is estimated. The VS proposed in [29] for the generic multivariate function can provide a variable-separation for a random field, and this can be used to get an affine representation for the model's inputs. The affine representation is crucial to achieve the decomposition of offline-online computation for many queries to model's outputs. In the framework of the VS method, we are usually required to solve a couple of nonlinear deterministic problems and nonlinear stochastic equations in the offline phase. This can affect the computation efficiency and bring challenges for numerical simulation, especially when identifying the stochastic basis functions $\zeta_k(\boldsymbol{\omega})$ in stochastic dimension spaces. To this end, we construct the optimal stochastic basis functions in a systematic enrichment manner based on the classic linearization techniques such as the Newton method. At each enrichment step, the stochastic basis function $\zeta_k(\boldsymbol{\omega})$ is determined by the linearized stochastic equation, which is deduced from the original nonlinear problems. Consequently, the computation complexity can be significantly reduced in both the offline and the online stage. We first describe the VS method for nonlinear stochastic problems in a general framework in this paper. Subsequently, the methodology is detailed and tested based on two nonlinear stochastic models that fit in the general framework, i.e., (i) the nonlinear elliptic equation, (ii) the steady Navier–Stokes equation.

The outline of the paper is as follows. In section 2, we give some preliminaries and notation for the paper and collect some necessary facts on nonlinear stochastic problems. In section 3, we first introduce the Newton's method for nonlinear problems. Then the VS method is presented for the nonlinear stochastic problems in an abstract framework. Section 4 is devoted to describing the details of the VS method for the nonlinear elliptic equation with random input. In section 5, we develop the VS method for the steady Navier–Stokes equation with random input. Two numerical examples are presented in section 6 to illustrate the performance of the proposed methods. Finally, we make some conclusions and comments.

2. Preliminaries and notation.

Let D denote a convex and bounded physical domain with Lipschitz continuous boundary ∂D. $\mathcal{V}$ is a Hilbert space defined on D with inner products $(\cdot,\cdot)_{\mathcal{V}}$. The associated norm is defined as $\|\cdot\|_{\mathcal{V}}^2 = (\cdot,\cdot)_{\mathcal{V}}$. Let $(\Omega, \mathcal{B}, P)$ be a complete probability space, where Ω is the space of elementary events, $\mathcal{B}$ is a σ-algebra on Ω, and P is the probability measure on $\mathcal{B}$. We denote the Hilbert space of the random variables with second order moments by $L_P^2(\Omega)$, which is defined by

$$L_P^2(\Omega) = \left\{ w : y \in \Omega \to w(y) \in \mathbb{R}; \int_\Omega w(y)^2 P(dy) < \infty \right\}.$$

The inner product in $L_P^2(\Omega)$ is given by $(w,v)_{L_P^2(\Omega)} := \int_\Omega w(y)v(y)P(dy)$, which induces the norm $\|w\|_{L^2}^2 = \|w\|_{L_P^2(\Omega)}^2 := (w,w)_{L_P^2(\Omega)}$.

Let $G : \mathcal{V} \times \Omega \to \mathcal{V}^*$ (the dual space of $\mathcal{V}$) be a mapping representing a nonlinear PDE with random inputs. We consider the nonlinear stochastic PDEs defined on a bounded physical domain. The problem can be read in an abstract form as, Given $\boldsymbol{\omega} \in \Omega$, find $u(x,\boldsymbol{\omega}) \in \mathcal{V}$ such that

$$G(u(x,\boldsymbol{\omega});\boldsymbol{\omega}) = 0 \ \forall\ x \in D,\ \ \boldsymbol{\omega} \in \Omega, \tag{2.1}$$

where $\boldsymbol{\omega} := (\omega_1, \ldots, \omega_d)$ is a sequence of d real-valued random variables, and $u(x, \boldsymbol{\omega})$ is the solution of the nonlinear SPDE. Let $S(u(\boldsymbol{\omega}); \boldsymbol{\omega})$ be the output of the model (2.1), where S is a bounded functional over $\mathcal{V}$.

The solution u belongs to Hilbert space $L^2(\Omega; \mathcal{V})$, which can be approximated by the tensor product space $\mathcal{V} \otimes L^2_P(\Omega)$. Here $\otimes$ denotes a tensor product for the Hilbert spaces. For simplicity of notation, we will denote $L^2_P(\Omega)$ by $\mathcal{S}$ in the rest of the paper. Then the inner product in $\mathcal{V} \otimes \mathcal{S}$ is defined by

$$(w, u)_{\mathcal{V}\otimes\mathcal{S}} = E[(w, u)_{\mathcal{V}}] := \int_{\Omega} (w, u)_{\mathcal{V}} P(dy).$$

Thus we define the norm as $\|u\|^2_{\mathcal{V}\otimes\mathcal{S}} := (u, u)_{\mathcal{V}\otimes\mathcal{S}}$.

The weak formulation of problem (2.1) reads, Find $u(x, \boldsymbol{\omega}) \in \mathcal{V}$ such that

$$g(u(x, \boldsymbol{\omega}), v; \boldsymbol{\omega}) = 0 \quad \forall\, v \in \mathcal{V}, \tag{2.2}$$

where $g(\cdot, \cdot; \boldsymbol{\omega})$ is defined as

$$g(w, v; \boldsymbol{\omega}) = \langle G(w; \boldsymbol{\omega}), v\rangle \quad \forall\, w, v \in \mathcal{V}.$$

Here we denote $\langle \cdot, \cdot \rangle$ as the duality pairing between $\mathcal{V}^*$ and $\mathcal{V}$. We assume that the mapping G is continuously differentiable and denote its (partial) Fréchet derivatives at $(z, \boldsymbol{\omega}) \in \mathcal{V} \times \Omega$ by $D_u G(z; \boldsymbol{\omega})$ and $D_{\boldsymbol{\omega}} G(z; \boldsymbol{\omega})$. Then, the partial Fréchet derivatives of $g(u, \cdot; \boldsymbol{\omega})$ with respect to u at $z \in \mathcal{V}$ are represented as

$$dg[z](w, v; \boldsymbol{\omega}) = \langle D_u G(z; \boldsymbol{\omega}) w, v\rangle \quad \forall\, w, v \in \mathcal{V}. \tag{2.3}$$

For the well-posedness of (2.2) (refer to [38]), $dg[u(\boldsymbol{\omega})](\cdot, \cdot; \boldsymbol{\omega})$ is required to be inf-sup stable, i.e., there exists a constant $\beta_0 > 0$ such that

$$\beta(\boldsymbol{\omega}) := \inf_{w\in\mathcal{V}} \sup_{v\in\mathcal{V}} \frac{dg[u(\boldsymbol{\omega})](w, v; \boldsymbol{\omega})}{\|w\|_{\mathcal{V}} \|v\|_{\mathcal{V}}} > \beta_0 \ \forall \boldsymbol{\omega} \in \Omega, \tag{2.4}$$

and continuous, i.e., there exists a positive constant $\gamma_0 < \infty$ such that

$$\gamma(\boldsymbol{\omega}) := \sup_{w\in\mathcal{V}} \sup_{v\in\mathcal{V}} \frac{dg[u(\boldsymbol{\omega})](w, v; \boldsymbol{\omega})}{\|w\|_{\mathcal{V}} \|v\|_{\mathcal{V}}} \leq \gamma_0 \ \forall \boldsymbol{\omega} \in \Omega. \tag{2.5}$$

To ensure the existence of a local branch of nonsingular solutions to problem (2.1), we use the following proposition.

PROPOSITION 2.1 (see [8, 38]). *Let $G : \mathcal{V} \times \Omega \to \mathcal{V}^*$ be a C^1 map. Assume that the following assumptions hold:*

1. *For some $\boldsymbol{\omega}_0 \in \Omega$, $u_0 \in \mathcal{V}$, $G(u_0; \boldsymbol{\omega}_0) = 0$.*
2. *$dg[u_0](\cdot, \cdot; \boldsymbol{\omega}_0)$ is continuous and satisfies the inf-sup condition, i.e., $D_u G(u_0; \boldsymbol{\omega}_0)$ is bijective.*

Then, there exist $r_0, r > 0$, and a unique $u(\boldsymbol{\omega}) \in B_r(u_0) \cap \mathcal{V}$ such that

$$G(u(\boldsymbol{\omega}); \boldsymbol{\omega}) = 0 \quad \forall \boldsymbol{\omega} \in B_{r_w}(\boldsymbol{\omega}_0) \cap \Omega,$$

where $B_r(u_0) \subset \mathcal{V}$ denotes the ball with center u_0 and radius $r > 0$, and $B_{r_w}(\boldsymbol{\omega}_0) \subset \Omega$ is the ball with radius $r_w > 0$, and centered at $\boldsymbol{\omega}_0$.

The proof of Proposition 2.1 is based on a straightforward application of the implicit function theorem [14, 43].

3. A variable-separation method.

Motivated by the VS approach proposed in [29], we develop a VS method for the nonlinear PDEs with random inputs. The VS method for nonlinear problems with random inputs is proposed based on the Newton's method [16, 30, 38]. Here we first present the Newton's method for nonlinear problems. Solving problem (2.2) by Newton's method requires nonlinear iterations with a linearized problem being solved at each step. For given $\boldsymbol{\omega} \in \Omega$, we initialize $u_0 \in \mathcal{V}$ as an approximation of u. At step k $(k = 1, 2, \cdots)$, we assume that the solution u_{k+1} can be written as

$$u_{k+1}(x) = u_k(x) + \delta u(x),$$

where δu is a correction of $u_k(x)$. Then we seek $\delta u \in \mathcal{V}$ such as

$$dg[u_k(\boldsymbol{\omega})](\delta u, v; \boldsymbol{\omega}) = -g(u_k(\boldsymbol{\omega}), v; \boldsymbol{\omega}) \quad \forall\ v \in \mathcal{V}. \tag{3.1}$$

We will stop the iteration procedure when $\|\delta u\|_{\mathcal{V}}$ is small enough. Provided that $dg[u_k(\boldsymbol{\omega})](\cdot, \cdot; \boldsymbol{\omega})$ is locally Lipschitz continuous and u_0 is sufficiently close to $u(\boldsymbol{\omega})$, Newton's method is quadratically convergent.

In order to evaluate the output $S(u(\boldsymbol{\omega}); \boldsymbol{\omega})$ of (2.2), we need to assemble and solve N (iteration number) linear systems (3.1) on the fine grid for any $\boldsymbol{\omega} \in \Omega$. This leads to a large number of calculations. Compared with the linear problems, it is more necessary and meaningful to construct an efficient surrogate model for the nonlinear stochastic problems. Therefore, we will introduce a VS method for the nonlinear PDEs with random inputs.

Let Ξ be a training set, which is a collection of a finite number of samples in Ω. Typically the training set is chosen by Monte Carlo methods, which is a straightforward method, and does not consider the property of the quantity of interest. It requires that the samples in Ξ are sufficiently scattered in the domain Ω. The quasi-random sampling using low-discrepancy sequences is a variant of random sampling methods, such as a Halton or Sobol sequence, which tends to provide more equidistributed samples in the parameter space. We note that $|\Xi|$ denotes the cardinality of the set Ξ.

3.1. The VS method for nonlinear stochastic problems in a general framework.

Let $\mathcal{V}_h \subset \mathcal{V}$ be a given finite dimensional approximation space. We find the numerical solution to problem (2.1) under the form

$$u(x, \boldsymbol{\omega}) \approx u_N(x, \boldsymbol{\omega}) := \sum_{i=1}^{N} \zeta_i(\boldsymbol{\omega}) h_i(x), \tag{3.2}$$

where $\zeta_i \in \mathcal{S}$ are stochastic functions and $h_i \in \mathcal{V}_h$ are deterministic functions. In the framework of the VS method, stochastic functions $\zeta_i(\boldsymbol{\omega})$ and deterministic functions $h_i(x)$ are generated by an incremental constructive manner.

Now we present the algorithm to obtain $\{h_i(x)\}_{i=1}^N$ and $\{\zeta_i(\boldsymbol{\omega})\}_{i=1}^N$ for the nonlinear problems (2.1). To this end, we initialize $i = 1$, and $\boldsymbol{\omega}_1$ is chosen randomly in Ξ. Then $h_1(x)$ is taken as the solution of (2.2) with $\boldsymbol{\omega} = \boldsymbol{\omega}_1$ by Newton's method. We take $u(x, \boldsymbol{\omega}) = \zeta_1(\boldsymbol{\omega}) h_1(x)$ and $v = h_1(x)$ in (2.2), and we have

$$g(\zeta_1(\boldsymbol{\omega}) h_1(x), h_1(x); \boldsymbol{\omega}) = 0. \tag{3.3}$$

Then $\zeta_1(\boldsymbol{\omega})$ is given by solving (3.3) with Newton's iteration method in practice.

At step $i \geq 2$, we choose

$$\boldsymbol{\omega}_i := \arg\max_{\boldsymbol{\omega}\in\Xi} \Delta_i(\boldsymbol{\omega}),$$

where $\Delta_i(\omega)$ corresponds to the norm $\|\cdot\|_{\mathcal{V}}$ of the residual of the nonlinear equation at parameter $\boldsymbol{\omega}$, and the details for the definition of $\Delta_i(\boldsymbol{\omega})$ are described in Appendix A.

Let $e(\boldsymbol{\omega}) = u(\boldsymbol{\omega}) - u_{i-1}(\boldsymbol{\omega})$. By (2.2), we get

$$g\big(u(\boldsymbol{\omega}) - u_{i-1}(\boldsymbol{\omega}) + u_{i-1}(\boldsymbol{\omega}), v; \boldsymbol{\omega}\big) = 0 \quad \forall v \in \mathcal{V}_h,$$

that is,

$$g\big(u_{i-1}(\boldsymbol{\omega}) + e(\boldsymbol{\omega}), v; \boldsymbol{\omega}\big) = 0 \quad \forall v \in \mathcal{V}_h,$$

where $e(\boldsymbol{\omega})$ can be seen as a correction. By (3.1), we get

$$dg[u_{i-1}(\boldsymbol{\omega})](e(\boldsymbol{\omega}), v; \boldsymbol{\omega}) = -g(u_{i-1}(\boldsymbol{\omega}), v; \boldsymbol{\omega}) \ \ \forall\ v \in \mathcal{V}. \tag{3.4}$$

We take $h_i(x)$ as the solution of (3.4) with $\boldsymbol{\omega} = \boldsymbol{\omega}_i$. We rewrite $e(\boldsymbol{\omega}) := h_i(x)\zeta_i(\boldsymbol{\omega})$ and take $v = h_i(x)$ in (3.4). Then $\zeta_i(\boldsymbol{\omega})$ is further determined by the following linear equation:

$$dg[u_{i-1}(\boldsymbol{\omega})](h_i(x)\zeta_i(\boldsymbol{\omega}), h_i(x); \boldsymbol{\omega}) = -g(u_{i-1}(\boldsymbol{\omega}), h_i(x); \boldsymbol{\omega}). \tag{3.5}$$

Remark 3.1. In order to reduce the computational complexity, we can take $\zeta_1(\boldsymbol{\omega})$ as the solution of a linearized system corresponding to (3.3) instead of solving (3.3) directly. Some numerical examples will be provided in section 6 to show that the proposed VS method can still give rise to a good approximation when determining $\zeta_1(\boldsymbol{\omega})$ by a linearized system with regard to (3.3).

To present the VS method for the nonlinear stochastic problems more clearly, we will present two examples that fit in this abstract framework, i.e., (i) the nonlinear elliptic equation and (ii) the steady Navier–Stokes equation, and introduce the details of VS methods for all of them.

4. Nonlinear elliptic equation. We consider the nonlinear stochastic elliptic equation

$$-\nabla \cdot (\kappa(u; \boldsymbol{\omega})\nabla u(x, \boldsymbol{\omega})) = \mathrm{f}(x, \boldsymbol{\omega}) \text{ in } D, \tag{4.1}$$

subject to the homogeneous boundary condition, and $\mathrm{f}(x, \boldsymbol{\omega}) \in L^2(D)$ for any $\boldsymbol{\omega} \in \Omega$. The diffusion coefficient $\kappa(u; \boldsymbol{\omega})$ depends on the unknown solution u. For well-posedness of (4.1), we assume that $\kappa(u; \boldsymbol{\omega})$ is a well-behaved positive function. Typically, $\kappa(u; \boldsymbol{\omega})$ is a polynomial in u or can be approximated by a polynomial in u.

Let $\mathcal{V} = H_0^1(D)$. The weak formulation of (4.1) reads, $\forall \boldsymbol{\omega} \in \Omega$, we find $u \in \mathcal{V}$ such that

$$a\big(\kappa(u; \boldsymbol{\omega}), u(\boldsymbol{\omega}), v; \boldsymbol{\omega}\big) = f(v; \boldsymbol{\omega}) \quad \forall\ v \in \mathcal{V}, \tag{4.2}$$

where the bilinear forms $a\big(\cdot, \cdot, \cdot; \boldsymbol{\omega}\big)$ and functional $f(v; \boldsymbol{\omega})$ are defined by

$$\begin{cases} a\big(\kappa(u; \boldsymbol{\omega}), u(\boldsymbol{\omega}), v; \boldsymbol{\omega}\big) := \displaystyle\int_D \kappa(u; \boldsymbol{\omega})\nabla u(\boldsymbol{\omega}) \cdot \nabla v dx, \\ f(v; \boldsymbol{\omega}) := \displaystyle\int_D \mathrm{f}(\boldsymbol{\omega}) v dx. \end{cases}$$

We assume that the linear functional $f(\cdot;\boldsymbol{\omega})$ is affine with respect to $\boldsymbol{\omega}$ such that

$$f(v;\boldsymbol{\omega}) = \sum_{i=1}^{m_f} \alpha^i(\boldsymbol{\omega}) f^i(v) \quad \forall v \in \mathcal{V}, \quad \forall \boldsymbol{\omega} \in \Omega. \tag{4.3}$$

Here, for $i = 1, \ldots, m_f$, each α^i is a stochastic function, and each $f^i : \mathcal{V} \longrightarrow R$ is a continuous functional independent of $\boldsymbol{\omega}$.

To fit into the general framework, the mapping $G : \mathcal{V} \times \Omega \to \mathcal{V}^*$ can be defined by

$$G(w;\boldsymbol{\omega}) = -\nabla \cdot (\kappa(w, x, \boldsymbol{\omega})\nabla w) - \mathrm{f}(x, \boldsymbol{\omega}) \quad \forall\ w \in \mathcal{V}. \tag{4.4}$$

Then the weak formulation of problem (4.4) reads

$$\langle G(w;\boldsymbol{\omega}), v\rangle = g(w, v;\boldsymbol{\omega}) := a\big(\kappa(w, x, \boldsymbol{\omega}), w, v;\boldsymbol{\omega}\big) - f(v;\boldsymbol{\omega}) \ \forall\ w, v \in \mathcal{V}. \tag{4.5}$$

Its Fréchet derivatives with respect to w at $z \in \mathcal{V}$ can be defined as

$$dg[z](w, v;\boldsymbol{\omega}) = a\big(\kappa_w(z, x, \boldsymbol{\omega})w, z, v;\boldsymbol{\omega}\big) + a\big(\kappa(z, x, \boldsymbol{\omega}), w, v;\boldsymbol{\omega}\big) \ \forall\ w, v \in \mathcal{V}, \tag{4.6}$$

where $\kappa_w(z, x, \boldsymbol{\omega})$ denotes the Fréchet derivatives of $\kappa(w, x, \boldsymbol{\omega})$ with respect to w at $z \in \mathcal{V}$. Let $\mathcal{V}_h \subseteq \mathcal{V}$ be a given finite dimensional space. By (3.1), (4.5), and (4.6), we have the Newton method of (4.2) such that for a given initial guess u_0, at step k, we seek $\delta u \in \mathcal{V}_h$ satisfying

$$\begin{aligned} &a\big(\kappa_u(u, x, \boldsymbol{\omega})\delta u, u, v;\boldsymbol{\omega}\big) + a\big(\kappa(u;\boldsymbol{\omega}), \delta u, v;\boldsymbol{\omega}\big) \\ &\quad = f(v;\boldsymbol{\omega}) - a\big(\kappa(u;\boldsymbol{\omega}), u(\boldsymbol{\omega}), v;\boldsymbol{\omega}\big) \quad \forall\ v \in \mathcal{V}_h, \end{aligned} \tag{4.7}$$

and then we have $u_{k+1} = u_k + \delta u$.

4.1. The VS method for stochastic nonlinear elliptic PDEs.

Now we present the VS method for the nonlinear stochastic elliptic equation to construct an approximation in the form (3.2).

Let $e(\boldsymbol{\omega}) = u(\boldsymbol{\omega}) - u_{k-1}(\boldsymbol{\omega})$, and let $r(v;\boldsymbol{\omega}) \in \mathcal{V}_h^*$ (the dual space of $\mathcal{V}_h$) be the residual

$$r(v;\boldsymbol{\xi}) := \begin{cases} f(v;\boldsymbol{\omega}), & k = 1, \\ f(v;\boldsymbol{\omega}) - a\big(\kappa(u_{k-1}, x, \boldsymbol{\omega}), u_{k-1}(\boldsymbol{\omega}), v;\boldsymbol{\omega}\big), & k \geq 2. \end{cases}$$

By (4.7), we get

$$a\big(\kappa_u(u_{k-1})e, u_{k-1}, v;\boldsymbol{\omega}\big) + a\big(\kappa(u_{k-1}), e, v;\boldsymbol{\omega}\big) = r(v;\boldsymbol{\omega}) \quad \forall v \in \mathcal{V}_h. \tag{4.8}$$

According to Riesz representation theory, there exists a function $\hat{e}(\boldsymbol{\omega}) \in \mathcal{V}_h$ such that

$$\big(\hat{e}(\boldsymbol{\omega}), v\big)_{\mathcal{V}} = r(v;\boldsymbol{\omega}) \quad \forall v \in \mathcal{V}_h. \tag{4.9}$$

Consequently, the dual norm of the residual $r(v;\boldsymbol{\omega})$ can be evaluated as follows:

$$\Delta_k(\boldsymbol{\omega}) := \|r(v;\boldsymbol{\omega})\|_{\mathcal{V}^*} := \sup_{v \in \mathcal{V}_h} \frac{r(v;\boldsymbol{\omega})}{\|v\|_{\mathcal{V}}} = \|\hat{e}(\boldsymbol{\omega})\|_{\mathcal{V}}. \tag{4.10}$$

For the computation of the error estimator $\|\hat{e}(\boldsymbol{\omega})\|_{\mathcal{V}}$, we apply an offline-online procedure presented in [39, 29].

At step k, we choose

$$\boldsymbol{\omega}_k := \begin{cases} \text{chosen randomly in } \Omega, & k = 1, \\ \arg\max\limits_{\boldsymbol{\omega}\in\Xi} \|\hat{e}(\boldsymbol{\omega})\|_{\mathcal{V}}, & k \geq 2. \end{cases}$$

Let $e_h(x)$ be the solution of (4.8) with $\boldsymbol{\omega} = \boldsymbol{\omega}_k$, and we take $h_k(x) = e_h(x)$ in (3.2).

We note that when $\kappa(u;\boldsymbol{\omega})$ is a polynomial in u, $\kappa(u_{k-1};\boldsymbol{\omega})$ and $\kappa_u(u_{k-1};\boldsymbol{\omega})$ can be rewritten as the following affine representations:

$$\begin{cases} \kappa(u_{k-1};\boldsymbol{\omega}) = \sum\limits_{i=1}^{M_k} \kappa_1^i(\boldsymbol{\omega}) l_1^i(x), \\ \kappa_u(u_{k-1};\boldsymbol{\omega}) = \sum\limits_{i=1}^{M_k'} \kappa_2^i(\boldsymbol{\omega}) l_2^i(x); \end{cases} \tag{4.11}$$

otherwise, we can use the VS approach for the multivariable function presented in [29] to get affine expansion approximation for $\kappa(u_{k-1};\boldsymbol{\omega})$ and $\kappa_u(u_{k-1};\boldsymbol{\omega})$.

Let $e(\boldsymbol{\omega}) := e_h(x)e_\xi(\boldsymbol{\omega})$ in (4.8). By (4.3), (4.8), and (4.11), we have

$$\begin{aligned} & e_\xi(\boldsymbol{\omega}) \left(\sum_{i=1}^{k-1}\sum_{j=1}^{M_k'} \zeta_i(\boldsymbol{\omega})\kappa_2^j(\boldsymbol{\omega}) a(l_2^j(x)e_h(x), h_i(x), v) + \sum_{j=1}^{M_k} \kappa_1^j(\boldsymbol{\omega}) a(l_1^j(x), e_h(x), v) \right) \\ & \quad = \sum_{i=1}^{m_f} \alpha^i(\boldsymbol{\omega}) f^i(v) - \sum_{i=1}^{k-1}\sum_{j=1}^{M_k} \zeta_i(\boldsymbol{\omega})\kappa_1^j(\boldsymbol{\omega}) a(l_1^j(x), h_i(x), v). \end{aligned} \tag{4.12}$$

We take $v = e_h(x)$, i.e., $v = h_k(x)$ in (4.12); then it follows that

$$e_\xi(\boldsymbol{\omega}) = \frac{r(h_k(x),\boldsymbol{\omega})}{a_1(\boldsymbol{\omega}) + a_2(\boldsymbol{\omega})}, \tag{4.13}$$

with

$$\begin{cases} a_1(\boldsymbol{\omega}) = \sum\limits_{j=1}^{M_k} \kappa_1^j(\boldsymbol{\omega}) a(l_1^j(x), h_k(x), h_k(x)), \\ a_2(\boldsymbol{\omega}) = \sum\limits_{i=1}^{k-1}\sum\limits_{j=1}^{M_k'} \zeta_i(\boldsymbol{\omega})\kappa_2^j(\boldsymbol{\omega}) a(l_2^j(x)h_k(x), h_i(x), h_k(x)), \\ r(h_k(x),\boldsymbol{\omega}) = \sum\limits_{i=1}^{m_f} \alpha^i(\boldsymbol{\omega}) f^i(h_k(x)) - \sum\limits_{i=1}^{k-1}\sum\limits_{j=1}^{M_k} \zeta_i(\boldsymbol{\omega})\kappa_1^j(\boldsymbol{\omega}) a(l_1^j(x), h_i(x), h_k(x)). \end{cases}$$

Then we take $\zeta_k(\boldsymbol{\omega}) = e_\xi(\boldsymbol{\omega})$ in (3.2).

Algorithm 1 describes the procedure for the VS method to solve the stochastic nonlinear elliptic equation. For practical simulation, we can take a small sample set Ξ in Algorithm 1.

Remark 4.1. For the first step, take $h_1(x)$ as the solution of (4.2) with $\boldsymbol{\omega} = \boldsymbol{\omega}_1$ by Newton's method. Letting $u(x,\boldsymbol{\omega}) \approx \zeta_1(\boldsymbol{\omega})h_1(x)$ and $v = h_1(x)$ in (4.2), we have

$$a\big(\kappa(\zeta_1(\boldsymbol{\omega})h_1(x),\boldsymbol{\omega})), \zeta_1(\boldsymbol{\omega})h_1(x), h_1(x);\boldsymbol{\omega}\big) = f(h_1(x);\boldsymbol{\omega}). \tag{4.14}$$

Algorithm 1. VS for nonlinear elliptic equations with stochastic influence.

```
Input: The stochastic nonlinear elliptic equation (4.1), a set of samples $\Xi \in \Omega$,
and the error tolerance $\varepsilon_0$.
Output: The separated representation $u_N(x,\boldsymbol{\omega}) := \sum_{i=1}^{N} h_i(x)\zeta_i(\boldsymbol{\omega})$.
1: Initialize the iteration counter $k = 1$, a random $\boldsymbol{\omega}_1 \in \Xi$;
    1.1: Calculate $h_1(x)$ by solving (4.2) with $\boldsymbol{\omega} = \boldsymbol{\omega}_1$ by Newton's method;
    1.2: Determine $\zeta_1(\boldsymbol{\omega})$ by (4.14);
    1.3: Update $\Xi$ with $\Xi = \Xi \setminus \boldsymbol{\omega}_1$, and take the approximation $u_1(x,\boldsymbol{\omega})$ as
           $u_1(x,\boldsymbol{\omega}) := h_1(x)\zeta_1(\boldsymbol{\omega})$;
2: Update $k \to k+1$, take $\boldsymbol{\omega}_k = \arg\max_{\boldsymbol{\omega}\in\Xi} \Delta_k(\boldsymbol{\omega})$, and take the residual
       $r_k(v;\boldsymbol{\omega}) := f(v;\boldsymbol{\omega}) - a\big(\kappa(u_{k-1},x,\boldsymbol{\omega}), u_{k-1}(\boldsymbol{\omega}), v;\boldsymbol{\omega}\big)$;
3: Calculate $h_k(x)$ by solving (4.8) with $\boldsymbol{\omega} = \boldsymbol{\omega}_k$, and $\zeta_k(\boldsymbol{\omega})$ by (4.13);
4: Update $\Xi$ with $\Xi = \Xi \setminus \boldsymbol{\omega}_k$, and take $u_k(x,\boldsymbol{\omega}) := \sum_{j=1}^{k} h_j(x)\zeta_j(\boldsymbol{\omega})$;
5: Take $\varepsilon_k := \max_{\boldsymbol{\omega}\in\Xi} \Delta_k(\boldsymbol{\omega})$;
6: Return to step 2 if $\varepsilon_k \geq \varepsilon_0$, otherwise terminate.
7: $N = k$.
```

Then $\zeta_1(\boldsymbol{\omega})$ can be determined by the nonlinear system (4.14), and we need to solve the nonlinear system (4.14) in the space of the random variables $\mathcal{S}$ to obtain $\zeta_1(\boldsymbol{\omega})$, which will bring some computational complexity. Here we can take $\zeta_1(\boldsymbol{\omega})$ as the solution of a linearized system corresponding to (4.14). For nonlinear elliptic equations with stochastic influence, we take $\kappa(u,\boldsymbol{\omega})) = \kappa(h_1(x))$, $u(x,\boldsymbol{\omega}) \approx \zeta_1(\boldsymbol{\omega})h_1(x)$, and $v = h_1(x)$ in (4.2). Then we have the following linearized problem with regard to $\zeta_1(\boldsymbol{\omega})$:

$$a\big(\kappa(h_1(x)), \zeta_1(\boldsymbol{\omega})h_1(x), h_1(x);\boldsymbol{\omega}\big) = f(h_1(x);\boldsymbol{\omega}). \tag{4.15}$$

By (4.3) and (4.15), $\zeta_1(\boldsymbol{\omega})$ can be given by

$$\zeta_1(\boldsymbol{\omega}) = \frac{\sum_{i=1}^{m_f} \alpha^i(\boldsymbol{\omega}) f^i(h_1(x))}{a\big(\kappa(h_1(x)), h_1(x), h_1(x)\big)}.$$

Therefore, to reduce the computational complexity, we can choose $\zeta_1(\boldsymbol{\omega})$ as the solution of (4.15) at step 1.2 of Algorithm 1.

Remark 4.2. We note that, for each step k, we can write $\zeta_k(\boldsymbol{\omega})$ in a matrix form such that

$$\zeta_k(\boldsymbol{\omega}) = \frac{\mathbf{F}^k\boldsymbol{\alpha}(\boldsymbol{\omega}) - \boldsymbol{\zeta}^k(\boldsymbol{\omega})\mathbf{A}_1^k\mathbf{k}_1^k(\boldsymbol{\omega})}{\boldsymbol{\zeta}^k(\boldsymbol{\omega})\mathbf{A}_2^k\mathbf{k}_2^k(\boldsymbol{\omega}) + \mathbf{a}_1^k\mathbf{k}_1^k(\boldsymbol{\omega})}, \tag{4.16}$$

where

$$\begin{cases} \boldsymbol{\alpha}(\boldsymbol{\omega}) = [\alpha^1(\boldsymbol{\omega}), \dots, \alpha^{m_f}(\boldsymbol{\omega})]^T, \\ \mathbf{F}^k = [f^1(h_k(x)), \dots, f^{m_f}(h_k(x))], \\ \boldsymbol{\zeta}^k(\boldsymbol{\omega}) = [\zeta_1(\boldsymbol{\omega}), \dots, \zeta_{k-1}(\boldsymbol{\omega})], \\ \mathbf{k}_1^k(\boldsymbol{\omega}) = [\kappa_1^1(\boldsymbol{\omega}), \dots, \kappa_1^{M_k}(\boldsymbol{\omega})]^T, \\ \mathbf{k}_2^k(\boldsymbol{\omega}) = [\kappa_2^1(\boldsymbol{\omega}), \dots, \kappa_2^{M'_k}(\boldsymbol{\omega})]^T, \end{cases}$$

and,

$$(\mathbf{A}_1^k)_{ji} = a(l_1^i(x), h_j(x), h_k(x)), \quad (\mathbf{A}_2^k)_{jr} = a(l_2^r(x)h_k(x), h_j(x), h_k(x)),$$
$$(\mathbf{a}_1^k)_i = a(l_1^i(x), h_k(x), h_k(x)), \text{ for } 1 \leq i \leq M_k,\ 1 \leq j \leq k-1, \text{ and } 1 \leq r \leq M'_k.$$

For $k = 1, \dots, N$, the matrixes $\mathbf{A}_1^k$ and $\mathbf{A}_1^k$, and the vectors $\mathbf{F}^k$, $\mathbf{a}_1^k$ are independent of parameter $\boldsymbol{\omega}$, their computation is once and in the offline phase. The online computation is to calculate (3.2) for any $\boldsymbol{\omega} \in \Omega$. This is efficient because the online computation only involves the separated representation (3.2).

5. Steady Navier–Stokes equation. We consider the following steady Navier–Stokes equations with uncertain parameters: $\forall \boldsymbol{\omega} \in \Omega$, we find the velocity $\mathbf{u}(x, \boldsymbol{\omega}) : D \to R^2$ and the pressure $p(x, \boldsymbol{\omega}) : D \to R$ such that

$$\begin{cases} -\nabla.\big(\nu(\boldsymbol{\omega})\nabla\mathbf{u}(x,\boldsymbol{\omega})\big) + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p(x,\boldsymbol{\omega}) = \mathbf{f}(x,\boldsymbol{\omega}) & \text{in } D, \\ \nabla\cdot\mathbf{u}(x,\boldsymbol{\omega}) = 0 & \text{in } D, \end{cases} \tag{5.1}$$

with the boundary conditions on ∂D given by $\mathbf{u} = \vec{w}$. Here $\nu(\boldsymbol{\omega})$ denotes the random viscosity. For simplicity, we consider the case of the homogeneous Dirichlet velocity boundary condition. Let $X := \big(H_0^1(D)\big)^2$ and $Q := L^2(D)$; then the weak formulation of (5.1) reads, $\forall \boldsymbol{\omega} \in \Omega$, we find $\{\mathbf{u}(\boldsymbol{\omega}), p(\boldsymbol{\omega})\} \in X \times Q$ such that

$$\begin{cases} a\big(\mathbf{u}(\boldsymbol{\omega}),\mathbf{v};\boldsymbol{\omega}\big) + c\big(\mathbf{u}(\boldsymbol{\omega}),\mathbf{u}(\boldsymbol{\omega}),\mathbf{v};\boldsymbol{\omega}\big) + b\big(\mathbf{v},p(\boldsymbol{\omega});\boldsymbol{\omega}\big) = f(\mathbf{v};\boldsymbol{\omega}) \quad \forall\ \mathbf{v} \in X, \\ b\big(\mathbf{u}(\boldsymbol{\omega}),q;\boldsymbol{\omega}\big) = 0 \quad \forall\ q \in Q, \end{cases} \tag{5.2}$$

where the bilinear forms $a\big(\cdot,\cdot;\boldsymbol{\omega}\big)$, $b\big(\cdot,\cdot;\boldsymbol{\omega}\big)$, the trilinear form $c\big(\cdot,\cdot,\cdot;\boldsymbol{\omega}\big)$, and the functional $f(\mathbf{v};\boldsymbol{\omega})$ are defined by

$$\begin{cases} a\big(\mathbf{u}(\boldsymbol{\omega}),\mathbf{v};\boldsymbol{\omega}\big) := \int_D \nu(\boldsymbol{\omega})\nabla\mathbf{u}(\boldsymbol{\omega}) : \nabla\mathbf{v}dx, \\ b\big(\mathbf{v},p(\boldsymbol{\omega});\boldsymbol{\omega}\big) := -\int_D p(\boldsymbol{\omega})\nabla\cdot\mathbf{v}dx, \\ c\big(\mathbf{u}(\boldsymbol{\omega}),\mathbf{u}(\boldsymbol{\omega}),\mathbf{v};\boldsymbol{\omega}\big) := \int_D (\mathbf{u}(\boldsymbol{\omega})\cdot\nabla\mathbf{u}(\boldsymbol{\omega}))\cdot\mathbf{v}dx, \\ f(\mathbf{v};\boldsymbol{\omega}) := \int_D \mathbf{f}(\boldsymbol{\omega})\cdot\mathbf{v}dx. \end{cases}$$

We note that the bilinear form $a(\cdot,\cdot;\boldsymbol{\omega})$ is affine with respect to $\boldsymbol{\omega}$ such that

$$a(\mathbf{w},\mathbf{v};\boldsymbol{\omega}) = \nu(\boldsymbol{\omega})a^0\big(\mathbf{w},\mathbf{v}\big) \quad \forall\mathbf{v},\mathbf{w} \in X,\ \forall\boldsymbol{\omega} \in \Omega, \tag{5.3}$$

where $\nu(\boldsymbol{\omega})$ is a stochastic function, and $a^0 : X \times X \longrightarrow R$ is a bilinear form independent of $\boldsymbol{\omega}$. We assume that the linear functional $f(\cdot;\boldsymbol{\omega})$ is affine with respect to $\boldsymbol{\omega}$ such that

$$f(\mathbf{v};\boldsymbol{\omega}) = \sum_{i=1}^{m_f} \alpha^i(\boldsymbol{\omega})f^i(\mathbf{v}) \quad \forall\mathbf{v} \in X, \quad \forall\boldsymbol{\omega} \in \Omega, \tag{5.4}$$

where $\{\alpha^i(\boldsymbol{\omega})\}_{i=1}^{m_f}$ are stochastic functions, and each $f^i : X \longrightarrow R$ is a continuous functional independent of $\boldsymbol{\omega}$.

Now we cast the steady Navier–Stokes equation into the general framework. In this case, we let $\mathcal{V} = X \times Q$ and define the mapping $G : \mathcal{V} \times \Omega \to \mathcal{V}^*$ as

$$\begin{aligned} \langle G(w;\boldsymbol{\omega}), v\rangle = g(w,v;\boldsymbol{\omega}) &= a\big(\mathbf{w},\mathbf{v};\boldsymbol{\omega}\big) + c\big(\mathbf{w},\mathbf{w},\mathbf{v};\boldsymbol{\omega}\big) + b\big(\mathbf{v},r;\boldsymbol{\omega}\big) \\ &\quad + b\big(\mathbf{w},q;\boldsymbol{\omega}\big) - f(\mathbf{v};\boldsymbol{\omega}) \quad \forall\ w,v \in \mathcal{V}, \end{aligned}$$

where $w = (\mathbf{w}, r)$ and $v = (\mathbf{v}, q)$. Then its Fréchet derivatives with respect to w at $z = (\mathbf{z}, s) \in \mathcal{V}$ can be defined as

$$\begin{aligned} dg[z](w, v; \boldsymbol{\omega}) = &\, a\big(\mathbf{w}, \mathbf{v}; \boldsymbol{\omega}\big) + c\big(\mathbf{w}, \mathbf{z}, \mathbf{v}; \boldsymbol{\omega}\big) + c\big(\mathbf{z}, \mathbf{w}, \mathbf{v}; \boldsymbol{\omega}\big) + b\big(\mathbf{v}, r; \boldsymbol{\omega}\big) \\ &+ b\big(\mathbf{w}, q; \boldsymbol{\omega}\big) \quad \forall\ w, v \in \mathcal{V}. \end{aligned}$$

We note that $dg[z](w, v; \boldsymbol{\omega})$ is independent on s due to the Navier–Stokes equations being linear with respect to the pressure variable.

Let $X_h \subset X$ and $Q_h \subset Q$ be the given finite dimensional approximation spaces. We can refer to [19, 37] for the details about the finite element approximation of the Navier–Stokes equation. As we know, the pair of finite dimensional spaces $\{\mathcal{V}_h, Q_h\}$ are required to satisfy the inf-sup condition (2.4). This will increase the computational complexity. However, the difficulty can be circumvented by modifying the variational problem, i.e., regularization or the penalty method [7, 19]. Here we attempt to regularize the variational problem for the Navier–Stokes equations.

Let $d(\cdot, \cdot; \boldsymbol{\omega}) : Q_h \times Q_h \to R$ be a continuous and coercive bilinear form for $\boldsymbol{\omega} \in \Omega$. We assume that $d(\cdot, \cdot; \boldsymbol{\omega})$ is affine with $\boldsymbol{\omega}$ such that

$$d(q, p; \boldsymbol{\omega}) = \sum_{i=1}^{m_c} k^i(\boldsymbol{\omega}) d^i(q, p) \quad \forall p, q \in Q_h,\ \forall \boldsymbol{\omega} \in \Omega.$$

For a small penalty parameter $\varepsilon > 0$, we rewrite the problem (5.2) as, $\forall \boldsymbol{\omega} \in \Omega$, we find $\{\mathbf{u}(\boldsymbol{\omega}), p(\boldsymbol{\omega})\} \in X_h \times Q_h$ such that

$$(5.5) \quad \begin{cases} a\big(\mathbf{u}(\boldsymbol{\omega}), \mathbf{v}; \boldsymbol{\omega}\big) + c\big(\mathbf{u}(\boldsymbol{\omega}), \mathbf{u}(\boldsymbol{\omega}), \mathbf{v}; \boldsymbol{\omega}\big) + b\big(\mathbf{v}, p(\boldsymbol{\omega}); \boldsymbol{\omega}\big) = f(\mathbf{v}; \boldsymbol{\omega}) \ \ \forall\ \mathbf{v} \in X_h, \\ b\big(\mathbf{u}(\boldsymbol{\omega}), q; \boldsymbol{\omega}\big) - \varepsilon d(p(\boldsymbol{\omega}), q; \boldsymbol{\omega}) = 0 \quad \forall\ q \in Q_h, \end{cases}$$

The penalty solution of (5.5) converges to the solution of the nonpenalized problem (5.2) as ε approaches zero [5, 6, 25, 41].

The Newton method of (5.2) reads, For an initial guess $(\mathbf{u}^0, p^0)$, at step k we seek $(\delta\mathbf{u}, \delta p) \in \mathcal{V}_h$ such that

$$(5.6) \quad \begin{cases} a\big(\delta\mathbf{u}, \mathbf{v}; \boldsymbol{\omega}\big) + c\big(\mathbf{u}_k, \delta\mathbf{u}, \mathbf{v}; \boldsymbol{\omega}\big) + c\big(\delta\mathbf{u}, \mathbf{u}_k, \mathbf{v}; \boldsymbol{\omega}\big) + b\big(\mathbf{v}, \delta p; \boldsymbol{\omega}\big) \\ \ = f(\mathbf{v}; \boldsymbol{\omega}) - a\big(\mathbf{u}_k, \mathbf{v}; \boldsymbol{\omega}\big) - c\big(\mathbf{u}_k, \mathbf{u}_k, \mathbf{v}; \boldsymbol{\omega}\big) - b\big(\mathbf{v}, p_k; \boldsymbol{\omega}\big) \quad \forall\ \mathbf{v} \in X_h, \\ b\big(\delta\mathbf{u}, q; \boldsymbol{\omega}\big) - \varepsilon d(\delta p, q; \boldsymbol{\omega}) = \varepsilon d(p_k, q; \boldsymbol{\omega}) - b\big(\mathbf{u}_k, q; \boldsymbol{\omega}\big) \quad \forall\ q \in Q_h, \end{cases}$$

and then we have $(\mathbf{u}_{k+1}, p_{k+1}) = (\mathbf{u}_k + \delta\mathbf{u}, p_k + \delta p)$. We note that the solution of the corresponding Stokes problem is a typical selection for the initial guess.

5.1. VS method for stochastic steady Navier–Stokes equation.

Now we present the VS method for the stochastic steady Navier–Stokes equation to find the numerical solutions under the form

$$(5.7) \qquad \begin{cases} \mathbf{u}(x, \boldsymbol{\omega}) \approx \mathbf{u}_N(x, \boldsymbol{\omega}) := \displaystyle\sum_{i=1}^{N} \zeta_i^u(\boldsymbol{\omega}) \mathbf{u}^i(x), \\ p(x, \boldsymbol{\omega}) \approx p_N(x, \boldsymbol{\omega}) := \displaystyle\sum_{i=1}^{N} \zeta_i^p(\boldsymbol{\omega}) p^i(x), \end{cases}$$

where $\zeta_i^u \in \mathcal{S}$ and $\zeta_i^p \in \mathcal{S}$ are stochastic functions, and $\mathbf{u}^i \in \mathcal{V}_h$ and $p^i \in \mathcal{V}_h$ are deterministic functions. We note that the Stokes problem corresponding to (5.2) reads, $\forall \boldsymbol{\omega} \in \Omega$, we find $\{\mathbf{u}(\boldsymbol{\omega}), p(\boldsymbol{\omega})\} \in X \times Q$ such that

$$\begin{cases} a\big(\mathbf{u}(\boldsymbol{\omega}), \mathbf{v}; \boldsymbol{\omega}\big) + b\big(\mathbf{v}, p(\boldsymbol{\omega}); \boldsymbol{\omega}\big) = f(\mathbf{v}; \boldsymbol{\omega}) \quad \forall\ \mathbf{v} \in X, \\ \qquad\qquad b\big(\mathbf{u}(\boldsymbol{\omega}), q; \boldsymbol{\omega}\big) = 0, \quad \forall\ q \in Q. \end{cases} \tag{5.8}$$

Here $\mathbf{u}^1(x)$ and $p^1(x)$ are taken as the solutions of the corresponding Stokes problem (5.9) such that

$$\begin{cases} a\big(\mathbf{u}^1(x), \mathbf{v}; \boldsymbol{\omega}_1\big) + b\big(\mathbf{v}, p^1(x); \boldsymbol{\omega}_1\big) = f(\mathbf{v}; \boldsymbol{\omega}_1) \quad \forall\ \mathbf{v} \in X_h, \\ b\big(\mathbf{u}^1(x), q; \boldsymbol{\omega}_1\big) - \varepsilon d(p^1(x), q; \boldsymbol{\omega}_1) = 0 \quad \forall\ q \in Q_h, \end{cases} \tag{5.9}$$

with $\boldsymbol{\omega}_1$ being selected randomly. We take $\mathbf{u}(\boldsymbol{\omega}) = \mathbf{u}^1(x)\zeta_1^u(\boldsymbol{\omega})$, $p(\boldsymbol{\omega}) = p^1(x)\zeta_1^p(\boldsymbol{\omega})$, $\mathbf{v} = \mathbf{u}^1(x)$, and $q = p^1(x)$ in (5.9). Then $\zeta_1^u(\boldsymbol{\omega})$ and $\zeta_1^p(\boldsymbol{\omega})$ are given by solving the following linear system:

$$\begin{cases} a_{11}(\boldsymbol{\omega})\zeta_1^u(\boldsymbol{\omega}) + a_{12}(\boldsymbol{\omega})\zeta_1^p(\boldsymbol{\omega}) = \sum_{i=1}^{m_f} \alpha^i(\boldsymbol{\omega}) f^i(\mathbf{u}^1(x)), \\ a_{21}(\boldsymbol{\omega})\zeta_1^u(\boldsymbol{\omega}) + a_{22}(\boldsymbol{\omega})\zeta_1^p(\boldsymbol{\omega}) = 0, \end{cases} \tag{5.10}$$

where

$$\begin{cases} a_{11}(\boldsymbol{\omega}) = \nu(\boldsymbol{\omega}) a^0\big(\mathbf{u}^1(x), \mathbf{u}^1(x)\big), \\ a_{12}(\boldsymbol{\omega}) = b\big(\mathbf{u}^1(x), p^1(x)\big), \\ a_{22}(\boldsymbol{\omega}) = -\varepsilon d\big(p^1(x), p^1(x)\big), \\ a_{21}(\boldsymbol{\omega}) = a_{12}(\boldsymbol{\omega}). \end{cases}$$

At the step $k \geq 2$, we let

$$\mathbf{e}_u(\boldsymbol{\omega}) := \mathbf{u}(\boldsymbol{\omega}) - \mathbf{u}_{k-1}(\boldsymbol{\omega}), \quad e_p(\boldsymbol{\omega}) := p(\boldsymbol{\omega}) - p_{k-1}(\boldsymbol{\omega}).$$

Based on (5.6), we take $r_1(\mathbf{v}; \boldsymbol{\omega}) \in X_h^*$ and $r_2(q; \boldsymbol{\omega}) \in Q_h^*$ as the residual such that

$$\begin{cases} r_1(\mathbf{v}; \boldsymbol{\omega}) := f(\mathbf{v}; \boldsymbol{\omega}) - a\big(\mathbf{u}_{k-1}, \mathbf{v}; \boldsymbol{\omega}\big) - c\big(\mathbf{u}_{k-1}, \mathbf{u}_{k-1}, \mathbf{v}; \boldsymbol{\omega}\big) - b\big(\mathbf{v}, p_{k-1}; \boldsymbol{\omega}\big), \\ r_2(q; \boldsymbol{\omega}) := -b\big(\mathbf{u}_{k-1}, q; \boldsymbol{\omega}\big) + \varepsilon d\big(p_{k-1}, q; \boldsymbol{\omega}\big), \end{cases} \tag{5.11}$$

and by (5.6), we get

$$\begin{cases} a\big(\mathbf{e}_u, \mathbf{v}; \boldsymbol{\omega}\big) + c\big(\mathbf{u}_{k-1}, \mathbf{e}_u, \mathbf{v}; \boldsymbol{\omega}\big) + c\big(\mathbf{e}_u, \mathbf{u}_{k-1}, \mathbf{v}; \boldsymbol{\omega}\big) + b\big(\mathbf{v}, e_p; \boldsymbol{\omega}\big) = r_1(\mathbf{v}; \boldsymbol{\omega}), \\ b\big(\mathbf{e}_u, q; \boldsymbol{\omega}\big) - \varepsilon d\big(e_p, q; \boldsymbol{\omega}\big) = r_2(q; \boldsymbol{\omega}) \ \ \forall\ \mathbf{v} \in X_h, \quad \forall\ q \in Q_h. \end{cases} \tag{5.12}$$

By the Riesz representation theory, there exist $\hat{\mathbf{e}}_{\mathbf{u}}(\boldsymbol{\omega}) \in X_h$ and $\hat{e}_p(\boldsymbol{\omega}) \in Q_h$ such that

$$\begin{cases} \big(\hat{\mathbf{e}}_u(\boldsymbol{\omega}), \mathbf{v}\big)_X = r_1(\mathbf{v}; \boldsymbol{\omega}) \quad \forall \mathbf{v} \in X_h, \\ \big(\hat{e}_p(\boldsymbol{\omega}), q\big)_Q = r_2(q; \boldsymbol{\omega}) \quad \forall q \in Q_h, \end{cases} \tag{5.13}$$

and we can evaluate the dual norm of the residual $r_1(\mathbf{v};\boldsymbol{\omega})$ and $r_2(q;\boldsymbol{\omega})$ by

$$\begin{cases} \|r_1(\mathbf{v};\boldsymbol{\omega})\|_{X^*} := \sup\limits_{\mathbf{v}\in X_h} \dfrac{r_1(\mathbf{v};\boldsymbol{\omega})}{\|\mathbf{v}\|_X} = \|\hat{\mathbf{e}}_u(\boldsymbol{\omega})\|_X, \\ \|r_2(q;\boldsymbol{\omega})\|_{Q^*} := \sup\limits_{q\in Q_h} \dfrac{r_2(q;\boldsymbol{\omega})}{\|q\|_Q} = \|\hat{e}_p(\boldsymbol{\omega})\|_Q. \end{cases} \tag{5.14}$$

The error estimator for the solution of (5.6) is defined by

$$\Delta_k(\boldsymbol{\omega}) := \sqrt{\|\hat{\mathbf{e}}_{\mathbf{u}}(\boldsymbol{\omega})\|_X^2 + \|\hat{e}_p(\boldsymbol{\omega})\|_Q^2}. \tag{5.15}$$

Then, we choose

$$\boldsymbol{\omega}_k := \arg\max_{\boldsymbol{\omega}\in\Xi} \Delta_k(\boldsymbol{\omega}).$$

Let $\mathbf{e}_u(x,\boldsymbol{\omega}_k)$ and $e_p(x,\boldsymbol{\omega}_k)$ be the solutions of (5.12) with $\boldsymbol{\omega} = \boldsymbol{\omega}_k$. Then $\mathbf{u}^k(x)$ and $p^k(x)$ in (5.7) are obtained by taking $\mathbf{u}^k(x) = \mathbf{e}_u(x,\boldsymbol{\omega}_k)$ and $p^k(x) = e_p(x,\boldsymbol{\omega}_k)$.

We take $\mathbf{e}_u(\boldsymbol{\omega}) := \mathbf{u}^k(x)\zeta_k^u(\boldsymbol{\omega})$ and $e_p(\boldsymbol{\omega}) := p^k(x)\zeta_k^p(\boldsymbol{\omega})$ in (5.12). For any $v \in X_h$, and for any $q \in Q_h$, we have

$$\begin{cases} a\big(\mathbf{u}^k(x)\zeta_k^u(\boldsymbol{\omega}),\mathbf{v};\boldsymbol{\omega}\big) + c\big(\mathbf{u}_{k-1}(\boldsymbol{\omega}),\mathbf{u}^k(x)\zeta_k^u(\boldsymbol{\omega}),\mathbf{v};\boldsymbol{\omega}\big) \\ +\, c\big(\mathbf{u}^k(x)\zeta_k^u(\boldsymbol{\omega}),\mathbf{u}_{k-1}(\boldsymbol{\omega}),\mathbf{v};\boldsymbol{\omega}\big) + b\big(\mathbf{v},p^k(x)\zeta_k^p(\boldsymbol{\omega});\boldsymbol{\omega}\big) = r_1(\mathbf{v};\boldsymbol{\omega}), \\ b\big(\mathbf{u}^k(x)\zeta_k^u(\boldsymbol{\omega}),q;\boldsymbol{\omega}\big) - \varepsilon d\big(p^k(x)\zeta_k^p(\boldsymbol{\omega}),q;\boldsymbol{\omega}\big) = r_2(q;\boldsymbol{\omega}). \end{cases} \tag{5.16}$$

By taking $\mathbf{v} = \mathbf{u}^k(x)$ and $q = p^k(x)$ in (5.16), and combining (5.7), (5.3), and (5.4), we get the linear system with regard to $\zeta_k^u(\boldsymbol{\omega})$, and $\zeta_k^p(\boldsymbol{\omega})$ as follows:

$$\begin{cases} a_{11}(\boldsymbol{\omega})\zeta_k^u(\boldsymbol{\omega}) + a_{12}(\boldsymbol{\omega})\zeta_k^p(\boldsymbol{\omega}) = r_1\big(\mathbf{u}^k(x);\boldsymbol{\omega}\big), \\ a_{21}(\boldsymbol{\omega})\zeta_k^u(\boldsymbol{\omega}) + a_{22}(\boldsymbol{\omega})\zeta_k^p(\boldsymbol{\omega}) = r_2\big(p^k(x);\boldsymbol{\omega}\big), \end{cases} \tag{5.17}$$

where

$$\begin{cases} a_{11}(\boldsymbol{\omega}) = \nu(\boldsymbol{\omega})a^0(\mathbf{u}^k,\mathbf{u}^k) + \sum_{i=1}^{k-1}\zeta_i^u(\boldsymbol{\omega})\big(c(\mathbf{u}^i,\mathbf{u}^k,\mathbf{u}^k) + c(\mathbf{u}^k,\mathbf{u}^i,\mathbf{u}^k)\big), \\ a_{12}(\boldsymbol{\omega}) = b\big(\mathbf{u}^k(x),p^k(x)\big), \qquad a_{21}(\boldsymbol{\omega}) = a_{12}(\boldsymbol{\omega}), \\ a_{22}(\boldsymbol{\omega}) = -\varepsilon d\big(p^k(x),p^k(x)\big), \end{cases}$$

and

$$\begin{aligned} r_1\big(\mathbf{u}^k(x);\boldsymbol{\omega}\big) &= \sum_{i=1}^{m_f}\alpha^i(\boldsymbol{\omega})f^i\big(\mathbf{u}^k(x)\big) - \sum_{i=1}^{k-1}\nu(\boldsymbol{\omega})\zeta_i^u(\boldsymbol{\omega})a\big(\mathbf{u}^i(x),\mathbf{u}^k(x)\big) \\ &\quad - \sum_{i=1}^{k-1}\sum_{j=1}^{k-1}\zeta_i^u(\boldsymbol{\omega})\zeta_j^u(\boldsymbol{\omega})c\big(\mathbf{u}^i(x),\mathbf{u}^j(x),\mathbf{u}^k(x)\big) \\ &\quad - \sum_{r=1}^{k-1}\zeta_r^p(\boldsymbol{\omega})b\big(\mathbf{u}^k(x),p^r(x)\big), \\ r_2\big(p^k(x);\boldsymbol{\omega}\big) &= -\sum_{i=1}^{k-1}\zeta_i^u(\boldsymbol{\omega})b\big(\mathbf{u}^i(x),p^k(x)\big) + \varepsilon\sum_{i=1}^{k-1}\zeta_i^p(\boldsymbol{\omega})d\big(p^i(x),p^k(x)\big). \end{aligned}$$

Therefore, $\zeta_k^u(\boldsymbol{\omega})$ and $\zeta_k^p(\boldsymbol{\omega})$ can be solved by the system (5.17).

We describe the main procedure for the VS method to solve the stochastic steady Navier–Stokes equation in Algorithm 2.

Algorithm 2. The VS method for the stochastic steady Navier–Stokes equation.

```
Input: The stochastic steady Navier–Stokes equation (5.1), a set of samples $\Xi \in \Omega$, and the error tolerance $\varepsilon_0$.
Output: The separated representation $\mathbf{u}_N(x,\boldsymbol{\omega}) := \sum_{i=1}^{N} \zeta_i^u(\boldsymbol{\omega})\mathbf{u}^i(x)$,
and $p_N(x,\boldsymbol{\omega}) := \sum_{i=1}^{N} \zeta_i^p(\boldsymbol{\omega})p^i(x)$.
1: Initialize the iteration counter $k = 1$, a random $\boldsymbol{\omega}_1 \in \Xi$;
   1.1: Calculate $\mathbf{u}^1(x)$ and $p^1(x)$ by solving (5.9) with $\boldsymbol{\omega} = \boldsymbol{\omega}_1$ by Newton's method;
   1.2: Determine $\zeta_1^u(\boldsymbol{\omega})$ and $\zeta_1^p(\boldsymbol{\omega})$ by the linear equation (5.10);
   1.3: Update $\Xi$ with $\Xi = \Xi \setminus \boldsymbol{\omega}_1$, take the approximation of $\mathbf{u}_1(x,\boldsymbol{\omega})$ as
        $\mathbf{u}_1(x,\boldsymbol{\omega}) := \mathbf{u}^1(x)\zeta_1^u(\boldsymbol{\omega})$, and $p_1(x,\boldsymbol{\omega}) := \zeta_1^p(\boldsymbol{\omega})p^1(x)$;
2: Update $k \to k+1$, and take $\boldsymbol{\omega}_k = \arg\max_{\boldsymbol{\omega}\in\Xi} \Delta_k(\boldsymbol{\omega})$; Then take the residual
   $r_1(\mathbf{v};\boldsymbol{\omega}) := f(\mathbf{v};\boldsymbol{\omega}) - a\big(\mathbf{u}_{k-1},\mathbf{v};\boldsymbol{\omega}\big) - c\big(\mathbf{u}_{k-1},\mathbf{u}_{k-1},\mathbf{v};\boldsymbol{\omega}\big) - b\big(\mathbf{v},p_{k-1};\boldsymbol{\omega}\big)$ and
   $r_2(q;\boldsymbol{\omega}) := -b\big(\mathbf{u}_{k-1},q;\boldsymbol{\omega}\big) + \varepsilon d\big(p_{k-1},q;\boldsymbol{\omega}\big)$;
3: Calculate $\mathbf{u}^k(x)$ and $p^k(x)$ by solving (5.12) with $\boldsymbol{\omega} = \boldsymbol{\omega}_k$, then determine
   $\zeta_k^u(\boldsymbol{\omega})$ and $\zeta_k^p(\boldsymbol{\omega})$ by the linear equation (5.17);
4: Update $\Xi$ with $\Xi = \Xi \setminus \boldsymbol{\omega}_k$, and take $\mathbf{u}_k(x,\boldsymbol{\omega}) := \sum_{i=1}^{k} \zeta_i^u(\boldsymbol{\omega})\mathbf{u}^i(x)$ and
   $p_k(x,\boldsymbol{\omega}) := \sum_{i=1}^{k} \zeta_i^p(\boldsymbol{\omega})p^i(x)$;
5: Take $\varepsilon_k := \max_{\boldsymbol{\omega}\in\Xi} \Delta_k(\boldsymbol{\omega})$;
6: Return to step 2 if $\varepsilon_k \geq \varepsilon_0$, otherwise terminate.
7: $N = k$.
```

6. Numerical results.

In this section, we describe two specific examples to illustrate the performance of the proposed VS methods. In section 6.1, we consider a nonlinear elliptic equation with random variables to illustrate the performance of the VS method. In section 6.2, a stochastic steady Navier–Stokes equation is considered to present the performance of the proposed method.

6.1. A nonlinear elliptic PDE with random variables.

In this subsection, to illustrate the performance of the proposed numerical Algorithm 1, we consider the following nonlinear model for numerical computation:

$$\begin{cases} -\mathrm{div}\big(\kappa(u,\boldsymbol{\omega})\nabla u(x,\boldsymbol{\omega})\big) = \mathrm{f}(x,\boldsymbol{\omega}) \quad \text{in} \quad D \times \Omega, \\ u(x,\boldsymbol{\omega}) = 0 \quad \text{on } \Gamma_1, \\ \kappa(u,\boldsymbol{\omega})\nabla u(x,\boldsymbol{\omega}) \cdot \mathbf{n} = 0 \quad \text{on other boundaries.} \end{cases} \tag{6.1}$$

Let $\kappa(u,\boldsymbol{\omega}) : D \times \Omega \longrightarrow \mathbb{R}$ be a diffusion coefficient function, which is defined by

$$\kappa(u,\boldsymbol{\omega}) = 1 + \omega_9^2 u^2 + e^{\omega_{10}} u^3.$$

Then its Fréchet derivatives with respect to u can be defined as

$$\kappa_u(u,\boldsymbol{\omega}) = 2\omega_9^2 u + 3e^{\omega_{10}} u^2.$$

Here we choose the physical domain $D = (0,1)^2$ and Dirichlet boundary $\Gamma_1 = (0,1)\times 1$, and $\mathbf{n}$ denotes the outward unit normal vector on $\partial D \setminus \Gamma_1$. The source term $\mathrm{f}(x,\boldsymbol{\omega})$ is taken as a random field, which is characterized by a two-point exponential covariance function cov[f], i.e.,

$$\mathrm{cov}[\mathrm{f}](x_1,y_1;x_2,y_2) = \sigma^2 \exp\left(-\frac{|x_1-x_2|^2}{2l_x^2} - \frac{|y_1-y_2|^2}{2l_y^2}\right),$$

where (x_i, y_i) $(i = 1, 2)$ is the spatial coordinate in D. Here the variance $\sigma^2 = 2$ and the correlation length $l_x = l_y = 0.2$. The random source term $\mathrm{f}(x, \boldsymbol{\omega})$ is obtained by truncating a Karhunen–Loève expansion, i.e.,

$$\mathrm{f}(x, \boldsymbol{\omega}) := E[\mathrm{f}] + \sum_{i=1}^{8} \sqrt{\gamma_i} b_i(x) \omega_i. \tag{6.2}$$

Here $E[\mathrm{f}] = 1$ and the random vector $\boldsymbol{\omega} := (\omega_1, \omega_2, \ldots, \omega_8) \in \mathbb{R}^8$. We assume that each ω_i $(i = 1, \ldots, 8)$ is uniformly distributed in the interval $[-1, 1]$.

We use 200×200 uniform grid for the partition of spatial domain to compute the reference solution and solve (4.8) in the offline phase. The Newton iteration method is used to compute the reference solution based on the bilinear Q_1 element (bilinear basis functions). We apply the VS method to get the variable separation representation of the solution for the nonlinear elliptic PDE. Given $\varepsilon_0 = 10^{-5}$ and $|\Xi| = 80$ samples selected from random domain Ω, the separated representation $u_N(x, \boldsymbol{\omega}) = \sum_{i=1}^{N} \zeta_i(\boldsymbol{\omega}) h_i(x)$ with $N = 20$ is obtained by Algorithm 1.

For the case of nonlinear initialization, $\zeta_1(\boldsymbol{\omega})$ is obtained by solving nonlinear system (4.14) using Newton's iteration method. Now we take $\zeta_1(\boldsymbol{\omega})$ as the solution of a linear system (4.15) and provide some numerical results to show that the proposed VS method can still render a good approximation. Given the same $\varepsilon_0 = 10^{-5}$ and $|\Xi| = 80$ samples, then we get the corresponding separated representation as

$$\hat{u}_N(x, \boldsymbol{\omega}) = \sum_{i=1}^{20} \zeta_i(\boldsymbol{\omega}) h_i(x).$$

For this numerical example, we choose the error estimators as

$$\varepsilon_k := \begin{cases} \max\limits_{\boldsymbol{\omega} \in \Xi} \|u(\boldsymbol{\omega})\|_{\mathcal{V}}, & k = 1, \\ \max\limits_{\boldsymbol{\omega} \in \Xi} \|u(\boldsymbol{\omega}) - u_{k-1}(\boldsymbol{\omega})\|_{\mathcal{V}}, & k \geq 2. \end{cases} \tag{6.3}$$

Based on the two cases, we depict the error estimators versus the different numbers of the separated terms in Figure 6.1. By the figure, we find that (1) the curves of

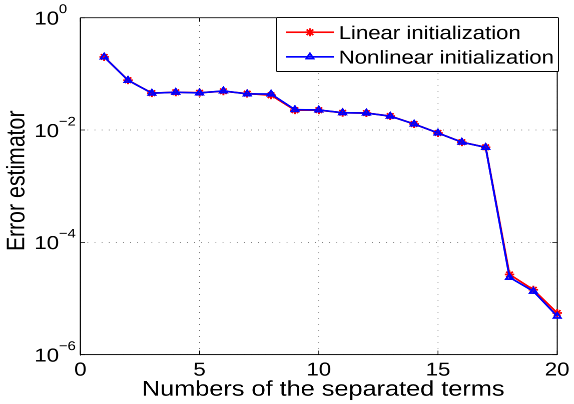


FIG. 6.1. *The error estimator corresponding to the numbers of the separated terms k in the VS method for the nonlinear elliptic PDE.*

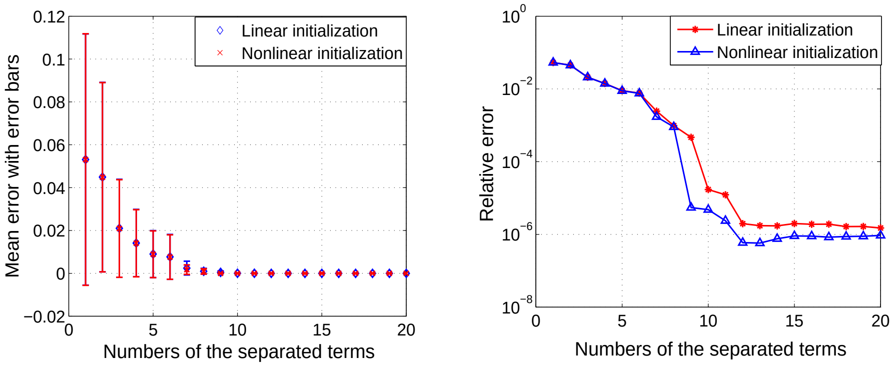
the error estimator are nearly identical for the two cases; (2) the error estimator decays gradually in the first few steps and then speedily when the iteration number k increases.

We choose 10^5 samples $\{\boldsymbol{\omega}_i\}_{i=1}^{10^5}$ randomly; the relative error is defined as follows:

$$\varepsilon^u = \frac{1}{M}\sum_{i=1}^{M}\frac{\|u(x,\boldsymbol{\omega}^{(i)}) - \tilde{u}(x,\boldsymbol{\omega}^{(i)})\|_{\mathcal{V}}^2}{\|u(x,\boldsymbol{\omega}^{(i)})\|_{\mathcal{V}}^2}, \tag{6.4}$$

where $M = 10^5$ and $\tilde{u}(x,\boldsymbol{\omega}^{(i)})$ is the solution by the VS method, and $u(x,\boldsymbol{\omega}^{(i)})$ is the reference solution solved by the Newton iteration method on the 200 × 200 grid. Based on the two representations $u_N(x,\boldsymbol{\omega})$ and $\hat{u}_N(x,\boldsymbol{\omega})$, we compute the relative errors defined by (6.4). Then we depict the mean of the relative errors with error bars $\big([(\text{mean} - \text{standard deviation})\ (\text{mean} + \text{standard deviation})]\big)$ and the average relative errors in log scale (Figure 6.2(b)) corresponding to the numbers of the separated terms k in Figure 6.2. From the figures we can see that (1) as the number of separated terms increases, the approximations become more accurate for the two cases; (2) Figure 6.2(a) shows that the error bars of the relative errors become more compact as the number of separated terms increases; (3) by Figure 6.2(b), we can see that the VS method with $\zeta_1(\boldsymbol{\omega})$ determined by nonlinear system (4.14) always achieves better approximation than the VS method with $\zeta_1(\boldsymbol{\omega})$ determined by a linear system (4.15); (4) the VS method for the two cases can render the robust and accurate approximation for the nonlinear elliptic PDE. Consquently, we have the following conclusion that the proposed VS method can still achieve a good approximation when we choose $\zeta_1(\boldsymbol{\omega})$ as the solution of a linearized system (4.15).

In Table 6.1, we list the average relative errors in Table 6.1 along with the average online CPU time based on the 10^5 random samples. From the table, we can conclude that (1) as the number of separated terms increases, the average online CPU time by the VS method is added slowly first, and then increases distinctly; (2) the magnitude of average online CPU time by the VS method is much smaller than that of the reference method; (3) the approximation obtained by the VS method achieves a good trade-off in both approximation accuracy and computation efficiency.



(a) Relative errors with error bars.

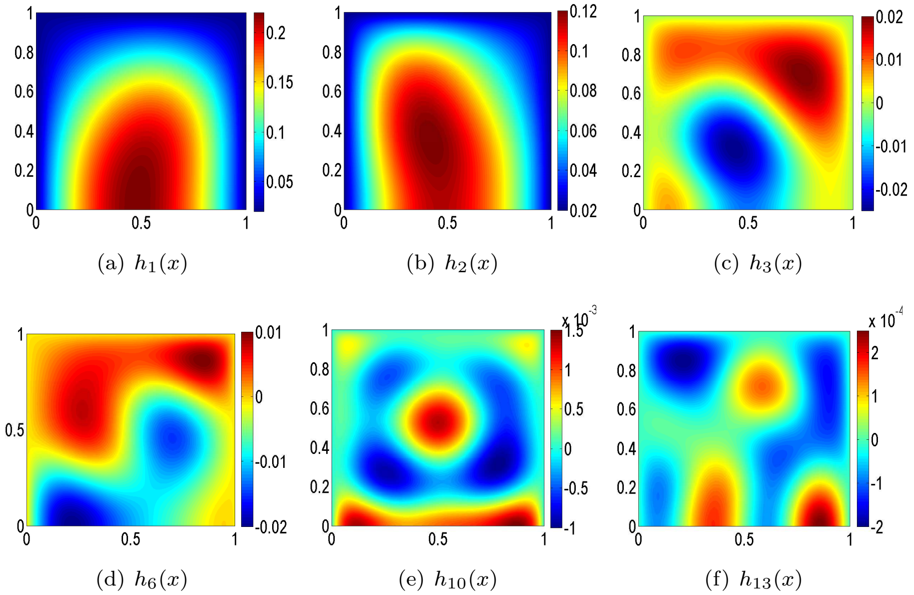
(b) Relative errors in log scale.

FIG. 6.2. *The mean of the relative errors with error bars and the average relative errors (log scale) corresponding to the numbers of the separated terms k for the nonlinear elliptic PDE.*

TABLE 6.1

Comparison of the CPU time and average relative errors for the reference and VS methods with different numbers of separated terms.

Strategies		Average error ε^u	Mean online CPU time
VS method	$N = 2$	4.49×10^{-2}	$1.62 \times 10^{-4}s$
	$N = 4$	1.41×10^{-2}	$1.88 \times 10^{-4}s$
	$N = 7$	1.72×10^{-3}	$1.95 \times 10^{-4}s$
	$N = 10$	4.79×10^{-6}	$2.29 \times 10^{-4}s$
	$N = 15$	9.06×10^{-7}	$7.46 \times 10^{-4}s$
	$N = 20$	8.74×10^{-7}	$2.81 \times 10^{-3}s$
Reference method		\	$7.86s$



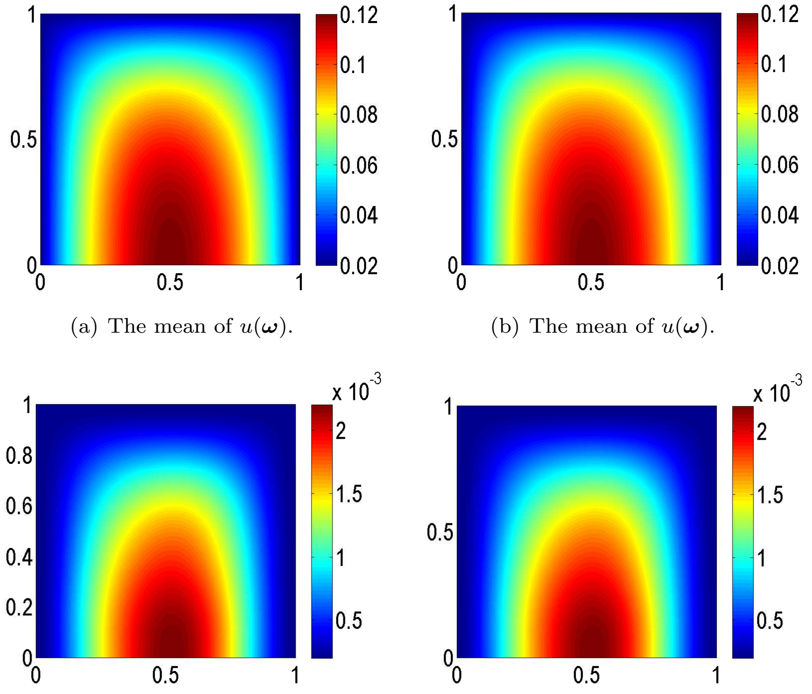
(a) $h_1(x)$ (b) $h_2(x)$ (c) $h_3(x)$

(d) $h_6(x)$ (e) $h_{10}(x)$ (f) $h_{13}(x)$

FIG. 6.3. *Spatial structure of the modes $\mathbf{u}^i(x)$ for velocity $\mathbf{u}(\boldsymbol{\omega})$ of the VS method in x-axis direction.*

In Figure 6.3 we depict the spatial structure of some modes $h_i(x)$ of $u(x, \boldsymbol{\omega})$ obtained by the VS method, which show that the first mode $h_1(x)$ represent the coarsest information of $u(x, \boldsymbol{\omega})$, and the last few modes capture the fine-scale information. Figure 6.4 demonstrates the mean and variance profiles of solution $u(x, \boldsymbol{\omega})$ for reference, and the VS method with the number of the separated terms $N = 15$. By the figure, we can see the mean profiles for the two methods are all nearly identical. Based on the reference and the VS method, Figure 6.5 depicts the probability density estimate of $u(x, \boldsymbol{\omega})$ at the single measurement location where the variance of $u(\hat{x}, \boldsymbol{\omega})$ is maximal (left) or minimal (right) for all $x \in D$. The figure shows that the VS method renders the same probability density as the reference solution.

6.2. Numerical results for a steady Navier–Stokes equation with random viscosity and random forcing term. In this section, we consider a steady Navier–Stokes equation with the homogeneous Dirichlet velocity boundary condition.



(a) The mean of $u(\boldsymbol{\omega})$. (b) The mean of $u(\boldsymbol{\omega})$.

(c) The variance of $u(\boldsymbol{\omega})$. (d) The variance of $u(\boldsymbol{\omega})$.

FIG. 6.4. *The mean and variance of the pressure and vorticity profiles. The first column are the reference solutions, the second column are the solutions by the VS method with the number of separated terms N being* 15.

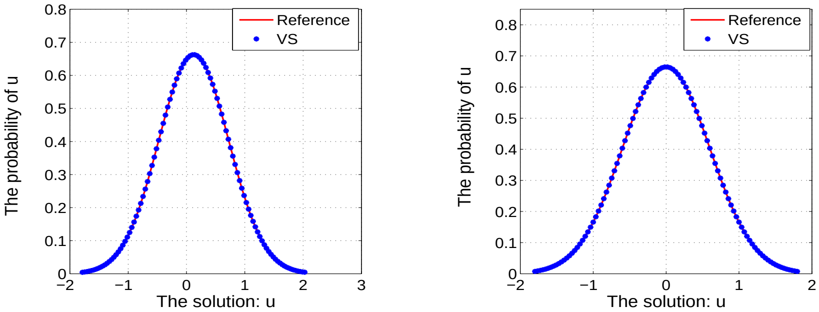


FIG. 6.5. *Probability density of $u(x_0, \boldsymbol{\omega})$ for reference, and the VS method with the number of the separated terms $N = 15$, where the variance of $u(x_0, \boldsymbol{\omega})$ is maximal (left) or minimal (right) $\forall x \in D$.*

The model is defined in the spatial domain $D = (0,1)^2$. Here we take $X := \left(H_0^1(D)\right)^2$, $Q := L^2(D)$, and $\mathcal{V} := X \times Q$. The random viscosity ν is given by

$$\nu(\boldsymbol{\omega}) = \nu_\varepsilon + \nu'(\boldsymbol{\omega}),$$

where $\nu'(\boldsymbol{\omega})$ has a log-normal distribution, and coefficient of variation $C_{\nu'} = 1.5$. Then the random viscosity is taken as

$$\nu(\boldsymbol{\omega}) = \nu_\varepsilon + \exp\left(\sigma \boldsymbol{\omega}_T\right), \ \sigma := \frac{\log C_{\nu'}}{2.85},$$

where $\nu_\varepsilon = \frac{1}{50}$, and

$$\boldsymbol{\omega}_T = \frac{1}{\sqrt{N_\nu}} \sum_{i=1}^{N_\nu} \omega_i. \tag{6.5}$$

Each ω_i $(i = 1, \dots, N_\nu)$ has a normal distribution with zero mean and unit variance, i.e., $\omega_i \sim N(0,1)$. $\boldsymbol{\omega}_T$ is in turn a normalized, centered, Gaussian random variable. The forcing term in (5.1) is also random (see, e.g., [42]). We take the vertical component of the rotational of the force field as a stochastic field $\Phi(x, \boldsymbol{\omega})$, which is characterized by a two-point exponential covariance function cov[Φ], i.e.,

$$\operatorname{cov}[\Phi](x_1, y_1; x_2, y_2) = \sigma^2 \exp\left(-\frac{|x_1 - x_2|^2}{2l_x^2} - \frac{|y_1 - y_2|^2}{2l_y^2}\right), \tag{6.6}$$

where (x_i, y_i) $(i = 1, 2)$ is the spatial coordinate in D. Here the variance $\sigma^2 = 2$ and correlation length $l_x = l_y = 0.2$. Without loss of generality, we may assume that the random field $\Phi(x, \boldsymbol{\omega})$ admits a Karhunen-Loève expansion, i.e.,

$$\Phi(x, \boldsymbol{\omega}) = \Phi_0 + \sum_{i=1}^{N_f} \omega_{N_\nu + i} \Phi_i,$$

where $\Phi_0 = 1$ and the random vector $\boldsymbol{\omega} := (\omega_1, \omega_2, \dots, \omega_{N_\nu + N_f}) \in \mathbb{R}^{N_\nu + N_f}$. Each ω_i $(i = N_\nu + 1, \dots, N_\nu + N_f)$ has a normal distribution with zero mean and unit variance, i.e., $\omega_i \sim N(0,1)$. Then $\mathbf{f}(x, \boldsymbol{\omega})$ is defined by

$$\mathbf{f}(x, \boldsymbol{\omega}) \approx \mathbf{f}^{N_f}(x, \boldsymbol{\omega}) := \mathbf{f}_0(x) + \sum_{i=1}^{N_f} \omega_{N_\nu + i} \mathbf{f}_i(x), \ \mathbf{f}_i(x) = \nabla \wedge \begin{bmatrix} 0 \\ 0 \\ \psi_i(x) \end{bmatrix},$$

where ψ_i is the solution of the following Poisson equation:

$$\begin{cases} -\triangle \psi_i(x) = \Phi_i(x) & \text{in } D, \\ \psi_i(x) = 0 & \text{in } \partial D. \end{cases} \tag{6.7}$$

Here we set $N_\nu = 9$ and $N_f = 9$ for the numerical simulation. This implies that the solution depends on $N_\nu + N_f = 18$ random variables.

For partition of the spatial domain, we apply 100 × 100 uniform grid and the stabilized rectangular elements $Q_1 - Q_1$ (bilinear basis functions for both velocity components and pressure) to compute the reference solution and solve (5.16) in the offline phase. Here, we utilize the approach suggested by Dohrmann and Bochev [17] to stabilize rectangular elements $Q_1 - Q_1$, where the penalty term in (5.5) is defined as

$$d\big(p, q; \boldsymbol{\omega}\big) = (p - \Pi p, q - \Pi q; \boldsymbol{\omega}),$$

where Π denotes the L^2 projection from the actual discrete pressure space Q_1 into the piecewise constant function space P_0. For this penalty method, we take $\varepsilon = 1$ in (5.5). Note that this projection is defined locally, that is, Πq is taken as a constant function in each element. Let $\mathbf{u}(x,\boldsymbol{\omega})$ and $p(x,\boldsymbol{\omega})$ be the reference solutions for velocity and pressure, respectively, solved by the Newton iteration method based on the system by penalty. Let $\mathbf{u}_H(x,\boldsymbol{\omega})$ and $p_H(x,\boldsymbol{\omega})$ be solved by the VS method proposed in section 5.1. Then the relative mean errors for velocity and pressure are defined, respectively, by

$$\varepsilon_u = \frac{1}{M}\sum_{i=1}^{M}\frac{\|\mathbf{u}(x,\boldsymbol{\omega}^{(i)}) - \mathbf{u}_H(x,\boldsymbol{\omega}^{(i)})\|^2_{L^2(D)}}{\|\mathbf{u}(x,\boldsymbol{\omega}^{(i)})\|^2_{L^2(D)}}, \tag{6.8}$$

$$\varepsilon_p = \frac{1}{M}\sum_{i=1}^{M}\frac{\|p(x,\boldsymbol{\omega}^{(i)}) - p_H(x,\boldsymbol{\omega}^{(i)})\|^2_{L^2(D)}}{\|p(x,\boldsymbol{\omega}^{(i)})\|^2_{L^2(D)}}. \tag{6.9}$$

We utilize the VS method proposed in section 5.1 to get the separated representations of the solution for this numerical example. We take $\varepsilon_0 = 10^{-4}$ in Algorithm 2 and select $|\Xi| = 60$ samples from the random domain Ω. Then $\{\zeta_i^u(\boldsymbol{\omega}), \mathbf{u}^i(x)\}_{i=1}^N$ and $\{\zeta_i^p(\boldsymbol{\omega}), p^i(x)\}_{i=1}^N$ are obtained by Algorithm 2 with $N = 30$.

In Figure 6.6, we depict the error estimator ε_k defined by equation $\varepsilon_k := \max_{\boldsymbol{\omega}\in\Xi}\Delta_k(\boldsymbol{\omega})$ versus the number of the separated terms k for the VS method. From the figure, we can see the error estimator decays fast as the iteration number k increases.

Figure 6.7 shows the spatial structure of some modes $\mathbf{u}^i(x)$ in the x-axis direction obtained by the VS method, and Figure 6.8 shows the spatial structure of some modes $\mathbf{u}^i(x)$ in the y-axis direction. From the figures, we find that, for both the velocity $\mathbf{u}$ and pressure p, the first mode $\mathbf{u}^1(x)$ represent the coarsest information, and the last few modes capture the fine-scale information.

Based on the representations (5.7), we select 10^5 samples $\{\boldsymbol{\omega}_i\}_{i=1}^{10^5}$ randomly and compute the relative errors defined by (6.8) and (6.9). In Figure 6.9, we show the mean of the relative errors with error bars $\big([$(mean $-$ standard deviation) (mean $+$ standard deviation)$]\big)$ for the VS method with different numbers of separated terms k. From the figure, we have two observations: (1) as the number of separated terms increases, the mean of the relative errors for both velocity $\mathbf{u}$ and pressure p decays distinctly;

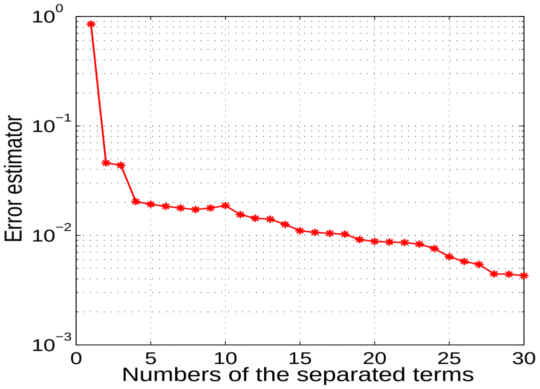
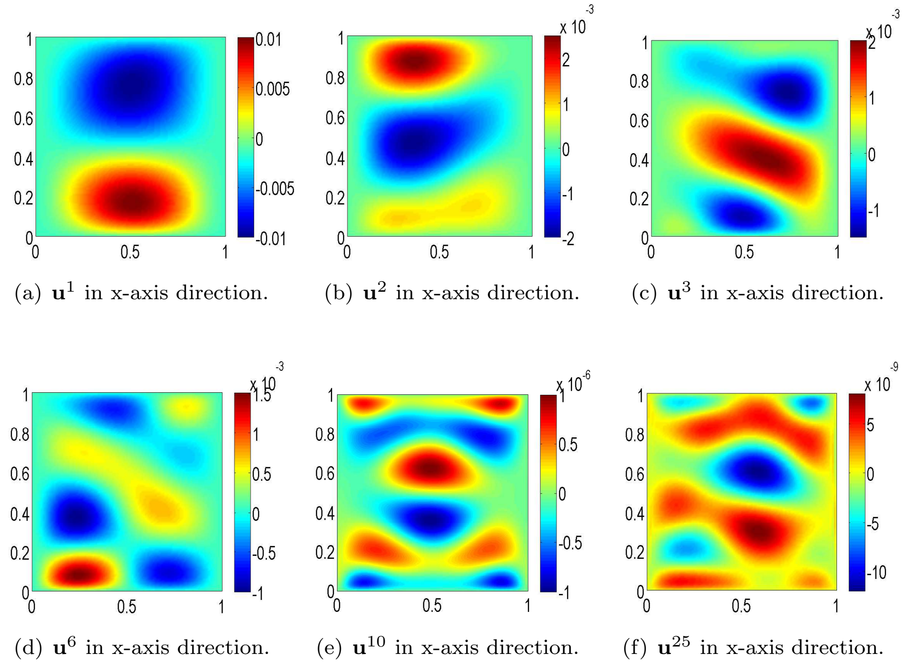


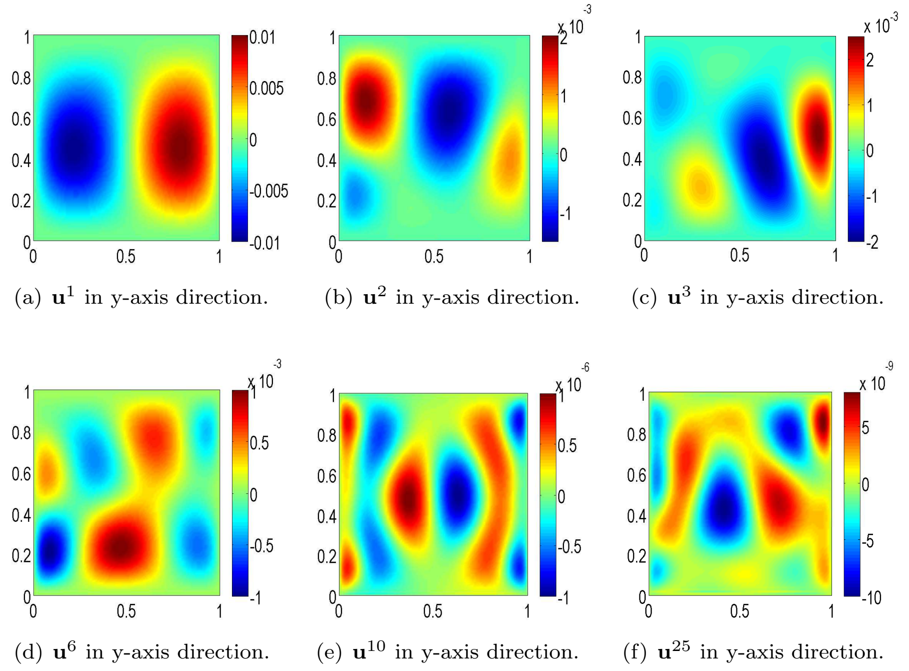
FIG. 6.6. *The error estimator ε_k corresponding to the numbers of the separated terms k in the VS method for the steady Navier–Stokes equation.*



(a) $\mathbf{u}^1$ in x-axis direction. (b) $\mathbf{u}^2$ in x-axis direction. (c) $\mathbf{u}^3$ in x-axis direction.

(d) $\mathbf{u}^6$ in x-axis direction. (e) $\mathbf{u}^{10}$ in x-axis direction. (f) $\mathbf{u}^{25}$ in x-axis direction.

FIG. 6.7. *Spatial structure of the modes* $\mathbf{u}^i(x)$ *for velocity* $\mathbf{u}(\boldsymbol{\omega})$ *of the VS method in x-axis direction.*



(a) $\mathbf{u}^1$ in y-axis direction. (b) $\mathbf{u}^2$ in y-axis direction. (c) $\mathbf{u}^3$ in y-axis direction.

(d) $\mathbf{u}^6$ in y-axis direction. (e) $\mathbf{u}^{10}$ in y-axis direction. (f) $\mathbf{u}^{25}$ in y-axis direction.

FIG. 6.8. *Spatial structure of the modes* $\mathbf{u}^i(x)$ *for velocity* $\mathbf{u}(\boldsymbol{\omega})$ *of the VS method in y-axis direction.*

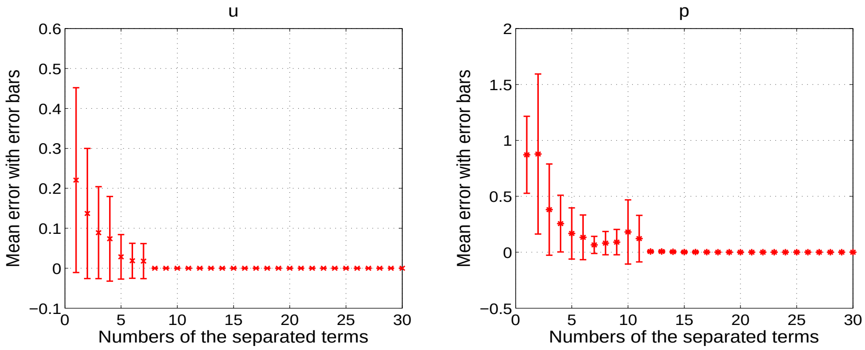


FIG. 6.9. *The mean of the relative errors with error bars corresponding to the numbers of the separated terms k in the VS method for the steady Navier–Stokes equation.*

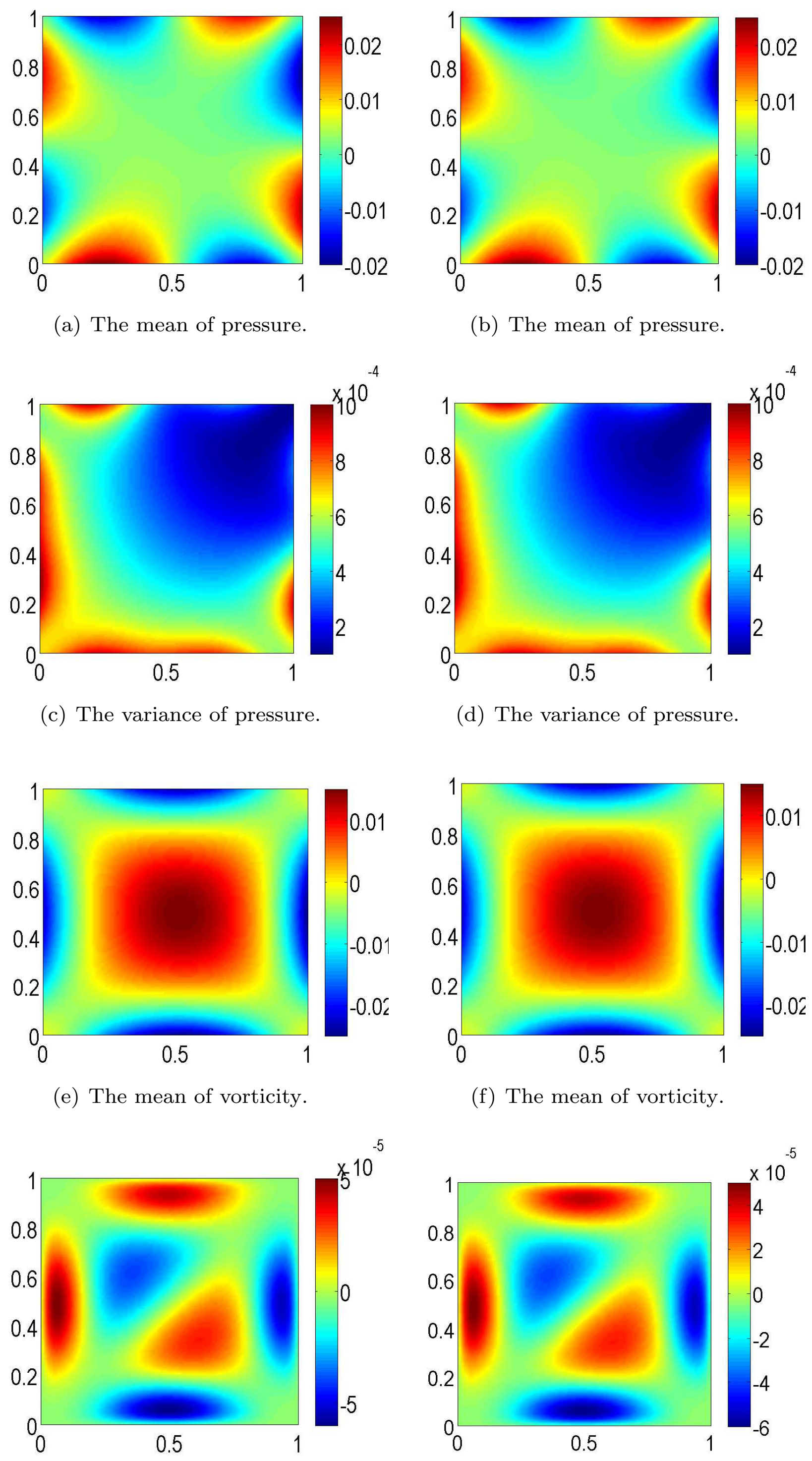
(2) for velocity $\mathbf{u}$, the error bars of the relative errors become more compact; for pressure p, there are some special cases, but on the whole, the error bars become more compact as the iteration number increases; (3) the VS method can provide a robust and accurate approximation for the steady Navier–Stokes problem with random influence. Let $\mathbf{u} = (u_{x_1}, u_{x_2})$. Here the two-dimensional vorticity (scalar) variable ξ is defined by

$$\xi := \frac{\partial u_{x_2}}{\partial x_1} - \frac{\partial u_{x_1}}{\partial x_2},$$

which acts in a direction orthogonal to the xy plane. In Figure 6.10, we depict the mean and variance of the pressure and vorticity profiles. From the figure, we can see that the VS method with the number of the separated terms being 15 gives almost the same mean profiles for velocity, vorticity, and pressure for the example.

To illustrate the efficiency of the proposed VS method, we compute the realizations corresponding to the 10^5 parameter samples. In Table 6.2, we list the mean of the relative error, CPU time (online CPU time $\mathcal{T}_{\text{on}}$, offline CPU time $\mathcal{T}_{\text{off}}$, total CPU time $\mathcal{T}_{\text{tot}}$, and average online CPU time $\bar{\mathcal{T}}_{\text{on}}$) needed for the VS method with the number of the separated terms being 8, 15, and 25, and average CPU time for the reference and VS methods. From the table, we can conclude that (1) as the iteration number k increases, the average relative errors become smaller, the CPU times (online CPU time $\mathcal{T}_{\text{on}}$, offline CPU time $\mathcal{T}_{\text{off}}$, total CPU time $\mathcal{T}_{\text{tot}}$, and average online CPU time T_{on}) needed for the VS method increase steadily too; (2) the average online CPU time by the VS method is much smaller than that of the reference method; (3) the VS method achieves a good trade-off in both approximation accuracy and computation efficiency for the stochastic steady Navier–Stokes equations.

Finally, we investigate the case that the viscosity parameter $\nu(\boldsymbol{\omega})$ depends on the different numbers of random variables. Given the same error tolerance $\varepsilon_0 = 10^{-3}$ and the same set of samples Ξ with $|\Xi| = 60$, here we consider the steady Navier–Stokes equations (5.1) with $N_\nu = 6, 9,$ and 15 in (6.5), and get the separated representations with $N = 25$ by Algorithm 2. In Figure 6.11, we depict the average relative errors for velocity $\mathbf{u}$ and pressure p versus the different iteration number for the three different models with $N_\nu = 6, 9,$ and 15. The figure shows that the viscosity parameter $\nu(\boldsymbol{\omega})$ depending on the different numbers of random variables have little impact on the performance of the VS method for the stochastic steady Navier–Stokes equations.



(a) The mean of pressure.

(b) The mean of pressure.

(c) The variance of pressure.

(d) The variance of pressure.

(e) The mean of vorticity.

(f) The mean of vorticity.

(g) The variance of vorticity.

(h) The variance of vorticity.

Fig. 6.10. *The mean and variance of the pressure and vorticity profiles. The first column is the reference solutions, and the second column is the solutions by the VS method with the number of the separated terms N being* 15.

Table 6.2
Comparison of average relative errors and the CPU time for the reference and VS methods with the number of the separated terms being 8, 15, and 25, based on 10^6 *parameter samples.*

Strategies	The VS method			Reference method
	$N = 8$	$N = 15$	$N = 25$	
ε_u	7.31×10^{-5}	3.19×10^{-8}	4.10×10^{-10}	\
ε_p	8.69×10^{-2}	1.70×10^{-3}	4.68×10^{-6}	\
$\mathcal{T}_{\text{off}}$	$384.64s$	$572.94s$	$840.45s$	\
$\mathcal{T}_{\text{on}}$	$32.31s$	$54.14s$	$68.59s$	\
$\mathcal{T}_{\text{tot}}$	$416.95s$	$627.08s$	$909.04s$	$5.02 \times 10^5 s$
T_{on}	$3.23 \times 10^{-4}s$	$5.41 \times 10^{-4}s$	$6.86 \times 10^{-4}s$	$5.02s$

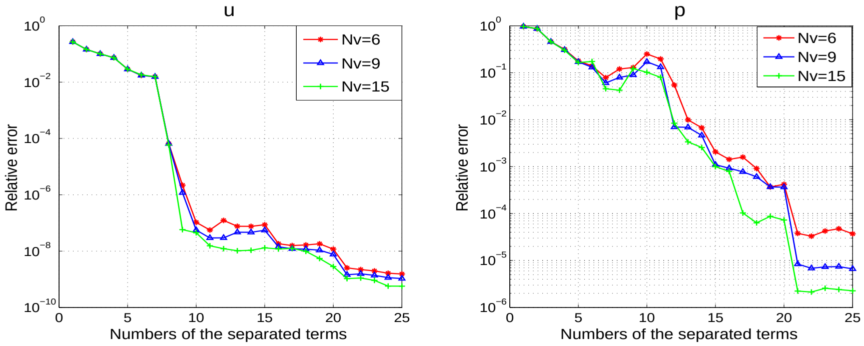


Fig. 6.11. *The mean of the relative errors corresponding to the numbers of the separated terms* k *in the VS method for the steady Navier–Stokes equation.*

7. Conclusions. In the work, we proposed the VS method for the nonlinear problems with random inputs to get the separated representations of the solutions. The proposed VS approach is devoted to constructing a low rank approximation of the Galerkin solution for nonlinear problems in a systematic enrichment manner. Compared with proper generalized decomposition, no iteration is performed at each enrichment step. To obtain the efficient and reliable approximation (1.1) for the nonlinear problems, we combined the VS method with the classic linearization techniques such as the Newton method to avoid solving nonlinear problems directly in the offline phase. At each enrichment step, the stochastic basis function $\zeta_k(\boldsymbol{\omega})$ was determined by the linearized stochastic equation, which is deduced from the original nonlinear problems; simultaneously, we also obtained the deterministic basis function by solving a linear equation. We note that the computation efficiency was dramatically improved by storing the stochastic basis functions in matrix form such as (4.16) and making best use of the precalculated terms. Specifically, for given $\boldsymbol{\omega} \in \Omega$, we calculate $\zeta_k(\boldsymbol{\omega})$ by (4.16) with the concrete numbers $\{\zeta_i(\boldsymbol{\omega})\}_{i=1}^{k-1}$, which have been calculated previously. Therefore, in this work, no stochastic approximation methods, such as the sparse low rank tensor approximation method used in [29], were performed to remove the mutual dependance of the stochastic basis functions. For the nonlinear problems with random inputs, the VS method was presented in an abstract framework. After that, we described the details of the VS method by two nonlinear models that fit in the abstract framework. Finally, we applied the proposed method to two numerical examples with random inputs and performed some careful numerical analysis for these numerical examples.

In the future, we will apply the proposed methods to the inverse problems with nonlinearities. It is desirable to explore rigorous convergence analysis for the VS method. Ongoing works are also focused on the extension to nonlinear unsteady problems. In addition, we will pay more attention to the applications of the proposed method to some models in dynamical systems, especially for some biological systems.

Appendix A. The definition of $\boldsymbol{\Delta_k(\omega)}$. An effective error estimator $\Delta_k(\boldsymbol{\omega})$ is crucial for both the efficiency and the reliability of the VS method. Here we will describe two ways for the definition of $\Delta_k(\boldsymbol{\omega})$. First, people usually utilize the Riesz representation to define the error estimator, e.g., (4.10) and (5.15), as presented in sections 4.1 and 5.1. For the computation of the error estimator $\Delta_k(\boldsymbol{\omega})$, we can utilize an offline-online procedure presented in [39, 28].

Now, we take the stochastic steady Navier–Stokes equation as an example to present the details of the offline-online procedure. By (5.3), (5.4), (5.11), and (5.12), we get

$$\begin{cases} \big(\hat{\mathbf{e}}_u(\boldsymbol{\omega}), \mathbf{v}\big)_X = -\displaystyle\sum_{i=1}^{k-1}\sum_{j=1}^{k-1} \zeta_i^u(\boldsymbol{\omega})\zeta_j^u(\boldsymbol{\omega}) c\big(\mathbf{u}^i(x), \mathbf{u}^j(x), \mathbf{v}\big) - \sum_{i=1}^{k-1} \nu(\boldsymbol{\omega})\zeta_i^u(\boldsymbol{\omega}) a\big(\mathbf{u}^i(x), \mathbf{v}\big) \\ \qquad\qquad + \displaystyle\sum_{i=1}^{m_f} \alpha^i(\boldsymbol{\omega}) f^i\big(\mathbf{v}\big) - \sum_{r=1}^{k-1} \zeta_r^p(\boldsymbol{\omega}) b\big(\mathbf{v}, p^r(x)\big) \ \forall\ \mathbf{v}\ \in X_h, \\ \big(\hat{e}_p(\boldsymbol{\omega}), q\big)_Q = -\displaystyle\sum_{i=1}^{k-1} \zeta_i^u(\boldsymbol{\omega}) b\big(\mathbf{u}^i(x), q\big) + \varepsilon \sum_{i=1}^{k-1} \zeta_i^p(\boldsymbol{\omega}) d\big(p^i(x), q\big) \ \forall\ q \in Q_h. \end{cases}$$

This implies that

$$\text{(A.1)} \qquad \begin{cases} \hat{\mathbf{e}}_u(\boldsymbol{\omega}) = \displaystyle\sum_{i=1}^{m_f} \alpha^i(\boldsymbol{\omega})\mathcal{F}^i + \sum_{i=1}^{k-1} \nu(\boldsymbol{\omega})\zeta_i^u(\boldsymbol{\omega})\mathcal{L}_i \\ \qquad\qquad + \displaystyle\sum_{i=1}^{k-1}\sum_{j=1}^{k-1} \zeta_i^u(\boldsymbol{\omega})\zeta_j^u(\boldsymbol{\omega})\mathcal{X}_j^i + \sum_{r=1}^{k-1} \zeta_r^p(\boldsymbol{\omega})\mathcal{B}_r, \\ \hat{e}_p(\boldsymbol{\omega}) = \displaystyle\sum_{i=1}^{k-1} \zeta_i^u(\boldsymbol{\omega})\mathcal{P}_i + \varepsilon \sum_{i=1}^{k-1} \zeta_i^p(\boldsymbol{\omega})\mathcal{C}_i, \end{cases}$$

where $\mathcal{F}^i$ is the Riesz representation of $f^i(\mathbf{v})$, i.e., $(\mathcal{F}^i, \mathbf{v})_X = f^i(\mathbf{v})$ for any $\mathbf{v} \in X_h$, $\mathcal{L}_i$ is the Riesz representation of $-a(\mathbf{u}^i, \mathbf{v})$, i.e., $(\mathcal{L}_i, \mathbf{v})_X = -a(\mathbf{u}^i, \mathbf{v})$ for any $\mathbf{v} \in X_h$, $\mathcal{X}_j^i$ is the Riesz representation of $-c\big(\mathbf{u}^i(x), \mathbf{u}^j(x), \mathbf{v}\big)$, i.e., $(\mathcal{X}_j^i, \mathbf{v})_X = -c\big(\mathbf{u}^i(x), \mathbf{u}^j(x), \mathbf{v}\big)$ for any $\mathbf{v} \in X_h$, $\mathcal{B}_r$ is the Riesz representation of $-b\big(\mathbf{v}, p^r(x)\big)$, i.e., $(\mathcal{B}_r, \mathbf{v})_X = -b\big(\mathbf{v}, p^r(x)\big)$ for any $\mathbf{v} \in X_h$, $\mathcal{P}_i$ is the Riesz representation of $-b(\mathbf{u}^i, q)$, i.e., $(\mathcal{P}_i, q)_Q = -b(\mathbf{u}^i, q)$ for any $q \in Q_h$, and $\mathcal{C}_i$ is the Riesz representation of $d\big(p^i(x), q\big)$, i.e., $(\mathcal{C}_i, q)_X = d\big(p^i(x), q\big)$ for any $q \in Q_h$. Then (A.1) gives rise to

$$
\begin{aligned}
\|\hat{\mathbf{e}}_u(\boldsymbol{\omega})\|_X^2 = {}& \sum_{n=1}^{m_f} \alpha^n(\boldsymbol{\omega}) \sum_{n'=1}^{m_f} \alpha^{n'}(\boldsymbol{\omega})(\mathcal{F}^n, \mathcal{F}^{n'})_X + \sum_{i=1}^{k-1} \nu(\boldsymbol{\omega}) \zeta_i^u(\boldsymbol{\omega}) \\
& \times \left(2 \sum_{n=1}^{m_f} \alpha^n(\boldsymbol{\omega})(\mathcal{F}^n, \mathcal{L}_i)_X + \sum_{i'=1}^{k-1} \nu(\boldsymbol{\omega}) \zeta_{i'}^u(\boldsymbol{\omega})(\mathcal{L}_{i'}, \mathcal{L}_i)_X \right) \\
& + \sum_{i=1}^{k-1} \sum_{j=1}^{k-1} \zeta_i^u(\boldsymbol{\omega}) \zeta_j^u(\boldsymbol{\omega}) \times \left(2 \sum_{n=1}^{m_f} \alpha^n(\boldsymbol{\omega})(\mathcal{F}^n, \mathcal{X}_j^i)_X \right. \\
& \left. + 2 \sum_{i'=1}^{k-1} \nu(\boldsymbol{\omega}) \zeta_{i'}^u(\boldsymbol{\omega})(\mathcal{L}_{i'}, \mathcal{X}_j^i)_X + \sum_{i'=1}^{k-1} \sum_{j'=1}^{k-1} \zeta_{i'}^u(\boldsymbol{\omega}) \zeta_{j'}^u(\boldsymbol{\omega})(\mathcal{X}_{j'}^{i'}, \mathcal{X}_j^i)_X \right) \\
& + \sum_{r=1}^{k-1} \zeta_r^p(\boldsymbol{\omega}) \left(2 \sum_{n=1}^{m_f} \alpha^n(\boldsymbol{\omega})(\mathcal{F}^n, \mathcal{B}_r)_X + 2 \sum_{i=1}^{k-1} \nu(\boldsymbol{\omega}) \zeta_i^u(\boldsymbol{\omega})(\mathcal{L}_i, \mathcal{B}_r)_X \right. \\
& \left. + 2 \sum_{i'=1}^{k-1} \sum_{j'=1}^{k-1} \zeta_{i'}^u(\boldsymbol{\omega}) \zeta_{j'}^u(\boldsymbol{\omega})(\mathcal{X}_{j'}^{i'}, \mathcal{B}_r)_X + \sum_{r'=1}^{k-1} \zeta_{r'}^p(\boldsymbol{\omega})(\mathcal{B}_{r'}, \mathcal{B}_r)_X \right),
\end{aligned} \tag{A.2}
$$

$$
\begin{aligned}
\|\hat{e}_p(\boldsymbol{\omega})\|_Q^2 = {}& \sum_{i=1}^{k-1} \sum_{i'=1}^{k-1} \zeta_i^u(\boldsymbol{\omega}) \zeta_{i'}^u(\boldsymbol{\omega})(\mathcal{P}_i, \mathcal{P}^{i'})_Q + \sum_{i=1}^{k-1} \zeta_i^p(\boldsymbol{\omega}) \\
& \times \left(2 \sum_{j=1}^{k-1} \zeta_j^u(\boldsymbol{\omega})(\mathcal{P}_j, \mathcal{C}_i)_Q + \sum_{i'=1}^{k-1} \zeta_{i'}^p(\boldsymbol{\omega})(\mathcal{C}_{i'}, \mathcal{C}_i)_Q \right).
\end{aligned} \tag{A.3}
$$

In the offline stage, we compute $\mathcal{F}^n$, $\mathcal{L}_i$, $\mathcal{X}_j^i$, $\mathcal{B}_i$, $\mathcal{C}_i$, and $\mathcal{P}_i$, where $1 \le i, j \le k-1$ and $1 \le n \le m_f$. We store all of the inner products used in (A.2) and (A.3) for the online stage. In the online stage, we use (A.2) and (A.3) to compute $\|\hat{e}_u(\boldsymbol{\omega})\|_X$ and $\|\hat{e}_p(\boldsymbol{\omega})\|_Q$ for any $\boldsymbol{\omega} \in \Xi$, and then obtain $\Delta_k(\boldsymbol{\omega})$ by (5.15).

By (A.1) we can find that, to evaluate the error estimator $\Delta_k(\boldsymbol{\omega})$ by (A.2) and (A.3) rapidly, $m_f + 4(N-1) + (N-1)^2$ linear problems are required to be computed on the fine grid. Therefore, the error estimator $\Delta_k(\boldsymbol{\omega})$ can be defined as the error in $\mathcal{V}$-norm directly, that is,

$$
\Delta_k(\boldsymbol{\omega}) = \sqrt{\|\mathbf{u}(\boldsymbol{\omega}) - \mathbf{u}_{k-1}(\boldsymbol{\omega})\|_X^2 + \|p(\boldsymbol{\omega}) - p_{k-1}(\boldsymbol{\omega})\|_Q^2},
$$

when $m_f + 4(N-1) + (N-1)^2 \ge |\Xi|$.

Similarly, we can define the error estimator $\Delta_k(\boldsymbol{\omega})$ for the nonlinear elliptic PDEs with random influences as

$$
\Delta_k(\boldsymbol{\omega}) = \|u(\boldsymbol{\omega}) - u_{k-1}(\boldsymbol{\omega})\|_{\mathcal{V}}.
$$

To identify the sample $\boldsymbol{\omega}_i = \arg\max_{\boldsymbol{\omega} \in \Xi} \Delta_i(\boldsymbol{\omega})$ quickly at each iteration step, the solutions $\{u(\boldsymbol{\omega}); \boldsymbol{\omega} \in \Xi\}$ can be calculated and stored first.

Acknowledgment. The authors would like to thank Peng Chen for the interesting discussions about the work.

REFERENCES

[1] I. Babuška, F. Nobile, and R. Tempone, *A stochastic collocation method for elliptic partial differential equations with random input data*, SIAM J. Numer. Anal., 45 (2007), pp. 1005–1034.

[2] I. Babuška, R. Tempone, and G. Zouraris, *Solving elliptic boundary value problems with uncertain coefficients by the finite element method: The stochastic formulation*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 1251–1294.

[3] G. Blatman and B. Sudret, *Sparse polynomial chaos expansions and adaptive stochastic finite elements using a regression approach*, C. R. Mécanique, 336 (2008), pp. 518–523.

[4] A. Buffa, Y. Maday, A. Patera, C. Prud'homme, and G. Turinici, *A priori convergence of the greedy algorithm for the parametrized reduced basis method*, ESAIM Math. Model. Numer. Anal., 46 (2012), pp. 595–603.

[5] M. Bercovier, *Perturbation of a mixed variational problem. Application to mixed finite element methods*, RAIRO Anal. Numer., 12 (1978), pp. 211–236.

[6] D. Boffi, F. Brezzi, and M. Fortin, *Mixed Finite Element Methods and Applications*, Springer Ser. Comput. Math. 44, Springer-Verlag, Berlin, 2013.

[7] P. B. Bochev and R. B. Lehoucq, *Regularization and stabilization of discrete saddle-point variational problems*, Electron. Trans. Numer. Anal., 22 (2006), pp. 97–113.

[8] G. Caloz and J. Rappaz, *Numerical Analysis for Nonlinear and Bifurcation Problems*, Handb. Numer. Anal. 5, Elsevier/North-Holland, Amsterdam, 1997, pp. 487–637.

[9] C. Canuto, T. Tonn, and K. Urban, *A posteriori error analysis of the reduced basis method for nonaffine parametrized nonlinear PDEs*, SIAM J. Sci. Comput., 47 (2009), pp. 2001–2022.

[10] P. Chen, A. Quarteroni, and G. Rozza, *A weighted reduced basis method for elliptic partial differential equations with random input data*, SIAM J. Numer. Anal., 51 (2013), pp. 3163–3185.

[11] P. Chen and A. Quarteroni, *Weighted reduced basis method for stochastic optimal control problems with elliptic PDE constraint*, SIAM/ASA J. Uncertain. Quantif., 2 (2014), pp. 364–396.

[12] Y. Chen, J. S. Hesthaven, Y. Maday, and J. Rodriguez, *Certified reduced basis methods and output bounds for the harmonic Maxwell's equations*, SIAM J. Sci. Comput., 32 (2010), pp. 970–996.

[13] S. Chaturantabut and D. C. Sorensen, *Nonlinear model reduction via discrete empirical interpolation*, SIAM J. Sci. Comput., 32 (2010), pp. 2737–2764.

[14] P. G. Ciarlet, *Linear and Nonlinear Functional Analysis with Applications*, SIAM, Philadelphia, 2014.

[15] H. Cho, D. Venturi, and G. E. Karniadakis, *Numerical methods for high-dimensional probability density function equations*, J. Comput. Phys., 305 (2016), pp. 817–837.

[16] P. Deuflhard, *Newton's Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*, Springer Ser. Comput. Math. 35, Springer, New York, 2011.

[17] C. Dohrmann and P. Bochev, *A stabilized finite element method for the Stokes problem based on polynomial pressure projections*, Internat. J. Numer. Methods Fluids, 46 (2004), pp. 183–201.

[18] M. Drohmann, B. Haasdonk, and M. Ohlberger, *Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation*, SIAM J. Sci. Comput., 34 (2012), pp. A937–A969.

[19] H. C. Elman, D. J. Silvester, and A. Wathen, *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*, Oxford University Press, New York, 2004.

[20] A. Falcó and A. Nouy, *Proper generalized decomposition for nonlinear convex problems in tensor Banach spaces*, Numer. Math., 121 (2012), pp. 503–530.

[21] P. Frauenfelder, C. Schwab, and R. Todor, *Finite elements for elliptic problems with stochastic coefficients*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 205–228.

[22] B. Ganapathysubramanian and N. Zabaras, *Sparse grid collocation schemes for stochastic natural convection problems*, J. Comput. Phys., 225 (2007), pp. 652–685.

[23] R. Ghanem, *Ingredients for a general purpose stochastic finite elements implementation*, Comput. Methods Appl. Mech. Engrg., 168 (1999), pp. 19–34.

[24] R. Ghanem and P. Spanos, *Stochastic Finite Elements: A Spectral Approach*, Dover, Mineola, NY, 2003.

[25] V. Girault and P. A. Raviart, *Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms*, Springer Ser. Comput. Math. 5, Springer-Verlag, Berlin, 1986.

[26] D. M. Ghiocel and R. G. Ghanem, *Stochastic finite-element analysis of seismic soil-structure interaction*, J. Engrg. Mech., 128 (2002), pp. 66–77.

[27] L. Jiang and Q. Li, *Model's sparse representation based on reduced mixed GMsFE basis methods*, J. Comput. Phys., 338 (2017), pp. 285–312.

[28] L. Jiang and Q. Li, *Model reduction method using variable-separation for stochastic saddle point problems*, J. Comput. Phys., 354 (2015), pp. 43–66.

[29] Q. Li and L. Jiang, *A variable-separation method based on sparse and low rank representation for stochastic partial differential equations*, J. Sci. Comput., 39 (2017), pp. A2879–A2910.

[30] M. G. Larson and F. Bengzon, *The Finite Element Method: Theory, Implementation, and Applications*, Springer, New York, 2013.

[31] H. Matthies and A. Keese, *Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 1295–1331.

[32] G. Migliorati, F. Nobile, E. von Schwerin, and R. Tempone, *Analysis of discrete L^2 projection on polynomial spaces with random evaluations*, Found. Comput. Math., 14 (2014), pp. 419–456.

[33] A. Nouy, *Recent developments in spectral stochastic methods for the numerical solution of stochastic partial differential equations*, Arch. Comput. Methods Eng., 16 (2009), pp. 251–285.

[34] A. Nouy and O. Le Maître, *Generalized spectral decomposition for stochastic nonlinear problems*, J. Comput. Phys., 228 (2009), pp. 202–235.

[35] A. Nouy, *Proper generalized decompositions and separated representations for the numerical solution of high dimensional stochastic problems*, Arch. Comput. Methods. Eng., 17 (2010), pp. 403–434.

[36] F. Nobile, R. Tempone, and C. Webster, *A sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM J. Numer. Anal., 46 (2008), pp. 2309–2345.

[37] A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*, Springer-Verlag, Berlin, 1994.

[38] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced Basis Methods for Partial Differential Equations*, Springer, New York, 2015.

[39] G. Rozza, D. Huynh, and A. Patera, *Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations*, Arch. Comput. Methods Eng., 15 (2008), pp. 229–275.

[40] D. Ryckelynck, *A priori hyperreduction method: An adaptive approach*, J. Comput. Phys., 202 (2005), pp. 346–366.

[41] R. Temam, *Navier-Stokes Equations: Theory and Numerical Analysis*, Stud. Math. Appl. 2, North-Holland, Amsterdam, 1979.

[42] L. Tamellini, O. Le Maître, and A. Nouy, *Model reduction based on proper generalized decomposition for the stochastic steady incompressible Navier–Stokes equations*, SIAM J. Sci. Comput., 36 (2014), pp. 1089–1117.

[43] E. Zeidler, *Nonlinear Functional Analysis and Its Applications*, Volume I*: Fixed-Point Theorems,* Springer, New York, 1985.

[44] D. Xiu and J. Hesthaven, *High-order collocation methods for differential equations with random inputs*, SIAM J. Sci. Comput., 27 (2005), pp. 1118–1139.

[45] D. Xiu, *Efficient collocational approach for parametric uncertainty analysis*, Commun. Comput. Phys., 2 (2007), pp. 293–309.