Domain Decomposition of Stochastic PDEs: Adaptations to Large Random Variables

Ajit Desai[a], Mohammad Khalil[b], Chris Pettit[c], Dominique Poirel[d], Abhijit Sarkar[a,*]

[a]*Department of Civil and Environmental Engineering, Carleton University, Ottawa, Ontario, Canada*
[b]*Sandia National Laboratories, Livermore, California, United States*
[c]*Department of Aerospace Engineering, United States Naval Academy, Annapolis, Maryland, United States*
[d]*Department of Mechanical and Aerospace Engineering, Royal Military College of Canada, Kingston, Ontario, Canada*

Abstract

In this paper, the authors aim to demonstrate that the intrusive approach converges faster in terms of accuracy and computational time compared to a non-intrusive approach for non-Gaussian random processes modeling the stochastic system parameters. We also illustrate the way to alleviate computational complexity arising due to the necessity of intrusive adjustments in the intrusive approach using general purpose deterministic FEM packages. Furthermore, we show that the computation time required to solve large-scale intrusive system arising in the context of stochastic PDEs with a large number of random variables can be substantially reduced by employing scalable domain decomposition solvers with the aid of high-performance computing. The solvers developed using FEniCS and UQTk, and implemented using MPI and PETSc are shown to scale up to 20 random variables and 4000 cores. The total floating-point operations required to solve such problem reached to 1.728 Petaflop. To the best knowledge of the authors, these algorithms have not been applied previously for uncertainty quantification at this scale.

Keywords:
Schur complement, Parallel preconditioner, Balancing domain decomposition by constraints, Dual-primal finite element tearing and interconnect method, Polynomial chaos expansion, Coarse grid

[*]Corresponding author. Tel.: +1 613 520 2600x6320; fax: +1 613 520 3951
Email address: `abhijit.sarkar@carleton.ca` (Abhijit Sarkar)

1. Introduction

2. Spectral Stochastic Finite Element Methods for Stochastic PDEs: Intrusive Versus Non-Intrusive Approach

The spectral stochastic finite element method (SSFEM) is a powerful numerical tool employed for UQ of stochastic PDEs [1, 2]. The SSFEM is based on polynomial chaos expansion (PCE), i.e., a series representation of random vectors in terms of orthogonal polynomials [1–3]. Here the term *spectral* is used because of the representation of a stochastic process as an infinite linear combination of orthogonal functions and their coefficients, which is analogous to a Fourier series representation of a function and it involves eigenvalue analysis of covariance function. Similar to the spectral method used in the numerical solution of deterministic differential equations[4], the SSFEM is a global approach, i.e. the selected basis functions are nonzero over the whole domain [1]. The application of SSFEM requires: (a) spatial discretization of a stochastic PDE using a finite element method, (b) stochastic discretization of the random system parameters and the solution process using the PCE, followed by a standard Galerkin projection along the random dimensions, and (c) the resulting system is solved for the PCE coefficients of the solution process using an intrusive approach or non-intrusive approach [1, 2, 5, 6].

This paper is organized in the following manner. Sec. 3 is dedicated for the spectral representation of input and output stochastic processes using the Karhunen-Loève expansion (KLE) and polynomial chaos expansion (PCE). This is followed by the formulation and implementation of the intrusive and non-intrusive SSFEM in the Sec. 4. Finally, in Sec. 5, we compare the performance of intrusive and non-intrusive approaches for non-Gaussian random system parameters in terms of error analysis in the solution coefficients.

3. Spectral Representation of Stochastic Processes

In numerous engineering applications, the system parameters and/or the initial and boundary conditions vary randomly across the physical domain of the problem. If sufficient statistical information is available, uncertainty in the system parameters can be modeled as random vectors or random processes characterized by multiple random variables [1, 2]. Representing uncertainty as a random vector necessitates its joint probability density function (PDF). On the other hand, the complete representation of uncertainty in the form of stochastic process requires n-th order (indexed by space or time or both) joint PDF of the process [1, 2, 6, 7]. Usually it is not possible to estimate the joint PDF of any arbitrary order of a random process in practical engineering applications. Therefore, the representation of an non-stationary and non-Gaussian stochastic process is often restricted to marginal PDF and the auto-correlation function [1, 7, 8].

The most widely utilized approaches for the spectral representation of stochastic processes are the KLE and PCE [1, 2, 9]. These methods are based on expanding the random function in a Fourier-type series [1, 2]. In the following sections, the mathematical details on the KLE and PCE representation of stochastic process are provided. In Sec. 3.1 and 3.2 we present the formulation of the KLE and PCE, respectively. Note that, in this paper, we will largely focus on the implementation aspects of these approaches. The theoretical details of the KLE and PCE are available in the numerous articles [1, 2, 6, 9–14].

3.1. Karhunen-Loève Expansion

The Karhunen-Loève expansion (KLE) is an useful tool available to characterize the random process with a set of random variables. It is based on the spectral decomposition of covariance kernel to obtain the set of eigenvalues and eigenfunctions used to characterize the stochastic process [1, 2, 6].

Consider $\alpha(\boldsymbol{x}, \xi(\theta))$ to be a stochastic process, a function of the position vector $\boldsymbol{x}$ defined over physical domain $\mathscr{D}$ and the set of random variables $\boldsymbol{\xi} = \{\xi_n\}_{n=1}^{\infty}$ which are functions of random event θ defined by the complete probability space $(\Omega, \mathscr{E}, \mathscr{P})$. Here Ω is a sample space, $\mathscr{E}$ is a set of events and $\mathscr{P}$ is a probability measure (refer [8] for more details). The KL expansion of such a stochastic process can be written as [1, 6],

$$\alpha(\boldsymbol{x}, \boldsymbol{\xi}(\theta)) = \bar{\alpha}(\boldsymbol{x}) + \sum_{n=1}^{\infty} \sqrt{\lambda_n}\, f_n(\boldsymbol{x})\xi_n(\theta), \tag{1}$$

where $\bar{\alpha}(\boldsymbol{x})$ is the expected value of the random process and $\{\xi_n\}$ is a set of uncorrelated (not necessarily independent) random variables. For a Gaussian stochastic process, $\{\xi_n\}_{n=1}^{\infty}$ is a set of independent Gaussian random variables. The eigenvalues $\{\lambda_n\}_{n=1}^{\infty}$ and eigenfunctions $\{f_n\}_{n=1}^{\infty}$ of the covariance function $C_{\alpha\alpha}(\boldsymbol{x}_1, \boldsymbol{x}_2)$ are obtained by solving the following integral equation [1]

$$\int_{\Omega} C_{\alpha\alpha}(\boldsymbol{x}_1, \boldsymbol{x}_2) f_n(\boldsymbol{x}_1) d\boldsymbol{x}_1 = \lambda_n\, f_n(\boldsymbol{x}_2). \tag{2}$$

Although an infinite number of KLE terms are showed in Eq. (1), for numerical simulation, a finite number of KLE modes (L) are sufficient for a smooth stochastic process [1, 2]. The truncated representation of stochastic process is represented as,

$$\alpha(\boldsymbol{x}, \boldsymbol{\xi}(\theta)) = \bar{\alpha}(\boldsymbol{x}) + \sum_{n=1}^{L} \sqrt{\lambda_n}\, f_n(\boldsymbol{x})\xi_n(\theta). \tag{3}$$

The number of KLE modes used in Eq. (3) is equal to the number of random variables required to approximate the stochastic process [1, 6].

The selection of the number of KLE modes is based on the correlation length of the underlying stochastic process. A strongly correlated (i.e., narrow bandwidth) process with a long correlation length needs fewer expansion terms to capture the uncertainty.

Conversely, if the correlation length is short, i.e., a weakly correlated (broad bandwidth) process demands more expansion terms [1, 2, 14]. Fig. 1 plots the relative partial sum of the eigenvalue contribution for the exponential covariance kernel given in Eq. (4). For the correlation length, $b = 1$ we need only 12 modes to capture 95% of relative contribution as oppose to 35 modes when the correlation length reduced to $b = 0.3$. Note that the number of random variables (i.e., the number of KLE modes) also refers to the dimension of the stochastic process in this paper.

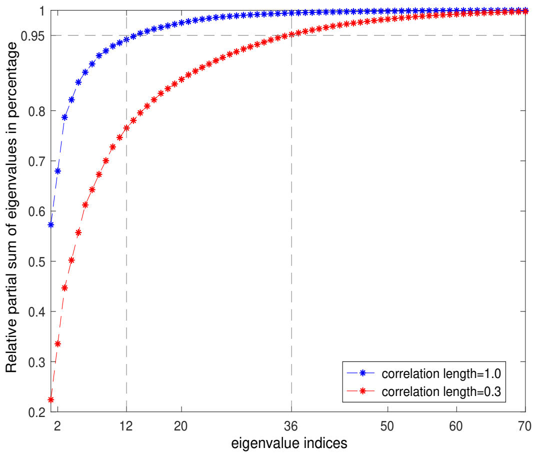

Figure 1: Relative partial sum of eigenvalues for exponential covariance kernel on a unit square domain.

Following the book by Ghanem and Spanos [1], a semi-analytical expression of the KLE is detailed for a Gaussian process. Consider an exponential covariance function for a Gaussian stochastic process defined over a two-dimensional physical domain $\mathscr{D}(x, y)$ over the interval $[-a \;\; a] \times [-a \;\; a]$ [1],

$$C_{\alpha\alpha}(x_1, y_1; x_2, y_2) = \sigma^2 e^{-|x_2-x_1|/b_1 \; -|y_2-y_1|/b_2}, \tag{4}$$

where b_1 and b_2 are the correlation lengths along x and y directions respectively and σ^2 denotes the variance of the underlying stochastic process.

Solving the integral given in Eq. (2) for the covariance kernel given in Eq. (4), the eigenvalues and eigenfunctions are obtained as [1],

$$\lambda_n = \lambda_i^x \otimes \lambda_i^y, \tag{5}$$

$$f_n(x, y) = g_i(x) \otimes h_i(y). \tag{6}$$

where $\otimes$ denotes the tensor product [3].

The two-dimensional eigenfunctions $\{f_n\}_{n=1}^{\infty}$ and eigenvalues $\{\lambda_n\}_{n=1}^{\infty}$ are computed as the tensor product of the one-dimensional eigenfunctions $\{g_i(x), h_i(y)\}$ and eigenvalues $\{\lambda_i^x, \lambda_i^y\}$ respectively [1, 10]. The analytical expressions for the one-dimensional eigenvalues and eigenfunctions for the exponential covariance kernel in Eq. (4) are [1, 10],

$$
\begin{aligned}
\lambda_i^x &= \sigma^2 \frac{2b_1}{1 + b_1^2 {\omega_i}^2}, \\
\lambda_i^y &= \sigma^2 \frac{2b_2}{1 + b_2^2 {\omega_i}^2},
\end{aligned} \tag{7}
$$

$$
g_i(\boldsymbol{x}) = h_i(\boldsymbol{x}) = \begin{cases} \frac{\cos(\omega_i \boldsymbol{x})}{\sqrt{a + \frac{\sin(2\omega_i a)}{2\omega_i}}}, & \text{for } i \text{ odd}, \\ \frac{\sin(\omega_i \boldsymbol{x})}{\sqrt{a - \frac{\sin(2\omega_i a)}{2\omega_i}}}, & \text{for } i \text{ even}, \end{cases}
$$

where ω_i's are the solution of the following transcendental equations given by [1, 10]

$$
\begin{aligned}
\frac{1}{b} - \omega_i \ \tan(\omega_i \ a) = 0, &\quad \text{for } i \text{ odd}, \\
\omega_i + \frac{1}{b} \ \tan(\omega_i \ a) = 0, &\quad \text{for } i \text{ even}.
\end{aligned} \tag{8}
$$

The numerical implementation of KLE can be summarized in the following steps.

- Solve the transcendental equations given in Eq. (8) for ω_i's.
- Substitute ω_i's in Eq. (7) and (8) to compute the one-dimensional eigenvalues and eigenfunctions respectively.
- Using Eq. (5) and (6) compute the two-dimensional eigenvalues $\{\lambda_n\}_{n=1}^{\infty}$ and eigenfunction $\{f_n\}_{n=1}^{\infty}$ respectively and
- using λ_n and f_n in Eq. (3), compute the KLE modes.

Note that the procedure presented above is for a two-dimensional physical domain. However, a similar procedure can be used for the cases with three-dimensional domains. Refer to [15] for the further details on the implementation of KLE with the code snippets.

3.2. Polynomial Chaos Expansion

The KLE is a powerful tool that can be exploited in characterizing a stochastic process with a known covariance function. However, the KLE cannot be used to represent a process with unknown covariance function. In such cases, the PCE can be utilized [1, 2].

A second order stochastic process $\alpha(\boldsymbol{x}, \boldsymbol{\xi}(\theta))$, i.e., a random process with finite variance can be expanded using PCE as [1],

$$\alpha(\boldsymbol{x}, \boldsymbol{\xi}(\theta)) = \sum_{j=0}^{\infty} \widehat{\alpha_j}(\boldsymbol{x})\ \Psi_j(\boldsymbol{\xi}(\theta)), \tag{9}$$

where $\Psi_j(\boldsymbol{\xi}(\theta))$ is a set of multidimensional orthogonal polynomials of the order p_α of L independent random variables $\boldsymbol{\xi} = \{\xi_1, \xi_2, \ldots, \xi_L\}$ and $\widehat{\alpha}_j(\boldsymbol{x})$ represents the unknown deterministic coefficients. Although the PCE is mainly founded on the independence assumption of the random variables, for some applications, there may exist significant dependence among the random variables. In such cases the Rosenblatt transformation [16], commonly employed for mapping dependent to independent variables can be exploited (see the following articles for further details [17, 18]).

The polynomials $\Psi_j(\boldsymbol{\xi})$ (note that the argument θ of $\boldsymbol{\xi}$ being dropped hereafter for notational simplicity) are orthogonal in a statistical sense, i.e., their inner product with respect to probability measure (PDF) is zero. Therefore, $\langle \Psi_i(\boldsymbol{\xi}), \Psi_j(\boldsymbol{\xi}) \rangle = 0$ for $i \neq j$ and for $i = j$, it can be computed as the statistical average as shown below [1],

$$\langle \Psi_i(\boldsymbol{\xi}), \Psi_j(\boldsymbol{\xi}) \rangle = \int \Psi_i(\boldsymbol{\xi}) \Psi_j(\boldsymbol{\xi})\ \mathrm{p}(\boldsymbol{\xi}) d\boldsymbol{\xi}, \tag{10}$$

where $\mathrm{p}(.)$ is the joint probability density function (PDF) of $\boldsymbol{\xi}$, obtained from the marginal PDF of $\mathrm{p_i}(\boldsymbol{\xi}_\mathrm{i})$ as: $\mathrm{p}(\boldsymbol{\xi}) = \prod_{i=1}^{L} \mathrm{p_i}(\xi_\mathrm{i})$.

Theoretically, we need an infinite number of PCE terms to accurately approximate the output process. However, for the numerical approximation, the truncated PCE with P_α expansion terms can be used to obtain a sufficiently accurate representation of the output. Note that the higher order moments in the probability density function can be the measure of accuracy. The truncated expansion given in Eq. (11) is shown to converge in the mean square sense [1]:

$$\alpha(\boldsymbol{x}, \boldsymbol{\xi}) = \sum_{j=0}^{P_\alpha} \widehat{\alpha_j}(\boldsymbol{x})\ \Psi_j(\boldsymbol{\xi}), \tag{11}$$

where the P_α is a function of the order of polynomials p_α and the number of random variables L as given below [1],

$$P_\alpha = \frac{(L + p_\alpha)!}{L!\, p_\alpha!} - 1. \tag{12}$$

The high-frequency random fluctuations in the stochastic process demand a large number of random variables and the strong nonlinear dependences of the solution process on the random system parameters necessitate the higher-order polynomials p_α in the PCE representation [6, 10, 19]. Note that the order of polynomials can differ in the input and output PC expansions but the number of random variables L should be the same [1, 2, 6]. In general, for better convergence, the output (solution) process is represented using higher order expansion than the inputs random process [2, 6]. This is because the uncertainty in the output process may amplify as the input uncertainty undergoes nonlinear transformation through the model. This demands higher order expansion terms to capture the nonlinear and non-Gaussian effects. For further details on the convergence studies refer to the following articles [6, 10, 19]. The implementational details and the associated code snippets of the PCE for a non-Gaussian stochastic process are presented in [15].

4. Spectral Stochastic Finite Element Methods

In this section, we briefly describe the formulations of the intrusive and non-intrusive spectral stochastic finite element methods. For an elementary exposition of both the methodologies, we consider a linear elliptic stochastic PDE on a spatial domain $\mathscr{D}$ with the known boundary condition on $\partial\mathscr{D}$. Consider a two-dimensional steady-state flow through random media with a spatially varying non-Gaussian diffusion coefficient c_d. The flow is modeled by a two-dimensional stochastic diffusion equation. This leads to a Poisson problem defined by a linear elliptic stochastic PDE as defined below:

$$-\nabla \cdot \big(\, c_d(\boldsymbol{x},\theta)\ \nabla u(\boldsymbol{x},\theta)\,\big) = F(\boldsymbol{x}), \qquad \mathscr{D}\times\Omega, \tag{13}$$

$$u(\boldsymbol{x},\theta) = 0, \qquad \partial\mathscr{D}\times\Omega, \tag{14}$$

where ∇ denotes the gradient which represents the differential operator with respect to the spatial variables $\boldsymbol{x}$, u is the solution process, θ is an element in the sample space Ω defined by the probability space $(\Omega, \mathscr{F}, \mathscr{P})$ [8]. For the sake convenience, $F(\boldsymbol{x})$ is modeled as a deterministic source term. However, the methodology presented herein can be easily extended to stochastic source function $F(\boldsymbol{x},\theta)$.

To solve a partial differential equation, for instance Eq. (13), using finite element methods the PDE needs to be expressed in the variational form [2, 20]. To do so, first, the Eq. (13) is multiplied by a test function v and performing integration by parts we obtain [20]:

$$\int_{\mathscr{D}} c_d(\boldsymbol{x},\theta)\ \nabla u\ .\ \nabla v\ d\boldsymbol{x} = \int_{\mathscr{D}} Fv\ d\boldsymbol{x} + \int_{\partial\mathscr{D}} \frac{\partial u}{\partial n} v\ ds, \tag{15}$$

where u is called the trial function, v is called the test function and $\partial u/\partial n$ is the derivative of u in the outward normal direction at the boundary. Note that v vanishes where u is known, i.e., on $\partial\mathscr{D}$. Therefore, the resulting variation form can be further simplified as:

$$\int_{\mathscr{D}} c_d(\boldsymbol{x},\theta)\ \nabla u\ .\ \nabla v\ d\boldsymbol{x} = \int_{\mathscr{D}} Fv\ d\boldsymbol{x}. \tag{16}$$

Consider the diffusion coefficient $c_d(\boldsymbol{x},\theta)$, being modeled as a lognormal stochastic process $l(\boldsymbol{x},\theta)$ obtained by an exponential of a Gaussian process $g(\boldsymbol{x},\theta)$ having the mean $g_0(\boldsymbol{x})$, variance σ^2 and exponential covariance function $C_{gg}(x,y)$ in Eq. (4). For illustration, $c_d(\boldsymbol{x},\theta)$ is characterized by using two random variables with second order PCE. Therefore, PCE of the lognormal stochastic process can be expressed as [1].

$$\begin{aligned} c_d(\boldsymbol{x},\theta) = l_0(\boldsymbol{x})\Big(&g_0 + \xi_1(\theta)g_1(\boldsymbol{x}) + \xi_2(\theta)g_2(\boldsymbol{x}) + \big(\xi_1^2(\theta)-1\big)\frac{g_1^2(\boldsymbol{x})}{2} \\ &+ \big(\xi_1(\theta)\xi_2(\theta)\big)g_1(\boldsymbol{x})g_2(\boldsymbol{x}) + \big(\xi_2^2(\theta)-1\big)\frac{g_2^2(\boldsymbol{x})}{2}\Big), \end{aligned} \tag{17}$$

where $\{g_i(\boldsymbol{x})\}_{i=1}^2$ are the scaled eigenfunctions of the covariance kernel $C_{gg}(x,y)$ and $l_0(\boldsymbol{x}) = \exp\left[g_0(\boldsymbol{x}) + \frac{1}{2}g_1^2(\boldsymbol{x}) + \frac{1}{2}g_2^2(\boldsymbol{x})\right]$ is the mean of the lognormal process.

Consequently, Eq. (16) can be re-written as

$$\begin{aligned} &\int_{\mathscr{D}} l_0(\boldsymbol{x})\Big(g_0 + \xi_1(\theta)g_1(\boldsymbol{x}) + \xi_2(\theta)g_2(\boldsymbol{x}) + \big(\xi_1^2(\theta)-1\big)\frac{g_1^2(\boldsymbol{x})}{2} \\ &+ \big(\xi_1(\theta)\xi_2(\theta)\big)g_1(\boldsymbol{x})g_2(\boldsymbol{x}) + \big(\xi_2^2(\theta)-1\big)\frac{g_2^2(\boldsymbol{x})}{2}\Big)\ \nabla u\ .\ \nabla v\ d\boldsymbol{x} = \int_{\mathscr{D}} Fv\ d\boldsymbol{x}. \end{aligned} \tag{18}$$

The finite element discretization with N nodes in the spatial domain leads to a system of linear equations with random coefficients θ denotes stochasticity [1]

$$\mathbf{A}(\theta)\mathbf{u}(\theta) = \mathbf{f}, \tag{19}$$

where $\mathbf{A}(\theta)$ is the random or stochastic system matrix, $\mathbf{u}(\theta)$ is the stochastic response vector and $\mathbf{f}$ is the deterministic source vector. In the deterministic setting the above system of equations can be solved for the mean values of the stochastic parameters. For example, the Gaussian process characterized by using $g_0 = 0$, $\sigma = 0.3$ and the exponential covariance function given in Eq. (4) with $b_1 = b_2 = b = 1$. Therefore, c_d can be expanded as [1] (for simplification using $L \to \infty$, refer to [6, 13] for details).

$$c_d(\boldsymbol{x},\theta) = l_0 = \exp\left[0 + \frac{1}{2}0.3^2\right] = 1.05. \tag{20}$$

Consider a unit square domain discretized using unstructured finite element mesh with 600 nodes and 1200 elements as shown in Fig. 2. The numerical simulations are performed using the unit source term. The solution field for the mean value of stochastic parameter is shown in Fig. 2. The corresponding FEniCS based python code snippet is shown in Listing [15].

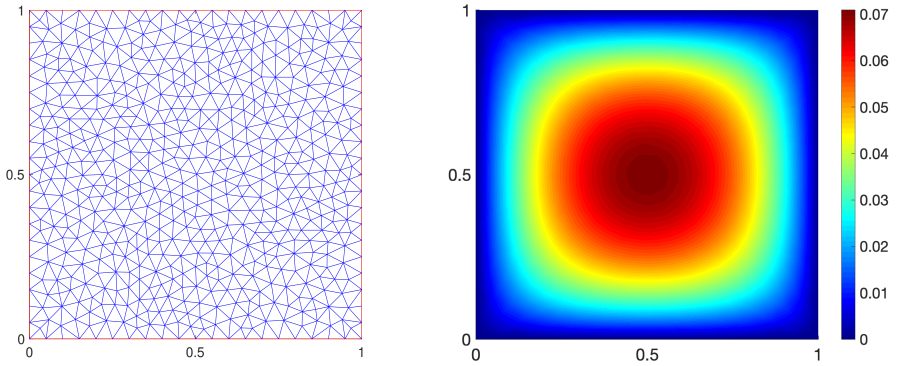

Figure 2: Finite element mesh and the solution field at the mean value of stochastic system parameters.

Assuming that the input data contains sufficient statistical information, we can represent the random system matrix and the stochastic solution process using the PCE as [1]:

$$\mathbf{A}(\theta) = \sum_{i=0}^{P_{\text{A}}} \hat{\mathbf{A}}_i \ \Psi_i(\boldsymbol{\xi}), \qquad \tilde{\mathbf{u}}(\theta) = \sum_{j=0}^{P_u} \hat{\mathbf{u}}_j \ \Psi_j(\boldsymbol{\xi}), \tag{21}$$

where the $\hat{\mathbf{A}}_i$'s are the PCE coefficients of the random system matrix, $\hat{\mathbf{u}}_j$'s are the PCE coefficients of the solution process and Ψ's are the multidimensional polynomials obtained as a function of L random variables $\boldsymbol{\xi} = \{\xi_1, \xi_2, \ldots, \xi_L\}$. P_{A} and P_u are the numbers of PCE terms required to characterize the stochastic system matrix and the stochastic response vector respectively [1]. The number of PCE terms P_α is a function of the order of polynomials p_α and the number of random variables L, which can be computed using Eq. (12) [1] and the subscript α represents A or u. In this study the input, namely the diffusion coefficient of the Poisson PDE is modeled by a lognormal process obtained by the exponential of a Gaussian process. For further details on the PCE of the lognormal process and KLE of the underlying Gaussian process refer to [15].

In the SSFEM approaches, the primary goal is to compute the PCE coefficients of the solution process $\hat{\mathbf{u}}_j$. Based on the approach utilized to compute $\hat{\mathbf{u}}_j$, the solution methodologies for SSFEM are generally classified into (a) intrusive (sampling-free) approach also referred to as the stochastic Galerkin (SG) approach [1, 5, 21–23] and (b) non-intrusive (sampling-based) approaches such as non-intrusive spectral projection (NISP) [2, 24–27] or stochastic collocation (SC) approach [28–31]. Note that, in this paper, we have focused on the SG and the NISP approach based on the sparse grid quadrature.

In the SG (intrusive) approach, the PCE for the random system parameters and the solution process, for example as given in Eq. (21) are substituted into the governing equation (for example Eq. (19)). In SG framework the strategy is to project the residual onto a finite-dimensional space spanned by appropriate orthogonal basis functions with respect to probability density $\mathrm{p}(\boldsymbol{\xi})$. Thus we seek the approximate solution as [3]

$$\begin{aligned} 0 &= \Big\langle \mathbf{A}(\theta)\mathbf{u}(\theta) - \mathbf{f}\ , \Psi_k(\boldsymbol{\xi}) \Big\rangle, \ \ k = 0, 1, \ldots, P_u, \qquad (22) \\ &= \int_{\mathfrak{D}} \Big[\sum_{i=0}^{P_{\mathrm{A}}} \hat{\mathbf{A}}_i \ \Psi_i(\boldsymbol{\xi}) \sum_{j=0}^{P_u} \hat{\mathbf{u}}_j \ \Psi_j(\boldsymbol{\xi}) - \mathbf{f} \Big] \Psi_k(\boldsymbol{\xi}) \mathrm{p}(\boldsymbol{\xi}) d\boldsymbol{\xi}, \ \ k = 0, 1, \ldots, P_u. \end{aligned}$$

Eq. (22) can also be approximately solved using the quadrature rule with quadrature points q_n and weights w_n as

$$\sum_{n=1}^{N} \Big[\sum_{i=0}^{P_{\mathrm{A}}} \hat{\mathbf{A}}_i \ \Psi_i(q_n) \sum_{j=0}^{P_u} \hat{\mathbf{u}}_j \ \Psi_j(q_n) - \mathbf{f} \Big] \Psi_k(q_n) \ \mathrm{p}(q_n) w_n = 0, \ \ k = 0, 1, \ldots, P_u. \qquad (23)$$

In the NISP framework (also referred as pseudo-spectral approach) the PCE coefficients of solution process are obtained by performing Galerkin projection on the PCE of the solution process given in Eq. (21). The orthogonality properties of PCE basis functions are utilized to compute the expansion coefficients of the solution process [1, 2] as,

$$\hat{\mathbf{u}}_k = \frac{1}{\big\langle \Psi_k^2(\boldsymbol{\xi}) \big\rangle} \sum_{n=1}^{N} \mathbf{u}(q_n) \Psi_k(q_n) \ \mathrm{p}(q_n) w_n, \ \ k = 0, 1, \ldots, P_u. \qquad (24)$$

The NISP approach is simple to implement; however, for the cases with a large number of random variables (which is the primary focus of this paper) the number of quadrature points required to solve the above integral increases significantly [3, 25, 26, 32].

In the framework of collocation approach, M collocation points are generated from the parameter space, and then enforce

$$\mathbf{u}(q_m) = \tilde{\mathbf{u}}(q_m), \ \ m = 1, \ldots, M \qquad (25)$$

to compute $\hat{\mathbf{u}}_j$. For a general basis function ϕ_k, performing above steps yields the following system [3]

$$\begin{bmatrix} \phi_0(q_1) & \ldots & \phi_{P_u}(q_1) \\ \vdots & \ddots & \vdots \\ \phi_0(q_M) & \ldots & \phi_{P_u}(q_M) \end{bmatrix} \begin{Bmatrix} \hat{\mathbf{u}}_0 \\ \vdots \\ \hat{\mathbf{u}}_{P_u} \end{Bmatrix} = \begin{Bmatrix} \mathbf{u}(q_1) \\ \vdots \\ \mathbf{u}(q_M) \end{Bmatrix}. \qquad (26)$$

Similar to the NISP approach, the SC approach is also straightforward to formulate. However, for a large number of random variables, the number and choice of collocation points as well as the choice of basis function become critical [3]. Note that the computational efforts required for the collocation approach is essentially equivalent to that of NISP approach when the number of collocation points M equal to quadrature points N.

The accuracy of intrusive (SG) approach is optimal in L^2 sense. Therefore, the SG approach can be computationally efficient over NISP or SC methods [3]. However, SG approach involves substantial modification of a deterministic simulation code [3]. Note that, later in this paper, we demonstrate that the requirement of modification of a deterministic simulation code due to intrusive adjustments in the SG approach can be alleviated by employing general purpose deterministic FEM packages (e.g. FEniCS [33]).

On the other hand, the SC approach does not encounter orthogonality and dependence requirement issues observed in SG approach. The SC approach can be applied to general parameter distribution (see [3, 26] for more details). However, the SC is a sampling-based approach and the number of required samples grow with the order and the number of random variables. Similarly, in the NISP approach the number of quadrature points grow with the order and number of random variables. Note that, the SC approach is not implemented in this paper, instead we use NISP approach based on the sparse grid quadrature. For more details on the SC approach and its comparison with NISP and SG approaches the author refer to the numerous articles available in the literature [3, 26, 28–31].

The rest of the paper is organized as follows: Sec. 4.1 outlines the formulation of intrusive SSFEM (SG). This is followed by the formulation of non-intrusive approach, specifically, the NISP with Smolyak sparse grid quadrature in Sec. 4.2. In Sec. 5 the performance comparison of the intrusive SSFEM with the non-intrusive SSFEM is conducted using a Poisson equation having a non-Gaussian diffusion coefficient, modeled by a lognormal process.

4.1. Intrusive SSFEM

In the intrusive SSFEM, the PCE of the system matrix with random coefficients $\mathbf{A}(\theta)$ and the solution process $\mathbf{u}(\theta)$, presented in Eq. (21) are directly substituted into the finite element discretization of stochastic PDE given in Eq. (19) leading to [1]

$$\epsilon = \sum_{i=0}^{P_{\mathrm{A}}} \hat{\mathbf{A}}_i \Psi_i(\boldsymbol{\xi}) \sum_{j=0}^{P_u} \hat{\mathbf{u}}_j \Psi_j(\boldsymbol{\xi}) - \mathbf{f} \neq 0, \tag{27}$$

where ϵ is the random residual.

Performing Galerkin projection, i.e., multiplying both sides of the Eq. (27) by $\Psi_k(\boldsymbol{\xi})$ with $k = 0, .., P_u$ and taking expectation both sides results in the following system of coupled equations [1],

$$\langle \epsilon, \Psi_k(\boldsymbol{\xi}) \rangle = 0 \ , \quad k = 0, 1, \ldots, P_u \tag{28}$$

$$\sum_{j=0}^{P_u} \sum_{i=0}^{P_{\mathrm{A}}} \langle \Psi_i(\boldsymbol{\xi}) \Psi_j(\boldsymbol{\xi}) \Psi_k(\boldsymbol{\xi}) \rangle \, \hat{\mathbf{A}}_i \hat{\mathbf{u}}_j = \langle \mathbf{f} \ \Psi_k(\boldsymbol{\xi}) \rangle \ , \ \ k = 0, 1, \ldots, P_u. \tag{29}$$

For notational convenience, we rewrite Eq. (29) using $\langle \Psi_i(\boldsymbol{\xi}) \Psi_j(\boldsymbol{\xi}) \Psi_k(\boldsymbol{\xi}) \rangle = \boldsymbol{\mathcal{C}}_{ijk}$ and $\langle \mathbf{f} \ \Psi_k(\boldsymbol{\xi}) \rangle = f_k$ leading to

$$\sum_{j=0}^{P_u} \sum_{i=0}^{P_{\mathrm{A}}} \boldsymbol{\mathcal{C}}_{ijk} \hat{\mathbf{A}}_i \ \hat{\mathbf{u}}_k = f_k \ , \quad k = 0, 1, \ldots, P_u. \tag{30}$$

For concise representation, the following notation is used,

$$A_{jk} = \sum_{i=0}^{P_{\mathrm{A}}} \boldsymbol{\mathcal{C}}_{ijk} \hat{\mathbf{A}}_i, \tag{31}$$

Thus, Eq. (30) can be further simplified as,

$$\sum_{j=0}^{P_u} A_{jk} \ \hat{\mathbf{u}}_k = f_k \ , \quad k = 0, 1, \ldots, P_u. \tag{32}$$

Equivalently Eq. (32) can be written as,

$$[\mathscr{A}]\{\mathscr{U}\} = \{\mathscr{F}\}, \tag{33}$$

where $\mathscr{A}$ in Eq. (33) is the system matrix, $\mathscr{U}$ is the vector of PCE coefficients of the solution process and $\mathscr{F}$ is the corresponding right hand side vector arising in the setting of intrusive SSFEM. Eq. (33) can be expanded as

$$\begin{bmatrix} A_{0,0} & A_{0,1} & \ldots & A_{0,P_u} \\ A_{1,0} & A_{1,1} & \ldots & A_{1,P_u} \\ \vdots & \vdots & \ddots & \vdots \\ A_{P_u,0} & A_{P_u,1} & \ldots & A_{P_u,P_u} \end{bmatrix} \begin{Bmatrix} \hat{\mathbf{u}}_0 \\ \hat{\mathbf{u}}_1 \\ \vdots \\ \hat{\mathbf{u}}_{P_u} \end{Bmatrix} = \begin{Bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{P_u} \end{Bmatrix}. \tag{34}$$

The solution of the above coupled linear system yields the PCE coefficients of the solution process. These PCE coefficients can be used to obtain the mean, variance, higher order statistics and PDF of the solution process [1].

The size of the system matrix $\mathscr{A}$ is $(N \times P_u, N \times P_u)$, where N is the number of degree-of-freedom related to the finite element mesh resolution and P_u is the number of PCE terms used in the representation of the solution process. Note that, P_u is a function of stochastic dimension L and order p_u of the PCE. From Eq. (31) and (32) it can be noted that, each block of the system matrix $\mathscr{A}$ is denoted by A_{jk} (a sub matrix of size $(N \times N)$) and it can be computed from the set of deterministic finite element matrices $\hat{\mathbf{A}}_i$. In this paper, the system matrix assembly procedure, i.e., implementation of Eq. (31) is performed by employing deterministic finite element assembly routines imported from the *FEniCS* general purpose FEM package [34, 35].

The intrusive system matrix $\mathscr{A}$ has the two-levels of sparsity (block-sparse structure), for example, see in Fig. 3 and 4 [5, 10, 21]. The first-level of sparsity is coming from the deterministic finite element matrices, $\hat{\mathbf{A}}_i$, which are sparse and the second-level of sparsity is due to the stochastic aspects of the problem arising from a few non-zero $\boldsymbol{\mathcal{C}}_{ijk}$ terms. The matrices in Fig. 3 and 4 are assembled for the stochastic diffusion equation with a lognormal diffusion coefficient represented using the underlying Gaussian stochastic process (refer to [15] for further details).

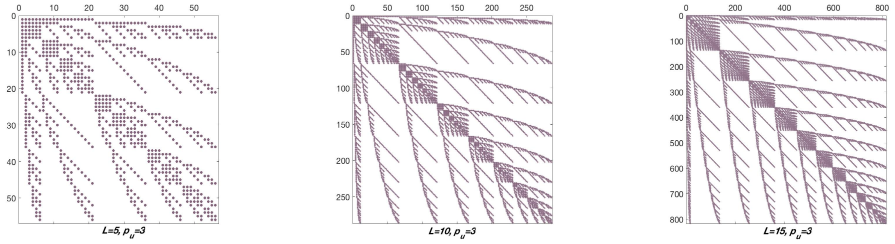

Figure 3: SSFEM system matrices for the fixed $N = 30$, $p_{\text{A}} = 2, p_u = 3$ with $L = 5$, $L = 10$ and $L = 15$.

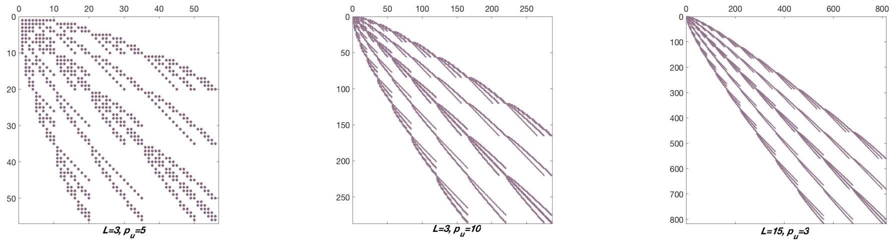

Figure 4: SSFEM system matrices for the fixed $N = 30$, $p_{\text{A}} = 2, L = 3$ with $p_u = 5$, $p_u = 10$ and $p_u = 15$.

The sparsity structure of each block $\hat{\mathbf{A}}_i$ in the intrusive system matrix depends on finite element discretization. Therefore, its sparsity pattern is not influenced by stochastic expansion terms such as L or p_u. However, the block-sparsity structure of the system matrix changes with (a) the number of random variables as shown in Fig. 3 (for fixed order of expansion and varying number of random variables) and (b) the order of expansion as shown in Fig. 4 (for fixed number of random variables and varying order of expansion). The orders and dimensions of the input and output stochastic processes influence the number of nonzero $\boldsymbol{\mathcal{C}}_{ijk}$ terms, and thus the number of non-zero blocks in $\mathscr{A}$. Consequently, the size of the intrusive system matrix and vector grows rapidly with the number of random variables and order of expansion. This is the primary challenge of the intrusive SSFEM with a large number of random variables and higher order expansions. Another difficulty of this approach is the need for intrusive adjustments in the associated deterministic PDE solver. Thus the existing deterministic FEM package cannot be readily utilized; this makes the intrusive approach more complicated and demanding. To overcome this challenge, in this paper the stochastic system assembly procedure is simplified by directly employing FEniCS deterministic assembly routines [34].

The implementational steps of intrusive SSFEM can be summarized as follows:

- Calculate and store the non-zero $\boldsymbol{\mathcal{C}}_{ijk}$ terms, i.e., moments of multidimensional polynomials and their respective indices i, j and k. This requires the computation of multidimensional inner products which can be obtained using one-dimensional moments and multi-indices [2]. The routines adopted from UQ Toolkit [36] are employed to compute the $\boldsymbol{\mathcal{C}}_{ijk}$ terms.

- The intrusive system matrix $\mathscr{A}$ is assembled using the deterministic system matrices $\hat{\mathbf{A}}_i$. For each of the input PCE index i in Eq. (31), the deterministic finite element assembly routines from an existing FEM package can be used to assemble $\hat{\mathbf{A}}_i$. for details).

- For the PCE indices i, j and k in Eq. (31), those matches with the pre-calculated non-zero i, j and k indices of $\boldsymbol{\mathcal{C}}_{ijk}$, multiply the i^{th} assembled deterministic matrix $\hat{\mathbf{A}}_i$ with the respective C_{ijk} and store it at the respective $[j, k]^{th}$ position of stochastic matrix $\mathscr{A}$.

Similar steps can be utilized to assemble the right hand side vector $\mathscr{F}$. Further details on the FEniCS-based implementation of intrusive SSFEM along with the code snippets for each major step in the process are outlined in [15].

4.2. Non-Intrusive SSFEM Approach

In this section, the PCE based non-intrusive spectral projection (NISP) approach is employed [2, 24, 25]. In the NISP approach, the Galerkin projection is directly performed on the PCE of the solution process given in Eq. (21). The orthogonality properties of PCE basis polynomials are utilized to compute the expansion coefficients of the solution process [1, 2] as,

$$\begin{aligned}\hat{\mathbf{u}}_k &= \frac{\langle \mathbf{u}(\theta)\Psi_k(\boldsymbol{\xi})\rangle}{\langle \Psi_k^2(\boldsymbol{\xi})\rangle} \\ &= \frac{1}{\langle \Psi_k^2(\boldsymbol{\xi})\rangle}\int_\Omega \mathbf{u}(\theta)\Psi_k(\boldsymbol{\xi})\mathrm{p}(\boldsymbol{\xi})d\boldsymbol{\xi}.\end{aligned} \tag{35}$$

The denominator in Eq. (35) can be evaluated analytically beforehand Thus the major computational effort lies in the evaluation of the multidimensional integral in the numerator. In the past, random sampling, tensor product quadrature or sparse grid quadrature based approaches were employed to solve the above integral [2, 24, 25, 30, 31]. The number of sampling points for random sampling and tensor product quadrature based approaches drastically increases with the increasing number of random variables and thereby renders these approaches inefficient [26, 30, 31]. In such cases, the sparse grid quadrature (for example, Eq. (36)) can be used to reduce the number of the solution evaluation points while maintaining the same level of accuracy [2, 25, 30–32].

$$\hat{\mathbf{u}}_k = \frac{1}{\langle \Psi_k^2(\boldsymbol{\xi})\rangle}\sum_{i=1}^{N_s}\omega_{p_i}\mathbf{u}(\boldsymbol{\xi}_i)\Psi_k(\boldsymbol{\xi}_i), \tag{36}$$

where N_s is the number of quadrature point with nodes $\boldsymbol{\xi}_i$ and corresponding weights ω_{p_i} (which incorporates PDF $\mathrm{p}(\boldsymbol{\xi}_i)$). In the current implementation, the Smolyak sparse grid quadrature scheme is employed to evaluate the integral in the numerator of Eq. (35) [30–32, 37]. For the sake of completeness we provide a minimal description of sparse grid techniques. Numerous articles are available in the literature on the formulation of Smolyak sparse grid and its applications to SSFEM [3, 26, 30–32, 37].

To illustrate the implementation of sparse grid quadrature based NISP approach, consider the finite element discretization of a stochastic PDE given in Eq. (19). Using $\{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_{n_s}\}$ the following deterministic system is solved at N_s sample points (in this case, sparse quadrature points) using an existing deterministic solver:

$$\mathbf{A}(\boldsymbol{\xi}_i(\theta))\mathbf{u}(\theta) = \mathbf{f}, \;\; i = 1, \ldots, N_s. \tag{37}$$

The evaluated $\mathbf{u}(\theta)$ at each sample point $\boldsymbol{\xi}_i$ is used in Eq. (35) to calculate the PCE coefficients of the solution process.

The sparse grid quadrature rule to integrate a multidimensional function $\mathcal{F} = \mathbf{u}(\theta)\Psi_k(\boldsymbol{\xi})$ in the numerator of Eq. (35) can be constructed using the univariate quadrature rule $\mathcal{Q}_l^{(1)}\mathcal{F}$ as follows [3, 32],

$$\mathcal{Q}_l^{(1)}\,\mathcal{F} = \sum_{q=1}^{N_s}\mathcal{F}(r_l^q)\;w_{p_l}^{\;q}, \tag{38}$$

where the subscript l of $\mathcal{Q}$ is the level of quadrature and the superscript of $\mathcal{Q}$ denotes the dimension d (in this case $d = 1$). N_s is the number of quadrature points in the sparse grid and r_l^q and wp_l^q are the nodes and respective weights for the l^{th} level sparse grid quadrature. The difference relation between two levels can be defined as [3]

$$\Delta_l^{(1)}\mathcal{F} = \left(\mathcal{Q}_l^{(1)} - \mathcal{Q}_{l-1}^{(1)}\right)\mathcal{F} \tag{39}$$

where $\mathcal{Q}_0^{(1)}\mathcal{F} = 0$ and $\Delta_l^{(1)}\mathcal{F}$ is also a quadrature formula with the nodes same as $\mathcal{Q}_l^{(1)}\mathcal{F}$ and weights are difference between those at the levels l and $l-1$. The set of nodal point for one-dimensional quadrature $\mathcal{Q}_l^{(1)}\mathcal{F}$ is denote by

$$\Theta_l^{(1)} = \{r_l^1, \ldots, r_l^{N_s}\}. \tag{40}$$

For illustration of difference relation in Eq. (39), consider an example from [3]. The quadrature rule $\mathcal{Q}_1^{(1)}$ for $d = 1$ and $l = 1$ with the nodes $\Theta_1^{(1)} = \{0, 1\}$ and weights = $\{0.5, 0.5\}$. Similarly the quadrature rule $\mathcal{Q}_2^{(1)}$ for $l = 2$ with the nodes $\Theta_2^{(1)} = \{0, 0.5, 1\}$ and weights = $\{0.25, 0.5, 0.25\}$. Then the nodes for the difference rule $\Delta_2^{(1)}$ are same as $\Theta_2^{(1)} = \{0, 0.5, 1\}$, but the weights = $\{-0.25, 0.5, -0.25\}$ are obtained from the difference $\mathcal{Q}_2^{(1)} - \mathcal{Q}_1^{(1)}$.

The sparse grid formula for d-dimensional integral at level l can be written using the difference relationship as [3]

$$\mathcal{Q}_l^{(d)}\,\mathcal{F} = \sum_{|l'| \leq l+d-1} \left(\Delta_{l_1}^{(1)} \otimes \cdots \otimes \Delta_{l_d}^{(1)}\right)\mathcal{F}. \tag{41}$$

where $l' = (l_1 \ldots l_d)$ with $|l'| = \sum_{i=1}^{d} l_i$. The nodal set Θ for the sparse grid can be obtained using one-dimensional sparse nodal set $\Theta_l^{(1)}$ as [3]

$$\Theta_l^{(d)} = \bigcup_{|l'| \leq l+d-1} \left(\Theta_{l_1}^{(1)} \times \cdots \times \Theta_{l_d}^{(1)}\right), \tag{42}$$

where $\bigcup$ denotes union of subsets.

For example, consider $\Theta_1^{(1)}$, $\Theta_2^{(1)}$ and $\Theta_3^{(1)}$ are the nodal sets at level $l = 1, 2$ and 3 and dimension $d = 1$. Then the sparse grid nodal set for $d = 2$ and $l = 3$ can be written as [3]

$$
\begin{aligned}
\Theta_{l=3}^{(d=2)} &= \bigcup_{|l'|\leq l+d-1} \left(\Theta_{l_1}^{(1)} \times \Theta_{l_2}^{(1)}\right) \\
&= \left(\Theta_{1}^{(1)} \times \Theta_{1}^{(1)}\right) \qquad (l_1 = 1, l_2 = 1) \\
&\cup \left(\Theta_{1}^{(1)} \times \Theta_{2}^{(1)}\right) \cup \left(\Theta_{2}^{(1)} \times \Theta_{1}^{(1)}\right) \\
&\cup \left(\Theta_{1}^{(1)} \times \Theta_{3}^{(1)}\right) \cup \left(\Theta_{2}^{(1)} \times \Theta_{2}^{(1)}\right) \cup \left(\Theta_{3}^{(1)} \times \Theta_{1}^{(1)}\right).
\end{aligned} \tag{43}
$$

Equivalent formulations of the Smolyak sparse grid using slightly different approach and also different notations can be found in the following articles [2, 3, 32].

For the implementation of the multidimensional sparse grid shown in Eq. (41) the growth rule in one-dimensional quadrature given in Eq. (38) must be defined. The fully nested Clenshaw-Curtis abscissas [3, 32] or the weakly-nested Gaussian abscissas [3, 32] can be used. In the current implementation, we have used the Gaussian abscissas [3, 32]. For illustration, Smolyak sparse grid with the Gaussian abscissas for the two-dimensional case with $l = 3$ and $l = 6$ are showed in Fig. 5 (see Sec. 6 for more details on sparse grid construction with Gaussian abscissas). For a moderate dimension and the moderate level of quadrature, the sparse grid can drastically reduce the number of quadrature points compared to the tensor product grid [26, 30, 31]. However, for high dimensional cases, the number of quadrature points substantially increases. For example, with $d = 20$ the number of quadrature points increase from 841 for $l = 3$ to 1014809 for $l = 6$. For additional details on the types of growth rules and their performance comparisons for non-intrusive SSFEM, refer the following articles and reference therein [3, 30–32].

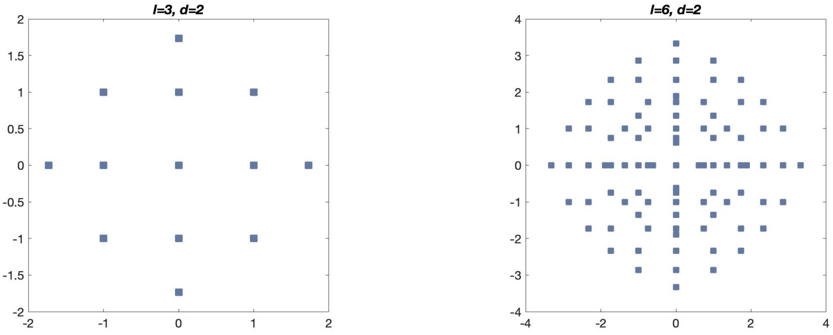

Figure 5: Number of quadrature points for a fixed $d = 2$ with $l = 3$ and $l = 6$.

The main advantage of the non-intrusive approach is that we do not have to modify the underlying deterministic PDE simulation code to use this method. Instead, the repetitive deterministic system solves are required at each sample point. This can be performed by directly employing any existing deterministic finite element solver as a black-box. Hence, the implementation of non-intrusive approaches demands a mush less coding effort than the intrusive approach. This feature makes the non-intrusive approaches more attractive option over the intrusive approach [26, 27, 38–40]. In this paper, the finite element solver from FEniCS FEM package [34, 35] is employed to compute solution at the required samples. Further details on the implementation of NISP approach with the code snippets are outlined in Sec. 6.

5. Comparison of Intrusive and Non-Intrusive SSFEM

To the author's best knowledge, there is no unanimous agreement in the UQ research community on which approach is superior: intrusive or non-intrusive method. Many articles suggest that the intrusive SSFEM method is more efficient concerning the accuracy and computational cost (albeit with substantial additional coding effort) as compared to the non-intrusive method for the following reasons [14, 38, 39, 41, 42]:

- To achieve the same level of accuracy, the computational cost of repeated system solution (e.g., at numerous quadrature points) of non-intrusive approach is considerably higher than that of the intrusive system arising from the stochastic Galerkin approach [38].
- The error due to the finite representation in intrusive approach is minimum leading to an optimal accuracy [1]. On the other hand, the accuracy of the non-intrusive approach highly depends on the choice of the sampling (quadrature, collocation) points [2, 14, 32, 40]. For the large-scale applications where each deterministic simulation is computationally demanding the sampling-based non-intrusive approaches are computationally costly [38]. In such cases, the intrusive approach can be advantageous if an efficient intrusive system solver is employed [38].

In this section, we implement both intrusive (SG) and non-intrusive (NISP) SSFEM approaches to compare their performances. The focus is given to compare the accuracies of these approaches for non-Gaussian random process modeling the stochastic system parameters with increasing number of random variables [10, 13]. In the numerous assessments performed in the past, attention is given to the uniform or Gaussian random variables [27, 38, 40]. However, comparative studies for non-Gaussian random processes modeling the stochastic system parameters are not widely available in the literature. Moreover, in this investigation, we conduct a detailed error analysis of the individual PCE coefficients along with the mean and variance of solution process to gain insights into the convergence behavior of both intrusive and non-intrusive approaches. This was lacking in the previous articles related to such assessments [27, 38, 39]. Note that, the adaptive PCE approaches reported in the literature [31, 43–45] to lower the increased computational cost of the SSFEM approaches in high-dimensional stochastic spaces, are not considered in this paper. Note that, such adaptations can be exploited in both intrusive and non-intrusive setting [31, 43–45]. Therefore, it may provide gains in both the approaches. But they are not pursued here although such studies will be worthy of future investigations.

Consider a two-dimensional steady-state flow through random media, modeled by a stochastic diffusion equation. This leads to a Poisson equation defined by a linear elliptic stochastic PDE as given in Eq. (13). The diffusion coefficient $c_d(\boldsymbol{x}, \theta)$ is modeled as a lognormal stochastic process obtained by an exponential of a Gaussian process having the mean $g(\boldsymbol{x}) = 0$, standard deviation $\sigma = 0.4$ and the exponential covariance function $C_{\alpha\alpha}$ (given in Eq. (4)) with correlation lengths $b_1 = b_2 = 1.0$. The lognormal process representation ensures that the diffusion coefficient c_d remains positive over the entire domain. The physical domain with unit-square geometry is discretized using an unstructured finite element mesh with $N = 133$ nodes and 264 linear triangular elements.

The PCE representations from Eq. (9) are used for the uncertain system parameters and the solution process. The procedure outlined in the Sec. 4.1 is followed to formulate the intrusive SSFEM system given in the form of Eq. (33). The PCE coefficients of the solution process are obtained by directly solving the intrusive system given in Eq. (33). Similarly, the non-intrusive procedure discussed in Sec. 4.2 is employed to obtain the solution coefficients from Eq. (35). The required deterministic samples for the sparse grid quadratures are computed by directly solving Eq. (37). Note that the results presented in this section are obtained by employing a direct serial solver for both intrusive and non-intrusive SSFEM.

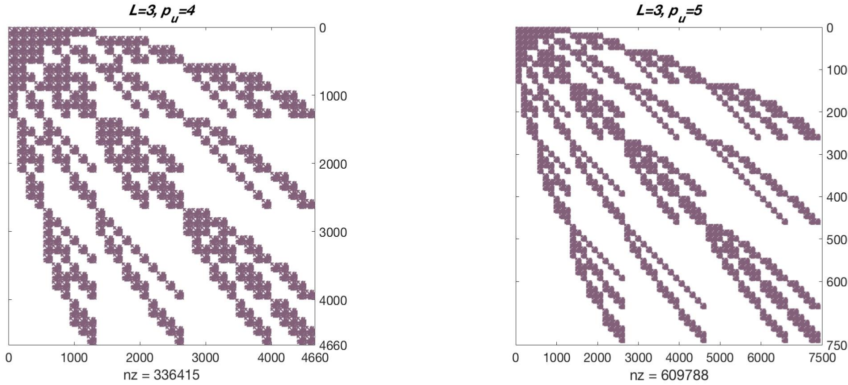

Figure 6: Block-sparse structures of the intrusive system matrices for a fixed mesh resolution with $L = 3$ and $p_u = 4, 5$

In Fig. 6 and 7, the block-diagonal structures of intrusive system matrices are presented. In Fig. 6, p_u is increased from 4 to 5 while keeping $L = 3$ fixed. In Fig. 7, L is

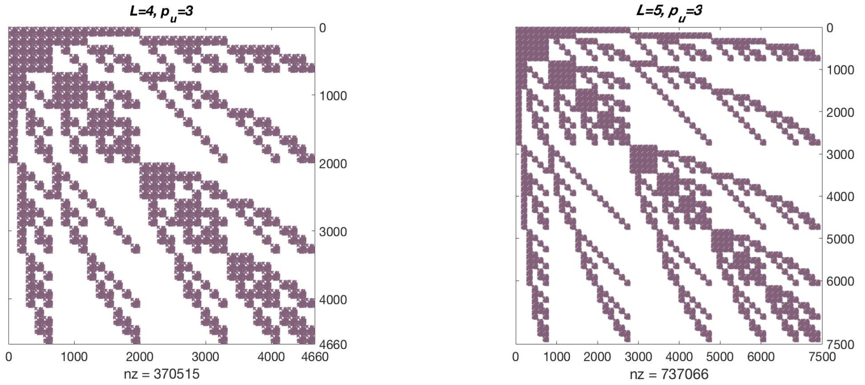

Figure 7: Block-sparse structures of the intrusive system matrices for a fixed mesh resolution with $p_u = 3$ and $L = 4, 5$

increased from 4 to 5 while keeping $p_u = 3$ fixed.

Note that the effect of increasing the order of expansion is different from that of the number of random variables. The system matrix resulting from varying L have more nonzero entries compared to the one resulting from a similar change in p_u as indicated by the number of nonzero entries nz in Fig. 6 and 7. For example, with $L = 5$ and $p_u = 3$ we have 0.73 million nonzero elements compared to $L = 3$ and $p_u = 5$ which has 0.60 million nonzero elements. Therefore, an increase in the number of random variables may pose a more significant challenge compared to increasing order of expansion or the finite element mesh resolution concerning memory and computation requirements. However, both the order p_u and dimension L influence the size and structure of the stochastic assembly matrix, thereby, affecting the condition number of the system matrix [5, 21].

To compare the performances of the intrusive approach with the non-intrusive approach, we consider the four different cases by selecting the number of random variables L as 2, 3, 4 and 5. The second order PCE is used for input PCE ensuring non-Gaussian terms are included in the series expansion. The order of the input PCE is kept constant for all the experiments. The minimum order of PCE used in the expansion of solution process is $p_u = 3$ which is higher than input PCE order, which captures the non-Gaussian effects [6, 10]. Then p_u is varied as 3, 4 and 5 to perform error analysis in the lower order PCE coefficients.

Note that the intrusive SSFEM code employed for all the simulation in this exercise is validated with the SSFEM code developed by Khalil [46], which is validated using method of manufactured solution (see Appendix-D.3 in [46] for more details). Additionally, the intrusive SSFEM serial solver used here is validated with the parallel domain decomposition based solver employed in [47] which is validated against MCS in [15]. For non-intrusive case, the deterministic samples are simulated using the code validated against an analytical case in FEniCS/puffin [34, 35]. The correctness of the quadrature points used to solve the integrals (in the numerator and the denominator of Eq. (35)) in the NISP approach are validated with the exact solution available for the $\left\langle \Psi_k^2(\boldsymbol{\xi}) \right\rangle$.

The relative error norms are computed using $p_u = 6$ in the intrusive SSFEM and used as the reference solution. For instance, the relative error in the i^{th} order solution coefficient can be computed as $||(\hat{\mathbf{u}}^i_{p_u=j} - \hat{\mathbf{u}}^i_{p_u=6})||/||\hat{\mathbf{u}}^i_{p_u=6}||$ with $j = 3, 4$ and 5 where $||.||$ denotes L_2 norm. For each case, the simulation is restricted to the maximum order of $p_u = 6$. This is because we were not able to fit the system matrix for the intrusive approach for the higher-order cases (i.e., $p_u > 6$) in the computer with quad-core processor and 16 gigabytes (GB) of random access memory (RAM) used for all the simulations.

The relative error norms in $0^{th}, 1^{st}$ and 2^{nd} order solution coefficients with $L = 2, 3, 4$ and 5 for both intrusive and non-intrusive SSFEM are plotted in Figs. 8 to 11. In the intrusive approach, as the order of expansion of the solution process p_u increases, the relative error in the lower order solution coefficients decreases. For example, if p_u increased from 3 to 4, the error in the lower order expansion terms, i.e., for $p_u = 0$, 1 or 2, decreases as expected. Similarly in the non-intrusive approach, as the level of quadrature l increases, the relative error in the individual solution PCE coefficients decreases. Note that, in the non-intrusive approach the solution coefficients are independent of each other as opposed to the intrusive approach, where the error in the lower order solution coefficients decrease as we include the higher-order expansion terms.

For the cases with lower order solution coefficients and a few random variables, for instance, 0^{th} order with $L = 2$ presented in Fig. 8, the relative errors are similar in both intrusive and non-intrusive approaches with $l = p_u$. For the higher-order solution coefficients with a large number of random variables, for instance, the 3^{rd} order coefficient with $L = 3$ presented in Fig. 9, the relative error is less in the case of intrusive approach compared to non-intrusive approach for $p_u = l$. The difference between the relative errors for the intrusive and non-intrusive approach grows as the p_u and the L increases. Therefore, in order to achieve the same level of accuracy in the PCE coefficients of the solution process, we need a higher level of the sparse quadrature in non-intrusive approach compared to the order of expansion in the intrusive approach (i.e., $l > p_u$). For example, in the case of 1^{st} order solution coefficient with 4 random variables presented in Fig. 10, to achieve the same level of accuracy as intrusive approach with $p_u = 3$, we need the fourth level of quadrature. Similarly, in the case of 2^{nd} order solution coefficient with 5 random variables shown in Fig. 11, to achieve the same level of accuracy as in intrusive approach with $p_u = 3$, we need the fifth level of quadrature.

For the same cases discussed above, the relative error norms in the standard deviation σ_u of the solution process (for all nodes) are shown in Fig. 12. Also, for $L = 4$ the error surface plots in the SD of the solution process using intrusive with $p_u = 3, 4$ and 5 and NISP with $l = 3, 4$ and 5 are compared in Fig. 13. These plots also suggest that, for the same level of accuracy in the standard deviation of the solution process with the intrusive SSFEM approach, we need a higher level of quadrature to solve the integral involved in the non-intrusive approach. Therefore, to model the uncertainty as a non-Gaussian

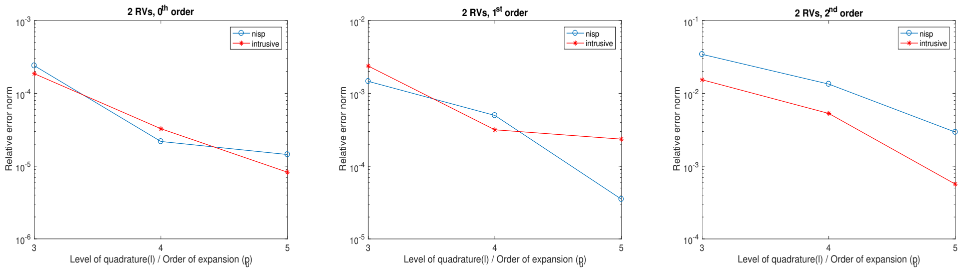

Figure 8: Error norms in 0^{th}, 1^{st} and 2^{nd} order PCE coefficients for $L = 2$.

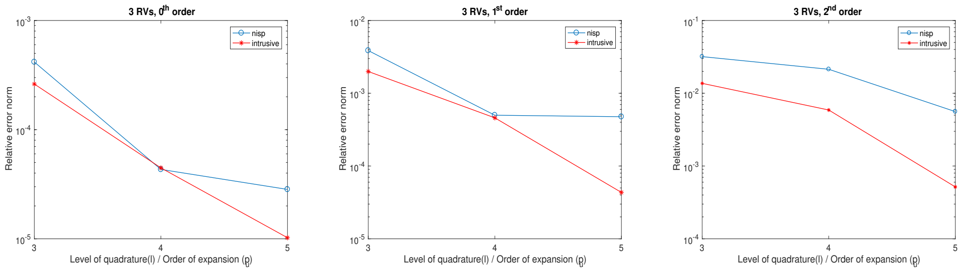

Figure 9: Error norms in 0^{th}, 1^{st} and 2^{nd} order PCE coefficients for $L = 3$.

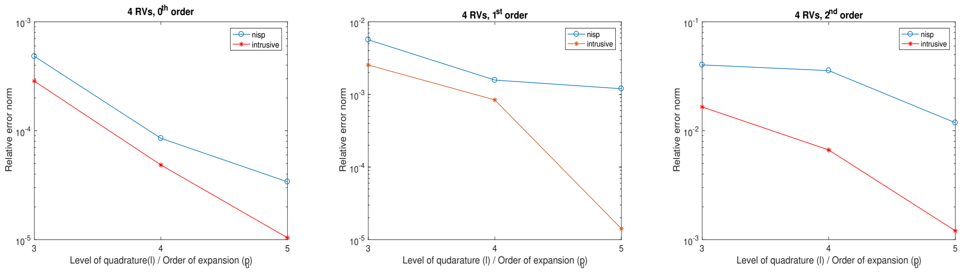

Figure 10: Error norms in 0^{th}, 1^{st} and 2^{nd} order PCE coefficients for $L = 4$.

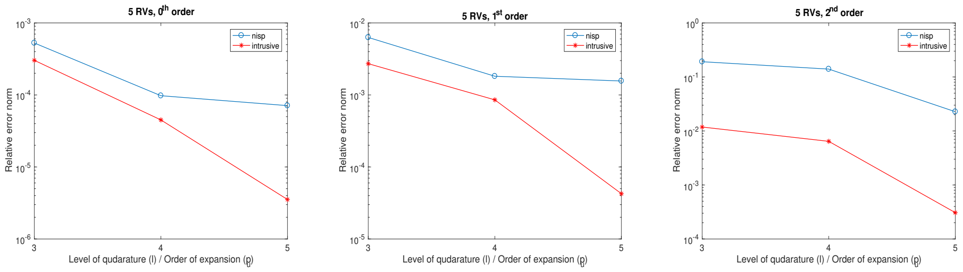

Figure 11: Comparison of relative error norms between intrusive and non-intrusive SSFEM in 0^{th}, 1^{st} and 2^{nd} order PCE coefficients for $L = 5$.

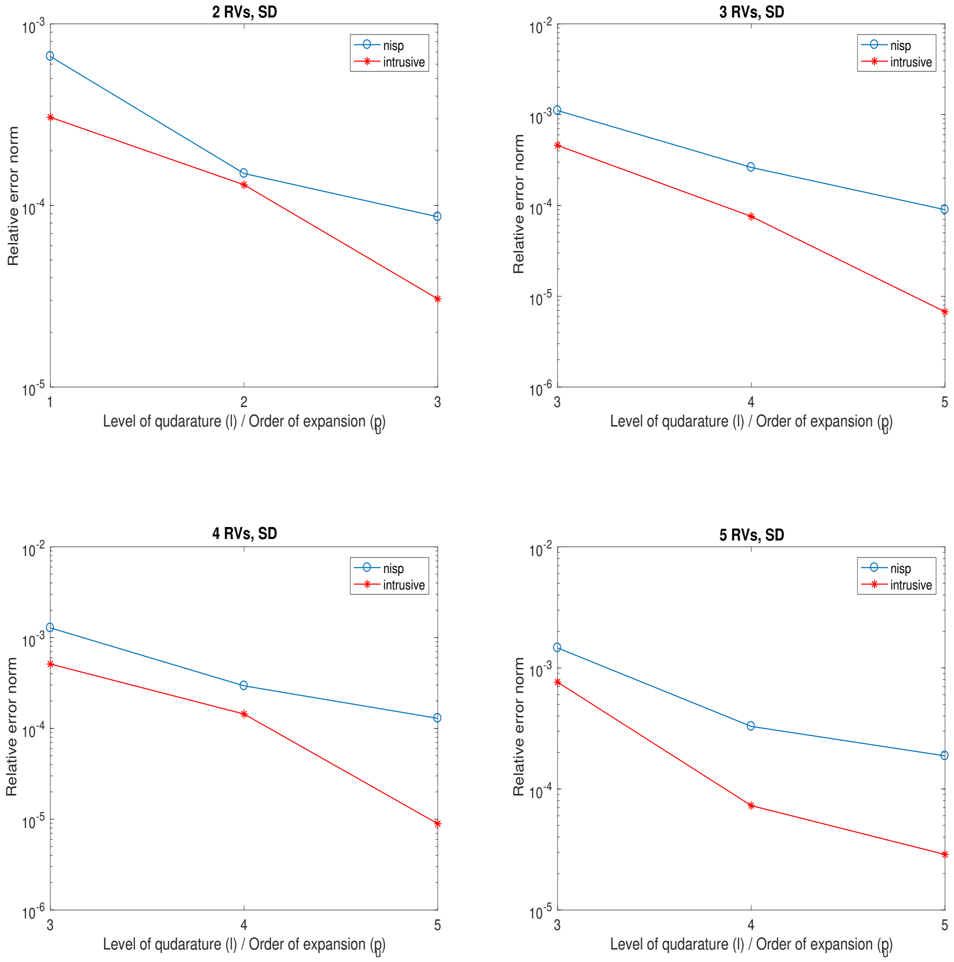

Figure 12: Comparison of relative error norms between intrusive and non-intrusive SSFEM in the standard deviation (computed for all nodes) of the solution process for $L = 2, 3, 4$ and 5.

stochastic process characterized by a large number of random variables we have to use the higher level sparse grid quadrature, i.e., $l >> p_u$. Consequently, we need a large number of samples in the non-intrusive approach. For instance, if we increase the level of quadrature from 3 to 4 or 5, there is a small increase in the number of quadrature points for low-dimensional cases, for example $L = 5$. However, the number of samples for high-dimensional cases, for example $L = 15$, increases quickly compared to the low-dimensional cases (i.e., $L = 5$) as shown in Table 1.

N_s	$L = 5$	$L = 15$
$l = 3$	241	11561
$l = 4$	781	120401
$l = 5$	2203	259607

Table 1: Number of sample (quadrature) points N_s in the sparse grid with level l and the number of random variables L.

In summary, from an implementational perspective the non-intrusive approach is favorable because one can directly employ any existing deterministic solver as a black box to simulate the required samples. On the other hand, the intrusive approach demands additional coding efforts. However, as demonstrated by the authors in [15, 47], the stochastic assembly procedure employed for intrusive SSFEM can utilize the readily available deterministic finite element assembly routines (such as FEniCS) which can substantially reduce the coding efforts. Also, as demonstrated above, for a large number of random variables and a high order of PCE, the intrusive approach has better control over the error in the individual solution coefficients compared with the non-intrusive approach. Therefore, for practical applications which may require modeling of non-Gaussian (for example, lognormal process) system parameters requiring with large-number of random variables, the intrusive approach can be advantageous.

Despite the fact that the PCE based approaches (both intrusive and non-intrusive) can outperform the other approaches for many applications, they pose the following challenges:

- The PCE based approaches have difficulty in capturing sudden changes in the response (solution process), for example, predicting shock dynamics in the presence of uncertainties [45, 48, 49] and responses of highly nonlinear aeroelastic systems which show an abrupt jump to a higher period limit cycle oscillations [27, 48, 50, 51].

- The PCE solution has non-uniform convergence and tends to break down for long-time integration which is called the long-term degeneracy problem [27, 44, 51, 52].

To address these challenge various adaptive PCE methods are developed [43–45, 49, 53]. One such method is adaptive multi-element generalized PCE [45, 49]. The main idea of

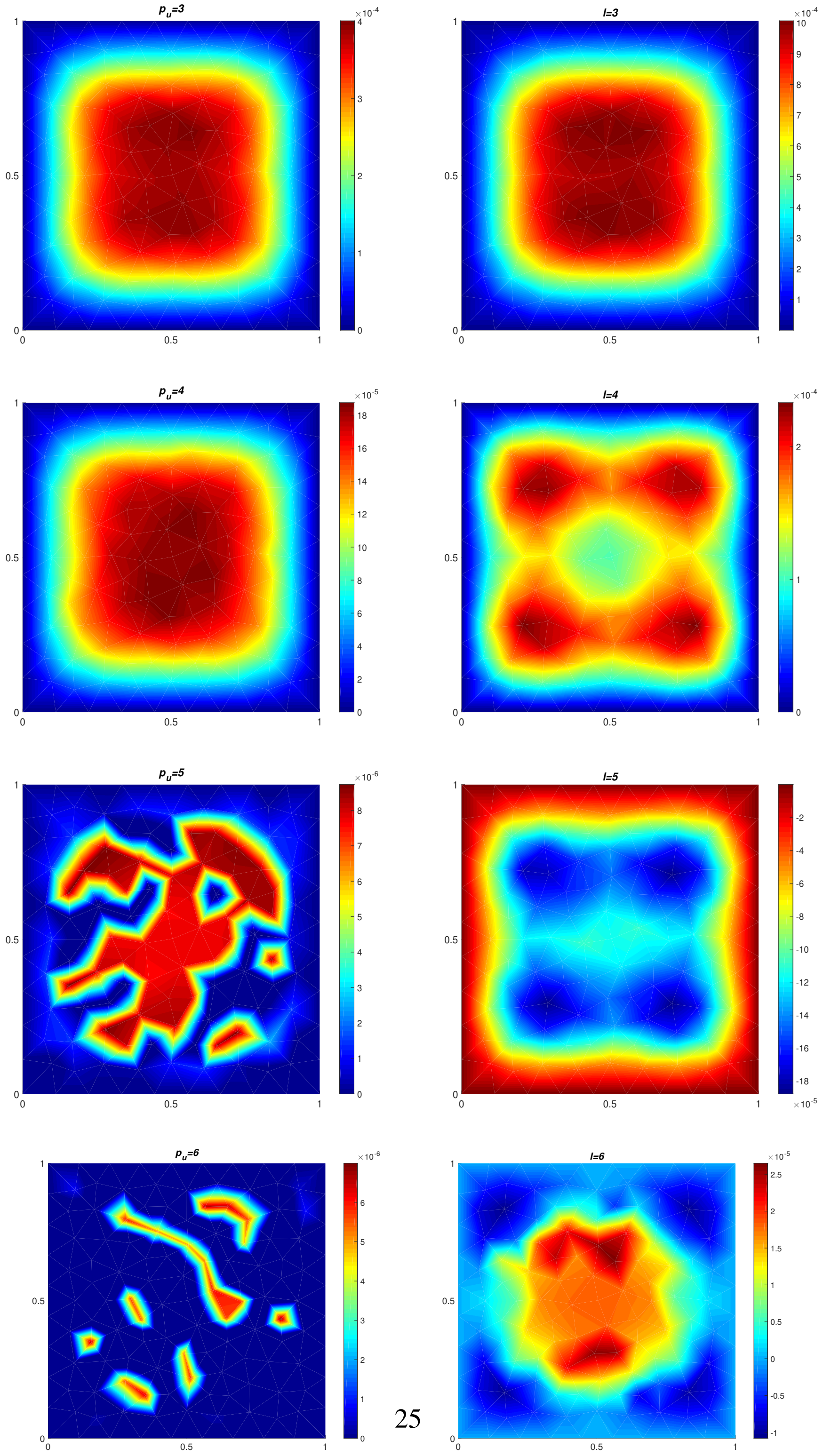

Figure 13: Comparison of error surface plots between intrusive and non-intrusive SSFEM for $L = 4$ in the standard deviation of the solution process using $p_u = 3, 4, 5$ and $l = 3, 4, 5$ with respect to standard deviation of the solution process using $p_u = 6$ and $l = 6$

the this method is to adaptively decompose the space of random inputs into multiple elements and subsequently employ PCE at the element level. Similar technique but a nonintrusive multi-element PCE formulation is employed in [44] in order to predict the stochastic response in the presence of discontinuities in the random space. In [53], a multi-resolution analysis based on multi-wavelet basis is applied to tackle uncertainty propagation in the complex and multidimensional stochastic problems. They conclude that the adaptive refinement of the multi-wavelet basis provides an attractive means to effectively tackle problems with steep or discontinuous dependence on random data. The nonintrusive B-spline stochastic projection method is developed in order to capture the sharp discontinuities along with the long term periodic behavior in [54]. In another approach by Gerritsma et al. [55], as time progresses the new stochastic variables are defined and corresponding orthogonal polynomials are constructed to alleviate long-term degeneracy issue. An algorithm based on constant phase interpolation was employed in [51, 56] to deal with the long time degeneracy problems.

Although we can readily accommodate the increased number of samples due to a large number of random variables in the non-intrusive approach, there is a substantial increase in the number of sample evaluations for the non-Gaussian input making it computationally costly compared to the intrusive approach. However, the memory required to assemble and solve the intrusive system increases as we increase the number of random variables. Therefore, for a computer with a fixed random access memory (RAM), there is an upper limit to the size of intrusive system we can accommodate. Nonetheless, if we can handle the increasing intrusive system size by distributing it among multiple nodes and employ an efficient parallel solver, we can get the solution coefficients quickly compared to the non-intrusive approach for the same level of accuracy. For these reasons, in this research, we have focused on: (a) the development of scalable domain decomposition solvers for intrusive SSFEM for two and three-dimensional stochastic PDEs and (b) an efficient parallel implementation of these solvers utilizing high-performance computing (HPC) to solve the problems with high resolution in both spatial and stochastic domains. Developing scalable solvers to tackle stochastic PDEs using SSFEM is an active area of research as evidenced by many articles published in the last couple of decades [21, 57–68].

6. Non-Intrusive SSFEM

Performing Galerkin projection onto the PC expansion of solution process given in Eq. (9) and then exploiting orthogonality properties of the basis functions, the PCE coefficients of the solution process can be evaluated as follows [1, 2, 69];

$$\hat{\mathbf{u}}_k = \frac{\langle \mathbf{u}(\theta)\Psi_k(\boldsymbol{\xi})\rangle}{\langle \Psi_j(\boldsymbol{\xi})\Psi_k(\boldsymbol{\xi})\rangle} = \frac{1}{\langle \Psi_k(\boldsymbol{\xi})^2\rangle}\int_\Omega \mathbf{u}(\theta)\Psi_k(\boldsymbol{\xi})\mathrm{p}(\boldsymbol{\xi})d\boldsymbol{\xi}, \tag{44}$$

where $\langle \Psi_j(\boldsymbol{\xi})\Psi_k(\boldsymbol{\xi})\rangle$ is non-zero only for $j = k$ and it can be obtained analytically beforehand [3, 15]. Therefore, the major computational efforts lies in the evaluation of the multidimensional integral in the numerator of Eq. (44) [2, 24, 25, 30, 31].

The efficient evaluation of the multidimensional integral in Eq. (44) is key part in the implementation of of this NISP approach especially for the high dimensional stochastic cases. Consider the FE discretization of a stochastic PDE given in Eq. (19). Using $\{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_{n_s}\}$ (where $\boldsymbol{\xi}_i$ is a set of L random variables), the following deterministic system is solved at n_s sample points using an existing deterministic solver as a black-box.

$$\mathbf{A}(\theta)\mathbf{u}(\theta) = \mathbf{f}. \tag{45}$$

Computed $\mathbf{u}(\theta)$ at each sample points are used to calculate the PCE coefficients of the solution process using Smolyak sparse grid quadrature [2, 25, 30–32, 36].

The sparse grid nodal set for $d = 2$ and $l = 3$ can be written as [3]

$$\begin{aligned}
\Theta_{l=3}^{(d=2)} &= \bigcup_{|l'| \le l+d-1} \left(\Theta_{l_1}^{(1)} \times \Theta_{l_2}^{(1)}\right) \qquad (46)\\
&= \left(\Theta_1^{(1)} \times \Theta_1^{(1)}\right) \quad (l_1 = 1, l_2 = 1)\\
&\cup \left(\Theta_1^{(1)} \times \Theta_2^{(1)}\right) \cup \left(\Theta_2^{(1)} \times \Theta_1^{(1)}\right)\\
&\cup \left(\Theta_1^{(1)} \times \Theta_3^{(1)}\right) \cup \left(\Theta_2^{(1)} \times \Theta_2^{(1)}\right) \cup \left(\Theta_3^{(1)} \times \Theta_1^{(1)}\right),
\end{aligned}$$

where $l' = (l_1 \ldots l_d)$ with $|l'| = \sum_{i=1}^{d} l_i$ (a specific example to obtain the sparse grid nodal set involving $\Theta_1^{(1)} \times \Theta_2^{(1)}$ is given below). For the implementation of the multi-dimensional sparse grid the growth rule in one-dimensional quadrature must be defined. In the current implementation, we have used the Gauss-Hermite quadrature rule [3, 32]. For further simplification, consider the Gauss-Hermite quadrature rule in one dimension with the nodes and weight specified in Table 2, for different level of quadrature.

level l	nodes $\Theta_l^{(1)}$	weights $\mathscr{W}_l^{(1)}$
$l = 1$	$\{0\}$	$\{1\}$
$l = 2$	$\{-1, 1\}$	$\{0.5, 0.5\}$
$l = 3$	$\{-1.7321, 0, 1.7321\}$	$\{0.167, 0.667, 0.167\}$

Table 2: Nodes and weights for Gauss-Hermite quadrature in one dimension and third level.

Using Eq. (46), Eq. (38) and Eq. (39), the nodes and weight for $l = 3$ and $d = 2$ are obtained as shown in Table 3.

nodes $\Theta_2^{(2)}$	weights $\mathscr{W}_2^{(2)}$
$\{-1.732, 0\}$	0.167
$\{-1.0, -1.0\}$	0.25
$\{-1.0, 0\}$	-0.5
$\{-1.0, 1.0\}$	0.25
$\{0, -1.732\}$	0.167
$\{0, -1.0\}$	-0.5
$\{0, 0\}$	1.333
$\{0, 1.0\}$	-0.5
$\{0, 1.732\}$	0.167
$\{1.0, -1.0\}$	0.25
$\{1.0, 0\}$	-0.5
$\{1.0, 1.0\}$	0.25
$\{1.732, 0\}$	0.167

Table 3: Nodes and weights for Gauss-Hermite quadrature in two dimension and third level.

The relations in Eq. (46) are used to get the set of nodes in Table 3. For instance, the nodes for $\left(\Theta_1^{(1)} \times \Theta_2^{(1)}\right)$ are obtained by taking tensor produt of $\Theta_1^{(1)} = \{0\}$ and $\Theta_2^{(1)} = \{-1, 1\}$, resulting into the following set of nodes = $\left[\{0, -1\}, \{0, 1\}\right]$ which correspond to the 6^{rd} and the 8^{th} row in the first column of Table 3. The difference relation given in Eq. (39) is used to get the corresponding weights [3] (refer to the example given in Sec. 4.2 for more details about the weights of the sparse grid quadrature).

6.0.1. Scalability using Large-Scale HPC Cluster

In a second set of experiments, we study the numerical scalability of the BDDC/NNC-PCGM solver for a highly dense mesh having 0.332 million nodes and 0.664 million elements. The number of cores utilized for these experiments vary from 1000 to 4000. The number of PCE terms ranges from 20 to 220 and the order of expansion $p_u = 3$ is kept constant. The scalability results with respect to the number of subdomains and the number of PCE terms are plotted in Fig. 14 and Fig. 15 respectively. These plots indicates that the solver is numerically scalable with large-scale computing clusters Similarly, the results in Fig. 14 shows an excellent strong parallel scaling with thousands of cores.

Furthermore, using PETSc-*log-view*, the total floating-point operations of all cores, average and maximum/minimum floating-point operations among cores are estimated against various numbers of cores used to solve the SSFEM system in the second set of experiments. The results are summarized in Table 4 and 5 below. Table 4 indicates, for a fixed problem size (both mesh resolution and the number of PCE terms are fixed), if we increase the number of cores, the total and average floating-point operations count decreases due to the decrease in workload per core as more cores are engaged. Table 5

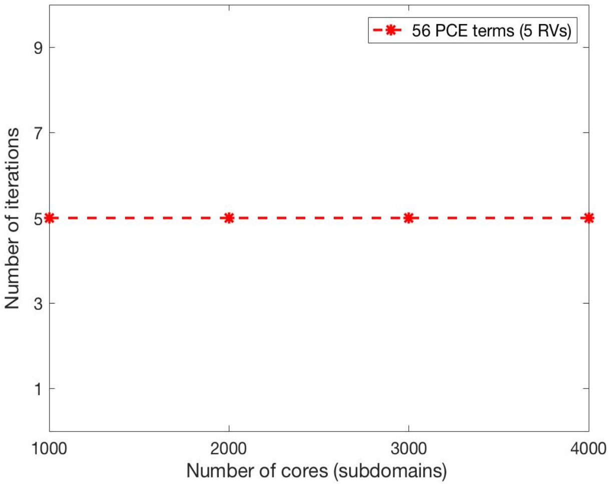

Figure 14: Iteration count versus number of subdomains for the fixed mesh resolution with fixed number of PCE terms.

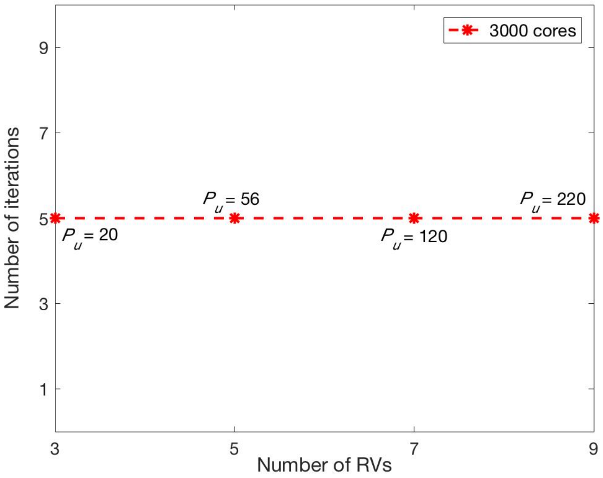

Figure 15: Iteration count versus number PCE terms for the fixed mesh resolution with fixed number of subdomains.

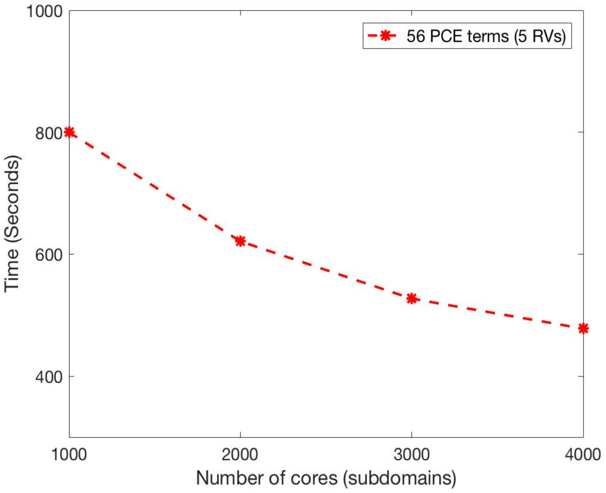

Figure 16: Execution time versus number of subdomains with the fixed mesh resolution and the fixed number of PCEs.

indicates, for a fixed number of cores ($n_s = 4000$), both average and total floating-point operations count increases with the increasing number of PCE terms, emphasizing the increased workload per core. The total floating-point operations required in the case of $P_u = 364$ reached 1.728 Petaflop. The results emphasize the necessity of using large-scale computing clusters and importance of scalable solvers for uncertainty propagation. Further increase in the number of RVs could demand a computing cluster capable of 10^{18} floating-point operations, pointing to emerging *exascale* computing [70, 71]. From the perspective of load balancing, the maximum to minimum floating-point operations ratio among the cores is far from 1 indicating workload imbalance. Further investigation is needed to fine-tune the parallel implementation to achieve efficient load balancing.

Fixed PCE ($P_u = 56$)	Total (Petaflop)	Average (Teraflop)	Max/Min
$N_s = 1000$	0.1684	0.1684	1.87000
$N_s = 2000$	0.1255	0.0627	1.99444
$N_s = 3000$	0.1195	0.0395	1.78914
$N_s = 4000$	0.1201	0.0300	1.85542

Table 4: Floating-point operations for different number of cores with the fixed number of PCE terms

6.1. *Comparison with Non-Intrusive SSFEM at High Dimensions*

To demonstrate the superiority of intrusive SSFEM equipped with scalable parallel DD solvers, consider a finite element mesh with 52704 nodes and 105410 triangular el-

Fixed cores ($N_s = 4000$)	Total (Petaflop)	Average (Teraflop)	Max/Min
$P_u = 56$	0.1232	0.0308	1.85542
$P_u = 120$	0.7884	0.1971	2.00349
$P_u = 220$	1.0448	0.2612	1.99514
$P_u = 364$	1.7281	0.4320	1.85249

Table 5: Floating-point operations for different number of PCE terms with the fixed number of cores

ements. The intrusive SSFEM systems formulated by using the fixed $p_u = 3$ and four different cases by selecting the number of RVs as $L = 5, 10, 15$ and 20 are considered. For the parallel solution, we employed BDDC/NNC solver using $80, 160, 240$ and 320 cores respectively. For the same cases, the sparse grid with the level of quadrature, $l = 3$ and $l = 4$ are employed to solve the problem non-intrusively. Note that the domain decomposition solver is not needed for the non-intrusive approach as each (deterministic) problem for a given sample fits in the memory of a given node. The number of cores employed in the intrusive approach and non-intrusive approach is kept the same, for instance, in the case of $L = 5$, we have used 80 cores to solve the intrusive system in parallel. Similarly, we consider that 80 samples can be evaluated in parallel for the non-intrusive approach. Note that, the total CPU time for the non-intrusive approach is computed by multiplying the total number of the samples by the average time required for one sample and then, dividing it by the number of cores.

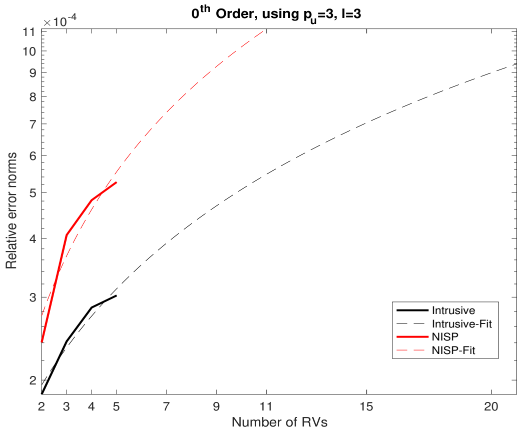

Figure 17: Extrapolated relative error convergence plots of intrusive and non-intrusive for 0^{th} order with third order and third level of quadrature.

Computational time for the intrusive and non-intrusive SSFEM for the four different cases by selecting the number of random variables $5, 10, 15$ and 20 is plotted in Fig. 18.

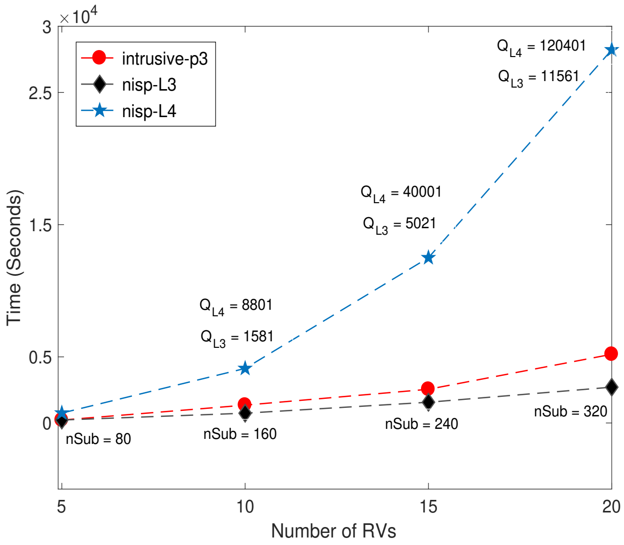

Figure 18: Comparison of execution time for intrusive and non-intrusive SSFEM with the fixed mesh resolution and $L = 5, 10, 15, 20$.

For the third-level sparse grid in non-intrusive approach, the total execution time for all four cases is less compared to that of intrusive approach with the third-order expansion (i.e., $l = p_u$). However, if we increase the level of sparse grid to 4 (i.e., $l > p_u$), the computational time required to evaluate the non-intrusive samples rises quickly, especially in the cases with large number of random variables, e.g., 15 or 20. As demonstrated earlier in Fig. 11, to achieve the same level of accuracy in PCE coefficients of the solution process we need a higher-level of the sparse grid quadrature in the non-intrusive SSFEM compared to the order of expansion in the intrusive SSFEM (i.e., $l > p_u$). Therefore, intrusive approach, equipped with the domain decomposition solvers shows computational advantages over non-intrusive approach for the high-dimensional, non-Gaussian stochastic input as considered in this specific application.

In summary, building on the basic probabilistic formulations of BDDC/NNC [72] and FETI-DP [73] solvers for SPDEs, this work extends the algorithms in order to tackle high-dimensional stochastic systems and perform scalability studies with respect to the number of input random variables and the corresponding PCE terms. In contrast to the references [72, 73], direct factorization of subdomain-level matrices is avoided and better memory management strategy is implemented for high-dimensional stochastic systems. Firstly, the assembly procedure developed by the authors in [15, 47], reduces the memory usage per core and thereby offers the ability to handle higher-dimensional stochastic systems compared to the references [72, 73]. Moreover, the local assembly procedure provides a method to utilize existing deterministic FEM packages such as

FEniCS, which drastically reduces the coding efforts required for the implementation of the DD-based intrusive SSFEM to a variety of PDEs. Secondly, the third-level iterative solvers used to tackle the subdomain-level local systems reduce the execution time and memory consumption; and hence they offer the ability to tackle stochastic systems with larger number of input random variables and PCE terms than the previous solvers [72, 73]. Finally, we demonstrated the computational superiority of intrusive SSFEM equipped with scalable parallel solvers over non-intrusive SSFEM for non-Gaussian and high-dimensional stochastic systems. These facts highlights the advantages of intrusive approach and demonstrate the necessity of scalable parallel solvers for extreme scale uncertainty quantification.

Acknowledgments

The first author acknowledges the support of an Ontario Trillium scholarship. The fourth author acknowledges the support of the Canadian Department of National Defence and a Discovery Grant from Natural Sciences and Engineering Research Council of Canada. The fifth author acknowledges the support of a Discovery Grant from Natural Sciences and Engineering Research Council of Canada and the Canada Research Chair Program. The computing infrastructure is supported by the Canada Foundation for Innovation (CFI), the Ontario Innovation Trust (OIT), Compute Canada and Calcul Québec.

References

[1] R. Ghanem, P. Spanos, Stochastic finite elements: a spectral approach, Springer-Verlag, New York, 1991.

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

[3] R. C. Smith, Uncertainty quantification: theory, implementation, and applications, Vol. 12, SIAM, 2013.

[4] C. Canuto, M. Y. Hussaini, A. Quarteroni, A. Thomas Jr, et al., Spectral methods in fluid dynamics, Springer Science & Business Media, 2012.

[5] R. Ghanem, R. Kruger, Numerical solution of spectral stochastic finite element systems, Computer Methods in Applied Mechanics and Engineering 129 (3) (1996) 289–303.

[6] R. Ghanem, Ingredients for a general purpose stochastic finite elements implementation, Computer Methods in Applied Mechanics and Engineering 168 (1) (1999) 19–34.

[7] A. Papoulis, S. U. Pillai, Probability, random variables, and stochastic processes, Tata McGraw-Hill Education, 2002.

[8] P. Billingsley, Probability and measure, John Wiley & Sons, 2008.

[9] D. Xiu, G. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations, SIAM Journal on Scientific Computing 24 (2) (2002) 619–644.

[10] R. Ghanem, Stochastic finite elements with multiple random non-Gaussian properties, Journal of Engineering Mechanics 125 (1) (1999) 26–40.

[11] I. Babuska, R. Tempone, G. E. Zouraris, Galerkin finite element approximations of stochastic elliptic partial differential equations, SIAM Journal on Numerical Analysis 42 (2) (2004) 800–825.

[12] I. Babuška, R. Tempone, G. E. Zouraris, Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation, Computer Methods in Applied Mechanics and Engineering 194 (12) (2005) 1251–1294.

[13] R. Ghanem, The nonlinear Gaussian spectrum of log-normal stochastic processes and variables, Journal of applied mechanics 66 (4) (1999) 964–973.

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

[15] A. Desai, Scalable domain decomposition algorithms for uncertainty quantification in high performance computing, Ph.D. thesis, Carleton University (2019).

[16] M. Rosenblatt, Remarks on a multivariate transformation, The annals of mathematical statistics 23 (3) (1952) 470–472.

[17] C. Soize, R. Ghanem, Physical systems with random uncertainties: chaos representations with arbitrary probability measure, SIAM Journal on Scientific Computing 26 (2) (2004) 395–410.

[18] S. Rahman, A polynomial chaos expansion in dependent random variables, Journal of Mathematical Analysis and Applications 464 (1) (2018) 749–775.

[19] S. Huang, S. Quek, K. Phoon, Convergence study of the truncated Karhunen–Loeve expansion for simulation of stochastic processes, International journal for numerical methods in engineering 52 (9) (2001) 1029–1043.

[20] P. G. Ciarlet, The finite element method for elliptic problems, SIAM, 2002.

[21] M. Pellissetti, R. Ghanem, Iterative solution of systems of linear equations arising in the context of stochastic finite elements, Advances in Engineering Software 31 (8) (2000) 607–616.

[22] A. Sarkar, N. Benabbou, R. Ghanem, Domain decomposition of stochastic PDEs: theoretical formulations, International Journal for Numerical Methods in Engineering 77 (5) (2009) 689–701.

[23] E. Ullmann, H. C. Elman, O. G. Ernst, Efficient iterative solvers for stochastic Galerkin discretizations of log-transformed random diffusion problems, SIAM Journal on Scientific Computing 34 (2) (2012) A659–A682.

[24] M. Reagana, H. Najm, R. Ghanem, O. Knio, Uncertainty quantification in reacting-flow simulations through non-intrusive spectral projection, Combustion and Flame 132 (3) (2003) 545–555.

[25] S. Hosder, R. W. Walters, R. Perez, A non-intrusive polynomial chaos method for uncertainty propagation in CFD simulations, AIAA Paper 891 (2006) 2006.

[26] M. Eldred, J. Burkardt, Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification, AIAA paper 976 (2009) 1–20.

[27] A. Desai, S. Sarkar, Analysis of a nonlinear aeroelastic system with parametric uncertainties using polynomial chaos expansion, Mathematical Problems in Engineering 2010.

[28] I. Babuška, F. Nobile, R. Tempone, A stochastic collocation method for elliptic partial differential equations with random input data, SIAM Journal on Numerical Analysis 45 (3) (2007) 1005–1034.

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

[30] B. Ganapathysubramanian, N. Zabaras, Sparse grid collocation schemes for stochastic natural convection problems, Journal of Computational Physics 225 (1) (2007) 652–685.

[31] F. Nobile, R. Tempone, C. G. Webster, A sparse grid stochastic collocation method for partial differential equations with random input data, SIAM Journal on Numerical Analysis 46 (5) (2008) 2309–2345.

[32] M. S. Eldred, Recent advances in non-intrusive polynomial chaos and stochastic collocation methods for uncertainty analysis and design, AIAA Paper 2274 (2009) 37.

[33] A. Logg, G. Wells, J. Hake, DOLFIN: A C++/Python finite element library, in: Automated Solution of Differential Equations by the Finite Element Method, Springer, 2012.
URL `https://bitbucket.org/fenics-project/dolfin`

[34] A. Logg, A. Mardal, G. Wells, Automated solution of differential equations by the finite element method: The FEniCS book, Vol. 84, Springer Science & Business Media, 2012.
URL `http://fenicsproject.org`

[35] J. Hoffman, L. Anders, FEniCS/puffin user manual, Tech. rep., KTH Royal Institute of Technology (2013).
URL `https://fenicsproject.org/pub/documents/puffin/`

[36] B. Debusschere, K. Sargsyan, C. Safta, UQTk version 2.1 user manual, Tech. rep., Sandia National Laboratory (SNL) (2013).
URL `http://www.sandia.gov/UQToolkit/`

[37] P. G. Constantine, M. S. Eldred, E. T. Phipps, Sparse pseudospectral approximation method, Computer Methods in Applied Mechanics and Engineering 229 (2012) 1–12.

[38] H. C. Elman, C. W. Miller, E. T. Phipps, R. S. Tuminaro, Assessment of collocation and Galerkin approaches to linear diffusion equations with random data, International Journal for Uncertainty Quantification 1 (2011) 1.

[39] R. Tipireddy, E. Phipps, R. Ghanem, A comparison of solution methods for stochastic partial differential equations, CSRI Summer Proceedings 2010 79.

[40] J. Bäck, F. Nobile, L. Tamellini, R. Tempone, Stochastic spectral Galerkin and collocation methods for PDEs with random coefficients: a numerical comparison, in: Spectral and High Order Methods for Partial Differential Equations, Springer, 2011, pp. 43–62.

[41] G. Stefanou, The stochastic finite element method: past, present and future, Computer Methods in Applied Mechanics and Engineering 198 (9) (2009) 1031–1051.

[42] L. Giraldi, A. Litvinenko, D. Liu, H. G. Matthies, A. Nouy, To be or not to be intrusive? the solution of parametric and stochastic equations—the "plain vanilla" Galerkin case, SIAM Journal on Scientific Computing 36 (6) (2014) A2720–A2744.

[43] R. Li, R. Ghanem, Adaptive polynomial chaos expansions applied to statistics of extremes in nonlinear random vibration, Probabilistic engineering mechanics 13 (2) (1998) 125–136.

[44] D. Lucor, G. E. Karniadakis, Adaptive generalized polynomial chaos for nonlinear random oscillators, SIAM Journal on Scientific Computing 26 (2) (2004) 720–735.

[45] X. Wan, G. Karniadakis, An adaptive multi-element generalized polynomial chaos method for stochastic differential equations, Journal of Computational Physics 209 (2) (2005) 617–642.

[46] M. Khalil, Bayesian inference for complex and large-scale engineering systems, Ph.D. thesis, Carleton University Ottawa (2013).

[47] A. Desai, M. Khalil, C. Pettit, D. Poirel, A. Sarkar, Scalable domain decomposition solvers for stochastic PDEs in high performance computing, Computer Methods in Applied Mechanics and Engineering 335 (2017) 194–222.

[48] G. Lin, C.-H. Su, G. E. Karniadakis, Predicting shock dynamics in the presence of uncertainties, Journal of Computational Physics 217 (1) (2006) 260–276.

[49] X. Wan, G. E. Karniadakis, Beyond Wiener–Askey expansions: handling arbitrary pdfs, Journal of Scientific Computing 27 (1-3) (2006) 455–464.

[50] L. Liu, Y. Wong, B. Lee, Non-linear aeroelastic analysis using the point transformation method, part 1: Freeplay model, Journal of Sound and Vibration 253 (2) (2002) 447–469.

[51] A. Desai, J. A. S. Witteveen, S. Sarkar, Uncertainty quantification of a nonlinear aeroelastic system using polynomial chaos expansion with constant phase interpolation, Journal of Vibration and Acoustics 135 (5) (2013) 051034.

[52] X. Wan, G. Karniadakis, Long-term behavior of polynomial chaos in stochastic flow simulations, Computer methods in applied mechanics and engineering 195 (41) (2006) 5582–5596.

[53] O. Le Maıtre, H. N. Najm, R. Ghanem, O. Knio, Multi-resolution analysis of Wiener-type uncertainty propagation schemes, Journal of Computational Physics 197 (2) (2004) 502–531.

[54] D. R. Millman, P. I. King, R. C. Maple, P. S. Beran, L. K. Chilton, Uncertainty quantification with a B-spline stochastic projection, AIAA journal 44 (8) (2006) 1845–1853.

[55] M. Gerritsma, J.-B. Van der Steen, P. Vos, G. Karniadakis, Time-dependent generalized polynomial chaos, Journal of Computational Physics 229 (22) (2010) 8333–8363.

[56] J. A. Witteveen, Efficient and robust uncertainty quantification for computational fluid dynamics and fluid-structure interaction, TU Delft, Delft University of Technology, 2009.

[57] O. Le Maıtre, O. Knio, B. Debusschere, H. Najm, R. Ghanem, A multigrid solver for two-dimensional stochastic diffusion equations, Computer Methods in Applied Mechanics and Engineering 192 (41) (2003) 4723–4744.

[58] D. B. Chung, M. A. Gutiérrez, L. L. Graham-Brady, F.-J. Lingen, Efficient numerical strategies for spectral stochastic finite element models, International journal for numerical methods in engineering 64 (10) (2005) 1334–1349.

[59] H. Elman, D. Furnival, Solving the stochastic steady-state diffusion problem using multigrid, IMA Journal of Numerical Analysis 27 (4) (2007) 675–688.

[60] C. E. Powell, H. C. Elman, Block-diagonal preconditioning for spectral stochastic finite-element systems, IMA Journal of Numerical Analysis 29 (2) (2009) 350–375.

[61] D. Ghosh, P. Avery, C. Farhat, A FETI-preconditioned conjugate gradient method for large-scale stochastic finite element problems, International Journal for Numerical Methods in Engineering 80 (6-7) (2009) 914–931.

[62] E. Ullmann, A Kronecker product preconditioner for stochastic Galerkin finite element discretization, SIAM Journal on Scientific Computing 32 (2) (2010) 923–946.

[63] B. Sousedík, R. Ghanem, E. Phipps, Hierarchical Schur complement preconditioner for the stochastic Galerkin finite element methods, Numerical Linear Algebra with Applications 21 (1) (2014) 136–151.

[64] W. Subber, S. Loisel, Schwarz preconditioners for stochastic elliptic PDEs, Computer Methods in Applied Mechanics and Engineering 272 (2014) 34–57.

[65] W. Subber, Domain decomposition methods for uncertainty quantification, Ph.D. thesis, Carleton University Ottawa (2012).

[66] C. E. Powell, D. Silvester, V. Simoncini, An efficient reduced basis solver for stochastic Galerkin matrix equations, SIAM Journal on Scientific Computing 39 (1) (2017) A141–A163.

[67] G. Stavroulakis, D. Giovanis, V. Papadopoulos, M. Papadrakakis, A GPU domain decomposition solution for spectral stochastic finite element method, Computer Methods in Applied Mechanics and Engineering 327 (2017) 392–410.

[68] S. Pranesh, D. Ghosh, A FETI-DP based parallel hybrid stochastic finite element method for large stochastic systems, Computers & Structures 195 (2018) 64–73.

[69] D. Xiu, Fast numerical methods for stochastic computations: a review, Communications in Computa-

tional Physics 5 (2-4) (2009) 242–272.

[70] S. Ashby, P. Beckman, J. Chen, P. Colella, B. Collins, D. Crawford, J. Dongarra, D. Kothe, R. Lusk, P. Messina, et al., The opportunities and challenges of exascale computing, Summary Report of the Advanced Scientific Computing Advisory Committee (ASCAC) Subcommittee (2010) 1–77.

[71] I. S. Duff, European exascale software initiative: Numerical libraries, solvers and algorithms, in: European Conference on Parallel Processing, Springer, 2011, pp. 295–304.

[72] W. Subber, A. Sarkar, A domain decomposition method of stochastic PDEs: An iterative solution techniques using a two-level scalable preconditioner, Journal of Computational Physics 257 (2014) 298–317.

[73] W. Subber, A. Sarkar, Dual-primal domain decomposition method for uncertainty quantification, Computer Methods in Applied Mechanics and Engineering 266 (2013) 112–124.