

# Flow-driven spectral chaos (FSC) method for simulating long-time dynamics of arbitrary-order non-linear stochastic dynamical systems

Hugo Esquivel<sup>a</sup>, Arun Prakash<sup>a</sup>, Guang Lin<sup>b,1</sup>

<sup>a</sup>*Lyles School of Civil Engineering, Purdue University, 550 W Stadium Ave, West Lafayette IN 47907, USA*

<sup>b</sup>*Department of Mathematics, School of Mechanical Engineering, Purdue University, 150 N University St, West Lafayette IN 47907, USA*

---

## Abstract

Uncertainty quantification techniques such as the time-dependent generalized polynomial chaos (TD-gPC) use an adaptive orthogonal basis to better represent the stochastic part of the solution space (aka random function space) in time. However, because the random function space is constructed using tensor products, TD-gPC-based methods are known to suffer from the curse of dimensionality. In this paper, we introduce a new numerical method called the *flow-driven spectral chaos* (FSC) which overcomes this curse of dimensionality at the random-function-space level. The proposed method is not only computationally more efficient than existing TD-gPC-based methods but is also far more accurate. The FSC method uses the concept of *enriched stochastic flow maps* to track the evolution of a finite-dimensional random function space efficiently in time. To transfer the probability information from one random function space to another, two approaches are developed and studied herein. In the first approach, the probability information is transferred in the mean-square sense, whereas in the second approach the transfer is done *exactly* using a new theorem that was developed for this purpose. The FSC method can quantify uncertainties with high fidelity, especially for the long-time response of stochastic dynamical systems governed by ODEs of arbitrary order. Six representative numerical examples, including a nonlinear problem (the Van-der-Pol oscillator), are presented to demonstrate the performance of the FSC method and corroborate the claims of its superior numerical properties. Finally, a parametric, high-dimensional stochastic problem is used to demonstrate that when the FSC method is used in conjunction with Monte Carlo integration, the curse of dimensionality can be overcome altogether.

**Keywords:** uncertainty quantification; long-time integration; stochastic flow map; (nonlinear) stochastic dynamical systems; flow-driven spectral chaos (FSC); TD-gPC.

---

## Note

This is an updated version of the journal article. Errata can be found on the last page of this document.

## Highlights

- FSC can propagate and quantify with high fidelity the long-time response of SODEs.
- FSC is partially insensitive to the curse of dimensionality.
- FSC is computationally more efficient than TD-gPC for the same accuracy level.
- FSC can transfer the probability information exactly at every instant of time.
- FSC can be used in the resolution of high-dimensional stochastic dynamical systems.

---

*Email addresses:* hesquive@purdue.edu (Hugo Esquivel), aprakas@purdue.edu (Arun Prakash), guanglin@purdue.edu (Guang Lin)

<sup>1</sup>Department of Statistics, Department of Earth, Atmospheric, and Planetary Sciences (by courtesy), Purdue University.

*Updated version of preprint submitted to Journal of Computational Physics (Elsevier)*

*July 22, 2022*

## 1. Introduction

The need for quantifying uncertainties for real-world applications arises in different fields, such as in physics, engineering, economics, sociology, etc. In structural engineering, for example, the source of random variability can arise from: material properties, imperfections in geometry, loading scenarios, boundary conditions, etc. Once this random variability is identified, it can be characterized mathematically using random variables, stochastic processes or, more generally, random fields in space and time. Various methods for solving stochastic differential equations have been proposed to date, among which we mention: Monte Carlo-based methods [1–3], collocation-based methods [4–8], perturbation-based methods [9–12], operator-based methods [13–16], and spectral-based methods [17–19]. This work is based on the spectral approach, for which we give a short historical overview below.

The *polynomial chaos* (PC), as originally introduced by Wiener in 1938 [20] and then further extended in [21, 22], is a spectral-based method to model stochastic processes with (independent) Gaussian random variables. Roughly speaking, the method uses Hermite polynomials as the underlying basis to expand a stochastic process in the space of random functions, and it is considered to be an extension of the theory of nonlinear functionals developed by Volterra in 1913 [23] for stochastic systems. Such an expansion is known to be convergent in the mean-square sense for stochastic processes with a finite second moment, thanks to the Cameron-Martin theorem [24]. Therefore, these processes are also termed *second-order stochastic processes* in the literature. Even though the PC method was applied to solve different stochastic problems at the time, it was later recognized that it suffered from non-uniform convergence for systems with non-Gaussian random variables. Lucor et al. [25] demonstrated that, under Wiener’s framework of Hermite functionals, the convergence rate of systems subjected to Gaussian input is exponential but substantially slower otherwise [26, 27]. Nonetheless, steady progress was made between the 1950s and 1980s towards generalizing Wiener’s ideas for systems with non-Gaussian inputs (e.g. [28–32]).

In the early 1990s, Ghanem and Spanos [33, 34] developed a method in the context of stochastic finite elements. The method essentially uses Wiener’s theory on polynomial chaos to decompose a second-order stochastic process into deterministic and non-deterministic parts. The non-deterministic part of the process can then be treated as an element of a Hilbert space, and thus, be approximated by its Galerkin projection onto a subspace spanned by a finite number of Hermite polynomials. Because the subspace still needs to be spanned by Hermite polynomials, this method is only capable of achieving exponential convergence for stochastic systems involving Gaussian random variables. Yet, the method was successfully applied by several researchers in the branch of continuum mechanics, including solid and fluid mechanics, in problems displaying random variability in their definition (e.g. [35–38]).

In 2002, Xiu and Karniadakis [26] introduced the *generalized polynomial chaos* (gPC) method to overcome the issue of convergence rate of the PC method. By employing an orthogonal basis from the Askey family—but concordant with the measure defined in the probability space—they showed that a process expanded with such a basis leads to exponential convergence to the solution. Thus, in the years that followed, the gPC method was demonstrated to be capable of solving a wider number of stochastic problems found in practice (e.g. [39–42]). The method, however, was later found not to be suitable for problems that feature strong nonlinear dependencies over the probability space as time progresses. For example, for long-time integration of stochastic dynamical systems, the gPC method fails to capture the probability moments accurately because the probability distribution of the solution changes significantly with time. In 2005, Wan and Karniadakis [43] developed the *multi-element generalized polynomial chaos* (ME-gPC) method to account for these nonlinear dependencies in time, such as the ability to handle stochastic discontinuities and long-time response of stochastic dynamical systems on-the-fly. The key idea of ME-gPC is to adaptively decompose the random space into elements until a pre-specified threshold for the relative error in variance is reached. Then, a stochastic spectral expansion is used on each random element to push the system’s state forward in time. This process is repeated every time the threshold is exceeded during the simulation. The ME-gPC method and its variants (e.g. [6, 7, 44–47]) have been proved to be capable of solving numerous problems in engineering and sciences (e.g. [48–53]).

The dynamically orthogonal PC (DO-PC) is another approach used for uncertainty quantification. It was formulated by Sapsis and Lermusiaux [54] in 2009 to study the response of continuous stochastic dynamical

systems more effectively. In this approach, the time rate of change of the spatio-temporal function space is ensured to be kept orthogonal to itself as the simulation proceeds. This condition, called the dynamically orthogonal (DO) condition, is enforced at every time step to derive an exact, closed set of evolution equations in time. With additional restrictions on the form of the solution representation, the DO-PC approach can recover both the POD (Proper Orthogonal Decomposition) method [55, 56] and the gPC method. Since its inception, the DO-PC has undergone further modifications and extensions to broaden its range of applications (e.g. [57–60]). An error analysis for the DO-PC method can be found in [61].

In 2010, the time-dependent gPC (TD-gPC) method was proposed by Gerritsma et al. [62] to address the issue of long-time integration in the gPC method. This was motivated by the fact that the probability distribution of the solution changes with time, which in turn requires that the random basis (of the solution space) is frequently updated during the simulation to ensure that the mean-square error is kept orthogonal to the discretized random function space. To keep the computational cost low, the random basis is adaptively updated whenever a preset threshold value is met during the simulation. Whenever this threshold value is met, a new set of orthogonal polynomials is generated from the monomials of the system’s state for use in subsequent time steps of the simulation. Heuveline and Schick [63] modified the TD-gPC method (mTD-gPC) to account for stochastic dynamical systems governed by second-order ODEs, and in doing so they also improved the accuracy of the method. In mTD-gPC, the stochastic part of the solution space is spanned (at the *reset times*) by performing a full tensor product between an evolving random function space (that depends upon the evolution of the system’s state) and the original random function space (which is spanned according to the gPC method). However, since both TD-gPC and mTD-gPC use tensor products to construct a suitable random basis, they both suffer from the curse of dimensionality because the number of basis vectors in both these approaches grows considerably fast with the dimensionality of the probability space (and sometimes this growth may be exponential if not addressed well). Heuveline and Schick [63] also developed a multi-element version of the mTD-gPC method called the *hybrid generalized polynomial chaos* as a means to keep the dimensionality of the random function space relatively low on each random element.

More recently, Luchtenburg et al. [64] developed a method for long-time uncertainty propagation in dynamical systems. The method consists of approximating the intermediate short-time flow maps by spectral polynomial bases, so that the system’s long-time flow map is constructed via a flow map composition. These short-time flow maps are represented by low-degree polynomial bases to account for the stretching and folding effect caused by the evolution of the system’s state in phase space. Ozen and Bal [65] introduced the dynamical gPC (DgPC) method to quantify uncertainties in the long-time response of stochastic dynamical systems. The method uses a generalization of the PCE (Polynomial Chaos Expansion) framework to construct a set of orthogonal polynomials from measures that evolve dynamically in time. They demonstrated that results obtained with DgPC compare well with other standard methods such as Monte Carlo. However, the method has limited applicability for large stochastic dynamical systems.

In this paper, a novel method called the *flow-driven spectral chaos* (FSC) is proposed to capture the long-time response of stochastic dynamical systems. The FSC method uses the concept of *enriched* stochastic flow maps to track the evolution of a finite-dimensional random function space efficiently in time. In this approach, the enriched stochastic flow map of the system is by definition a flow map that pushes forward the first few time derivatives of the solution (including the solution itself) in an augmented random phase space. Unlike mTD-gPC (or gPC), the number of basis vectors needed to construct the orthogonal bases in FSC does not grow with the dimensionality of the probability space. Therefore, the FSC method does not suffer from the curse of dimensionality at the random-function-space level. However, as with all spectral-based methods, it does suffer from the curse of dimensionality at the random-space level, because the number of quadrature points needed to compute the inner products accurately can grow exponentially with the dimensionality of the probability space. Nevertheless, as we show in Section 9, when the FSC method is used in conjunction with Monte Carlo integration to compute the inner products, the curse of dimensionality can be eliminated altogether. Thus, the FSC method presents a major advance over gPC-based methods since for the same level of accuracy in the solution it is computationally far more efficient.

This paper is organized as follows. Section 2 introduces the setting and notation used in this manuscript, and then a quick overview of the standard gPC method is provided. Section 3 discusses the Gram-Schmidt process for random function spaces and also outlines a *new theorem* that has been developed for orthog-

onalyzing a sequence of independent random functions. This theorem is then utilized in the FSC scheme (Section 6) to transfer the probability information of the system's state *exactly* at the current time of the simulation. Section 4 reviews the concept of stochastic flow map, followed by the definition of *enriched* stochastic flow map in Section 5. In Section 6 we describe the proposed FSC method in detail using two different approaches (FSC-1 and FSC-2) for the transfer of the probability information. Six numerical examples are then presented in Section 7, followed by a discussion of the numerical results in Section 8. In Section 9 we solve a parametric, high-dimensional stochastic problem to demonstrate (from a numerical standpoint) that using the FSC method, in conjunction with Monte Carlo integration to compute the inner products, it is possible to overcome the curse of dimensionality at both the random-function-space level and the random-space level—thus eliminating it altogether. In Appendices A and B we present in detail the discretization of the two random function spaces needed to simulate the stochasticity of a Van-der-Pol oscillator and the system described in Section 9 using the spectral approach. Finally, Appendix C presents a comparison between the time-complexity analyses of our new theorem and the traditional Gram-Schmidt process in order to assess the computational cost of both approaches algebraically.

## 2. Setting and notation

*Spaces.* The spaces that we use in this work are defined below.

**Definition 1** (Temporal space). Let the topological space  $(\mathfrak{T}, \mathcal{O})$  be called *temporal space*, where  $\mathfrak{T} = [0, T]$  is a closed interval representing the temporal domain of the system,  $T$  is a positive real number symbolizing the duration of the simulation, and  $\mathcal{O} = \mathcal{O}_{\mathbb{R}} \cap \mathfrak{T}$  is the topology on  $\mathfrak{T}$  with  $\mathcal{O}_{\mathbb{R}}$  denoting the standard topology over  $\mathbb{R}$ .

*Remark 1.* Although this temporal space can be specialized further to be a Hilbert space, in this manuscript we only need the topological structure of it to assist Definition 3 in regard to continuity of functions in time. In simple terms, this temporal space defines the time interval of interest for running the stochastic simulations.

**Definition 2** (Random space). Let  $(\Omega, \mathbf{\Omega}, \lambda)$  be a (complete) *probability space*, where  $\Omega$  is the sample space,  $\mathbf{\Omega} \subset 2^{\Omega}$  is the  $\sigma$ -algebra on  $\Omega$  (aka the collection of events), and  $\lambda : \mathbf{\Omega} \rightarrow [0, 1]$  is the probability measure on  $\mathbf{\Omega}$ . Let  $\xi : (\Omega, \mathbf{\Omega}) \rightarrow (\mathbb{R}^d, \mathcal{B}_{\mathbb{R}^d})$  be a measurable function (aka random variable) given by  $\xi = \xi(\omega)$ , with  $\mathcal{B}_{\mathbb{R}^d}$  denoting the Borel  $\sigma$ -algebra over  $\mathbb{R}^d$ . Furthermore, let the measure space  $(\Xi, \mathbf{\Xi}, \mu)$  be called *random space*, where  $\Xi = \xi(\Omega) \subset \mathbb{R}^d$  is a set representing the random domain of the system,  $\mathbf{\Xi} = \mathcal{B}_{\mathbb{R}^d} \cap \Xi$  is the  $\sigma$ -algebra on  $\Xi$ , and  $\mu : \mathbf{\Xi} \rightarrow [0, 1]$  is the probability measure on  $\mathbf{\Xi}$  defined by the pushforward of  $\lambda$  by  $\xi$ , that is  $\mu = \xi_*(\lambda)$ . Here  $d$  symbolizes the dimensionality of the random space.

*Remark 2.* In addition to the standard definition of a ‘probability space’, we define a ‘random space’ in Definition 2 to address cases where the probability space may be abstract. The random variable  $\xi$  relates these two spaces and aids in computation.

From these two definitions it is clear that more structure can be added to these spaces; for example, a metric, a norm, an inner product, etc. However, we opt not to do so herein to keep the above definitions as simple as possible, and more importantly, because they are not needed in this manuscript.

**Definition 3** (Temporal function space). Let  $\mathcal{T}(n) = C^n(\mathfrak{T}, \mathcal{O}; \mathbb{R})$  be a continuous  $n$ -differentiable function space. This *temporal function space* is the space of all functions  $f : (\mathfrak{T}, \mathcal{O}) \rightarrow (\mathbb{R}, \mathcal{O}_{\mathbb{R}})$  that have continuous first  $n$  derivatives on  $(\mathfrak{T}, \mathcal{O})$ .

*Remark 3.* The temporal function space is thus defined to indicate the level of differentiability that some temporal functions need to possess in Sections 4 and 5—especially those concerning with the state of the dynamical system under consideration.

**Definition 4** (Random function space). Let  $\mathcal{X} = (L^2(\Xi, \mathbf{\Xi}, \mu; \mathbb{R}), \langle \cdot, \cdot \rangle)$  be a Lebesgue square-integrable space equipped with its standard inner product  $\langle \cdot, \cdot \rangle : L^2(\Xi, \mathbf{\Xi}, \mu; \mathbb{R}) \times L^2(\Xi, \mathbf{\Xi}, \mu; \mathbb{R}) \rightarrow \mathbb{R}$  given by  $\langle f, g \rangle =$

$\int fg d\mu$ . This *random function space* (aka RFS in this manuscript) is the space of all (equivalence classes of) measurable functions  $f : (\Xi, \Xi) \rightarrow (\mathbb{R}, \mathcal{B}_{\mathbb{R}})$  that are square-integrable with respect to  $\mu$ . This space is known to form a Hilbert space because it is complete under the metric induced by the inner product. In addition, let  $\{\Psi_j : (\Xi, \Xi) \rightarrow (\mathbb{R}, \mathcal{B}_{\mathbb{R}})\}_{j=0}^{\infty}$  be a complete orthogonal basis in  $\mathcal{X}$ , such that  $\Psi_0(\xi) = 1$  for all  $\xi \in \Xi$ .

*Remark 4.* In the literature, the ‘random function space’ just defined is also called ‘random space’ to simplify the terminology of the space. However, in this work, a distinction between the two spaces is needed. We claim that the FSC method *does not* suffer from the curse of dimensionality at the *random-function-space* level, because the number of basis vectors that we use to span  $\mathcal{X}$  does not depend upon the dimensionality of the random space—in contrast to other spectral methods such as gPC, TD-gPC, etc. which use tensor products to construct  $\mathcal{X}$ . However, as with all spectral methods the FSC method *does* suffer from the curse of dimensionality at the *random-space* level, because we still have the issue that the higher the dimensionality of the random space is, the more difficult is to compute the inner products accurately. We emphasize, however, that this is still an open area of research and that there are several numerical techniques available in the literature that deal with this issue, e.g. [66–69]. In Section 9, for example, we show that Monte Carlo integration can be used to address the curse of dimensionality at the random-space level, and that together with the FSC method, it can eliminate the curse of dimensionality of the proposed spectral approach at both random levels.

From Definition 4 it follows that any function  $f \in \mathcal{X}$  can be represented in a Fourier series of the form:

$$f = \sum_{j=0}^{\infty} f^j \Psi_j,$$

where  $f^j$  denotes the  $j$ -th coefficient of the series with the superscript not denoting an exponentiation.

Moreover, let  $\Upsilon_{ij} = \langle \Psi_i, \Psi_j \rangle$  be the  $(i, j)$ -th component of the inner-product tensor associated with the chosen orthogonal basis in  $\mathcal{X}$ . Then, because of the orthogonality property of the basis and the selection of the first basis vector to be identically equal to one ( $\Psi_0 \equiv 1$ ), one obtains:

$$\Upsilon_{ij} = \langle \Psi_i, \Psi_j \rangle \delta_{ij} = \begin{cases} 1 & \text{for } i = j = 0 \\ \langle \Psi_i, \Psi_i \rangle & \text{for } i = j \text{ with } i, j > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Following the notation and conventions of multilinear and tensor algebra, we note that  $\Upsilon = \Upsilon_{ij} \Psi^i \otimes \Psi^j : \mathcal{X}^2 \rightarrow \mathbb{R}$  is a symmetric tensor of type  $(0, 2)$  given by  $\Upsilon[f, g] = \Upsilon_{ij} f^i g^j$ , where  $\Psi^i : \mathcal{X} \rightarrow \mathbb{R}$  is the  $i$ -th dual basis vector in  $\mathcal{X}'$  defined by

$$\Psi^i[h] := [\Psi^i, h] = \frac{\langle \Psi_i, h \rangle}{\langle \Psi_i, \Psi_i \rangle} \equiv h^i.$$

Here  $[\cdot, \cdot] : \mathcal{X}' \times \mathcal{X} \rightarrow \mathbb{R}$  represents the dual pairing between  $\mathcal{X}$  and  $\mathcal{X}'$  satisfying the property:  $[\Psi^i, \Psi_j] = \delta^i_j$  with  $\delta^i_j$  denoting the Kronecker delta. The second equality follows from the Riesz representation theorem [70], which means that the map  $\Psi^i \mapsto \Psi_i / \langle \Psi_i, \Psi_i \rangle$  is an isometric isomorphism between  $\mathcal{X}'$  and  $\mathcal{X}$ .

**Definition 5** (Solution space and root space). Let  $\mathcal{U} = \mathcal{T}(n) \otimes \mathcal{X}$  and  $\mathcal{V} = \mathcal{T}(0) \otimes \mathcal{X}$  be the *solution space* and the *root space* of the system, respectively. Then, as a result of these definitions, we have:  $\mathcal{X} \subset \mathcal{U} \subset \mathcal{V}$ .

*Remark 5.* These two definitions are used below to relate the solution space and the root space via the partial differential operator  $\mathcal{L}$ . They are also used in the manuscript to simplify the notation of these spaces.

Throughout this paper, we assume that the components of the  $d$ -tuple random variable  $\xi = (\xi^1, \dots, \xi^d)$  are mutually independent and that the random domain  $\Xi$  is a hypercube of  $d$  dimensions obtained by performing a  $d$ -fold Cartesian product of intervals  $\bar{\Xi}_i := \xi^i(\Omega)$ . Letting  $\mu^i(d\xi^i) =: d\mu^i$  denote the probability measure of  $d\xi^i$  around  $\xi^i \in \bar{\Xi}_i$ , one can then define the measure in  $\mathcal{X}$  by

$$\mu = \bigotimes_{i=1}^d \mu^i, \quad \text{or equivalently,} \quad d\mu \equiv \mu(d\xi) = \prod_{i=1}^d \mu^i(d\xi^i) \equiv d\mu^1 \cdots d\mu^d.$$

*Problem statement.* In this work, we consider the following stochastic problem (assumed well-posed).

Find the real-valued stochastic process  $u : \mathfrak{T} \times \Xi \rightarrow \mathbb{R}$  in  $\mathcal{U}$ , such that ( $\mu$ -a.e.):

$$\mathcal{L}[u] = f \quad \text{on } \mathfrak{T} \times \Xi \quad (1a)$$

$$\{\mathcal{B}_k[u](0, \cdot) = b_k\}_{k=1}^n \quad \text{on } \{0\} \times \Xi, \quad (1b)$$

where  $\mathcal{L} : \mathcal{U} \rightarrow \mathcal{V}$  is a partial differential operator of order  $(n, 0)$ ,  $\mathcal{B}_k[\cdot](0, \cdot) : \mathcal{U} \rightarrow \mathcal{X}$  is a partial differential operator of order  $(n-1, 0)$  that upon differentiation evaluates the resulting function at  $t = 0$ ,  $f : \mathfrak{T} \times \Xi \rightarrow \mathbb{R}$  is a function in  $\mathcal{V}$  given by  $f = f(t, \xi)$ , and  $b_k : \Xi \rightarrow \mathbb{R}$  is a function in  $\mathcal{X}$  given by  $b_k = b_k(\xi)$ .

The operators  $\mathcal{L}$  and  $\mathcal{B}_k$  take differentiations only in time and can be, in general, nonlinear. For the case when  $(n, d) = (2, 3)$  and  $\mathcal{L}$  and  $\mathcal{B}_k$  are linear operators, we get:  $\xi = (\xi^1, \xi^2, \xi^3)$  and

$$\begin{aligned} \mathcal{L}[u](t, \xi) &= a_2(t, \xi) \ddot{u}(t, \xi) + a_1(t, \xi) \dot{u}(t, \xi) + a_0(t, \xi) u(t, \xi) \\ \mathcal{B}_1[u](0, \xi) &= b_{11}(\xi) \dot{u}(0, \xi) + b_{10}(\xi) u(0, \xi) \\ \mathcal{B}_2[u](0, \xi) &= b_{21}(\xi) \dot{u}(0, \xi) + b_{20}(\xi) u(0, \xi), \end{aligned}$$

where  $a_0, a_1, a_2 \in \mathcal{V}$  with  $a_2 \neq 0$ , and  $b_{10}, b_{11}, b_{20}, b_{21} \in \mathcal{X}$  such that  $b_{10}b_{21} - b_{11}b_{20} \neq 0$ . Observe that in these expressions,  $\dot{u} := \partial_t u$  and  $\ddot{u} := \partial_t^2 u$  denote the first and second partial derivatives of  $u$  with respect to time. This example also shows that, for the more general case, the stochasticity of the system can enter via the operators  $\mathcal{L}$  and  $\mathcal{B}_k$ , and the source functions  $f$  and  $b_k$ .

Because  $u$  is already assumed to be an element of  $\mathcal{U}$  in (1), the stochastic systems that we are interested in are those whose underlying process is of second-order only. A stochastic process  $u$  is said to be of second-order if its second moment is finite, or equivalently, if  $u(t, \cdot) \in \mathcal{X}$  for all  $t \in \mathfrak{T}$ .

In this sense, since  $u \in \mathcal{U}$ , it can be represented by the Fourier series:

$$u(t, \xi) = \sum_{j=0}^{\infty} u^j(t) \Psi_j(\xi), \quad (2)$$

where  $u^j$  is a temporal function in  $\mathcal{T}(n)$  denoting the  $j$ -th random mode of  $u$ . This series, usually referred to as *stochastic spectral expansion* in the literature [18, 19], will be used herein as the *solution representation* of the underlying process to seek.

It is worth mentioning that if we demand  $u$  to be sufficiently smooth in the solution space, especially in  $\mathcal{X}$ , the expansion given by (2) will lead to exponential convergence to the solution, since  $\{\Psi_j\}_{j=0}^{\infty}$  is an orthogonal basis with respect to the probability measure  $\mu$  in  $\mathcal{X}$ . This particular selection of the basis for the underlying process is known as the *optimal basis*, and it can be obtained by using any orthogonalization technique such as the Gram-Schmidt process [71].

A system governed by (1) can also be expressed in modeling notation as

$$y = \mathcal{M}[u][x] \quad \text{subject to initial condition} \quad \mathcal{I}[u], \quad (1^*)$$

where  $\mathcal{M}[u] : \mathcal{V}^r \rightarrow \mathcal{V}^s$  represents the mathematical model of the system defined by (1a),  $x = (x_1, \dots, x_r) : \mathfrak{T} \times \Xi \rightarrow \mathbb{R}^r$  is the  $r$ -tuple input of  $\mathcal{M}[u]$ , and  $y = (y_1, \dots, y_s) : \mathfrak{T} \times \Xi \rightarrow \mathbb{R}^s$  is the  $s$ -tuple output of  $\mathcal{M}[u]$  (aka the  $s$ -tuple observable in physics or the  $s$ -tuple response in engineering). In addition,  $\mathcal{I}[u]$  represents the initial condition for  $\mathcal{M}[u]$  which is given by (1b). The objective of this mathematical model is to propagate and quantify the effects of input uncertainty  $x$  on system's output  $y$ . Note that here  $x$  is to be understood as the model's input and not necessarily as the system's input. Therefore, the components of  $x$  might not only include the source function  $f$  in (1) but also the coefficients of operator  $\mathcal{L}$ .

*Discretization of random function space (standard gPC method).* For this work, let us simply consider a  $p$ -discretization of the random function space  $\mathcal{X}$  as follows. Let  $\mathcal{X}^{[P]} = \text{span}\{\Psi_j\}_{j=0}^P$  be a finite subspace of  $\mathcal{X}$  with  $P+1 \in \mathbb{N}_1$  denoting the dimensionality of the subspace, and let  $u^{[P]}(t, \cdot)$  be an element of  $\mathcal{X}^{[P]}$ .

Then, from (2) it follows that:

$$u(t, \xi) \approx u^{[P]}(t, \xi) = \sum_{j=0}^P u^j(t) \Psi_j(\xi), \quad (3)$$

provided that  $\{\Psi_j\}_{j=0}^\infty$  is well-graded to carry out the approximation of  $u$  this way.

If  $d$  denotes the dimensionality of the random space, and  $p$  is the maximal order polynomial in  $\{\Psi_j\}_{j=0}^P$ , then the total number of terms that we obtain after expanding (3) can be determined as

$$P + 1 = \binom{d+p}{p} = \frac{(d+p)!}{d!p!}. \quad (4)$$

This expression shows that the total number of terms used in (3) grows combinatorially fast as a function of  $d$  and  $p$ , and thus, it suffers to some extent from the *curse of dimensionality*. In practice, the usefulness of representing the solution with such a construction (i.e. by means of a *total-order tensor product*) is limited for problems where  $d$  and  $p$  are less than 10 or so [19]. For higher-dimensional spaces, more general *sparse tensor products* can be utilized to help alleviate better the curse of dimensionality, e.g. by means of *Smolyak-based tensor products*. However, for low dimensional spaces, *full tensor products* can still be used whenever  $d$  is 2 or 3. In full tensor products, the total number of terms increases exponentially fast as a function of  $d$  and  $p$ . That is,  $P + 1 = (p + 1)^d$ .

*Remark 6.* An orthogonal basis in  $\mathfrak{X}^{[P]}$  can be constructed as products of univariate orthogonal polynomials in the following way. Let  $\{\Psi_j^{(i)} : \Xi_i \rightarrow \mathbb{R}\}_{j=0}^\infty$  be an orthogonal basis with respect to  $\mu^i$ , where  $i \in \{1, 2, \dots, d\}$ . These bases are usually chosen to be univariate polynomials along the  $i$ -th dimension satisfying the condition that  $\Psi_0^{(i)}(\xi^i) = 1$  for all  $\xi^i \in \Xi_i$ . Then, one defines:

$$\Psi_j \equiv \Psi_{\pi(k)} := \bigotimes_{i=1}^d \Psi_{k_i}^{(i)} = \Psi_{k_1}^{(1)} \otimes \dots \otimes \Psi_{k_d}^{(d)}, \quad (5)$$

where  $k = (k_1, \dots, k_d) \in \mathbb{N}_0^d$  is a multi-index with  $|k| = k_1 + \dots + k_d$ , and  $\Psi_{\pi(k)}$  is a function given by

$$\Psi_{\pi(k)}(\xi) = \prod_{i=1}^d \Psi_{k_i}^{(i)}(\xi^i) = \Psi_{k_1}^{(1)}(\xi^1) \dots \Psi_{k_d}^{(d)}(\xi^d).$$

In these expressions,  $\pi : \mathbb{N}_0^d \rightarrow \mathbb{N}_0$  is an ordering map that sorts the elements in ascending order based on the multi-index degree  $|k|$ , followed by a reverse-lexicographic ordering for those elements that share the same multi-index degree. In case of resorting to a *total-order tensor product*, the condition  $|k| \leq p$  is enforced in (5) to make  $p$  (which is always taken less than  $\max |k|$ ) be the maximal order polynomial in  $\{\Psi_j\}_{j=0}^P$ .

*Remark 7.* In Section 6 we will see that in our FSC method the basis  $\{\Psi_j\}_{j=0}^P$  is not constructed by performing a tensor product like in Remark 6, but by recurring instead to the time derivatives of the solution itself. This is in contrast to the standard TD-gPC method [62], which uses tensor products to construct a basis based on the monomials of the solution, and for which it is known suffers from the curse of dimensionality.

For notational convenience, expansion (3) will simply be written hereafter as

$$u(t, \xi) = \sum_j u^j(t) \Psi_j(\xi), \quad (6)$$

where a summation sign is implied over the repeated index  $j$ , and  $j \in \{0, 1, \dots, P\}$  unless indicated otherwise. Observe that the superscript  $^{[P]}$  in  $u$  was dropped to avoid unnecessary complexity in notation.

Substituting (6) into (1) gives

$$\mathcal{L}[u^j \Psi_j] = f \quad \text{on } \mathfrak{T} \times \Xi \quad (7a)$$

$$\{\mathcal{B}_k[u^j \Psi_j](0, \cdot) = b_k\}_{k=1}^n \quad \text{on } \{0\} \times \Xi. \quad (7b)$$

Projecting (7) onto  $\mathfrak{X}^{[P]}$  yields a system of  $P+1$  ordinary differential equations of order  $n$  in the variable  $t$ , where the unknowns are the random modes  $u^j = u^j(t)$  and their first  $n-1$  time derivatives:

$$\Psi^i[\mathcal{L}[u^j \Psi_j]] = \Psi^i[f] \quad \text{on } \mathfrak{T} \quad (8a)$$

$$\{\Psi^i[\mathcal{B}_k[u^j \Psi_j](0, \cdot)] = \Psi^i[b_k]\}_{k=1}^n \quad \text{on } \{0\} \quad (8b)$$

with  $i, j \in \{0, 1, \dots, P\}$ . This is the so-called *orthogonal projection* of (7) onto  $\mathfrak{X}^{[P]}$ , and it ensures that the mean-square error resulting from the finite representation of  $u$  using (3) is orthogonal to  $\mathfrak{X}^{[P]}$  [34].

A closer look at (8) indicates that the system of equations that we are dealing with at this point is no longer ‘stochastic’ but ‘deterministic’ since the randomness of the stochastic system has effectively been absorbed by the application of the dual vectors  $\{\Psi^i \in \mathfrak{X}'\}_{i=0}^P$ . In other words, system (8) does not depend on the tuple  $(t, \xi)$  but only on  $t$  at this stage of the analysis.

*Discretization of temporal function space.* Because (8) is a system of ordinary differential equations with initial conditions, any suitable time integration method can be used to find its solution at discrete times; giving therefore rise to an  $(h, p)$ -discretization for  $\mathcal{T}(n)$  in general.

*Probability moments.* In probability theory, the real-valued expectation,  $\mathbf{E} : \mathfrak{X} \rightarrow \mathbb{R}$ , is a linear map that outputs the expected value of a real-valued random variable, and it is given by:

$$\mathbf{E}[f] = \int f \, d\mu.$$

In contrast, the real-valued covariance,  $\text{Cov} : \mathfrak{X}^2 \rightarrow \mathbb{R}$ , is a symmetric, bilinear map that measures the joint variability of two real-valued random variables. It is defined by:

$$\text{Cov}[f, g] = \mathbf{E}[(f - \mathbf{E}[f])(g - \mathbf{E}[g])].$$

These two maps can be used as the building block to construct other maps, such as the variance of  $f$  which is defined as  $\text{Var}[f] = \text{Cov}[f, f]$ . Higher probability moments (e.g. skewness, kurtosis, etc.) are not considered in this work. However, we do so for the sake of brevity and without loss of generality, chiefly because higher probability moments are not guaranteed to exist for second-order stochastic processes.

Now, let  $z = y_k$  be the  $k$ -th component of output  $y = \mathcal{M}[u][x]$  (from (1\*)). If  $z \in \mathcal{V}$ , then it can be expanded with a polynomial chaos similar to the one set forth in (3) to obtain:

$$z(t, \xi) \approx z^{[P]}(t, \xi) = \sum_{j=0}^P z^j(t) \Psi_j(\xi) \equiv z^j(t) \Psi_j(\xi),$$

where  $P$  does not need to be the same as in (3), and the  $j$ -th random mode of  $z$  is given by:

$$z^j(t) = \frac{\langle \Psi_j, z(t, \cdot) \rangle}{\langle \Psi_j, \Psi_j \rangle}.$$

This representation of  $z$  will allow us to compute the probability moments of interest with minimal computational effort, as demonstrated below.

The expectation of  $z$ ,  $\mathbf{E}[z] : \mathfrak{T} \rightarrow \mathbb{R}$ , is easy to compute and it is given by the first random mode of  $z$ :

$$\mathbf{E}[z](t) := \int z(t, \cdot) \, d\mu = z^j(t) \int \Psi_j \, d\mu = z^j(t) \langle \Psi_j, \Psi_0 \rangle = z^0(t). \quad (9)$$

The autocovariance of  $z$ ,  $\text{Cov}[z, z] : \mathfrak{T}^2 \rightarrow \mathbb{R}$ , is defined as:

$$\begin{aligned} \text{Cov}[z, z](t, s) &:= \mathbf{E}[(z(t, \cdot) - \mathbf{E}[z](t))(z(s, \cdot) - \mathbf{E}[z](s))] \\ &= \mathbf{E}[(z^j(t) \Psi_j - z^0(t))(z^k(s) \Psi_k - z^0(s))] && \text{with } j, k \in \{0, 1, \dots, P\} \\ &= \mathbf{E}[z^j(t) z^k(s) \Psi_j \Psi_k], && \text{with } j, k \in \{1, 2, \dots, P\} \end{aligned}$$



and thus, upon further simplification we get

$$\text{Cov}[z, z](t, s) = \sum_{j=1}^P \sum_{k=1}^P z^j(t) z^k(s) \int \Psi_j \Psi_k d\mu = \sum_{j=1}^P \sum_{k=1}^P z^j(t) z^k(s) \langle \Psi_j, \Psi_k \rangle = \sum_{j=1}^P \Upsilon_{jj} z^j(t) z^j(s).$$

The variance of  $z$ ,  $\text{Var}[z] : \mathfrak{T} \rightarrow \mathbb{R}_0^+$ , is nothing but:

$$\text{Var}[z](t) := \text{Cov}[z, z](t, t) = \sum_{j=1}^P \Upsilon_{jj} z^j(t) z^j(t). \quad (10)$$

*Probability distributions.* In this paper, we employ four different probability distributions to characterize the stochasticity in the systems defined in Sections 7 and 9, namely: uniform, beta, gamma and normal. Because the measures associated with these distributions are absolutely continuous with respect to the Lebesgue measure, they possess probability density functions,  $f : \Xi \rightarrow \mathbb{R}_0^+$ , given by:

$$\begin{aligned} \text{Uniform} &\sim f(\xi) = \frac{1}{b-a} \text{ on } \Xi = [a, b], \quad \text{Beta}(\alpha, \beta) \sim f(\xi) = \frac{(\xi-a)^{\alpha-1} (b-\xi)^{\beta-1}}{(b-a)^{\alpha+\beta-1} B(\alpha, \beta)} \text{ on } \Xi = [a, b] \\ \text{Gamma}(\alpha, \beta) &\sim f(\xi) = \frac{\beta^\alpha}{\Gamma(\alpha)} (\xi-a)^{\alpha-1} \exp(-\beta(\xi-a)) \text{ on } \Xi = [a, \infty), \quad \text{and} \\ \text{Normal}(\mu, \sigma^2) &\sim f(\xi) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\xi-\mu}{\sigma}\right)^2\right] \text{ on } \Xi = \mathbb{R}. \end{aligned}$$

*Numerical integration.* The numerical integration of an inner product can be carried out at least in two different ways: using *grid-based integration* [66–68] or *Monte Carlo-based integration* [69]. The difference between the two lies in how the quadrature points are chosen from the domain of the integral. In the latter, the quadrature points are randomly sampled from the domain to seek an approximate evaluation of the integral, whereas in the former the quadrature points are selected to be the intersecting points of some predefined regular grid. It is well-known that when this grid is the Gaussian grid associated with the measure, the grid-based integration method produces the most accurate approximation of the integral.

In grid-based integration, we can either use *full grids* or *sparse grids* to perform the numerical evaluation of the integral. Using one or the other will depend on the level of accuracy we want to achieve and the computational cost we are willing to pay to estimate the numerical value of the integral. Popular sparse grids based on the work by S.A. Smolyak [72] deal with the curse of dimensionality well. However, in situations where the dimensionality of the integral domain is high, Monte Carlo is usually the preferred integration technique since the convergence rate to the sought integral is dimension independent.

For the numerical examples presented in Section 7, we use the Gaussian quadrature rule based on full grids to approximate the inner products. The reason behind this choice is that in those numerical examples, the dimensionality of the random space is at most 2. For this low-dimensional random space, we can heedlessly define a full grid in the random domain to estimate the integrals with high accuracy. Consequently, the aforementioned inner products are computed with the following expression:

$$\langle f, g \rangle := \int f g d\mu \approx \mathcal{Q}^{[Q]}[fg] := \sum_{i=1}^Q f(\xi_i) g(\xi_i) w_i,$$

where  $w_i \in \mathbb{R}^+$  is the quadrature weight associated with the Gaussian quadrature point  $\xi_i \in \Xi$ , and  $Q \in \mathbb{N}_1$  denotes the number of quadrature points involved in approximating the evaluation of the inner product. Here the quadrature points are selected from the Gaussian grid associated with the measure  $\mu$ .

*Remark 8.* We note that when  $fg$  is a sufficiently smooth integrand, we have:

$$\mathcal{Q}^{[Q]}[fg] \rightarrow \int f g d\mu \quad \text{as } Q \rightarrow \infty,$$

and if  $fg$  is a polynomial, then there exists a  $Q \in \mathbb{N}_1$  such that  $\mathcal{Q}^{[Q+j]}[fg]$  evaluates the integral exactly for all  $j \in \mathbb{N}_0$ .

Moreover, we use Monte Carlo integration to approximate the inner products that emerge from solving the parametric, high-dimensional stochastic problem described in Section 9. In this case we choose Monte Carlo integration because the dimensionality of the random space is up to 10.

### 3. The Gram-Schmidt process for random function spaces

Suppose that we have a non-orthogonal basis in  $\mathcal{X}$  given by  $\{\Phi_j\}_{j=0}^\infty$ . The objective of the *Gram-Schmidt process* is to use this basis to construct an orthogonal basis in the same space with the recursive formula:

$$\Psi_j := \Phi_j - \sum_{k=0}^{j-1} \frac{\langle \Phi_j, \Psi_k \rangle}{\langle \Psi_k, \Psi_k \rangle} \Psi_k \quad \forall j \in \mathbb{N}_0. \quad (11)$$

Recall that in Section 2 we prescribed the condition that the first basis vector is identically equal to one. This condition gives rise to the following theorem which is valid for any square-integrable function space defined on a probability space. We point out that the benefit of using this theorem is that if both the expectation vector and the covariance matrix of the non-orthogonal basis are known beforehand, the orthogonalization process can (in general) be performed faster than the traditional Gram-Schmidt process.<sup>2</sup> This is, for example, a typical situation in the area of stochastic modeling where the probability information of the stochastic input is—as often as not—known beforehand, and thence one might be interested in constructing a stochastic-input space based on the available information. This need is fulfilled by Theorem 1.

**Theorem 1.** *Let  $\mathcal{X}$  be a random function space, and let  $\{\Phi_j\}_{j=1}^\infty$  be an ordered set of linearly independent functions in  $\mathcal{X}$  such that the constant functions are not in the set. Then,  $\{\Psi_j\}_{j=0}^\infty$  is an orthogonal basis in  $\mathcal{X}$  given by:*

$$\Psi_j := \Phi_j - \mathbf{E}[\Phi_j] \Psi_0 - \sum_{k=1}^{j-1} \frac{\det \Delta_k(j)}{\det \square_k} \Psi_k \quad \text{with} \quad \Psi_0 \equiv 1, \quad (12)$$

where  $\square_k \in \mathcal{M}(k \times k, \mathbb{R})$  is the covariance matrix for the first  $k$  elements of  $\{\Phi_j\}_{j=1}^\infty$ :

$$\square_k = \begin{bmatrix} \text{Cov}[\Phi_1, \Phi_1] & \cdots & \text{Cov}[\Phi_1, \Phi_k] \\ \vdots & \ddots & \vdots \\ \text{Cov}[\Phi_k, \Phi_1] & \cdots & \text{Cov}[\Phi_k, \Phi_k] \end{bmatrix},$$

and  $\Delta_k : \{k+1, k+2, \dots\} \rightarrow \mathcal{M}(k \times k, \mathbb{R})$  is a map defined by

$$\Delta_k(j) = \begin{bmatrix} \text{Cov}[\Phi_1, \Phi_1] & \cdots & \text{Cov}[\Phi_1, \Phi_k] \\ \vdots & \ddots & \vdots \\ \text{Cov}[\Phi_{k-1}, \Phi_1] & \cdots & \text{Cov}[\Phi_{k-1}, \Phi_k] \\ \text{Cov}[\Phi_j, \Phi_1] & \cdots & \text{Cov}[\Phi_j, \Phi_k] \end{bmatrix}$$

with  $\Delta_1(j) = \text{Cov}[\Phi_j, \Phi_1]$  and

$$\Delta_2(j) = \begin{bmatrix} \text{Cov}[\Phi_1, \Phi_1] & \text{Cov}[\Phi_1, \Phi_2] \\ \text{Cov}[\Phi_j, \Phi_1] & \text{Cov}[\Phi_j, \Phi_2] \end{bmatrix}.$$

In these expressions  $k \in \mathbb{N}_1$ .

---

<sup>2</sup>Please see Appendix C for a time-complexity analysis for Theorem 1 and the traditional Gram-Schmidt process.

*Proof sketch.* The proof of this theorem follows from the Gram-Schmidt process applied to the set  $\{\Phi_j\}_{j=1}^\infty$ . To begin with, let us consider first the cases when  $j \in \{0, 1, 2, 3\}$ . Expression (12) already tells us that for  $j = 0$ ,  $\Psi_0 \equiv 1$ . It is for this reason that all constant functions are excluded from  $\{\Phi_j\}_{j=1}^\infty$ . For  $j \in \{1, 2, 3\}$ , we proceed as follows.

□ **When  $j = 1$ .** It is easy to see that (11) yields

$$\Psi_1 = \Phi_1 - \frac{\langle \Phi_1, \Psi_0 \rangle}{\langle \Psi_0, \Psi_0 \rangle} \Psi_0 = \Phi_1 - \mathbf{E}[\Phi_1] \Psi_0, \quad (13)$$

because  $\langle \Phi_1, \Psi_0 \rangle = \mathbf{E}[\Phi_1]$  and  $\langle \Psi_0, \Psi_0 \rangle = 1$ .

□ **When  $j = 2$ .** From (11) we get

$$\Psi_2 = \Phi_2 - \frac{\langle \Phi_2, \Psi_0 \rangle}{\langle \Psi_0, \Psi_0 \rangle} \Psi_0 - \frac{\langle \Phi_2, \Psi_1 \rangle}{\langle \Psi_1, \Psi_1 \rangle} \Psi_1. \quad (14)$$

Replacing (13) into (14) gives

$$\Psi_2 = \Phi_2 - \mathbf{E}[\Phi_2] \Psi_0 - \frac{\text{Cov}[\Phi_1, \Phi_2]}{\text{Cov}[\Phi_1, \Phi_1]} \Psi_1 = \Phi_2 - \mathbf{E}[\Phi_2] \Psi_0 - \frac{\det \Delta_1(2)}{\det \square_1} \Psi_1,$$

after noting that  $\langle \Phi_2, \Phi_1 - \mathbf{E}[\Phi_1] \Psi_0 \rangle$  simplifies to  $\text{Cov}[\Phi_1, \Phi_2]$ , and  $\langle \Phi_1 - \mathbf{E}[\Phi_1] \Psi_0, \Phi_1 - \mathbf{E}[\Phi_1] \Psi_0 \rangle$  is nothing but  $\text{Cov}[\Phi_1, \Phi_1]$ .

□ **When  $j = 3$ .** In a similar fashion, (11) yields

$$\begin{aligned} \Psi_3 &= \Phi_3 - \mathbf{E}[\Phi_3] \Psi_0 - \frac{\text{Cov}[\Phi_1, \Phi_3]}{\text{Cov}[\Phi_1, \Phi_1]} \Psi_1 - \frac{\text{Cov}[\Phi_1, \Phi_1] \text{Cov}[\Phi_3, \Phi_2] - \text{Cov}[\Phi_1, \Phi_2] \text{Cov}[\Phi_3, \Phi_1]}{\text{Cov}[\Phi_1, \Phi_1] \text{Cov}[\Phi_2, \Phi_2] - \text{Cov}[\Phi_1, \Phi_2] \text{Cov}[\Phi_2, \Phi_1]} \Psi_2 \\ &= \Phi_3 - \mathbf{E}[\Phi_3] \Psi_0 - \frac{\det \Delta_1(3)}{\det \square_1} \Psi_1 - \frac{\det \Delta_2(3)}{\det \square_2} \Psi_2. \end{aligned}$$

The following formulas can be derived algebraically using mathematical induction. For brevity, we only provide an insight into how this can be done.

□ **Formula for  $\langle \Psi_k, \Psi_k \rangle$ .** Consider, for example, the case when  $k = 2$ :

$$\langle \Psi_2, \Psi_2 \rangle = \mathbf{E}[(\Psi_2)^2] = \mathbf{E} \left[ \left( \Phi_2 - \mathbf{E}[\Phi_2] \Psi_0 - \frac{\text{Cov}[\Phi_1, \Phi_2]}{\text{Cov}[\Phi_1, \Phi_1]} \Psi_1 \right)^2 \right]. \quad (15)$$

Substituting (13) into (15) yields

$$\langle \Psi_2, \Psi_2 \rangle = \frac{\text{Cov}[\Phi_1, \Phi_1] \text{Cov}[\Phi_2, \Phi_2] - \text{Cov}[\Phi_1, \Phi_2] \text{Cov}[\Phi_2, \Phi_1]}{\text{Cov}[\Phi_1, \Phi_1]} = \frac{\det \square_2}{\det \square_1}.$$

In general, it is possible to show that the formula for  $\langle \Psi_k, \Psi_k \rangle$  is given by:

$$\langle \Psi_k, \Psi_k \rangle = \frac{\det \square_k}{\det \square_{k-1}} \quad \forall k \in \mathbb{N}_1, \quad (16)$$

where we have set:  $\det \square_0 = 1$ .

□ **Formula for  $\langle \Phi_j, \Psi_k \rangle$ .** As before, let us consider the case when  $k = 2$  for the sake of illustration:

$$\langle \Phi_j, \Psi_2 \rangle = \left\langle \Phi_j, \Phi_2 - \mathbf{E}[\Phi_2] \Psi_0 - \frac{\text{Cov}[\Phi_1, \Phi_2]}{\text{Cov}[\Phi_1, \Phi_1]} \Psi_1 \right\rangle. \quad (17)$$

Replacing (13) into (17) gives

$$\langle \Phi_j, \Psi_2 \rangle = \frac{\text{Cov}[\Phi_1, \Phi_1] \text{Cov}[\Phi_j, \Phi_2] - \text{Cov}[\Phi_1, \Phi_2] \text{Cov}[\Phi_j, \Phi_1]}{\text{Cov}[\Phi_1, \Phi_1]} = \frac{\det \Delta_2(j)}{\det \square_1}.$$

In general, the formula for  $\langle \Phi_j, \Psi_k \rangle$  is given by:

$$\langle \Phi_j, \Psi_k \rangle = \frac{\det \Delta_k(j)}{\det \square_{k-1}} \quad \forall k \in \mathbb{N}_1, \quad (18)$$

where once again we have set:  $\det \square_0 = 1$ . Here,  $j \in \{k+1, k+2, \dots\}$  by definition of  $\Delta_k$ .

Therefore, substituting (16) and (18) into (11) yields expression (12).  $\square$

This theorem will be implemented in Section 6 to transfer the probability information exactly at the current time of the simulation.

**Example 1.** Let us consider a one-dimensional random space with domain  $\Xi = [-1, 1]$  and measure  $\mu$  given by  $d\mu(\xi) = \frac{3}{2}\xi^2 d\xi$ , and suppose that we are interested in finding an orthogonal basis in one of the infinitely many  $\mathcal{X}^{[5]}$  spaces that we can construct by applying Theorem 1 over a set of 5 monomials in the variable  $\xi$ . To this end, let  $\{\Phi_j(\xi) = \xi^j\}_{j=1}^5$  be this set. Then, its expectation vector and covariance matrix are:

$$[\mathbf{E}[\Phi_j]] = \begin{bmatrix} 0 \\ \frac{3}{5} \\ 0 \\ \frac{3}{7} \\ 0 \end{bmatrix}, \quad [\text{Cov}[\Phi_i, \Phi_j]] = \begin{bmatrix} \text{Cov}[\Phi_1, \Phi_1] & \cdots & \text{Cov}[\Phi_1, \Phi_5] \\ \vdots & \ddots & \vdots \\ \text{Cov}[\Phi_5, \Phi_1] & \cdots & \text{Cov}[\Phi_5, \Phi_5] \end{bmatrix} = \begin{bmatrix} \frac{3}{5} & 0 & \frac{3}{7} & 0 & \frac{1}{3} \\ 0 & \frac{12}{175} & 0 & \frac{8}{105} & 0 \\ \frac{3}{7} & 0 & \frac{1}{3} & 0 & \frac{3}{11} \\ 0 & \frac{8}{105} & 0 & \frac{48}{539} & 0 \\ \frac{1}{3} & 0 & \frac{3}{11} & 0 & \frac{3}{13} \end{bmatrix}.$$

From (12) it follows that an orthogonal basis in the chosen  $\mathcal{X}^{[5]}$  space is given by:

$$\begin{aligned} \Psi_0(\xi) &= 1, & \Psi_1(\xi) &= \xi, & \Psi_2(\xi) &= \xi^2 - \frac{3}{5}, & \Psi_3(\xi) &= \xi^3 - \frac{5}{7}\xi, \\ \Psi_4(\xi) &= \xi^4 - \frac{10}{9}\xi^2 + \frac{5}{21}, & \Psi_5(\xi) &= \xi^5 - \frac{14}{11}\xi^3 + \frac{35}{99}\xi. \end{aligned}$$

By way of illustration, the determinant ratios that appear in (12) were computed with expressions such as this:

$$\frac{\det \Delta_2(4)}{\det \square_2} = \frac{\begin{vmatrix} \text{Cov}[\Phi_1, \Phi_1] & \text{Cov}[\Phi_1, \Phi_2] \\ \text{Cov}[\Phi_4, \Phi_1] & \text{Cov}[\Phi_4, \Phi_2] \end{vmatrix}}{\begin{vmatrix} \text{Cov}[\Phi_1, \Phi_1] & \text{Cov}[\Phi_1, \Phi_2] \\ \text{Cov}[\Phi_2, \Phi_1] & \text{Cov}[\Phi_2, \Phi_2] \end{vmatrix}} = \frac{\begin{vmatrix} \frac{3}{5} & 0 \\ 0 & \frac{8}{105} \end{vmatrix}}{\begin{vmatrix} \frac{3}{5} & 0 \\ 0 & \frac{12}{175} \end{vmatrix}} = \frac{10}{9}.$$

#### 4. Stochastic flow map

Provided sufficient regularity, a stochastic system governed by (1) can be expressed in explicit form as

$$\partial_t^n u(t, \xi) = \ell(t, \xi, s(t, \xi)) \quad \text{on } \mathfrak{T} \times \Xi \quad (19a)$$

$$\{\partial_t^{k-1} u(0, \xi) = c_k(\xi)\}_{k=1}^n \quad \text{on } \{0\} \times \Xi, \quad (19b)$$

where  $s = (u, \partial_t u, \dots, \partial_t^{n-1} u) \in \prod_{j=1}^n \mathcal{T}(n-j+1) \otimes \mathcal{X}$  is the configuration state of the system over  $\mathfrak{T} \times \Xi$ ,  $\ell : \mathfrak{T} \times \Xi \times \mathbb{R}^n \rightarrow \mathbb{R}$  is a noisy, non-autonomous function (which can also be regarded as a function in  $\mathcal{V}$ ) concordant with (1a), and  $c_k : \Xi \rightarrow \mathbb{R}$  is a function in  $\mathcal{X}$  concordant with (1b).

If the solution is analytic on  $\mathfrak{T}$  for all  $\xi \in \Xi$ , then it can be represented by the Taylor series:

$$u(t_i + h, \xi) = \sum_{j=0}^{\infty} \frac{h^j}{j!} \partial_t^j u(t_i, \xi),$$

where  $h := t - t_i$  is the time-step size used for the simulation around  $t_i$  (once  $t$  is fixed), and  $t_i \in \mathfrak{T}$  is the time instant of the simulation. Below we use this representation to define a *local* stochastic flow map for the state of the system under consideration. We notice, however, that to do so, we need to assume that  $u \in \mathcal{T}(n + M - 1) \otimes \mathcal{X} \subset \mathcal{U}$ , where  $M \in \mathbb{N}_1$  denotes the order of the flow map we want to implement. For most problems encountered in physics and engineering, this requirement does not represent a major drawback if  $M$  is taken relatively small.

**First-order ODE** Specializing (19) for a first-order ODE ( $n = 1$ ) yields

$$\partial_t u(t, \xi) = \ell(t, \xi, u(t, \xi)) \quad \text{on } \mathfrak{T} \times \Xi \quad (20a)$$

$$u(0, \xi) = c(\xi) \quad \text{on } \{0\} \times \Xi. \quad (20b)$$

Differentiating (20a) with respect to time three times gives:

$$\partial_t^2 u := D_t \ell = \partial_t \ell + \partial_u \ell \partial_t u \quad (21a)$$

$$\partial_t^3 u := D_t^2 \ell = \partial_t^2 \ell + 2 \partial_{tu}^2 \ell \partial_t u + \partial_u \ell \partial_t^2 u + \mathbf{g}_3 \quad (21b)$$

$$\partial_t^4 u := D_t^3 \ell = \partial_t^3 \ell + 3 \partial_{tut}^3 \ell \partial_t u + 3 \partial_{tu}^2 \ell \partial_t^2 u + \partial_u \ell \partial_t^3 u + \mathbf{g}_4, \quad (21c)$$

where  $\mathbf{g}_3 = \partial_u^2 \ell (\partial_t u)^2$ , and  $\mathbf{g}_4 = 3 \partial_{tuu}^3 \ell (\partial_t u)^2 + \partial_u^3 \ell (\partial_t u)^3 + 3 \partial_u^2 \ell \partial_t^2 u \partial_t u$ .

A stochastic flow map of order 4,  $\varphi(4) : \mathbb{R} \times \mathcal{X} \rightarrow \mathcal{X}$ , can then be given by:

$$\varphi(4)(h, s(t_i, \cdot)) := u(t_i + h, \cdot) - O(h^5) = \sum_{j=0}^4 \frac{h^j}{j!} \partial_t^j u(t_i, \cdot),$$

where the time derivatives of  $u$  at  $t = t_i$  are computed with (20a) and (21).

**Second-order ODE** Likewise, specializing (19) for a second-order ODE ( $n = 2$ ) yields

$$\partial_t^2 u(t, \xi) = \ell(t, \xi, u(t, \xi), \dot{u}(t, \xi)) \quad \text{on } \mathfrak{T} \times \Xi \quad (22a)$$

$$\{u(0, \xi) = c_1(\xi), \dot{u}(0, \xi) = c_2(\xi)\} \quad \text{on } \{0\} \times \Xi, \quad (22b)$$

where  $\dot{u} := \partial_t u$ . Differentiating (22a) with respect to time three times gives:

$$\partial_t^3 u := D_t \ell = \partial_t \ell + \partial_u \ell \partial_t u + \partial_{\dot{u}} \ell \partial_t^2 u \quad (23a)$$

$$\partial_t^4 u := D_t^2 \ell = \partial_t^2 \ell + 2 \partial_{tu}^2 \ell \partial_t u + (2 \partial_{t\dot{u}}^2 \ell + \partial_u \ell) \partial_t^2 u + \partial_{\dot{u}} \ell \partial_t^3 u + \mathbf{h}_4 \quad (23b)$$

$$\partial_t^5 u := D_t^3 \ell = \partial_t^3 \ell + 3 \partial_{tut}^3 \ell \partial_t u + 3 (\partial_{t\dot{u}}^3 \ell + \partial_{tu}^2 \ell) \partial_t^2 u + (3 \partial_{t\dot{u}}^2 \ell + \partial_u \ell) \partial_t^3 u + \partial_{\dot{u}} \ell \partial_t^4 u + \mathbf{h}_5, \quad (23c)$$

where  $\mathbf{h}_4 = \partial_u^2 \ell (\partial_t u)^2 + 2 \partial_{u\dot{u}}^2 \ell \partial_t u \partial_t^2 u + \partial_{\dot{u}}^2 \ell (\partial_t^2 u)^2$ , and

$$\begin{aligned} \mathbf{h}_5 = & 3 \partial_{tuu}^3 \ell (\partial_t u)^2 + 3 (\partial_{t\dot{u}}^3 \ell + \partial_{tu}^2 \ell) (\partial_t^2 u)^2 + \partial_u^3 \ell (\partial_t u)^3 + \partial_{\dot{u}}^3 \ell (\partial_t^2 u)^3 \\ & + 3 (2 \partial_{t\dot{u}}^3 \ell + \partial_u^2 \ell) \partial_t u \partial_t^2 u + 3 \partial_{u\dot{u}}^2 \ell \partial_t u \partial_t^3 u + 3 \partial_{\dot{u}}^2 \ell \partial_t^2 u \partial_t^3 u \\ & + 3 \partial_{u\dot{u}}^3 \ell (\partial_t u)^2 \partial_t^2 u + 3 \partial_{u\dot{u}}^3 \ell \partial_t u (\partial_t^2 u)^2. \end{aligned}$$

A stochastic flow map of order 4,  $\varphi(4) : \mathbb{R} \times \mathcal{X}^2 \rightarrow \mathcal{X}^2$ , can then be defined as:

$$\varphi(4)(h, s(t_i, \cdot)) := (u(t_i + h, \cdot), \dot{u}(t_i + h, \cdot)) - O(h^5) = \left( \sum_{j=0}^4 \frac{h^j}{j!} \partial_t^j u(t_i, \cdot), \sum_{j=0}^4 \frac{h^j}{j!} \partial_t^{j+1} u(t_i, \cdot) \right), \quad (24)$$

where the second and higher time derivatives of  $u$  at  $t = t_i$  are computed with (22a) and (23).

*Remark 9.* In these expressions we observe that if (19) is a linear ODE, then  $\mathbf{g}_3 = \mathbf{g}_4 = \mathbf{h}_4 = \mathbf{h}_5 \equiv 0$ . If in addition it is autonomous, expressions (21) and (23) reduce, respectively, to:

$$\partial_t^2 u = \partial_u \ell \partial_t u, \quad \partial_t^3 u = \partial_u \ell \partial_t^2 u \quad \text{and} \quad \partial_t^4 u = \partial_u \ell \partial_t^3 u. \quad (21^*)$$

$$\partial_t^3 u = \partial_u \ell \partial_t u + \partial_u \ell \partial_t^2 u, \quad \partial_t^4 u = \partial_u \ell \partial_t^2 u + \partial_u \ell \partial_t^3 u \quad \text{and} \quad \partial_t^5 u = \partial_u \ell \partial_t^3 u + \partial_u \ell \partial_t^4 u. \quad (23^*)$$

**High-order ODE** In general, a stochastic flow map of order  $M$ ,  $\varphi(M) : \mathbb{R} \times \mathcal{X}^n \rightarrow \mathcal{X}^n$ , can be taken as:

$$\varphi(M)(h, s(t_i, \cdot)) := (u(t_i + h, \cdot), \dots, \partial_t^{k-1} u(t_i + h, \cdot), \dots, \partial_t^{n-1} u(t_i + h, \cdot)) - O(h^{M+1}), \quad (25)$$

where its  $k$ -th component,  $\varphi^k(M) : \mathbb{R} \times \mathcal{X}^n \rightarrow \mathcal{X}$ , is given by:

$$\varphi^k(M)(h, s(t_i, \cdot)) := \partial_t^{k-1} u(t_i + h, \cdot) - O(h^{M+1}) = \sum_{j=0}^M \frac{h^j}{j!} \partial_t^{j+k-1} u(t_i, \cdot)$$

with  $k \in \{1, 2, \dots, n\}$ .

In expression (25), the  $n$ -th time derivative of  $u$  at  $t = t_i$  is computed with (19a), and if  $M = 4$ , then the next time derivatives of  $u$  are given by:

$$\partial_t^{n+1} u := \frac{d\ell}{dt} = \frac{\partial \ell}{\partial t} + \frac{\partial \ell}{\partial s^k} \partial_t^k u, \quad (26a)$$

$$\partial_t^{n+2} u := \frac{d^2 \ell}{dt^2} = \frac{\partial^2 \ell}{\partial t^2} + 2 \frac{\partial^2 \ell}{\partial t \partial s^k} \partial_t^k u + \frac{\partial \ell}{\partial s^k} \partial_t^{k+1} u + \frac{\partial^2 \ell}{\partial s^k \partial s^l} \partial_t^k u \partial_t^l u, \quad (26b)$$

and

$$\begin{aligned} \partial_t^{n+3} u := \frac{d^3 \ell}{dt^3} = & \frac{\partial^3 \ell}{\partial t^3} + 3 \frac{\partial^3 \ell}{\partial t^2 \partial s^k} \partial_t^k u + 3 \frac{\partial^2 \ell}{\partial t \partial s^k} \partial_t^{k+1} u + \frac{\partial \ell}{\partial s^k} \partial_t^{k+2} u \\ & + 3 \frac{\partial^3 \ell}{\partial t \partial s^k \partial s^l} \partial_t^k u \partial_t^l u + 3 \frac{\partial^2 \ell}{\partial s^k \partial s^l} \partial_t^{k+1} u \partial_t^l u + \frac{\partial^3 \ell}{\partial s^k \partial s^l \partial s^m} \partial_t^k u \partial_t^l u \partial_t^m u, \end{aligned} \quad (26c)$$

where  $s^k = \partial_t^{k-1} u$  is the  $k$ -th component of  $s$ , and a summation sign is implied over every repeated index  $k, l, m \in \{1, 2, \dots, n\}$ .

For an autonomous,  $n$ -th-order linear ODE, the first  $M - 1$  time derivatives of (19a) reduce to:

$$\partial_t^{n+m} u = \sum_{k=1}^n \frac{\partial \ell}{\partial s^k} \partial_t^{m+k-1} u \quad \forall m \in \{1, 2, \dots, M-1\}.$$

It is worth noting that the goal of a (local) stochastic flow map is to push the state of the system one-time step forward in  $\mathcal{X}^n$  (i.e. in the *random phase space* of the system<sup>3</sup>), provided that the time-step size used is greater than zero. For example, Fig. 1 depicts the case of a system governed by a second-order ODE whose state motion in  $\mathcal{X}^2$  starts at  $t = 0$  and ends at  $t = T$ ; pretty much in the same way a two-dimensional, time-dependent stochastic input would evolve in  $\mathcal{X}^2$  if its map were known beforehand.

*Remark 10.* In practice, it is easier to compute the time derivatives of  $\ell$  approximately, by using any of the standard numerical methods available in the literature, such as the central difference method. When the central difference method is, say, used to compute approximately the first time derivative of  $\ell$  at  $t = t_i$ , we get:

$$\frac{d\ell}{dt}(t_i, \cdot, s(t_i, \cdot)) \approx \frac{\ell(t_i + h, \cdot, s(t_i + h, \cdot)) - \ell(t_i - h, \cdot, s(t_i - h, \cdot))}{2h}$$

for some small  $h \in \mathbb{R} \setminus \{0\}$ . Therefore, the condition that we made earlier that  $u \in \mathcal{T}(n + M - 1) \otimes \mathcal{X}$  now drops to its natural condition that  $u \in \mathcal{U}$ .

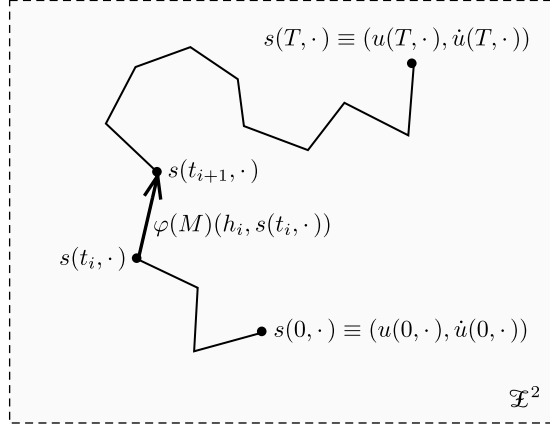


Figure 1: Evolution of a second-order dynamical system via a stochastic flow map of order  $M$  (with  $h_i > 0$ )

**Example 2.** Consider a specialization of the stochastic system given by (1) for the case when  $n = 2$ , and without loss of generality, suppose that  $\mathcal{L}$ ,  $\mathcal{B}_1$  and  $\mathcal{B}_2$  are linear operators. Then, under this setting, one obtains:

$$a_2(t, \xi) \ddot{u}(t, \xi) + a_1(t, \xi) \dot{u}(t, \xi) + a_0(t, \xi) u(t, \xi) = f(t, \xi) \quad \text{on } \mathfrak{T} \times \Xi \quad (27a)$$

$$\begin{cases} b_{11}(\xi) \dot{u}(0, \xi) + b_{10}(\xi) u(0, \xi) = b_1(\xi) \\ b_{21}(\xi) \dot{u}(0, \xi) + b_{20}(\xi) u(0, \xi) = b_2(\xi) \end{cases} \quad \text{on } \{0\} \times \Xi, \quad (27b)$$

where the dimensionality of the random space,  $d$ , can be assumed to be any natural number.

Provided sufficient regularity, an explicit form of (27) can be established by:

$$\partial_t^2 u(t, \xi) := \ell(t, \xi, u(t, \xi), \dot{u}(t, \xi)) = \bar{f}(t, \xi) + \bar{a}_0(t, \xi) u(t, \xi) + \bar{a}_1(t, \xi) \dot{u}(t, \xi) \quad \text{on } \mathfrak{T} \times \Xi \quad (28a)$$

$$\{u(0, \xi) = \bar{b}_1(\xi), \dot{u}(0, \xi) = \bar{b}_2(\xi)\} \quad \text{on } \{0\} \times \Xi, \quad (28b)$$

where  $\bar{f} = f/a_2$ ,  $\bar{a}_0 = -a_0/a_2$  and  $\bar{a}_1 = -a_1/a_2$  are elements of  $\mathcal{V}$ , and  $\bar{b}_1 = (b_1 b_{21} - b_2 b_{11})/(b_{10} b_{21} - b_{11} b_{20})$  and  $\bar{b}_2 = (b_2 b_{10} - b_1 b_{20})/(b_{10} b_{21} - b_{11} b_{20})$  are elements of  $\mathcal{X}$ .

This is, of course, a non-autonomous stochastic system because  $\ell$  depends explicitly on the time variable  $t$ , and what is more, it features a time-dependent stochastic input  $x = (x_1, x_2, x_3) \in \mathcal{V}^3$  given by:

$$x_1(t, \xi) = \bar{f}(t, \xi), \quad x_2(t, \xi) = \bar{a}_0(t, \xi), \quad x_3(t, \xi) = \bar{a}_1(t, \xi).$$

A stochastic flow map,  $\varphi(4) = (\varphi^1(4), \varphi^2(4))$ , for the system in hand can then be specified with (24) to produce:

$$\varphi^1(4)(h, s(t_i, \cdot)) = u(t_i, \cdot) + h \dot{u}(t_i, \cdot) + \frac{1}{2} h^2 \partial_t^2 u(t_i, \cdot) + \frac{1}{6} h^3 \partial_t^3 u(t_i, \cdot) + \frac{1}{24} h^4 \partial_t^4 u(t_i, \cdot) \quad (29a)$$

$$\varphi^2(4)(h, s(t_i, \cdot)) = \dot{u}(t_i, \cdot) + h \partial_t^2 u(t_i, \cdot) + \frac{1}{2} h^2 \partial_t^3 u(t_i, \cdot) + \frac{1}{6} h^3 \partial_t^4 u(t_i, \cdot) + \frac{1}{24} h^4 \partial_t^5 u(t_i, \cdot), \quad (29b)$$

where the second time derivative  $\partial_t^2 u(t_i, \cdot)$  is computed with (28a), and the next time derivatives  $\{\partial_t^j u(t_i, \cdot)\}_{j=3}^5$  with (23) from where one obtains:  $\hbar_4 = \hbar_5 \equiv 0$  and

$$\partial_t \ell = \partial_t \bar{f} + \partial_t \bar{a}_0 u + \partial_t \bar{a}_1 \dot{u}, \quad \partial_t^2 \ell = \partial_t^2 \bar{f} + \partial_t^2 \bar{a}_0 u + \partial_t^2 \bar{a}_1 \dot{u}, \quad \partial_t^3 \ell = \partial_t^3 \bar{f} + \partial_t^3 \bar{a}_0 u + \partial_t^3 \bar{a}_1 \dot{u}, \quad (30a)$$

$$\partial_u \ell = \bar{a}_0, \quad \partial_{tu} \ell = \partial_t \bar{a}_0, \quad \partial_{ttu}^3 \ell = \partial_t^2 \bar{a}_0, \quad (30b)$$

$$\partial_{\dot{u}} \ell = \bar{a}_1, \quad \partial_{t\dot{u}}^2 \ell = \partial_t \bar{a}_1, \quad \partial_{tt\dot{u}}^3 \ell = \partial_t^2 \bar{a}_1. \quad (30c)$$

<sup>3</sup>Note that  $\mathcal{X}^n$  is the  $n$ -fold cartesian product of  $\mathcal{X}$ , where  $n$  denotes the order of the system's governing ODE with respect to time.

## 5. Enriched stochastic flow map

In the enriched version of the stochastic flow map we are not only concerned with pushing the state of the system one-time step forward in  $\mathcal{X}^n$ , but also  $\ell$  (as displayed in Eq. (19a)) and its first  $M-1$  time derivatives. Because of this, we define the *enriched stochastic flow map* of order  $M$ ,  $\hat{\varphi}(M) : \mathbb{R} \times \mathcal{X}^{n+M} \rightarrow \mathcal{X}^{n+M}$ , such that its  $k$ -th component is given by:

$$\hat{\varphi}^k(M)(h, \hat{s}(t_i, \cdot)) := \hat{s}^k(t_i + h, \cdot) = \begin{cases} \varphi^k(M)(h, s(t_i, \cdot)) & \text{for } k \in \{1, 2, \dots, n\} \\ D_t^{k-n-1} \ell(t_i + h, \cdot, s(t_i + h, \cdot)) & \text{otherwise} \end{cases}$$

with  $k \in \{1, 2, \dots, n+M\}$ . Here  $\hat{s} = (u, \partial_t u, \dots, \partial_t^{n+M-1} u) \in \prod_{j=1}^{n+M} \mathcal{T}(n+M-j) \otimes \mathcal{X}$  is called the *enriched configuration state* of the system over  $\mathcal{T} \times \Xi$ . Observe that  $\hat{s}^{n+1} := \partial_t^n u = \ell$  is defined by (19a), and that if  $M = 4$ , then  $\{\hat{s}^k := \partial_t^{k-1} u = D_t^{k-n-1} \ell\}_{k=n+2}^{n+4}$  is given by the expressions prescribed in (26).

**Example 3.** Consider the stochastic system presented in Example 2. The associated enriched stochastic flow map for that system,  $\hat{\varphi}(4)$ , is found to be:

$$\hat{\varphi}^1(4)(h, s(t_i, \cdot)) := \varphi^1(4)(h, s(t_i, \cdot)) = s^1(t_i + h, \cdot) = u(t_i + h, \cdot) - O(h^5), \quad (31a)$$

$$\hat{\varphi}^2(4)(h, s(t_i, \cdot)) := \varphi^2(4)(h, s(t_i, \cdot)) = s^2(t_i + h, \cdot) = \dot{u}(t_i + h, \cdot) - O(h^5), \quad (31b)$$

$$\hat{\varphi}^3(4) := \ell = \partial_t^2 u, \quad \hat{\varphi}^4(4) := D_t \ell = \partial_t^3 u, \quad \hat{\varphi}^5(4) := D_t^2 \ell = \partial_t^4 u \quad \text{and} \quad \hat{\varphi}^6(4) := D_t^3 \ell = \partial_t^5 u, \quad (31c)$$

where  $\hat{\varphi}^1(4)$  and  $\hat{\varphi}^2(4)$  are computed with (29),  $\hat{\varphi}^3(4)$  with (28a), and  $\{\hat{\varphi}^k(4)\}_{k=4}^6$  with (23). In Example 2 we found that from (23) one obtains  $\hbar_4 = \hbar_5 \equiv 0$  and (30).

## 6. Flow-driven spectral chaos (FSC) method

As already mentioned in the introduction, the FSC method uses the concept of enriched stochastic flow maps to track the evolution of the stochastic part of the solution space efficiently in time. In Section 4, we assumed that the solution of the system governed by (19) was analytic on the temporal domain. This assumption implies that  $u$  can be represented by a Taylor series centered at  $t = t_i \in \mathcal{T}$  for all  $\xi \in \Xi$ :

$$u(t, \xi) = \sum_{j=0}^{\infty} \frac{(t - t_i)^j}{j!} \partial_t^j u(t_i, \xi), \quad (32)$$

where  $(t - t_i)^j / j!$  is nothing but a temporal function in  $\mathcal{T}$ , and  $\partial_t^j u(t_i, \xi)$  is a random function in  $\mathcal{X}$ . From this it follows that if  $\{\partial_t^j u(t_i, \xi)\}_{j=0}^{\infty}$  is orthogonalized with respect to the measure in  $\mathcal{X}$ , one can write expression (32) in the following way:

$$u(t, \xi) = \sum_{j=0}^{\infty} u^j(t) \Psi_j(\xi). \quad (2)$$

For a system driven by a stochastic flow map of order  $M$ , it is apparent that the infinitely many basis vectors in  $\{\partial_t^j u(t_i, \cdot)\}_{j=0}^{\infty}$  do not all need to be orthogonalized, but only the  $n+M$  components of the enriched stochastic flow map:  $\{\hat{\varphi}^k(M)(0, \hat{s}(t_i, \cdot)) \equiv \partial_t^{k-1} u(t_i, \cdot)\}_{k=1}^{n+M}$ . In this sense, the order of the stochastic flow map determines the maximum number of basis vectors to use in the simulation. This explains why the FSC method does not suffer from the curse of dimensionality at the random-function-space level, even when the probability space is high-dimensional.

However, to reduce the numerical cost associated with the orthogonalization of  $n+M$  basis vectors as indicated above, one can choose to start the FSC analysis by considering first the lowest value for  $M$ , i.e.  $M = 1$ . Then, if more accurate results are needed, the  $M$ -value can be incremented progressively, provided that it does not exceed the order of the stochastic flow map we are targeting. Hence, in the FSC method the discretization of the random function space  $\mathcal{X}^{[P]}$  is bounded by  $n+1 \leq P \leq n+M$  as long as we presume that the system is driven by a stochastic flow map of order  $M$ .



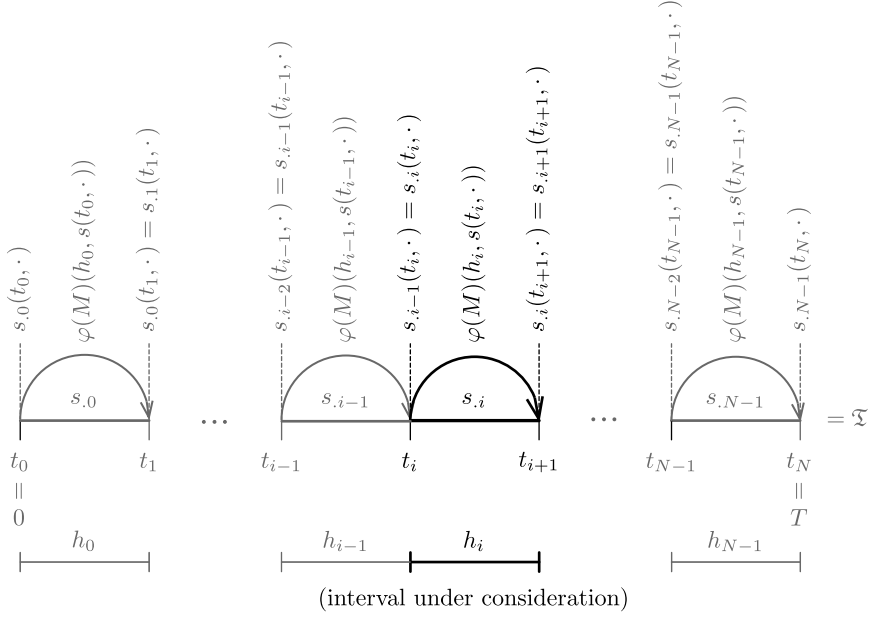


Figure 2: Evolution of a dynamical system via a stochastic flow map of order  $M$  (with  $h_i > 0$ )

*Remark 11.* We note that once this  $M$  is fixed to solve the problem in hand numerically, the probability information of the system's state can be pushed exactly in time if the discretization level of  $\mathcal{X}$  is such that  $P = n + M$ . However, as we will see in Section 8, accurate results are also achievable for a lower discretization level of  $\mathcal{X}$ , and thus, we do not always need to run our simulations with a relatively high value of  $P$ .

*FSC scheme.* Let us consider the stochastic system given by (1). Let  $\{\mathfrak{T}_i\}_{i=0}^{N-1}$  be a partition of the temporal domain, where  $\mathfrak{T}_i \neq \emptyset$  represents the  $i$ -th interval of the partition, and define  $s_i = s|_{\text{cl}(\mathfrak{T}_i) \times \Xi}$  to be the restriction of  $s$  to  $\mathfrak{R}_i := \text{cl}(\mathfrak{T}_i) \times \Xi$ . (Please see Fig. 2, and recall that  $s$  represents the configuration state of the system over  $\mathfrak{T} \times \Xi$ .) Then, if the system is driven by a stochastic flow map of order  $M$ , proceed as below.

1. Loop across the temporal domain from  $i = 0$  to  $i = N - 1$ .

(a) Define a solution representation for the configuration state  $s_i$  in the following way.

- Take  $\Phi_{0,i} \equiv 1$  and  $\{\Phi_{j,i} := \hat{\varphi}^j(M)(0, \hat{s}(t_i, \cdot))\}_{j=1}^P$  to be an ordered set of linearly independent functions in  $\mathcal{X}$  with  $n + 1 \leq P \leq n + M$ . Observe that  $\hat{\varphi}(M)(0, \hat{s}(t_i, \cdot)) \equiv \hat{s}_i(t_i, \cdot) = \hat{s}_{i-1}(t_i, \cdot)$  for  $i \geq 1$ . However, if  $i = 0$ , then  $\hat{\varphi}(M)(0, \hat{s}(t_0, \cdot)) \equiv \hat{s}(0, \cdot)$ . (*Note:* When the initial conditions over  $\mathfrak{R}_i$  are linearly dependent, please see Remark 12.) It is worth mentioning that, for computational efficiency, we have chosen  $h_i = 0$  for the definition of  $\{\Phi_{j,i}\}_{j=1}^P$ , but in principle  $h_i$  is any number between 0 and the length of  $\mathfrak{T}_i$ . For higher accuracy,  $h_i$  is recommended to be taken as the distance between  $t_i$  and the midpoint in  $\mathfrak{T}_i$ . This recommendation is especially important when the length of  $\mathfrak{T}_i$  is relatively large, such as in multi-time-step simulations.
- Orthogonalize the set  $\{\Phi_{j,i}\}_{j=0}^P$  using the Gram-Schmidt process [71], so that the resulting set  $\{\Psi_{j,i}\}_{j=0}^P$  is an orthogonal basis in  $\mathcal{X}$ . That is, for  $j \in \{0, 1, \dots, P\}$ :

$$\Psi_{j,i} := \Phi_{j,i} - \sum_{k=0}^{j-1} \frac{\langle \Phi_{j,i}, \Psi_{k,i} \rangle}{\langle \Psi_{k,i}, \Psi_{k,i} \rangle} \Psi_{k,i}.$$

(Equivalently, Theorem 1 can be employed in this step.)

- Define  $\mathcal{F}_i^{[P]} = \text{span}\{\Psi_{j,i}\}_{j=0}^P$  to be a  $p$ -discretization of  $\mathcal{F}$  over the region  $\mathfrak{R}_i$ . Since  $\mathcal{F}_i^{[P]}$  is an evolving function space, expansion (3) is now to be read as:

$$u_{.i}(t, \xi) \approx u_{.i}^{[P]}(t, \xi) = \sum_{j=0}^P u_{.i}^j(t) \Psi_{j,i}(\xi) \equiv u_{.i}^j(t) \Psi_{j,i}(\xi). \quad (3^*)$$

From this it follows that the  $l$ -th component of the configuration state,  $s_{.i}^l$ , can be computed by taking the  $(l-1)$ -th time derivative of this representation. Here  $l \in \{1, 2, \dots, n\}$ .

- (b) Transfer the random modes of  $s_{.i-1} = (u_{.i-1}, \partial_t u_{.i-1}, \dots, \partial_t^{n-1} u_{.i-1})$  to  $s_{.i} = (u_{.i}, \partial_t u_{.i}, \dots, \partial_t^{n-1} u_{.i})$  at  $t = t_i$ , given that  $i \geq 1$ . To this end, any of the following two approaches can be adopted.

**FSC-1.** This approach transfers the probability information in the mean-square sense, by ensuring that the equalities shown below hold in the mean-square sense for each of the components of  $s_{.i}$  and  $s_{.i-1}$  at  $t = t_i$  (summation sign implied only over repeated index  $k$ ):

$$s_{.i}^l(t_i, \xi) = s_{.i-1}^l(t_i, \xi) \iff (s^l)^{k_{.i}}(t_i) \Psi_{k,i}(\xi) = (s^l)^{k_{.i-1}}(t_i) \Psi_{k,i-1}(\xi). \quad (33)$$

Projecting (33) onto  $\mathcal{F}_i^{[P]}$  gives the random modes of each of the components of  $s_{.i}$  at  $t = t_i$ :

$$(s^l)^{j_{.i}}(t_i) = \sum_{k=0}^P \frac{\langle \Psi_{j,i}, \Psi_{k,i-1} \rangle}{\langle \Psi_{j,i}, \Psi_{j,i} \rangle} (s^l)^{k_{.i-1}}(t_i), \quad (8b^*)$$

where  $l \in \{1, 2, \dots, n\}$  and  $j \in \{0, 1, \dots, P\}$ . This approach was first introduced by Heuveline and Schick in [63].

**FSC-2.** This approach transfers the probability information exactly. In order to do so, Theorem 1 is implemented to obtain the random modes of each of the components of  $s_{.i}$  at  $t = t_i$ . Thus, from (12) it follows that:

$$\Phi_{l,i} = \mathbf{E}[\Phi_{l,i}] \Psi_{0,i} + \sum_{j=1}^{l-1} \frac{\det \Delta_j(l)}{\det \square_j} \Psi_{j,i} + \Psi_{l,i} \quad \text{with} \quad \Psi_{0,i} \equiv 1,$$

which after taking  $\{\Phi_{l,i} := \varphi^l(M)(0, s(t_i, \cdot)) \equiv s_{.i}^l(t_i, \cdot) = s_{.i-1}^l(t_i, \cdot)\}_{l=1}^n$  yields

$$(s^l)^{j_{.i}}(t_i) = \begin{cases} \mathbf{E}[s_{.i-1}^l(t_i, \cdot)] & \text{for } j = 0 \\ \frac{\det \Delta_j(l)}{\det \square_j} & \text{for } 0 < j < l \\ 1 & \text{for } j = l \\ 0 & \text{otherwise,} \end{cases} \quad (8b^*)$$

where  $l \in \{1, 2, \dots, n\}$  and  $j \in \{0, 1, \dots, P\}$ . This approach was first introduced by Gerritsma et al. in [62], but it had not been generalized for high-order stochastic ODEs until now.

If  $i = 0$ , the initial conditions are computed with (8b) directly.

- (c) Substitute (3\*) into (1) to obtain (7).
- (d) Project (7a) onto  $\mathcal{F}_i^{[P]}$  to obtain (8a) subject to (8b\*). Note that if  $i = 0$ , (8a) is subject to (8b).
- (e) Integrate (8) over time, as long as a suitable time integration method has been selected for solving the resulting system of equations. This step requires to find the random modes of each of the components of the configuration state  $s_{.i}$  at  $t = t_{i+1}$ ; that is,  $\{(s^l)^{j_{.i}}(t_{i+1})\}_{l=1, j=0}^{n,P}$ .
- (f) Compute both the mean and the variance of each of the components of output  $y = \mathcal{M}[u][x]$  over  $\mathfrak{R}_i$ , by recurring to the formulas prescribed by (9) and (10).

## 2. Post-process results.

*Remark 12.* When the initial conditions are linearly dependent over the region  $\mathfrak{R}_i$ , any of the following two approaches can be used:

- If at the start of the simulation the initial conditions over  $\mathfrak{R}_1$  are deterministic, the first  $n$  vectors in  $\{\Phi_{j,0} := \hat{\varphi}^j(M)(0, \hat{s}(t_0, \cdot))\}_{j=1}^P$  are required to be removed from the set, for they are linearly dependent to  $\Phi_{0,0} \equiv 1$ . However, if more generally the initial conditions are linearly dependent over  $\mathfrak{R}_i$ , one can find a linear map  $A$  such that its image produces a set of linearly independent vectors for use in the definition of  $\Phi_{j,i}$ . That is, if the rank of  $A$  is  $r+1$ , then  $A : \mathcal{X}^{n+1} \rightarrow \mathcal{X}^{r+1}$  is a linear map given by:

$$(1, \varphi(M)(0, s(t_i, \cdot))) \mapsto (b^0 \equiv 1, b^1, \dots, b^r) = A(1, \varphi(M)(0, s(t_i, \cdot))).$$

For example, if  $n = 4$  and  $r = 2$ , the map  $A : \mathcal{X}^5 \rightarrow \mathcal{X}^3$  can be given in matrix form by:

$$\begin{bmatrix} b^0 \\ b^1 \\ b^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & A_1^1 & A_2^1 & A_3^1 & A_4^1 \\ 0 & A_1^2 & A_2^2 & A_3^2 & A_4^2 \end{bmatrix} \begin{bmatrix} 1 \\ \varphi^1(M)(0, s(t_i, \cdot)) \\ \varphi^2(M)(0, s(t_i, \cdot)) \\ \varphi^3(M)(0, s(t_i, \cdot)) \\ \varphi^4(M)(0, s(t_i, \cdot)) \end{bmatrix}$$

with  $A_k^j \in \mathbb{R}$  and such that  $\text{rank}(A) = r+1 = 3$ . In general, this means that the ordered set needed in step 1(a) reduces to:

$$\{\Phi_{j,i}\}_{j=0}^{P-n+r} \equiv \{b^k\}_{k=0}^r \cup \{\hat{\varphi}^k(M)(0, \hat{s}(t_i, \cdot))\}_{k=n+1}^P.$$

- As an alternative, the gPC method can be employed to advance the state of the system one-time step forward in the simulation, and then switch over FSC once this is done. The disadvantage of this alternative is that it only works at early times of the simulation.

*Remark 13.* A stopping condition can be defined within the FSC scheme to help reduce the computational cost associated with the creation of a new random function space in the simulation. However, a major drawback of incorporating a stopping condition within the scheme is that, if not chosen well, it can lead to significant degradation of the solution over time. Stopping conditions such as those addressed by Gerritsma et al. [62] were implemented in this work, but led to high errors in the response at the end of the simulations. This is the primary reason why we did not provide one within the FSC scheme.

**Example 4.** Consider Example 2 one more time. If **FSC-1** is used to transfer the probability information of the system's state at  $t = t_i$ , we get the following expressions for the random modes of  $s_i = (u_i, \dot{u}_i)$ :

$$u_{j,i}^j(t_i) = \sum_{k=0}^P \frac{\langle \Psi_{j,i}, \Psi_{k,i-1} \rangle}{\langle \Psi_{j,i}, \Psi_{j,i} \rangle} u_{k,i-1}^k(t_i) \quad \text{and} \quad \dot{u}_{j,i}^j(t_i) = \sum_{k=0}^P \frac{\langle \Psi_{j,i}, \Psi_{k,i-1} \rangle}{\langle \Psi_{j,i}, \Psi_{j,i} \rangle} \dot{u}_{k,i-1}^k(t_i)$$

with  $j \in \{0, 1, \dots, P\}$ . However, if **FSC-2** is employed instead, the probability information is transferred with the following expressions. For the random modes of  $u_i$ , the expressions are:

$$u_{j,i}^j(t_i) = \begin{cases} \mathbf{E}[u_{i-1}(t_i, \cdot)] & \text{for } j = 0 \\ 1 & \text{for } j = 1 \\ 0 & \text{for } 2 \leq j \leq P, \end{cases}$$

and those corresponding to the random modes of  $\dot{u}_i$  are:

$$\dot{u}_{j,i}^j(t_i) = \begin{cases} \mathbf{E}[\dot{u}_{i-1}(t_i, \cdot)] & \text{for } j = 0 \\ \frac{\text{Cov}[u_{i-1}(t_i, \cdot), \dot{u}_{i-1}(t_i, \cdot)]}{\text{Cov}[u_{i-1}(t_i, \cdot), u_{i-1}(t_i, \cdot)]} & \text{for } j = 1 \\ 1 & \text{for } j = 2 \\ 0 & \text{for } 3 \leq j \leq P. \end{cases}$$

From this example, it is easy to see why the probability information is transferred faster if **FSC-2** is used.

## 7. Numerical examples

To better investigate the performance of the FSC method in the resolution of problems involving stochastic input data, six problems are selected and described in this section.

### 7.1. Problem 1: A linear system governed by a 1st-order stochastic ODE

We consider the problem of a falling body under stochastic air resistance as follows.

Find the velocity of the falling body  $v : \mathfrak{T} \times \Xi \rightarrow \mathbb{R}$  in  $\mathcal{U}$ , such that ( $\mu$ -a.e.):

$$\begin{aligned} m\dot{v} + kv &= mg && \text{on } \mathfrak{T} \times \Xi \\ v(0, \cdot) &= \mathbf{v} && \text{on } \{0\} \times \Xi, \end{aligned}$$

where  $m = 4$  kg is the mass of the falling body,  $k : \Xi \rightarrow \mathbb{R}^+$  is the air resistance given by  $k(\xi) = \xi$ ,  $g = 9.81$  m/s<sup>2</sup> is the gravity acceleration,  $\mathbf{v} = 50$  m/s is the initial condition of the falling body, and  $\mathfrak{T} = [0, 150]$  s. Here  $\dot{v} := \partial_t v$  denotes the acceleration of the falling body.

Two probability distributions are considered for this system. The first distribution is a *uniform distribution*,  $\text{Uniform} \sim \xi \in \Xi = [a, b]$ , and the second distribution is a *beta distribution*,  $\text{Beta}(\alpha, \beta) \sim \xi \in \Xi = [a, b]$ . The parameters for both distributions are:  $(a, b) = (1, 2)$  kg/s and  $(\alpha, \beta) = (2, 5)$ .

### 7.2. Problem 2: A linear system governed by a 2nd-order stochastic ODE with one random variable

We consider the problem of a single-degree-of-freedom system with stochastic stiffness under free vibration.

Find the displacement of the system  $u : \mathfrak{T} \times \Xi \rightarrow \mathbb{R}$  in  $\mathcal{U}$ , such that ( $\mu$ -a.e.):

$$\begin{aligned} m\ddot{u} + ku &= 0 && \text{on } \mathfrak{T} \times \Xi \\ \{u(0, \cdot) = \mathbf{u}, \dot{u}(0, \cdot) = \mathbf{v}\} &&& \text{on } \{0\} \times \Xi, \end{aligned}$$

where  $m = 100$  kg is the mass of the system,  $k : \Xi \rightarrow \mathbb{R}^+$  is the stiffness of the system given by  $k(\xi) = \xi$ ,  $\mathbf{u} = 0.05$  m and  $\mathbf{v} = 0.20$  m/s are the initial conditions of the system, and  $\mathfrak{T} = [0, 150]$  s. Note that  $\dot{u} := \partial_t u$  and  $\ddot{u} := \partial_t^2 u$  denote the velocity and acceleration of the system, respectively.

Three probability distributions are considered for this system. The first distribution is a *uniform distribution*,  $\text{Uniform} \sim \xi \in \Xi = [a, b]$ , the second distribution is a *beta distribution*,  $\text{Beta}(\alpha_1, \beta_1) \sim \xi \in \Xi = [a, b]$ , and the third distribution is a *gamma distribution*,  $\text{Gamma}(\alpha_2, \beta_2) \sim \xi \in \Xi = [a, \infty)$ . The parameters for these three distributions are selected to be:  $(a, b) = (340, 460)$  N/m and  $(\alpha_1, \beta_1, \alpha_2, \beta_2) = (2, 5, 10, 0.1)$ .

### 7.3. Problem 3: A linear system governed by a 2nd-order stochastic ODE with two random variables

In this section we consider the single-degree-of-freedom system previously defined, but in addition to having a stochastic stiffness for the system we also have a stochastic mass. In this setting, the mass of the system,  $m : \Xi \rightarrow \mathbb{R}^+$ , and the stiffness of the system,  $k : \Xi \rightarrow \mathbb{R}^+$ , are given by  $m(\xi) = \xi^1$  and  $k(\xi) = \xi^2$ , respectively, with 1 and 2 not denoting an exponentiation. We note that  $\xi = (\xi^1, \xi^2)$  is a 2-tuple random variable, and thus,  $\Xi$  is a two-dimensional random domain.

For this system, two probability distributions are explored. The first distribution is a *uniform-uniform distribution*,  $\text{Uniform} \otimes \text{Uniform} \sim \xi \in \Xi = [a_1, b_1] \times [a_2, b_2]$ , and the second distribution is a *uniform-beta distribution*,  $\text{Uniform} \otimes \text{Beta}(\alpha, \beta) \sim \xi \in \Xi = [a_1, b_1] \times [a_2, b_2]$ . The parameters for both distributions are taken as:  $(a_1, b_1) = (85, 115)$  kg,  $(a_2, b_2) = (340, 460)$  N/m and  $(\alpha, \beta) = (2, 5)$ .

#### 7.4. Problem 4: A linear system governed by a 3rd-order stochastic ODE

Here we consider the problem of a linear mechanical system governed by a 3rd-order stochastic ODE. Find the displacement of the system  $u : \mathfrak{T} \times \Xi \rightarrow \mathbb{R}$  in  $\mathcal{U}$ , such that ( $\mu$ -a.e.):

$$\begin{aligned} \partial_t^3 u + \frac{1}{2} \partial_t^2 u + k \partial_t u + u &= 0 & \text{on } \mathfrak{T} \times \Xi \\ \{u(0, \cdot) = u, \partial_t u(0, \cdot) = v, \partial_t^2 u(0, \cdot) = a\} & & \text{on } \{0\} \times \Xi, \end{aligned}$$

where  $k : \Xi \rightarrow \mathbb{R}$  is a mechanical parameter given by  $k(\xi) = \xi$ ;  $u = 1$  m,  $v = -1$  m/s and  $a = 2$  m/s<sup>2</sup> are the initial conditions of the system; and  $\mathfrak{T} = [0, 150]$  s.

Three probability distributions are investigated for this system. The first distribution is a *uniform distribution*,  $\text{Uniform} \sim \xi \in \Xi = [a, b]$ , the second distribution is a *beta distribution*,  $\text{Beta}(\alpha, \beta) \sim \xi \in \Xi = [a, b]$ , and the third distribution is a *normal distribution*,  $\text{Normal}(\mu, \sigma^2) \sim \xi \in \Xi = \mathbb{R}$ . The parameters for these three distributions are:  $(a, b) = (2, 3)$  N/m,  $(\mu, \sigma) = (2.5, 0.125)$  N/m and  $(\alpha, \beta) = (2, 5)$ .

#### 7.5. Problem 5: A linear system governed by a 4th-order stochastic ODE

Next, we consider the problem of a linear mechanical system governed by a 4th-order stochastic ODE. Find the displacement of the system  $u : \mathfrak{T} \times \Xi \rightarrow \mathbb{R}$  in  $\mathcal{U}$ , such that ( $\mu$ -a.e.):

$$\begin{aligned} \partial_t^4 u + k \partial_t^2 u + u &= 0 & \text{on } \mathfrak{T} \times \Xi \\ \{u(0, \cdot) = u, \partial_t u(0, \cdot) = v, \partial_t^2 u(0, \cdot) = a, \partial_t^3 u(0, \cdot) = j\} & & \text{on } \{0\} \times \Xi, \end{aligned}$$

where  $k : \Xi \rightarrow \mathbb{R}$  is a mechanical parameter given by  $k(\xi) = \xi$ ;  $u = 1$  m,  $v = -1$  m/s,  $a = 2$  m/s<sup>2</sup> and  $j = -3$  m/s<sup>3</sup> are the initial conditions of the system; and  $\mathfrak{T} = [0, 150]$  s.

The same three probability distributions mentioned in Problem 4 are considered here, but the parameters of the distributions now take the following values:  $(a, b) = (3, 5)$  kg,  $(\mu, \sigma) = (4, 0.2)$  kg and  $(\alpha, \beta) = (2, 5)$ .

#### 7.6. Problem 6: A nonlinear system governed by a 2nd-order stochastic ODE

In this last section, we study the stochastic behavior of a Van-der-Pol oscillator. Because this oscillator is highly nonlinear over the temporal-random space, we use it herein as a toy problem to test the performance of the FSC method more thoroughly. For reference, the spectral discretization of this problem is derived in detail in Appendix A. The problem can be stated as follows.

Find the displacement of the oscillator  $u : \mathfrak{T} \times \Xi \rightarrow \mathbb{R}$  in  $\mathcal{U}$ , such that ( $\mu$ -a.e.):

$$m\ddot{u} - (1 - \rho u^2) c\dot{u} + ku = 0 \quad \text{on } \mathfrak{T} \times \Xi \quad (34a)$$

$$\{u(0, \cdot) = u, \dot{u}(0, \cdot) = v\} \quad \text{on } \{0\} \times \Xi, \quad (34b)$$

where  $m = 100$  kg is the mass of the oscillator,  $c$  is a *uniformly-distributed* random variable representing the strength of the damping and defined in  $[150, 450]$  kg/s,  $\rho = 150$  m<sup>-2</sup> is the contributing factor to the nonlinearity of the oscillator,  $k = 400$  N/m is the stiffness of the oscillator,  $u \sim \text{Beta}(2, 5)$  is a *beta-distributed* random variable defined in  $[0.05, 0.25]$  m,  $v = 2u - 0.10$  is another *beta-distributed* random variable expressed in m/s, and  $\mathfrak{T} = [0, 150]$  s. Observe that  $\dot{u} := \partial_t u$  and  $\ddot{u} := \partial_t^2 u$  denote the velocity and acceleration of the oscillator, respectively.

## 8. Discussion on numerical results

In this section, we demonstrate and compare the performance of the FSC scheme using the two approaches developed in Section 6 for the transfer of the probability information, namely FSC-1 and FSC-2. We do this for each of the problems described in Section 7, followed by a computational-cost comparison between FSC and mTD-gPC [63] for Problem 2 of Section 7.2.

The local error,  $\epsilon : \mathcal{T} \rightarrow \mathcal{T}$ , and the global error,  $\epsilon_G : \mathcal{T} \rightarrow \mathbb{R}$ , are defined with these expressions:

$$\begin{aligned}\epsilon[f](t) &= |f(t) - f_{\text{exact}}(t)| \\ \epsilon_G[f] &= \frac{1}{T} \int_{\mathfrak{T}} |f(t) - f_{\text{exact}}(t)| dt \approx \frac{\Delta t}{T} \sum_{i=0}^N |f(t_i) - f_{\text{exact}}(t_i)|,\end{aligned}$$

where  $\Delta t$  is the time-step size used for the simulation,  $t_i \in \mathfrak{T}$  is the time instant of the simulation, and  $N$  denotes the number of time steps employed in the simulation (with  $t_0 = 0$  and  $t_N = N \Delta t = T$ ).

In this work we use the Runge-Kutta method [73] of fourth-order (aka RK4 method) to push the state of the system forward in time. The time-step size used is  $\Delta t = 0.001$  s (unless indicated otherwise), which means that  $N = 150\,000$  time steps are used in the simulations<sup>4</sup>. The time-step size is taken this small to attenuate as much as possible the errors coming from the discretization of  $\mathcal{T}(n)$ . As pointed out in Remark 13, we update the stochastic part of the solution space at every time step in an attempt to curtail the degradation of the solution over time. Finally, the gPC method (with  $P = 6$ ) is implemented for the first second of the simulation to ensure that the stochasticity of the system's state is well developed for the analysis with FSC or mTD-gPC.

To evaluate the inner products numerically, we use the following quadrature rules on each random axis:

$$\begin{aligned}\text{Uniform} &\sim \text{Gauss-Legendre}(100 \text{ points}), & \text{Beta} &\sim \text{Gauss-Jacobi}(80 \text{ points}), \\ \text{Gamma} &\sim \text{Gauss-Laguerre}(140 \text{ points}) & \text{and} & \text{Normal} \sim \text{Gauss-Hermite}(110 \text{ points}).\end{aligned}$$

All problems are run in MATLAB R2016b [74] on a 2017 MacBook Pro with quad-core *3.1 GHz Intel Core i7* processor (hyper-threading technology enabled), *16 GB 2133 MHz LPDDR3* memory and *1 TB PCI-Express SSD* storage (APFS-formatted), running macOS Mojave (version 10.14.6).

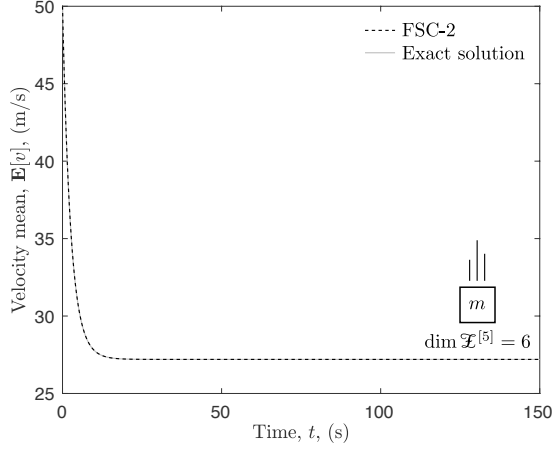
### 8.1. Numerical results for the five linear systems

Figs. 3 to 7 show the evolution of the mean and the variance of one of the system's responses using FSC-2 and the exact solution<sup>5</sup> for sake of comparison. In particular, Fig. 3 shows the evolution of the system's velocity for Problem 1, Figs. 4 to 6 the evolution of the system's displacement for Problems 2 to 4, and Fig. 7 the evolution of the system's jerk for Problem 5. As observed, responses obtained with both the FSC-1 and FSC-2 methods approach the exact solution with high fidelity. We emphasize that these responses are obtained by using only a few number of basis vectors with  $P$  set equal to  $4 + n$  after the first second of the simulation. (Recall that  $n$  denotes the order of the governing ODE with respect to time.)

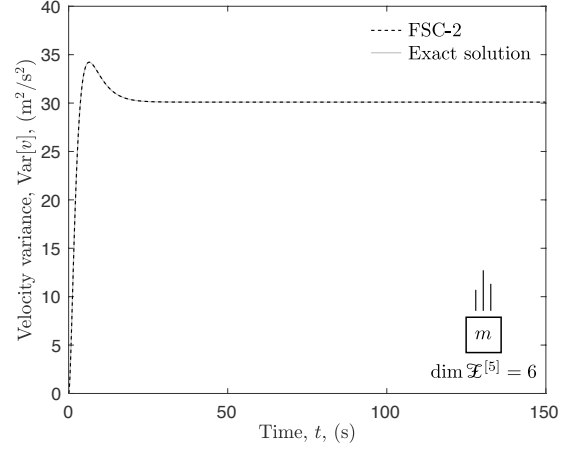
Figs. 8 to 10 present the local errors in mean and variance of each of the responses mentioned above but only for Problems 1 and 2 for sake of brevity. The errors are depicted for both FSC-1 and FSC-2. To compare, we also include the case when  $P = n$  even though the FSC scheme requires that  $P$  is taken at least equal to  $n + 1$ . We do so to test the implications of spanning the RFS with the state variables of the system only. The cases when  $P = n + 2$  and  $P = n + 4$  are also provided for the sake of comparison. From these figures, it is apparent that as the number of basis vectors increases, so does the accuracy of the results. In particular, when the FSC-1 approach is used, the following observation can be made. By increasing the number of basis vectors from  $n + 1$  to  $n + 3$ , the accuracy of the results improves significantly by 6 orders of magnitude. However, when the number of basis vectors is increased from  $n + 3$  to  $n + 5$ , the results either do not improve noticeably or worsen a bit (as in Fig. 10). This is in contrast to the FSC-2 approach. When FSC-2 is used, the accuracy of the results improves not only significantly but also consistently as the number of basis vectors increases. The figures also indicate that FSC-2 can achieve in general a higher level of accuracy than FSC-1 as time progresses in the simulation. However, we do notice that whenever  $P = n$ , no difference between the two approaches can be discerned.

<sup>4</sup>For numerical reasons, if in the plots an asterisk is displayed next to a probability distribution, it means that the simulation was conducted for the first 100 seconds only. In such cases,  $N = 100\,000$  for a time-step size of 0.001 seconds.

<sup>5</sup>To obtain both the 'exact' mean and the 'exact' variance, MATLAB's Symbolic Math Toolbox [74] is first used to find the response of interest analytically (e.g. the solution of the stochastic ODE). Then, the `vpaintegral` is called with `RelTol` set equal to  $10^{-16}$  to compute the mean and the variance of the response numerically at every time instant of the simulation.

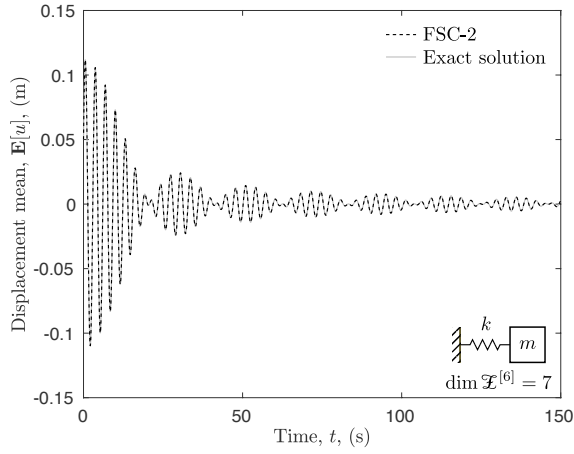


(a) Mean

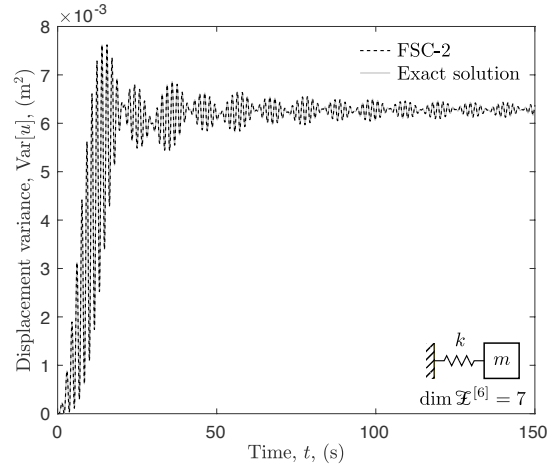


(b) Variance

Figure 3: *Problem 1* — Evolution of  $\mathbf{E}[v]$  and  $\text{Var}[v]$  for the case when the  $p$ -discretization level of RFS is  $\mathcal{X}^{[5]}$  and  $\mu \sim \text{Uniform}$



(a) Mean



(b) Variance

Figure 4: *Problem 2* — Evolution of  $\mathbf{E}[u]$  and  $\text{Var}[u]$  for the case when the  $p$ -discretization level of RFS is  $\mathcal{X}^{[6]}$  and  $\mu \sim \text{Uniform}$

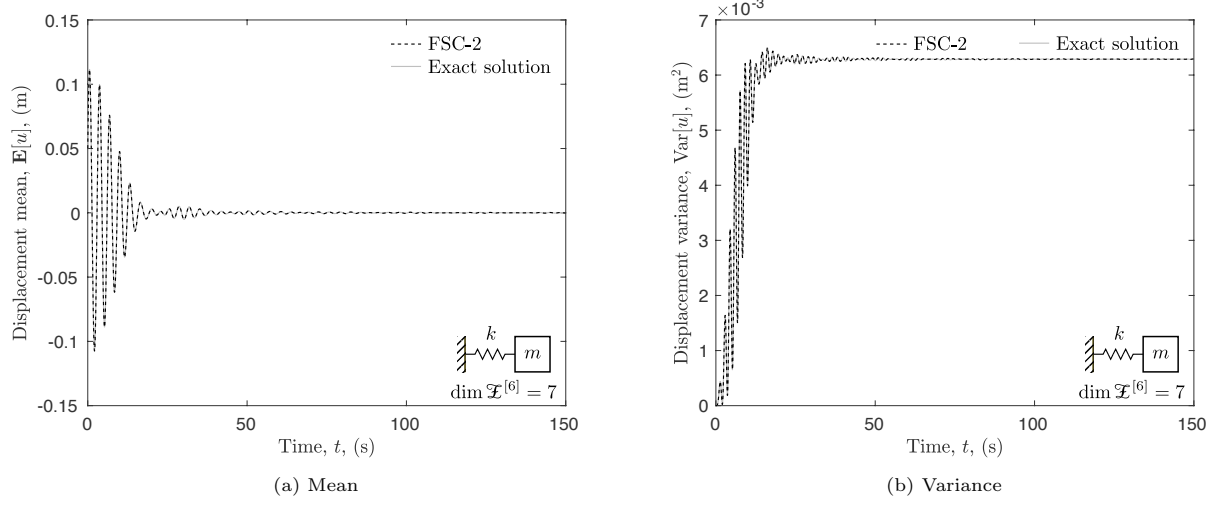


Figure 5: *Problem 3* — Evolution of  $\mathbf{E}[u]$  and  $\text{Var}[u]$  for the case when the  $p$ -discretization level of RFS is  $\mathcal{F}^{[6]}$  and  $\mu \sim \text{Uniform} \otimes \text{Uniform}$

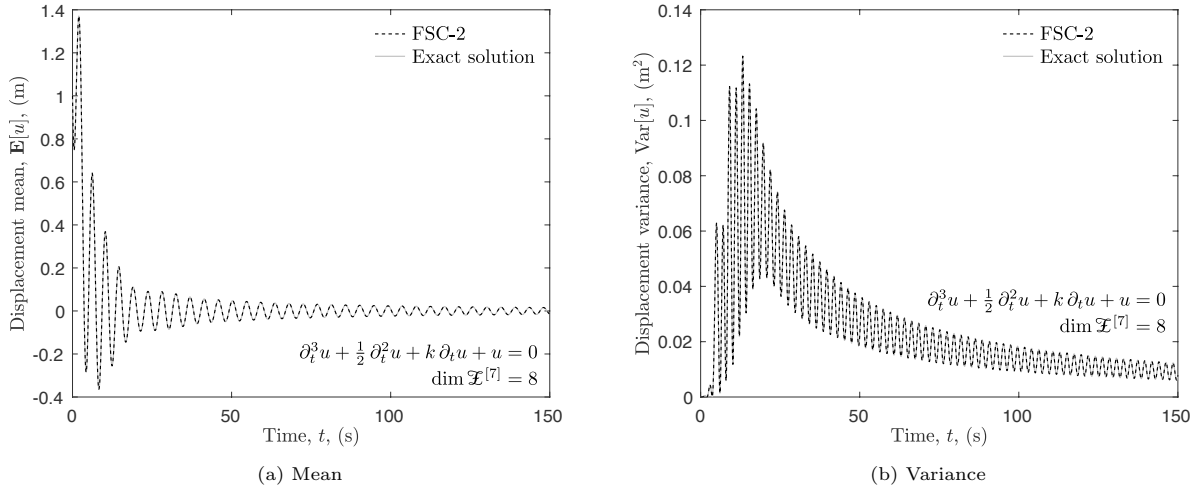


Figure 6: *Problem 4* — Evolution of  $\mathbf{E}[u]$  and  $\text{Var}[u]$  for the case when the  $p$ -discretization level of RFS is  $\mathcal{F}^{[7]}$  and  $\mu \sim \text{Uniform}$



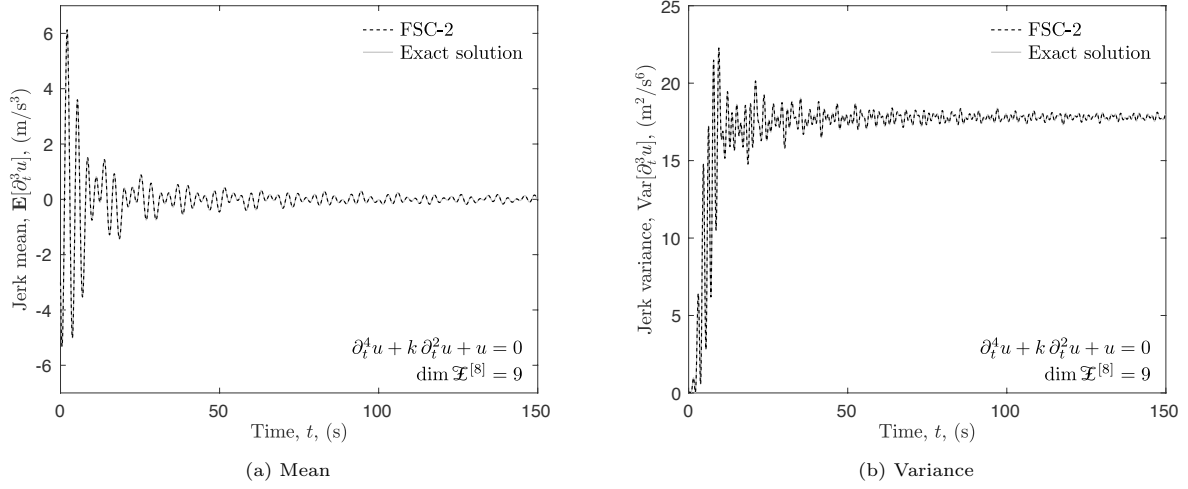
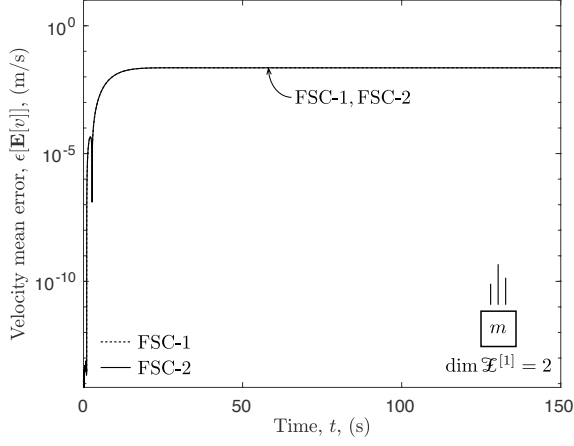


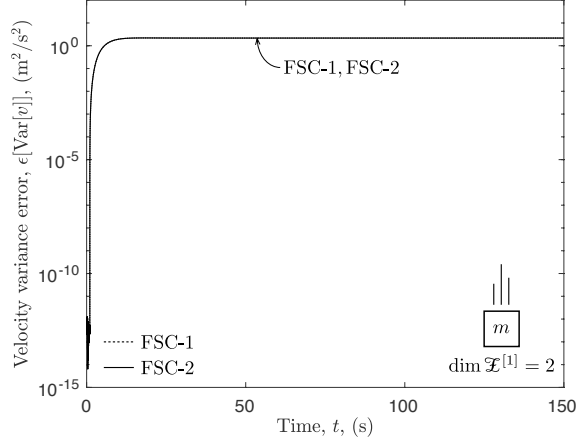
Figure 7: *Problem 5* — Evolution of  $\mathbf{E}[\partial_t^3 u]$  and  $\text{Var}[\partial_t^3 u]$  for the case when the  $p$ -discretization level of RFS is  $\mathcal{F}^{[8]}$  and  $\mu \sim \text{Uniform}$

Figs. 11 to 15 present the convergence of global errors as a function of the number of basis vectors. Included in these figures are the cases when the random parameters take different probability distributions as specified in Sections 7.1 to 7.5. In general, exponential convergence to the solution is achieved when  $n+1$ ,  $n+2$  and  $n+3$  basis vectors are used. However, in the case of using  $n+4$  basis vectors, the accuracy of the results does not improve noticeably for FSC-1 but it does for FSC-2. In fact, a significant difference between the two approaches can be discerned after using  $n+4$  basis vectors. Moreover, as it can be deduced from the results, that there is no gain in employing  $n+5$  basis vectors in the simulations because it does not lead to an increase in the accuracy of the solution. It is interesting to point out that in all figures, the results from FSC-1 and FSC-2 happen to be indistinguishable from each other whenever  $n+1$  or  $n+2$  basis vectors are used in the simulations. This is in contrast to Fig. 11 which shows that exponential convergence to the solution cannot be attained if the number of basis vectors is increased from  $n+2$  to  $n+3$  (i.e. from 3 to 4). This peculiar result can be explained by noting that the system's response is non-oscillatory, which makes the track of the RFS deviate continuously as the simulation proceeds. Figs. 12 and 13 further show that when the probability space is one-dimensional (Problem 2), better results are obtained than when it is two-dimensional (Problem 3). This is because more quadrature points are needed in Problem 3 to achieve the same level of accuracy. We also note from Fig. 13 that albeit the probability space is two-dimensional, a few numbers of basis vectors are needed to obtain accurate results. Remarkably, using only 5 or 6 basis vectors for FSC-2 already produces a global error of approximately  $10^{-10}$ . Finally, Fig. 15 indicates that when the response is unbounded (e.g. when  $\xi$  is assumed normally distributed), FSC-2 has the ability to control better the error propagation of the solution as the simulation proceeds.

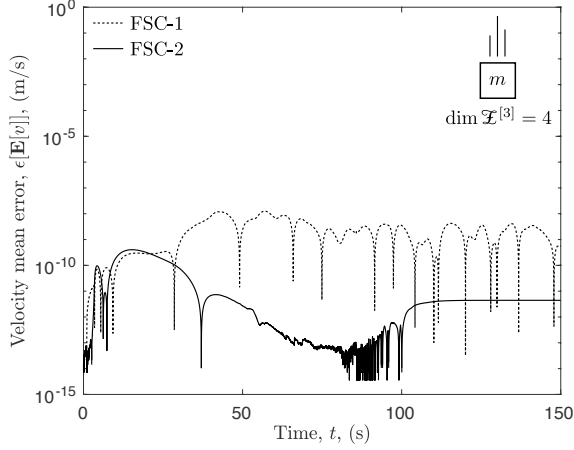
Fig. 16 presents the convergence of global errors for Problem 2 as a function of the number of basis vectors and for different time-step sizes. The goal of this figure is to show the implications of increasing the time-step size used by default ( $\Delta t = 0.001$  s) in regard to the accuracy of the results. Though here we only depict the case when  $\xi$  is uniformly distributed, similar trends are obtained when other distributions are used. In particular, this figure shows that the discretization of the temporal function space plays an important role when it comes to the FSC-2 approach but not when it comes to the FSC-1 approach. The reason for this is that in FSC-1 the errors coming from the RFS discretization are substantially larger than those coming from the discretization of the temporal function space. This leads to the perception that decreasing the time-step size in FSC-1 does not have a direct effect on improving the accuracy of the results derived from the RFS discretization. Furthermore, we observe that more accurate results are obtained with FSC-2 if the time-step size used for the simulation is progressively decreased.



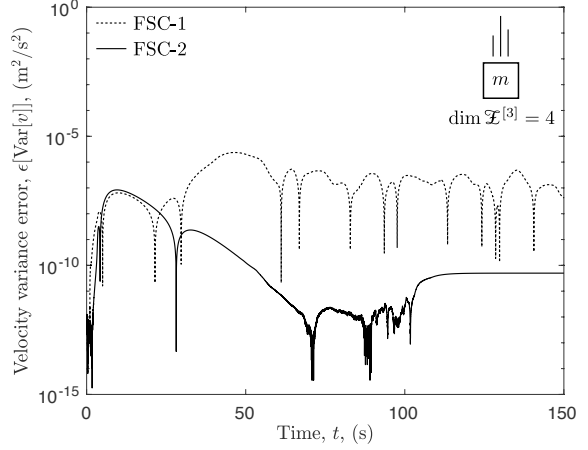
(a) Mean error for  $\mathfrak{I}^{[1]}$



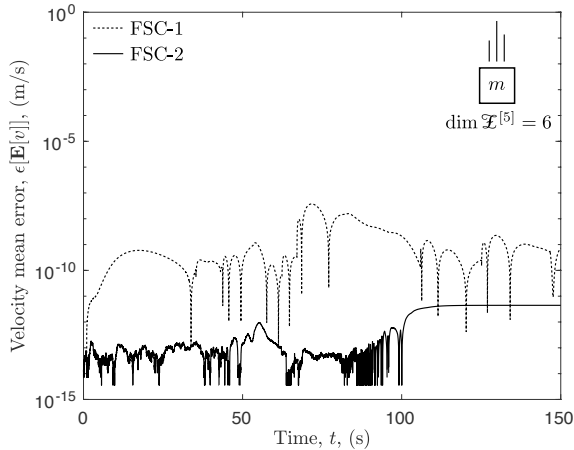
(b) Variance error for  $\mathfrak{I}^{[1]}$



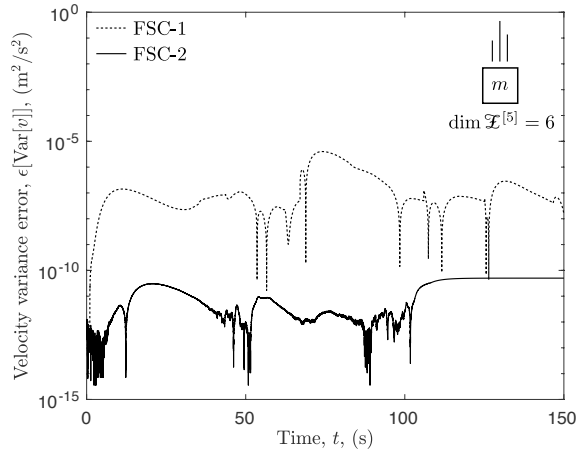
(c) Mean error for  $\mathfrak{I}^{[3]}$



(d) Variance error for  $\mathfrak{I}^{[3]}$

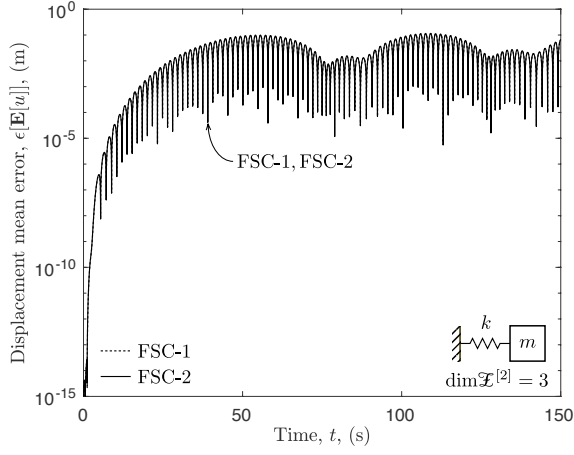


(e) Mean error for  $\mathfrak{I}^{[5]}$

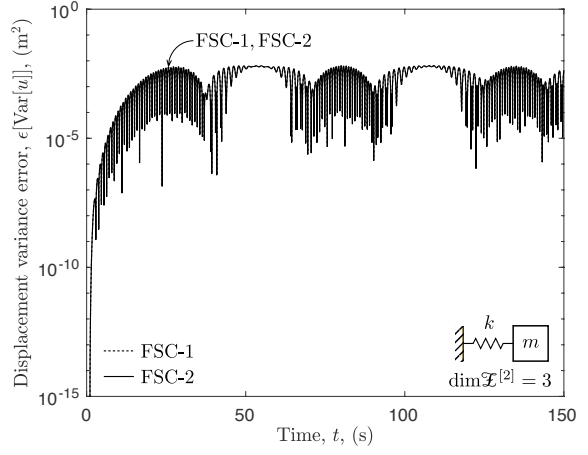


(f) Variance error for  $\mathfrak{I}^{[5]}$

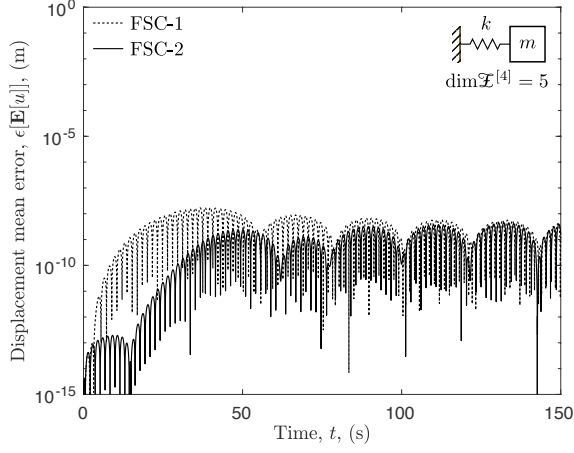
Figure 8: *Problem 1* — Local error evolution of  $\mathbf{E}[v]$  and  $\text{Var}[v]$  for different  $p$ -discretization levels of RFS and for  $\mu \sim \text{Uniform}$



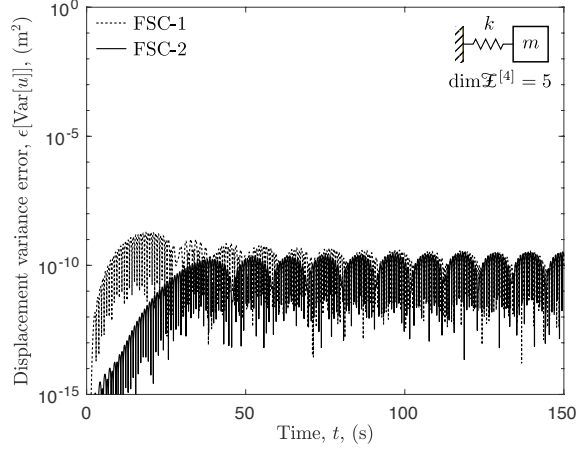
(a) Mean error for  $\mathfrak{X}^{[2]}$



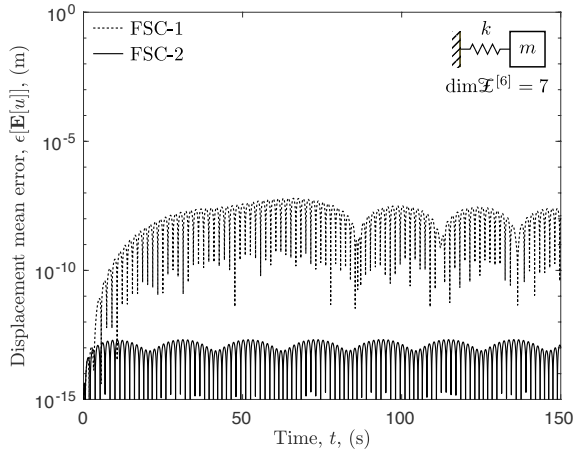
(b) Variance error for  $\mathfrak{X}^{[2]}$



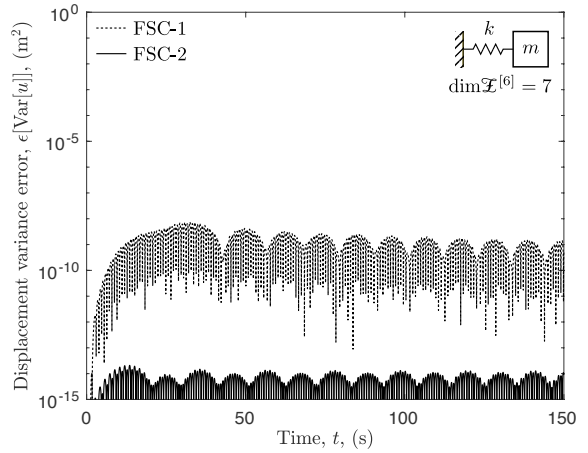
(c) Mean error for  $\mathfrak{X}^{[4]}$



(d) Variance error for  $\mathfrak{X}^{[4]}$

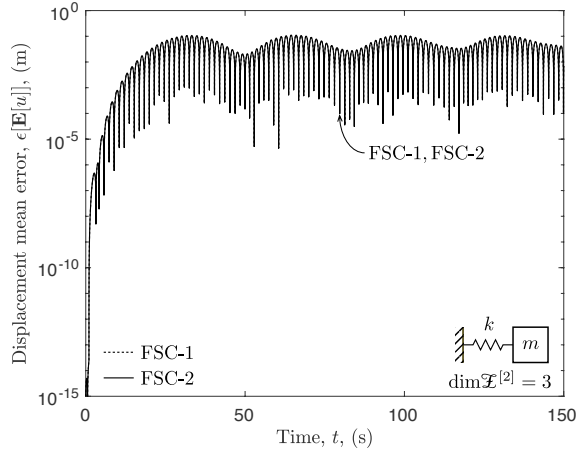


(e) Mean error for  $\mathfrak{X}^{[6]}$

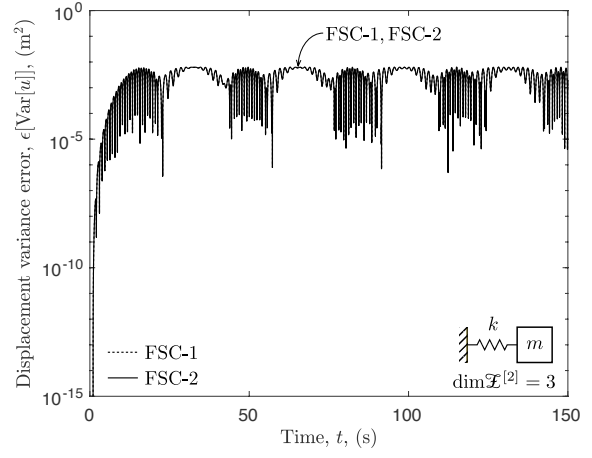


(f) Variance error for  $\mathfrak{X}^{[6]}$

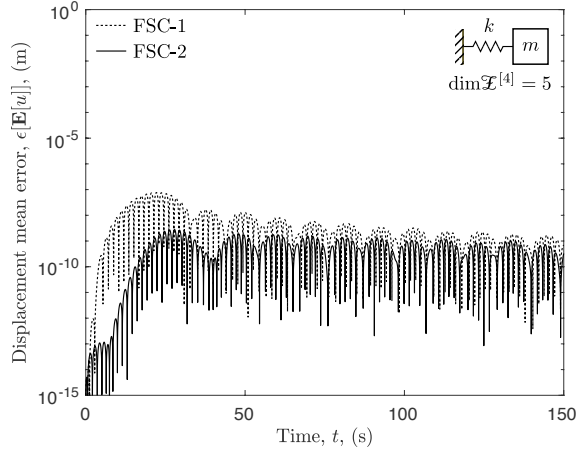
Figure 9: *Problem 2* — Local error evolution of  $\mathbf{E}[u]$  and  $\text{Var}[u]$  for different  $p$ -discretization levels of RFS and for  $\mu \sim \text{Uniform}$



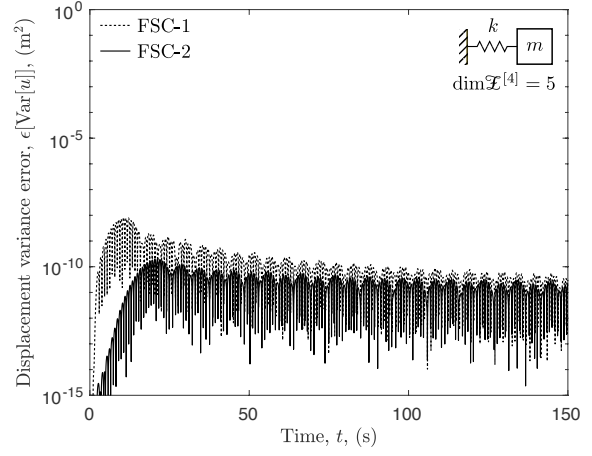
(a) Mean error for  $\mathcal{X}^{[2]}$



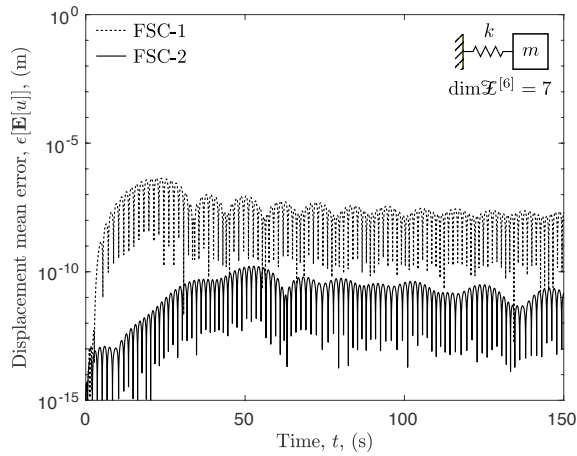
(b) Variance error for  $\mathcal{X}^{[2]}$



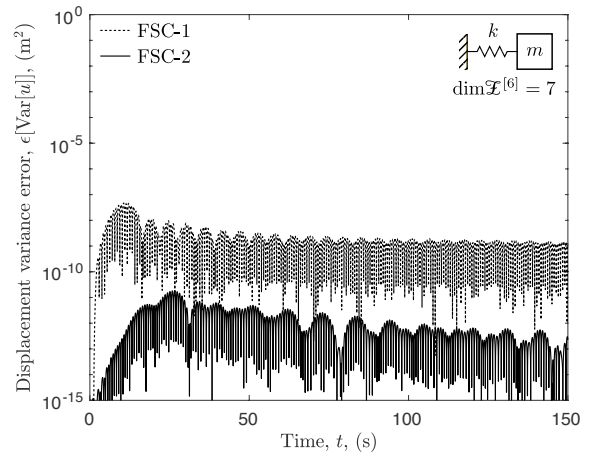
(c) Mean error for  $\mathcal{X}^{[4]}$



(d) Variance error for  $\mathcal{X}^{[4]}$

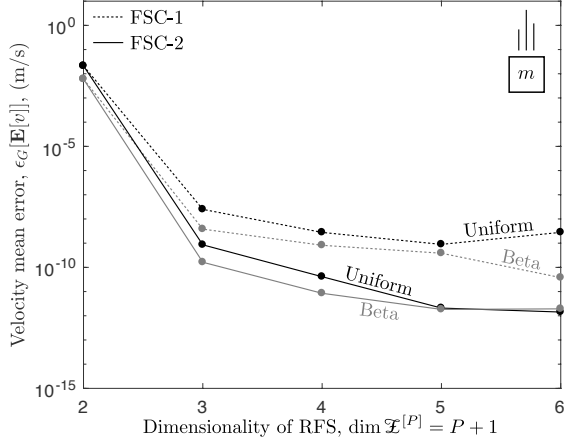


(e) Mean error for  $\mathcal{X}^{[6]}$

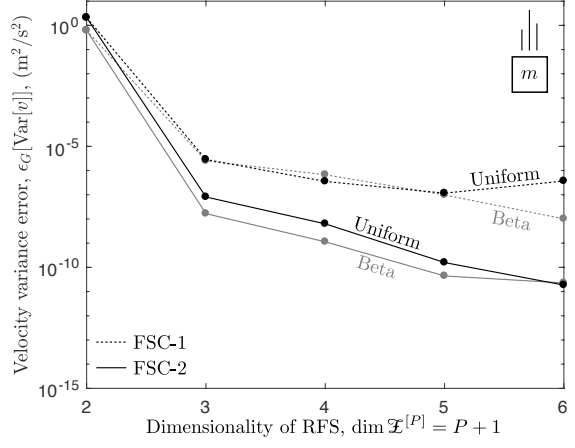


(f) Variance error for  $\mathcal{X}^{[6]}$

Figure 10: *Problem 3* — Local error evolution of  $\mathbf{E}[u]$  and  $\text{Var}[u]$  for different  $p$ -discretization levels of RFS and for  $\mu \sim \text{Uniform} \otimes \text{Uniform}$

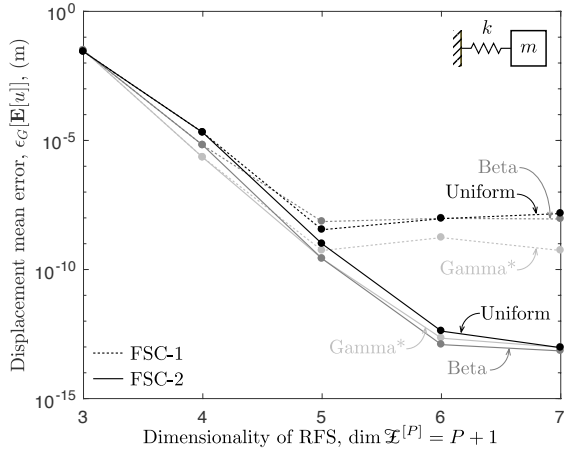


(a) Mean error

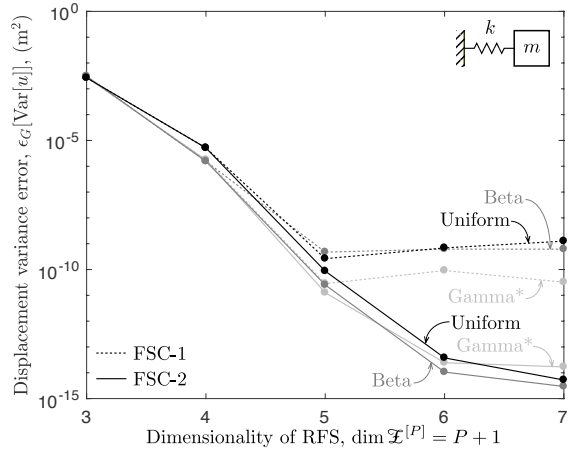


(b) Variance error

Figure 11: *Problem 1* — Global error of  $\mathbf{E}[v]$  and  $\text{Var}[v]$  for different  $p$ -discretization levels of RFS



(a) Mean error



(b) Variance error

Figure 12: *Problem 2* — Global error of  $\mathbf{E}[u]$  and  $\text{Var}[u]$  for different  $p$ -discretization levels of RFS

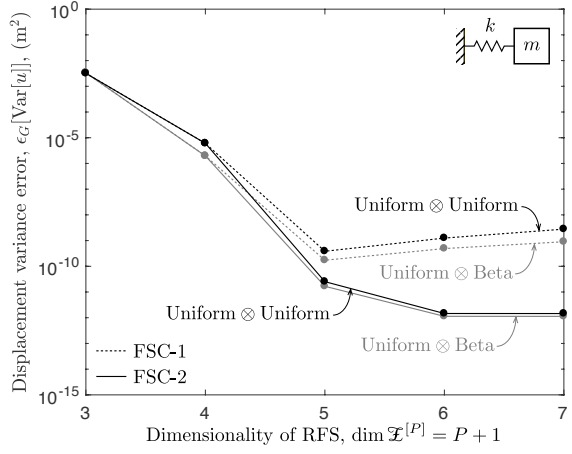
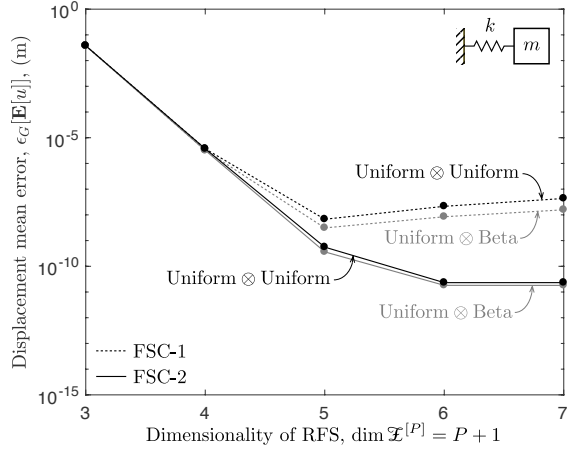


Figure 13: *Problem 3* — Global error of  $\mathbf{E}[u]$  and  $\text{Var}[u]$  for different  $p$ -discretization levels of RFS

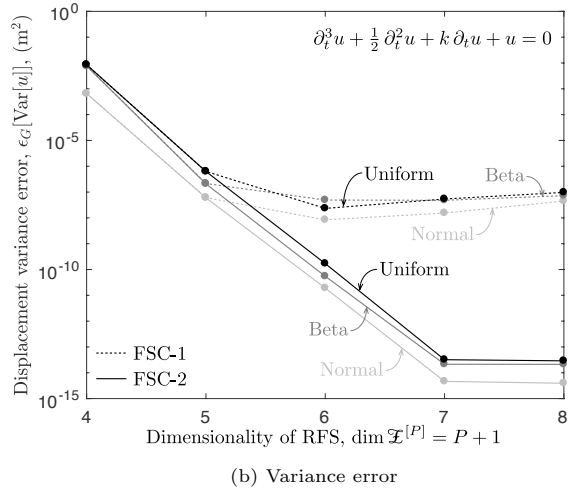
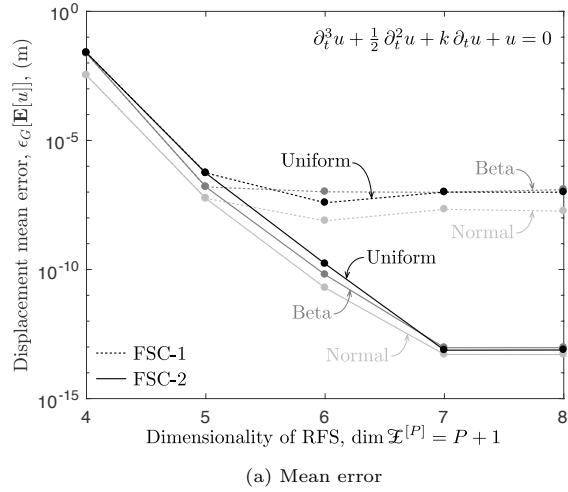
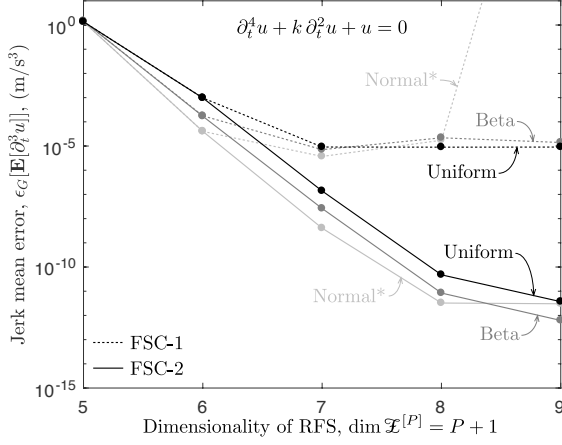
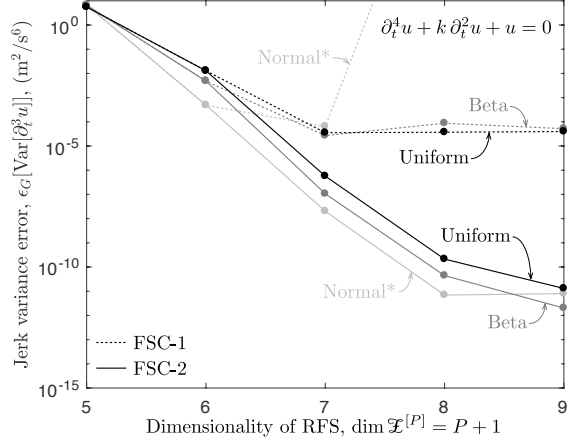


Figure 14: *Problem 4* — Global error of  $\mathbf{E}[u]$  and  $\text{Var}[u]$  for different  $p$ -discretization levels of RFS

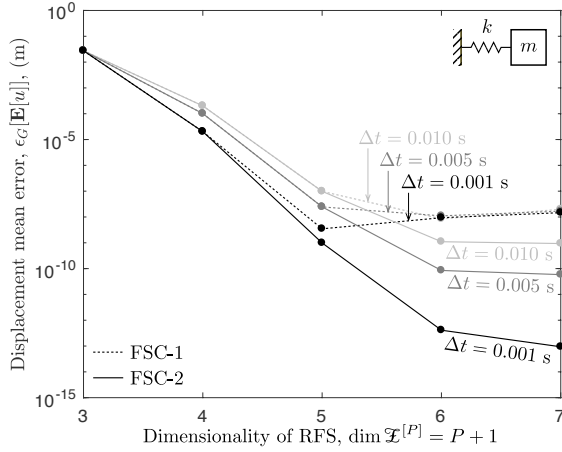


(a) Mean error

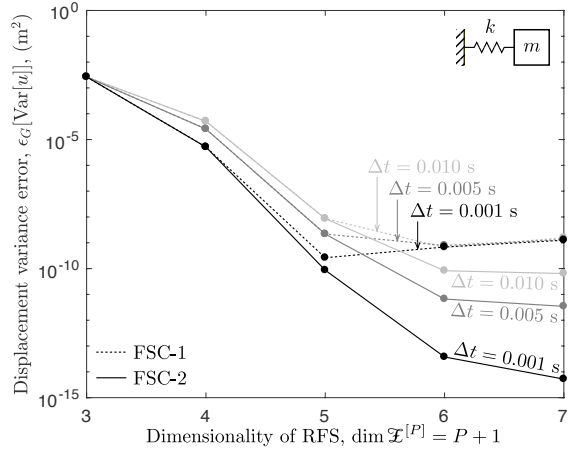


(b) Variance error

Figure 15: *Problem 5* — Global error of  $\mathbf{E}[\partial_t^3 u]$  and  $\text{Var}[\partial_t^3 u]$  for different  $p$ -discretization levels of RFS



(a) Mean error



(b) Variance error

Figure 16: *Problem 2* — Global error of  $\mathbf{E}[u]$  and  $\text{Var}[u]$  for different  $p$ -discretization levels of RFS and time-step sizes ( $\mu \sim \text{Uniform}$ )

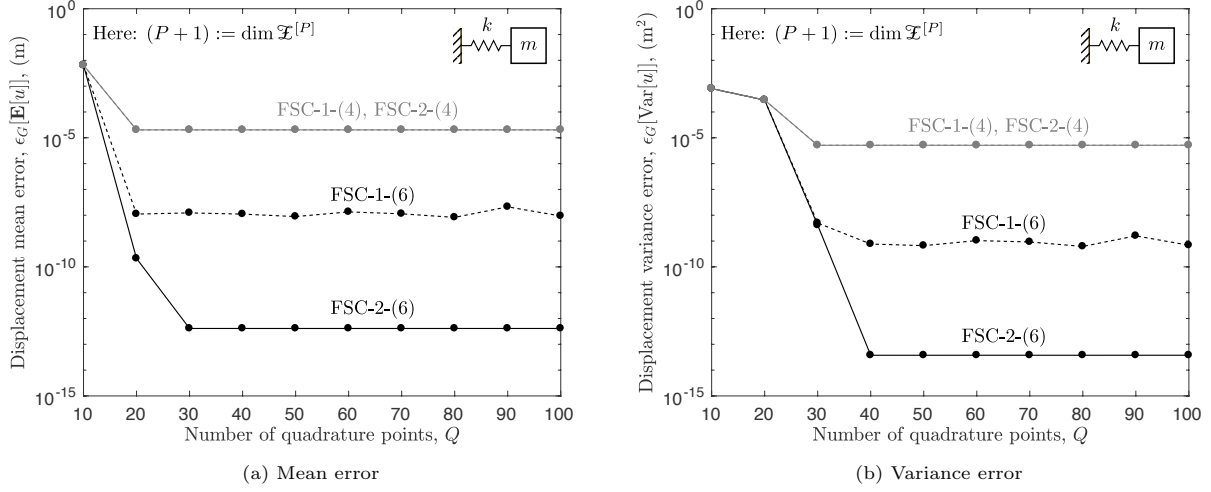


Figure 17: *Problem 2* — Global error of  $\mathbf{E}[u]$  and  $\text{Var}[u]$  as a function of the number of quadrature points employed to estimate the inner products ( $\mu \sim \text{Uniform}$ )

Fig. 17 depicts the convergence of global errors for Problem 2 as a function of the number of quadrature points utilized to estimate the inner products that appear when using the spectral approach. We see that when 6 basis vectors are used to perform the simulation, FSC-2 produces a more accurate result than FSC-1 if the number of quadrature points is sufficiently large. However, no discernible differences between the two FSC approaches are observed if 4 basis vectors are used. In fact, when 10 quadrature points (or even 20 quadrature points for the case of the variance) are utilized, the global error does not improve as a function of the number of basis vectors used or the FSC approach chosen. This is because the global error is in this case dominated by the error coming from the Gaussian quadrature rule rather than by the errors coming from the discretization of the random function space and the transfer of the probability information.

Finally, Fig. 18 plots the global errors as a function of the computational cost associated with FSC-1, FSC-2 and mTD-gPC. This cost is expressed here in terms of the wall-clock time taken to complete the simulation. Labels  $P2Q0$ ,  $P2Q1$  and  $P2Q2$  are defined in Ref. [63] (Pg. 45), and they correspond to the cases when 6, 12 and 18 basis vectors are employed in the simulations, respectively. Some conclusions can be deduced from this figure. First, with respect to achieving a similar level of error, both FSC approaches run much faster than mTD-gPC. In particular, if a global error of approximately  $10^{-8}$  is desired for the simulation, FSC runs about 6 times faster than mTD-gPC. This outcome can be explained by noticing that mTD-gPC requires more than twice the number of basis vectors than FSC. Second, for the same number of basis vectors, the FSC method is able to produce results that are at least 6 orders of magnitude more accurate than mTD-gPC. For example, if the FSC-2 approach is employed with 6 basis vectors, the results are 11 orders of magnitude more accurate than the mTD-gPC counterpart. Therefore, the FSC methods are not only superior in terms of computational efficiency than mTD-gPC, but they also have the ability to encode the probability information a lot better as the simulation proceeds. However, it is important to note that by increasing the number of basis vectors from 5 to 7 for FSC-1 and 12 to 18 for mTD-gPC, the results worsen noticeably by an order of magnitude or so. This is primarily due to the limited precision of the machine and the fact that the probability information is being transferred in the mean-square sense. Moreover, Fig. 18 also reveals that in general FSC-2 runs slightly faster than FSC-1. In fact, the more basis vectors we use, the higher this difference in speed is. Finally, we also observe that when 7 basis vectors are used, FSC-2 is 5 orders of magnitude more accurate than FSC-1.

## 8.2. Numerical results for the nonlinear system

For this section we use the RK4 method with  $\Delta t = 0.005$  s to speed up the simulations. Fig. 19 depicts the evolution of the mean and variance of the oscillator's displacement using FSC-2 and a Monte Carlo



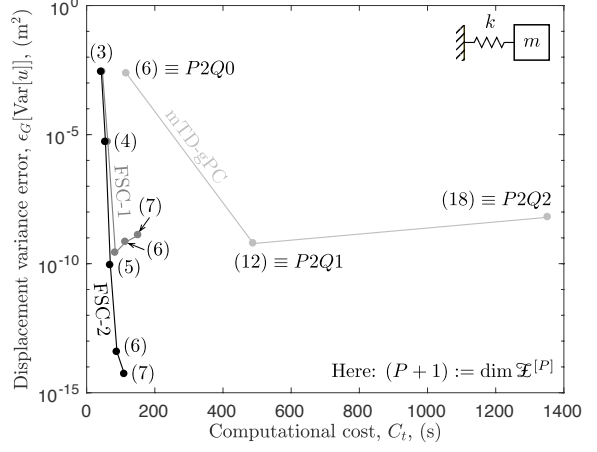
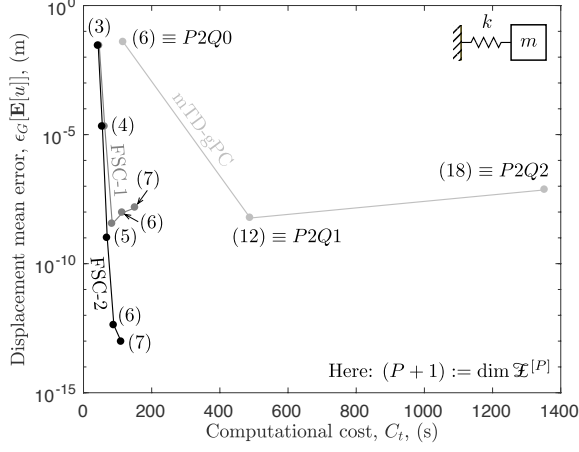


Figure 18: *Problem 2* — Global error of  $\mathbf{E}[u]$  and  $\text{Var}[u]$  versus computational cost for different  $p$ -discretization levels of RFS and for  $\mu \sim \text{Uniform}$

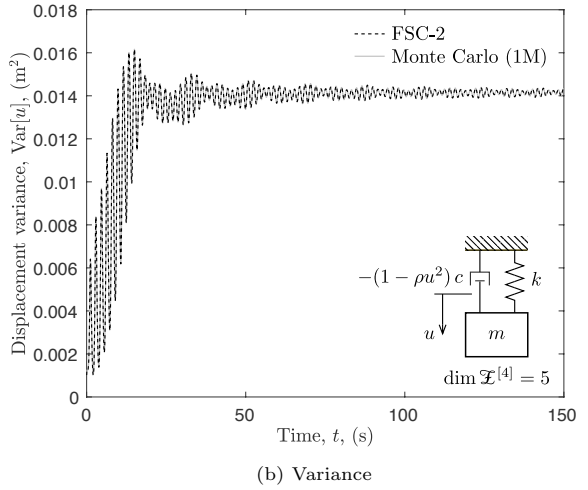
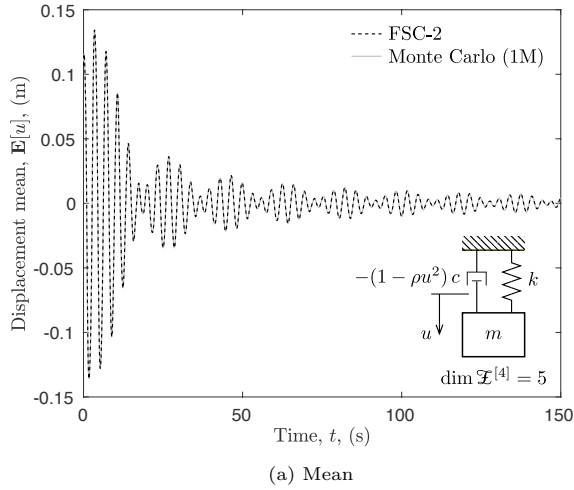


Figure 19: *Problem 6 (the Van-der-Pol oscillator)* — Evolution of  $\mathbf{E}[u]$  and  $\text{Var}[u]$  for the case when the  $p$ -discretization level of RFS is  $\mathfrak{F}^{[4]}$  and  $\mu \sim \text{Uniform} \otimes \text{Beta}$

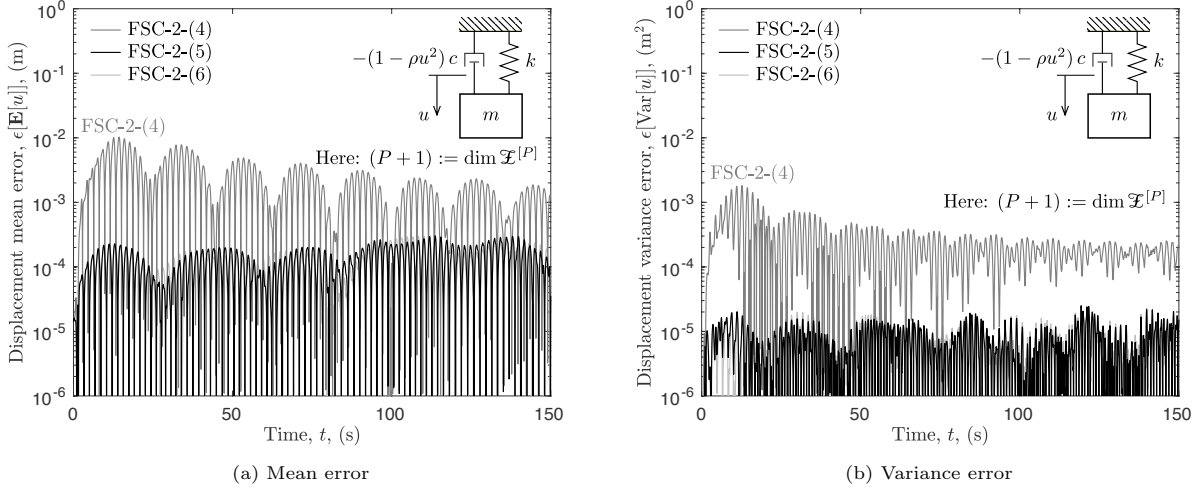


Figure 20: *Problem 6 (the Van-der-Pol oscillator)* — Local error evolution of  $\mathbf{E}[u]$  and  $\text{Var}[u]$  for different  $p$ -discretization levels of RFS with respect to the 1-million Monte Carlo simulation ( $\mu \sim \text{Uniform} \otimes \text{Beta}$ )

simulation with one million realizations. This figure shows that by using only 5 basis vectors, FSC-2 has the ability to reproduce the Monte Carlo response with high fidelity. This is quite a remarkable result, since not only the problem is highly nonlinear over the temporal-random space but it also features a two-dimensional probability space.

Fig. 20 plots the local errors in mean and variance of the oscillator's displacement. We observe that by using 4 basis vectors, an error of  $10^{-2}$  and  $10^{-3}$  can be achieved for the mean and variance, respectively. If more basis vectors are used, say 5 or 6, the corresponding errors cut down two orders of magnitude. This suggests that 5 basis vectors are sufficient to reproduce a Monte Carlo simulation with one million realizations to a comparable level of accuracy.

## 9. A parametric, high-dimensional stochastic problem

In this section, we show that the FSC method does not suffer from the curse of dimensionality at the random-function-space level by solving a parametric, high-dimensional problem. Further, we show that by using the FSC method in conjunction with Monte Carlo integration to compute the inner products, one can overcome the curse of dimensionality at the random-space level as well—effectively eliminating the curse of dimensionality altogether. For this, we investigate the following problem.

### 9.1. Problem statement

Find the displacement of the system  $u : \mathfrak{T} \times \Xi \rightarrow \mathbb{R}$  in  $\mathcal{U}$ , such that ( $\mu$ -a.e.):

$$\ddot{u} + ku = f \quad \text{on } \mathfrak{T} \times \Xi \quad (35a)$$

$$\{u(0, \cdot) = \mathbf{u}, \dot{u}(0, \cdot) = \mathbf{v}\} \quad \text{on } \{0\} \times \Xi, \quad (35b)$$

where  $k, f : \mathfrak{T} \times \Xi \rightarrow \mathbb{R}$  and  $\mathbf{u}, \mathbf{v} : \Xi \rightarrow \mathbb{R}$  are random variables given by:

$$\begin{aligned} k(t, \xi) &= \frac{1}{2400} (\xi^1 + 40) (\xi^6 + \xi^7 + 40) (3 - \exp(-\frac{1}{77}(\xi^2 + 7)t)), \\ f(t, \xi) &= \frac{1}{1200} (\xi^3 + 7) (\xi^8 + 40) ((\xi^9 + \xi^{10}) \sin(\frac{1}{7}\pi t) + 3), \\ \mathbf{u}(\xi) &= \frac{1}{7}(\xi^4 + 8) \quad \text{and} \quad \mathbf{v}(\xi) = \frac{1}{8}(\xi^5 + 7). \end{aligned}$$

In this problem we assume the following probability distribution:  $\text{Beta}^{\otimes 3} \otimes \text{Uniform}^{\otimes d-3} \sim \xi \in \Xi = [-1, 1]^d$ . For the beta distribution, we take  $(\alpha, \beta) = (2, 5)$  as the parameters of the distribution. Since the

random space is a parametric  $d$ -dimensional space, here we study specifically the cases when  $d = 5, 7, 10$ . In particular, when  $d = 5$ , we take  $\xi^6 = \xi^7 = \xi^8 = \xi^9 = \xi^{10} = 0$ , and when  $d = 7$ , we put  $\xi^8 = \xi^9 = \xi^{10} = 0$ . We recall that  $\xi = (\xi^1, \dots, \xi^d)$  for a  $d$ -dimensional space. The spectral discretization of this problem is derived in detail in Appendix B.

## 9.2. Discussion on numerical results

To run all the simulations, we employ the RK4 method with a time-step size of 0.01 s. In Table 2 we have provided the non-orthogonalized version of the random basis for use within the FSC scheme. Because the random space is in this case high-dimensional, the inner products are computed with a Monte Carlo integration using  $10^5$  quadrature points sampled from the random domain. For reference, each FSC simulation is compared below against a Monte Carlo simulation with one million realizations to determine if convergence was achieved for the FSC-2 scheme.

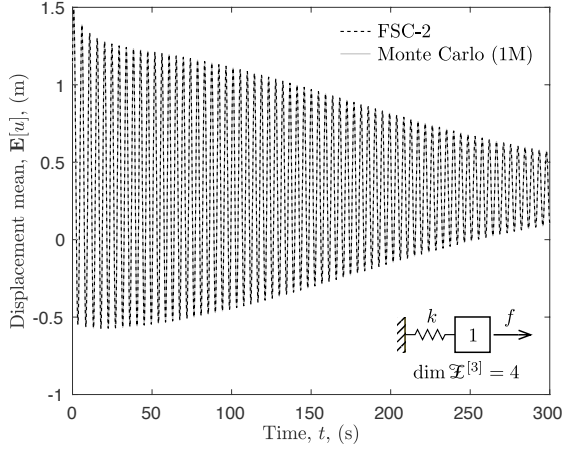
In Fig. 21 we plot the evolution of the mean and variance of the system's displacement for the three stochastic problems considered in this section. Two observations are in order. First, for the smallest RFS that one can construct using the FSC method (i.e.  $\mathcal{X}^{[3]}$  for this problem), the FSC solution is capable of reproducing reasonably well the Monte Carlo solution. Second, as the dimensionality of the random space is increased, we see that there is no need to increase the dimensionality of the RFS to achieve good results. This is different from other spectral methods such as gPC or TD-gPC, which would have required a lot more basis vectors to span the RFS. For example, if a total-order tensor product were utilized to solve the 10-dimensional problem, expression (4) would have indicated the need of using 66 basis vectors to enrich the random basis with quadratic functions. However, as we see in Fig. 21, only 4 basis vectors are needed in the simulations with FSC to achieve solutions that are nearly indistinguishable from the Monte Carlo counterpart. This proves—at least from a numerical standpoint—our assertion that the FSC method does not suffer from the curse of dimensionality at the random-function-space level.

To study now how the number of quadrature points affects the estimation of the inner products with a Monte Carlo integration—and consequently, the FSC results—we run the FSC-2 simulations using five realizations of  $10^2$ ,  $10^3$ ,  $10^4$  and  $10^5$  quadrature points. The resulting solutions are then compared against a reference solution obtained from performing a Monte Carlo simulation with one million realizations. In Fig. 22 we present this study for the three stochastic problems considered in this section by computing the global errors obtained from the mean and variance of the system's displacement. The first observation that we can make is that the variability of the five realizations is in general higher when using fewer quadrature points. This is not surprising since in Monte Carlo integration is well recognized that the fewer the number of quadrature points, the more variation in the results is expected from different realizations of quadrature points. A second observation is that the accuracy of the results improves as the number of quadrature points increases. This is once again expected for it is a direct consequence of the law of large numbers in probability theory. This parametric, high-dimensional stochastic problem has therefore shown that when the dimensionality of the probability space is high, the inner products can be computed with a Monte Carlo integration technique without producing adverse effects, such as non-convergence issues, in the FSC scheme.

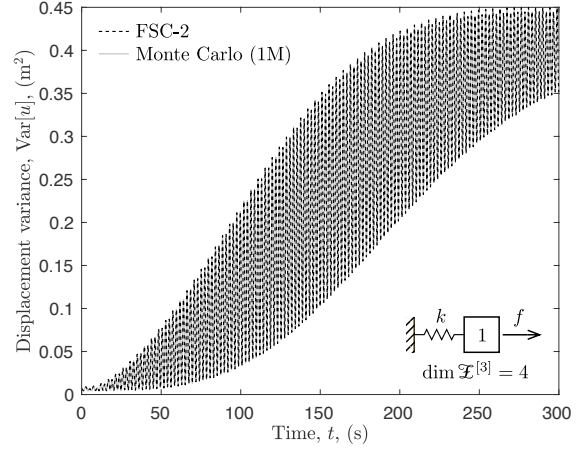
From this section it follows that if we combine the FSC method with a Monte Carlo integration technique, the curse of dimensionality can be fully annihilated at both random levels, namely: at the random-space level and at the random-function-space level. This makes the FSC method more suitable for solving higher-dimensional stochastic problems with the spectral approach.

## 10. Conclusion

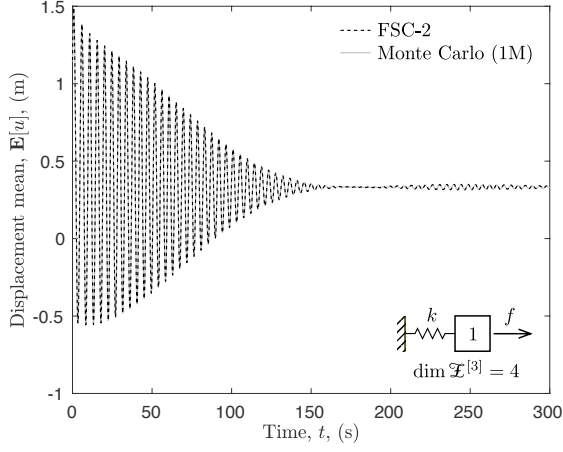
A novel numerical method called the *flow-driven spectral chaos* (FSC) is proposed in this paper to quantify uncertainties in the long-time response of stochastic dynamical systems. In the FSC method, we use the concept of enriched stochastic flow maps to track the evolution of the stochastic part of the solution space efficiently in time. This track is motivated by the fact that the solution of a stochastic ODE and its probability distribution change significantly as a function of time. (It is well-known that when the state of a system is pushed one-time step forward in the simulation, the random basis loses unavoidably its optimality.)



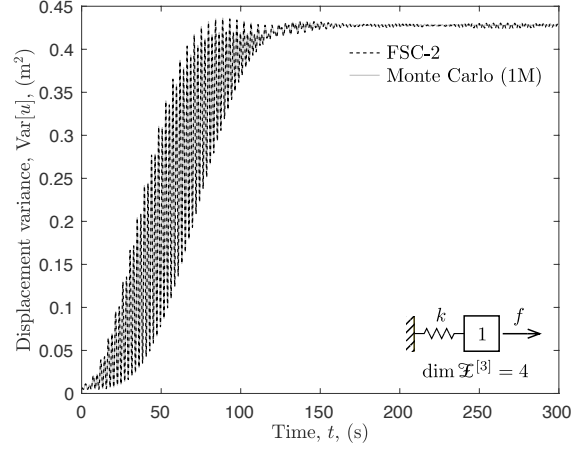
(a) Mean for  $d = 5$



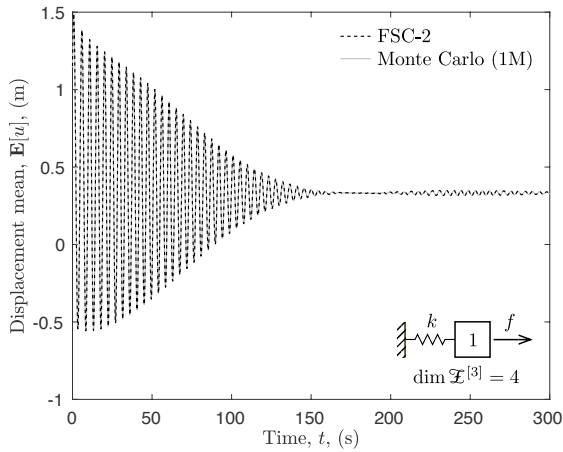
(b) Variance for  $d = 5$



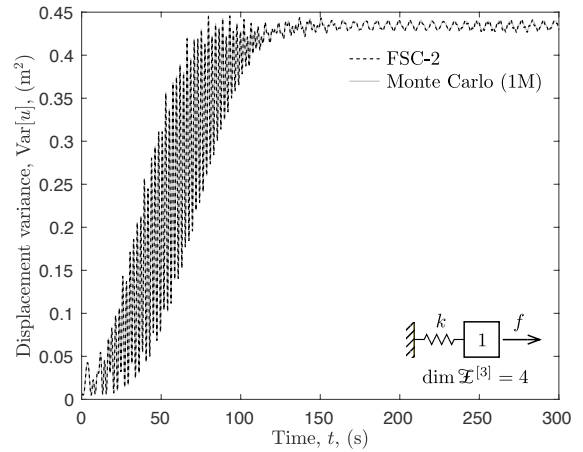
(c) Mean for  $d = 7$



(d) Variance for  $d = 7$



(e) Mean for  $d = 10$



(f) Variance for  $d = 10$

Figure 21: Evolution of  $\mathbf{E}[u]$  and  $\text{Var}[u]$  for the case when the  $p$ -discretization level of RFS is  $\mathcal{X}^{[3]}$  and  $\mu \sim \text{Beta}^{\otimes 3} \otimes \text{Uniform}_{\otimes d-3}$

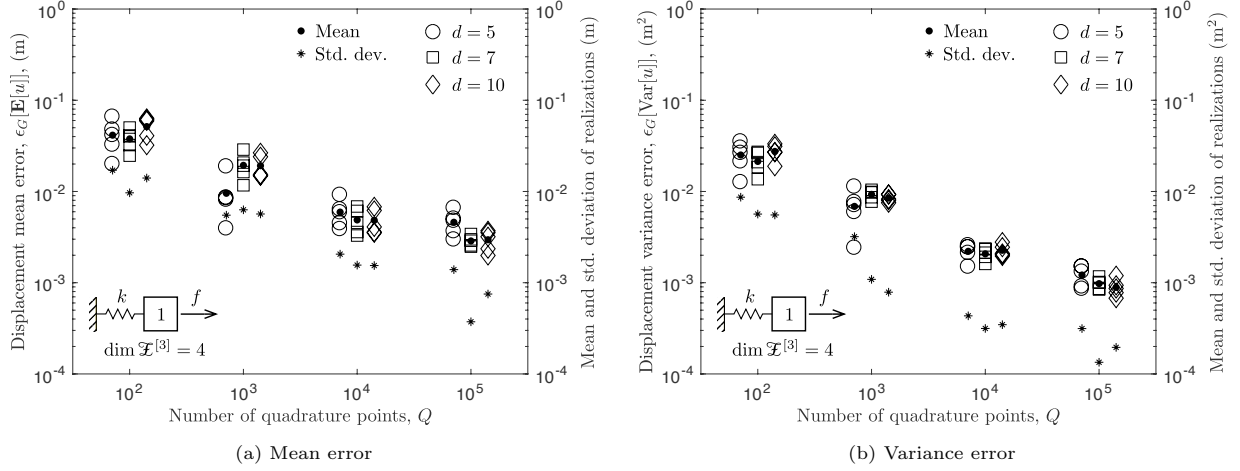


Figure 22: Global error of  $\mathbf{E}[u]$  and  $\text{Var}[u]$  as a function of the number of quadrature points employed to estimate the inner products with a Monte Carlo integration ( $\mu \sim \text{Beta}^{\otimes 3} \otimes \text{Uniform}^{\otimes d-3}$ )

Thus, to resolve this long-time integration issue, we span the stochastic part of the solution space via the time derivatives of the solution at the current time for use within the time step of the simulation. This new way of approaching the problem follows upon noting that a Taylor-series expansion can decompose a stochastic process into an infinite series of functions in the form of a product of a temporal function and a random function (this is of course provided that the process is assumed analytic on the temporal domain). The random functions thus generated are subsequently orthogonalized to serve as a random basis for the solution space. To transfer the probability information at any instant of time, two approaches are developed herein, FSC-1 and FSC-2. The first approach (FSC-1) enforces the probability information to be transferred in the mean-square sense, whereas the second approach (FSC-2) ensures that the probability information is transferred exactly. As discussed in Section 8, the FSC-2 approach has not only the ability to produce results that are more accurate than FSC-1, but also the ability to transfer the probability information faster than FSC-1. This is especially true if the order of the ODE is low (provided we keep the dimensionality of the random function space fixed). Therefore, we suggest using the FSC-2 approach when referring to the FSC method.

We have shown that the FSC method is insensitive to the curse of dimensionality at the random-function-space level. This is because in practice the stochastic flow map of a system is chosen to be finite-accurate, which allows the aforementioned Taylor-series expansion to be truncated up to a specific order. This is in contrast to other methods such as gPC [26] and TD-gPC [62], which use the concept of polynomial chaos and tensor products to find a suitable random basis for the solution space, and which are known to suffer from the curse of dimensionality to some extent. This curse of dimensionality has been regarded as a fundamental issue in methods based on the spectral approach since the introduction of the PC method back in 1938. This paper has addressed this fundamental issue at the random-function-space level.

The FSC method has been applied to six representative problems in this paper. The first five deal with systems governed by a linear stochastic ODE, while the last one is a system governed by a nonlinear stochastic ODE (the Van-der-Pol oscillator). These ODEs were selected to range from first to fourth order so that the performance of the FSC method could be investigated more thoroughly. Based on our findings, we can conclude that FSC outperforms TD-gPC in both accuracy and computational efficiency for solving stochastic dynamical systems with complex physics. Furthermore, in Section 9 we solved three high-dimensional stochastic problems to demonstrate that the curse of dimensionality can be overcome at both, the random-function-space level and the random-space level, by using the FSC method together with Monte Carlo integration to compute the inner products.

## A. The Van-der-Pol oscillator

### A.1. Discretization of the random function space

The random function space for the Van-der-Pol oscillator described in Section 7.6 is discretized here for sake of reference. Let  $\mathcal{F}^{[P]}$  be a finite subspace of  $\mathcal{F}$ . Then, it is clear that the oscillator's displacement can be represented in  $\mathcal{F}^{[P]}$  with:

$$u(t, \xi) \approx u^{[P]}(t, \xi) = \sum_{j=0}^P u^j(t) \Psi_j(\xi) \equiv u^j(t) \Psi_j(\xi). \quad (36)$$

Similarly, let  $\tilde{\mathcal{F}}$  be a finite subspace of  $\mathcal{F}$  to represent the stochastic-input space of the system *exactly*. This subspace is defined as  $\tilde{\mathcal{F}} = \text{span}\{\tilde{\Psi}_m\}_{m=0}^2$ , where  $\tilde{\Psi}_0 \equiv 1$ ,  $\tilde{\Psi}_1 = c - \mathbf{E}[c]$  and  $\tilde{\Psi}_2 = u - \mathbf{E}[u]$ . Hence, we have:

$$c(\xi) = \sum_{m=0}^2 c^m \tilde{\Psi}_m(\xi) \equiv c^m \tilde{\Psi}_m(\xi) \quad \text{and} \quad u(\xi) = \sum_{m=0}^2 u^m \tilde{\Psi}_m(\xi) \equiv u^m \tilde{\Psi}_m(\xi), \quad (37)$$

whence  $c^0 = \mathbf{E}[c]$ ,  $c^1 = 1$ ,  $c^2 = 0$ ,  $u^0 = \mathbf{E}[u]$ ,  $u^1 = 0$  and  $u^2 = 1$  are the coefficients of the finite series.

*Remark 14.* In this problem,  $\dim \tilde{\mathcal{F}} = 3$  because  $c$  and  $u$  are the only independent random variables in the model's input. In a more general setting,  $\dim \tilde{\mathcal{F}} = \tilde{N} + 1$ , where  $\tilde{N}$  represents the number of independent random variables specified in input  $x$  of mathematical model  $\mathcal{M}[u]$ .

Replacing (36) and (37) into (34), and then projecting onto  $\mathcal{F}^{[P]}$ , yields a system of  $P + 1$  second-order ordinary differential equations in the variable  $t$  (with initial conditions):

$$m \mathbb{M}_{j000}^i \ddot{u}^j - \mathbb{M}_{j00m}^i \dot{u}^j c^m + \rho \mathbb{M}_{jklm}^i \dot{u}^j u^k u^l c^m + k \mathbb{M}_{j000}^i u^j = 0 \quad \text{on } \mathfrak{T} \quad (38a)$$

$$\{\mathbb{M}_{j000}^i u^j(0) = \mathbb{M}_{000m}^i u^m, \mathbb{M}_{j000}^i \dot{u}^j(0) = 2 \mathbb{M}_{000m}^i u^m - 0.10 \mathbb{M}_{0000}^i\} \quad \text{on } \{0\}, \quad (38b)$$

where  $\mathbb{M} = \mathbb{M}_{jklm}^i \Psi_i \otimes \Psi_j \otimes \Psi_k \otimes \Psi_l \otimes \tilde{\Psi}_m : \mathcal{F}^{[P]'} \times \mathcal{F}^{[P]} \times \mathcal{F}^{[P]} \times \mathcal{F}^{[P]} \times \tilde{\mathcal{F}} \rightarrow \mathbb{R}$  is the discretized *random tensor*<sup>6</sup> for the whole dynamical system (assumed to exist) given by (a summation sign is implied over every repeated index)

$$\mathbb{M}[\alpha, w, x, y, z] = \mathbb{M}_{jklm}^i \alpha_i w^j x^k y^l z^m,$$

whence  $i, j, k, l \in \{0, 1, \dots, P\}$ ,  $m \in \{0, 1, 2\}$ , and

$$\mathbb{M}_{jklm}^i = \frac{\langle \Psi_i, \Psi_j \Psi_k \Psi_l \tilde{\Psi}_m \rangle}{\langle \Psi_i, \Psi_i \rangle}.$$

This  $\mathbb{M}$  is a tensor of type  $(1, 4)$  and symmetric in the indices  $j, k$  and  $l$ .

However, since  $\mathbb{M}_{j000}^i$  is nothing but  $\delta^i_j$  (the Kronecker delta), system (38) can be restated as follows:

$$\begin{aligned} m \ddot{u}^i - \mathbb{M}_{j00m}^i \dot{u}^j c^m + \rho \mathbb{M}_{jklm}^i \dot{u}^j u^k u^l c^m + k u^i &= 0 \quad \text{on } \mathfrak{T} \\ \{u^i(0) = \mathbb{M}_{000m}^i u^m, \dot{u}^i(0) = 2 \mathbb{M}_{000m}^i u^m - 0.10 \delta^i_0\} &\quad \text{on } \{0\}. \end{aligned}$$

### A.2. Random basis for use in the FSC scheme

Table 1 presents the non-orthogonalized version of the random basis that we use in the simulations with FSC over the region  $\mathfrak{R}_i = \text{cl}(\mathfrak{T}_i) \times \Xi$ .

<sup>6</sup>This tensor is also known as *multiplication tensor* in the literature, see e.g. [18, 19].

Table 1: Non-orthogonalized version of the random basis used for the Van-der-Pol oscillator (Section 7.6)

$\Phi_{0,i}(\xi) := 1$
$\Phi_{1,i}(\xi) := \dot{\varphi}^1(M)(0, \hat{s}(t_i, \xi)) = u_{.i}(t_i, \xi) = u_{.i-1}(t_i, \xi)$
$\Phi_{2,i}(\xi) := \dot{\varphi}^2(M)(0, \hat{s}(t_i, \xi)) = \dot{u}_{.i}(t_i, \xi) = \dot{u}_{.i-1}(t_i, \xi)$
$\Phi_{3,i}(\xi) := \dot{\varphi}^3(M)(0, \hat{s}(t_i, \xi)) = \partial_t^2 u_{.i}(t_i, \xi) = \frac{1}{m} ((1 - \rho u^2_{.i}(t_i, \xi)) c(\xi) \dot{u}_{.i}(t_i, \xi) - k u_{.i}(t_i, \xi))$
$\Phi_{4,i}(\xi) := \dot{\varphi}^4(M)(0, \hat{s}(t_i, \xi)) = \partial_t^3 u_{.i}(t_i, \xi) =$ $\frac{1}{m} (-2\rho c(\xi) u_{.i}(t_i, \xi) \dot{u}^2_{.i}(t_i, \xi) + (1 - \rho u^2_{.i}(t_i, \xi)) c(\xi) \partial_t^2 u_{.i}(t_i, \xi) - k \dot{u}_{.i}(t_i, \xi))$
$\Phi_{5,i}(\xi) := \dot{\varphi}^5(M)(0, \hat{s}(t_i, \xi)) = \partial_t^4 u_{.i}(t_i, \xi) = \frac{1}{m} (-2\rho c(\xi) \dot{u}^3_{.i}(t_i, \xi)$ $- 6\rho c(\xi) u_{.i}(t_i, \xi) \dot{u}_{.i}(t_i, \xi) \partial_t^2 u_{.i}(t_i, \xi) + (1 - \rho u^2_{.i}(t_i, \xi)) c(\xi) \partial_t^3 u_{.i}(t_i, \xi) - k \partial_t^2 u_{.i}(t_i, \xi))$

## B. A parametric, high-dimensional stochastic problem

### B.1. Discretization of the random function space

The random function space for the parametric, high-dimensional problem described in Section 9 is discretized here. This time, however, because there is at least one random variable enclosed in the argument of  $\exp(\cdot)$ , deriving a single random tensor (as we did in Appendix A) to represent the entire stochasticity of the system at hand is not feasible. Instead, we derive below a collection of random tensors for each of the random variables that appear in (35) separately, namely:  $\mathbf{k}(t)$ ,  $\mathbf{f}(t)$ ,  $\mathbf{u}$  and  $\mathbf{v}$ .

As in Appendix A, we take  $\mathcal{X}^{[P]}$  to be a finite subspace of  $\mathcal{X}$ , and the solution representation for (35) as that stipulated by (36). Therefore, replacing (36) into (35), and then projecting onto  $\mathcal{X}^{[P]}$  gives

$$\ddot{u}^i + k^i_j u^j = f^i \quad \text{on } \mathcal{T} \quad (39a)$$

$$\{u^i(0) = \mathbf{u}^i, \dot{u}^i(0) = \mathbf{v}^i\} \quad \text{on } \{0\}, \quad (39b)$$

where  $i, j \in \{0, \dots, P\}$  (summation sign implied over repeated index  $j$ ), and

$$k^i_j(t) = \langle \Psi_i, k(t, \cdot) \Psi_j \rangle / \langle \Psi_i, \Psi_i \rangle, \quad f^i(t) = \langle \Psi_i, f(t, \cdot) \rangle / \langle \Psi_i, \Psi_i \rangle,$$

$$\mathbf{u}^i = \langle \Psi_i, \mathbf{u} \rangle / \langle \Psi_i, \Psi_i \rangle \quad \text{and} \quad \mathbf{v}^i = \langle \Psi_i, \mathbf{v} \rangle / \langle \Psi_i, \Psi_i \rangle.$$

*Remark 15.* Note that the discretized random tensor associated with  $k$  is of type  $(1, 1)$ , i.e.  $\mathbf{k}(t) = k^i_j(t) \Psi_i \otimes \Psi_j : \mathcal{X}^{[P]'} \times \mathcal{X}^{[P]} \rightarrow \mathbb{R}$  (assumed to exist), whereas those associated with  $f$ ,  $\mathbf{u}$  and  $\mathbf{v}$  are of type  $(1, 0)$ , i.e.  $\mathbf{f}(t) = f^i(t) \Psi_i$ ,  $\mathbf{u} = \mathbf{u}^i \Psi_i$ ,  $\mathbf{v} = \mathbf{v}^i \Psi_i : \mathcal{X}^{[P]'} \rightarrow \mathbb{R}$ . Because of this, the last three random tensors can also be regarded as random vectors in  $\mathcal{X}^{[P]}$ , and so we have the following identification:  $\mathbf{f}(t) \mapsto f(t, \cdot)$ ,  $\mathbf{u} \mapsto \mathbf{u}$  and  $\mathbf{v} \mapsto \mathbf{v}$ .

### B.2. Random basis for use in the FSC scheme

Table 2 presents the non-orthogonalized version of the random basis that we use in the simulations with FSC over the region  $\mathcal{R}_i = \text{cl}(\mathcal{T}_i) \times \Xi$ .

Table 2: Non-orthogonalized version of the random basis used for the  $d$ -dimensional stochastic problem (Section 9)

$\Phi_{0,i}(\xi) := 1$
$\Phi_{1,i}(\xi) := \hat{\varphi}^1(M)(0, \hat{s}(t_i, \xi)) = u_{.i}(t_i, \xi) = u_{.i-1}(t_i, \xi)$
$\Phi_{2,i}(\xi) := \hat{\varphi}^2(M)(0, \hat{s}(t_i, \xi)) = \dot{u}_{.i}(t_i, \xi) = \dot{u}_{.i-1}(t_i, \xi)$
$\Phi_{3,i}(\xi) := \hat{\varphi}^3(M)(0, \hat{s}(t_i, \xi)) = \partial_t^2 u_{.i}(t_i, \xi) = f(\xi) - k(t_i, \xi) u_{.i}(t_i, \xi)$
$\Phi_{4,i}(\xi) := \hat{\varphi}^4(M)(0, \hat{s}(t_i, \xi)) = \partial_t^3 u_{.i}(t_i, \xi) = \partial_t f(t_i, \xi) - \partial_t k(t_i, \xi) u_{.i}(t_i, \xi) - k(t_i, \xi) \dot{u}_{.i}(t_i, \xi)$
$\Phi_{5,i}(\xi) := \hat{\varphi}^5(M)(0, \hat{s}(t_i, \xi)) = \partial_t^4 u_{.i}(t_i, \xi) = \partial_t^2 f(t_i, \xi) - \partial_t^2 k(t_i, \xi) u_{.i}(t_i, \xi) - 2 \partial_t k(t_i, \xi) \dot{u}_{.i}(t_i, \xi) - k(t_i, \xi) \partial_t^2 u_{.i}(t_i, \xi)$

### C. Time-complexity analysis for Theorem 1 and the traditional Gram-Schmidt process

In this last section we derive the number of elementary operations<sup>7</sup> that we need to perform in order to obtain an orthogonal basis from a set of linearly independent random functions. We do this for Theorem 1 and the traditional Gram-Schmidt process for sake of comparison. For both analyses, we assume right away that  $\Psi_0 \equiv 1$  and  $\langle \Psi_0, \Psi_0 \rangle = 1$ . Therefore, for the purpose of this section, the basis to orthogonalize takes the form:  $\{\Phi_j\}_{j=1}^P$ , and the resulting orthogonalized basis the form:  $\{\Psi_j\}_{j=0}^P$  with  $\Psi_0 \equiv 1$ .

Below,  $Q$  denotes the number of quadrature points used to estimate the inner products with a quadrature rule such as the Gaussian quadrature rule.

*Theorem 1.* In Table 3 we have disaggregated the number of elementary operations into the different steps involved in Theorem 1. If we assume that both the expectation vector and the covariance matrix are known beforehand, the number of operations needed in Theorem 1 is

$$N_{\text{op}} = \sum_{i=3}^5 N_{\text{op},i} = \begin{cases} 0 & \text{for } P = 0 \\ P(P+1)Q + \frac{1}{30}P^5 + \frac{1}{6}P^4 - \frac{1}{3}P^3 + \frac{1}{3}P^2 - \frac{6}{5}P + 1 & \text{for } P \geq 1. \end{cases} \quad (40)$$

Otherwise, the number of operations needed is

$$N_{\text{op}} = \sum_{i=1}^5 N_{\text{op},i} = \begin{cases} 0 & \text{for } P = 0 \\ \frac{7}{2}P(P + \frac{11}{7})Q + \frac{1}{30}P^5 + \frac{1}{6}P^4 - \frac{1}{3}P^3 - \frac{1}{6}P^2 - \frac{27}{10}P + 1 & \text{for } P \geq 1. \end{cases} \quad (41)$$

*Traditional Gram-Schmidt process.* As with Theorem 1, in Table 4 we have disaggregated into steps the number of elementary operations needed in the traditional Gram-Schmidt process (from (11)), yielding

$$N'_{\text{op}} = \sum_{i=1}^3 N'_{\text{op},i} = \begin{cases} 0 & \text{for } P = 0 \\ \frac{5}{2}(P^2 + \frac{9}{5}P - \frac{6}{5})Q - 2P + 1 & \text{for } P \geq 1. \end{cases} \quad (42)$$

<sup>7</sup>By elementary operations we mean: addition, subtraction, multiplication and division on the real numbers.



Table 3: Number of elementary operations involved in each step of Theorem 1 to get  $\{\Phi_j\}_{j=1}^P \mapsto \{\Psi_j\}_{j=0}^P$

<b>Theorem 1:</b>	
①	Compute mean vector $N_{\text{op},1} := \#(\{\mathbf{E}[\Phi_j]\}_{j=1}^P) = P(2Q - 1) \quad \text{for } P \geq 1$
②	Compute covariance matrix $N_{\text{op},2} := \#(\{\text{Cov}[\Phi_i, \Phi_j]\}_{i,j=1}^P) = \frac{1}{2}P(P+1)(5Q-1) \quad \text{for } P \geq 1$
③	Compute the numerators of determinant ratio $N_{\text{op},3} := \#(\{\det \Delta_k(j)\}_{j=2,k=1}^{P,j-1}) = \frac{1}{30}P(P-1)(P-2)(P^2+3P+7) \quad \text{for } P \geq 2$
④	Compute the denominators of determinant ratio $N_{\text{op},4} := \#(\{\det \square_k\}_{k=1}^{P-1}) = \frac{1}{6}(P-1)(P-2)(P^2+P+3) \quad \text{for } P \geq 2$
⑤	Compute orthogonal basis $N_{\text{op},5} := \#(\{\Psi_j\}_{j=0}^P) = \frac{1}{2}P((P+1)(2Q+1)-2) \quad \text{for } P \geq 1$
<b>Remark:</b>	
•	Computation of mean $\# \left( \mathbf{E}[f] \approx \sum_{i=1}^Q f(\xi_i) w_i \right) = 2Q - 1$
•	Computation of covariance $\# \left( \text{Cov}[f, g] \approx \sum_{i=1}^Q (f(\xi_i) - \mathbf{E}[f])(g(\xi_i) - \mathbf{E}[g]) w_i \right) = 5Q - 1$
•	Computation of determinant $\#(\det \mathbf{A} = \det(\mathbf{LU}) = \det \mathbf{U}) = 2 \sum_{i=1}^n (n-i)(n-i+1) + n - 1 = \frac{1}{3}(2n^3 + n - 3)$
Here: $\mathbf{A} \in \mathcal{M}(n \times n, \mathbb{R})$ , and $\mathbf{LU}$ is the LU-decomposition of $\mathbf{A}$	

Table 4: Number of elementary operations involved in each step of the traditional Gram-Schmidt process to get  $\{\Phi_j\}_{j=1}^P \mapsto \{\Psi_j\}_{j=0}^P$

<b>Traditional Gram-Schmidt process:</b>	
①	Compute the numerators of inner-product ratio $N'_{\text{op},1} := \#(\{\langle \Phi_j, \Psi_k \rangle\}_{j=1,k=0}^{P,j-1}) = \frac{1}{2}P(2PQ + (P+1)(Q-1)) \quad \text{for } P \geq 1$
②	Compute the denominators of inner-product ratio $N'_{\text{op},2} := \#(\{\langle \Psi_k, \Psi_k \rangle\}_{k=0}^{P-1}) = (P-1)(3Q-1) \quad \text{for } P \geq 1$
③	Compute orthogonal basis $N'_{\text{op},3} := \#(\{\Psi_j\}_{j=0}^P) = \frac{1}{2}P((P+1)(2Q+1)-2) \quad \text{for } P \geq 1$

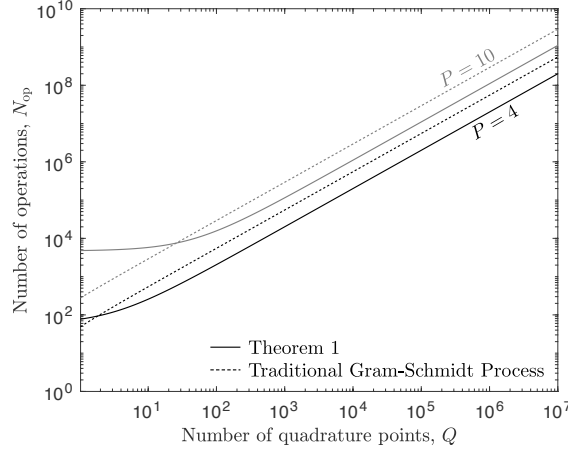


Figure 23: Time-complexity analysis for Theorem 1 and the traditional Gram-Schmidt process

*Performance comparison.* Combining the result obtained in (40) with that in (42) produces the plot depicted in Fig. 23. From this figure we can see that as the number of quadrature points increases, the time complexity of Theorem 1 decreases relative to that of the traditional Gram-Schmidt process. In other words, when the number of quadrature points is sufficiently large, Theorem 1 surpasses in efficiency the traditional Gram-Schmidt process. In particular, if  $P = 3, 4$ , Theorem 1 is up to 2.75 times faster. However, if the expectation vector or the covariance matrix are not known beforehand, Theorem 1 is slower than the traditional Gram-Schmidt process, because the  $Q$ -coefficient in (41) is always larger than that in (42) for all values of  $P$ .

## Acknowledgements

The first author gratefully acknowledges the scholarship granted by Colciencias and Atlántico Department (Colombia) under Call 673 to pursue a Ph.D. degree in Structural Engineering at Purdue University, West Lafayette, IN. The authors also gratefully acknowledge the support of the National Science Foundation (CNS-1136075, DMS-1555072, DMS-1736364, CMMI-1634832, and CMMI-1560834), Brookhaven National Laboratory subcontract 382247, ARO/MURI grant W911NF-15-1-0562, and U.S. Department of Energy (DOE) Office of Science Advanced Scientific Computing Research program DE-SC0021142.

## Credit author statement

**Hugo Esquivel:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing–‘original draft’, Writing–‘review and editing’. **Arun Prakash:** Funding acquisition, Project administration, Resources, Supervision, Writing–‘review and editing’. **Guang Lin:** Funding acquisition, Project administration, Resources, Supervision, Writing–‘review and editing’.

## References

- [1] D. P. Kroese, T. Taimre, Z. I. Botev, Handbook of Monte Carlo methods, Vol. 706, John Wiley & Sons, 2013.
- [2] G. Fishman, Monte Carlo: Concepts, algorithms, and applications, Springer Science & Business Media, 2013.
- [3] R. Y. Rubinstein, D. P. Kroese, Simulation and the Monte Carlo method, Vol. 10, John Wiley & Sons, 2016.
- [4] 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.
- [5] 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.

- [6] J. Foo, X. Wan, G. E. Karniadakis, The multi-element probabilistic collocation method (ME-PCM): Error analysis and applications, *Journal of Computational Physics* 227 (22) (2008) 9572–9595.
- [7] J. Foo, G. E. Karniadakis, Multi-element probabilistic collocation method in high dimensions, *Journal of Computational Physics* 229 (5) (2010) 1536–1557.
- [8] A. L. Teckentrup, P. Jantsch, C. G. Webster, M. Gunzburger, A multilevel stochastic collocation method for partial differential equations with random input data, *SIAM/ASA Journal on Uncertainty Quantification* 3 (1) (2015) 1046–1074.
- [9] J. D. Collins, W. T. Thomson, The eigenvalue problem for structural systems with statistical properties, *AIAA journal* 7 (4) (1969) 642–648.
- [10] W. K. Liu, T. Belytschko, A. Mani, Probabilistic finite elements for nonlinear structural dynamics, *Computer Methods in Applied Mechanics and Engineering* 56 (1) (1986) 61–81.
- [11] W. K. Liu, T. Belytschko, A. Mani, Random field finite elements, *International journal for numerical methods in engineering* 23 (10) (1986) 1831–1845.
- [12] M. Kleiber, T. D. Hien, *The stochastic finite element method: Basic perturbation technique and computer implementation*, Wiley, 1992.
- [13] M. Shinozuka, G. Deodatis, Response variability of stochastic finite element systems, *Journal of Engineering Mechanics* 114 (3) (1988) 499–519.
- [14] F. Yamazaki, M. Shinozuka, G. Dasgupta, Neumann expansion for stochastic finite element analysis, *Journal of Engineering Mechanics* 114 (8) (1988) 1335–1354.
- [15] G. Deodatis, Weighted integral method. I: Stochastic stiffness matrix, *Journal of Engineering Mechanics* 117 (8) (1991) 1851–1864.
- [16] G. Deodatis, M. Shinozuka, Weighted integral method. II: Response variability and reliability, *Journal of Engineering Mechanics* 117 (8) (1991) 1865–1877.
- [17] D. Xiu, *Numerical Methods for Stochastic Computations: A Spectral Method Approach*, Princeton University Press, 2010.
- [18] O. P. Le Maître, O. M. Knio, *Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics*, Springer Science & Business Media, 2010.
- [19] T. J. Sullivan, *Introduction to Uncertainty Quantification*, Vol. 63, Springer, 2015.
- [20] N. Wiener, The homogeneous chaos, *American Journal of Mathematics* 60 (4) (1938) 897–936.
- [21] N. Wiener, A. Wintner, The discrete chaos, *American Journal of Mathematics* 65 (2) (1943) 279–298.
- [22] N. Wiener, *Nonlinear problems in random theory*, MIT press, 1958.
- [23] V. Volterra, M. Tomassetti, F. Zarlatti, *Leçons sur les équations intégrales et les équations intégro-différentielles*, Gauthier-Villars Paris, 1913.
- [24] R. H. Cameron, W. T. Martin, The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals, *Annals of Mathematics* (1947) 385–392.
- [25] D. Lucor, D. Xiu, G. Karniadakis, Spectral representations of uncertainty in simulations: Algorithms and applications, in: *Proceedings of the International Conference on Spectral and High Order Methods (ICOSAHOM-01)*, Uppsala, Sweden, 2001.
- [26] D. Xiu, G. E. Karniadakis, The Wiener-Askey polynomial chaos for stochastic differential equations, *SIAM journal on scientific computing* 24 (2) (2002) 619–644.
- [27] R. Field, M. Grigoriu, A new perspective on polynomial chaos, *Computational Stochastic Mechanics*, Spanos & Deodatis eds, Millpress (2003) 199–205.
- [28] K. Itô, Multiple Wiener integral, *Journal of the Mathematical Society of Japan* 3 (1) (1951) 157–169.
- [29] H. Ogura, Orthogonal functionals of the Poisson process, *IEEE Transactions on Information Theory* 18 (4) (1972) 473–481.
- [30] A. Segall, T. Kailath, Orthogonal functionals of independent-increment processes, *IEEE Transactions on Information Theory* 22 (3) (1976) 287–298.
- [31] M. Schetzen, Nonlinear system modeling based on the Wiener theory, *Proceedings of the IEEE* 69 (12) (1981) 1557–1573.
- [32] D. D. Engel, *The multiple stochastic integral*, Vol. 265, American Mathematical Soc., 1982.
- [33] R. Ghanem, P. D. Spanos, Polynomial chaos in stochastic finite elements, *Journal of Applied Mechanics* 57 (1) (1990) 197–202.
- [34] R. G. Ghanem, P. D. Spanos, *Stochastic Finite Elements: A Spectral Approach*, Springer-Verlag New York, 1991.
- [35] R. Ghanem, P. D. Spanos, A stochastic Galerkin expansion for nonlinear random vibration analysis, *Probabilistic Engineering Mechanics* 8 (3-4) (1993) 255–264.
- [36] R. Ghanem, Stochastic finite elements with multiple random non-Gaussian properties, *Journal of Engineering Mechanics* 125 (1) (1999) 26–40.
- [37] O. M. Knio, H. N. Najm, R. G. Ghanem, et al., A stochastic projection method for fluid flow. I: Basic formulation, *Journal of computational Physics* 173 (2) (2001) 481–511.
- [38] O. P. Le Maître, M. T. Reagan, H. N. Najm, R. G. Ghanem, O. M. Knio, A stochastic projection method for fluid flow. II: Random process, *Journal of computational Physics* 181 (1) (2002) 9–44.
- [39] D. Xiu, G. E. Karniadakis, Modeling uncertainty in flow simulations via generalized polynomial chaos, *Journal of computational physics* 187 (1) (2003) 137–167.
- [40] B. V. Asokan, N. Zabarar, Using stochastic analysis to capture unstable equilibrium in natural convection, *Journal of Computational Physics* 208 (1) (2005) 134–153.
- [41] O. Knio, O. Le Maître, Uncertainty propagation in CFD using polynomial chaos decomposition, *Fluid Dynamics Research* 38 (9) (2006) 616.
- [42] H. N. Najm, Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics, *Annual review of fluid mechanics* 41 (2009) 35–52.

- [43] X. Wan, G. E. Karniadakis, An adaptive multi-element generalized polynomial chaos method for stochastic differential equations, *Journal of Computational Physics* 209 (2) (2005) 617–642.
- [44] X. Wan, G. E. Karniadakis, Beyond Wiener-Askey expansions: Handling arbitrary PDFs, *Journal of Scientific Computing* 27 (1-3) (2006) 455–464.
- [45] X. Wan, G. E. Karniadakis, Multi-element generalized polynomial chaos for arbitrary probability measures, *SIAM Journal on Scientific Computing* 28 (3) (2006) 901–928.
- [46] X. Ma, N. Zabarar, An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations, *Journal of Computational Physics* 228 (8) (2009) 3084–3113.
- [47] J. D. Jakeman, A. Narayan, D. Xiu, Minimal multi-element stochastic collocation for uncertainty quantification of discontinuous functions, *Journal of Computational Physics* 242 (2013) 790–808.
- [48] X. Wan, G. E. Karniadakis, Long-term behavior of polynomial chaos in stochastic flow simulations, *Computer methods in applied mechanics and engineering* 195 (41-43) (2006) 5582–5596.
- [49] N. Agarwal, N. R. Aluru, A domain adaptive stochastic collocation approach for analysis of MEMS under uncertainties, *Journal of Computational Physics* 228 (20) (2009) 7662–7688.
- [50] G. Kewlani, K. Iagnemma, A multi-element generalized polynomial chaos approach to analysis of mobile robot dynamics under uncertainty, in: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2009, pp. 1177–1182.
- [51] P. Prempraneerach, F. S. Hover, M. S. Triantafyllou, G. E. Karniadakis, Uncertainty quantification in simulations of power systems: Multi-element polynomial chaos methods, *Reliability Engineering & System Safety* 95 (6) (2010) 632–646.
- [52] S. Oladyshkin, H. Class, R. Helmig, W. Nowak, A concept for data-driven uncertainty quantification and its application to carbon dioxide storage in geological formations, *Advances in Water Resources* 34 (11) (2011) 1508–1518.
- [53] G. Kewlani, J. Crawford, K. Iagnemma, A polynomial chaos approach to the analysis of vehicle dynamics under uncertainty, *Vehicle system dynamics* 50 (5) (2012) 749–774.
- [54] T. P. Sapsis, P. F. Lermusiaux, Dynamically orthogonal field equations for continuous stochastic dynamical systems, *Physica D: Nonlinear Phenomena* 238 (23) (2009) 2347–2360.
- [55] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, 1965.
- [56] P. Holmes, J. L. Lumley, G. Berkooz, C. W. Rowley, *Turbulence, coherent structures, dynamical systems and symmetry*, Cambridge university press, 2012.
- [57] M. Choi, T. P. Sapsis, G. E. Karniadakis, A convergence study for SPDEs using combined polynomial chaos and dynamically-orthogonal schemes, *Journal of Computational Physics* 245 (2013) 281–301.
- [58] M. Ueckermann, P. F. Lermusiaux, T. P. Sapsis, Numerical schemes for dynamically orthogonal equations of stochastic fluid and ocean flows, *Journal of Computational Physics* 233 (2013) 272–294.
- [59] M. Cheng, T. Y. Hou, Z. Zhang, A dynamically bi-orthogonal method for time-dependent stochastic partial differential equations. I: Derivation and algorithms, *Journal of Computational Physics* 242 (2013) 843–868.
- [60] M. Cheng, T. Y. Hou, Z. Zhang, A dynamically bi-orthogonal method for time-dependent stochastic partial differential equations. II: Adaptivity and generalizations, *Journal of Computational Physics* 242 (2013) 753–776.
- [61] E. Musharbash, F. Nobile, T. Zhou, Error analysis of the dynamically orthogonal approximation of time dependent random PDEs, *SIAM Journal on Scientific Computing* 37 (2) (2015) A776–A810.
- [62] 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.
- [63] V. Heuveline, M. Schick, A hybrid generalized polynomial chaos method for stochastic dynamical systems, *International Journal for Uncertainty Quantification* 4 (1) (2014).
- [64] D. M. Luchtenburg, S. L. Brunton, C. W. Rowley, Long-time uncertainty propagation using generalized polynomial chaos and flow map composition, *Journal of Computational Physics* 274 (2014) 783–802.
- [65] H. C. Ozen, G. Bal, Dynamical polynomial chaos expansions and long time evolution of differential equations with random forcing, *SIAM/ASA Journal on Uncertainty Quantification* 4 (1) (2016) 609–635.
- [66] P. J. Davis, P. Rabinowitz, *Methods of numerical integration*, Courier Corporation, 2007.
- [67] E. Novak, K. Ritter, High dimensional integration of smooth functions over cubes, *Numerische Mathematik* 75 (1) (1996) 79–97.
- [68] H.-J. Bungartz, M. Griebel, Sparse grids, *Acta numerica* 13 (2004) 147–269.
- [69] G. Leobacher, F. Pillichshammer, *Introduction to quasi-Monte Carlo integration and applications*, Springer, 2014.
- [70] W. Rudin, *Real and complex analysis*, McGraw-Hill, 1987.
- [71] W. Cheney, D. Kincaid, *Linear algebra: Theory and applications*, 2nd Edition, Jones & Bartlett Learning, 2010.
- [72] S. A. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, in: *Doklady Akademii Nauk*, Vol. 148, Russian Academy of Sciences, 1963, pp. 1042–1045.
- [73] J. C. Butcher, J. Butcher, *The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods*, Vol. 512, Wiley New York, 1987.
- [74] The MathWorks Inc., MATLAB (2016).  
URL [www.mathworks.com](http://www.mathworks.com)

## Errata

This updated version of the article fixes a typo found on Page 5<sup>8</sup>.

Throughout this paper, we assume that the components of the  $d$ -tuple random variable  $\xi = (\xi^1, \dots, \xi^d)$  are ~~pairwise~~ mutually independent and that the random domain  $\Xi$  is a hypercube of  $d$  dimensions obtained by performing a  $d$ -fold Cartesian product of intervals  $\bar{\Xi}_i := \xi^i(\Omega)$ .

---

<sup>8</sup>Also on Page 5 of the journal article.