EFFICIENT SPECTRAL-CHAOS METHODS FOR UNCERTAINTY QUANTIFICATION IN LONG-TIME RESPONSE OF STOCHASTIC DYNAMICAL SYSTEMS

by

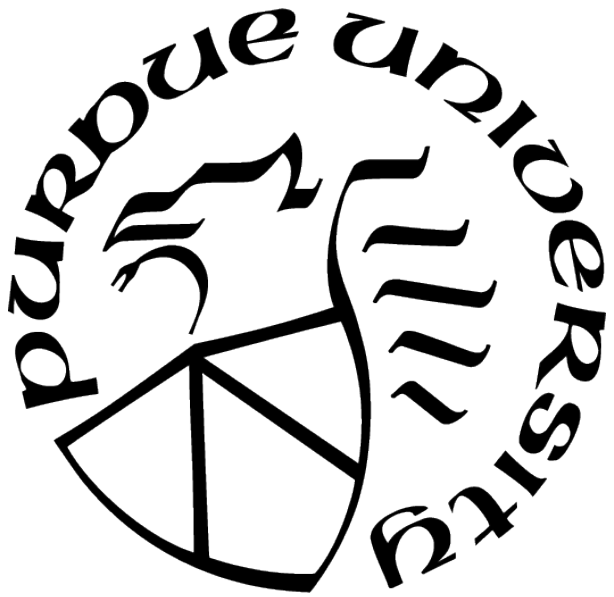
Hugo Raul Esquivel Otero

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy



Lyles School of Civil Engineering

West Lafayette, Indiana

May 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Arun Prakash, Co-Chair

Lyles School of Civil Engineering

Dr. Guang Lin, Co-Chair

Department of Mathematics & School of Mechanical Engineering

Dr. Alina Alexeenko

School of Aeronautics and Astronautics & Davidson School of Chemical Engineering

Dr. Ayhan Irfanoglu

Lyles School of Civil Engineering

Dr. Shirley Dyke

School of Mechanical Engineering & Lyles School of Civil Engineering

Approved by:

Dr. Dulcy Abraham

To Lord Jesus Christ and my family.

ACKNOWLEDGMENTS

First of all, the completion of this work would have not been possible without the unconditional support of my family—especially that from my mother Hilda and sister Ana. I am truly grateful to Dr. Arun Prakash and Dr. Guang Lin—my supervisors—for having shared their expertise and feedback with me during my journey as a doctoral researcher at Purdue. Their support and guidance are greatly appreciated throughout all these years of hard work. I also want to extend my sincere thanks to my doctoral dissertation committee for their advice and suggestions on how to further improve the final draft of my dissertation. I am deeply indebted to Colciencias and Atlántico Department for having granted me a scholarship under Call 673 to pursue a doctoral degree in Structural Engineering at Purdue. Their financial funding for my first three years at Purdue is greatly appreciated and acknowledged. Thanks should also go to Colfuturo for their wonderful job at administering the scholarship. Especially, I want to thank Constanza Londoño for having been so responsive and supportive in many times of need. I would also like to extend my deepest gratitude to the Lyles School of Civil Engineering for having supported my doctoral research through the grant of many teaching assistantships and graduate award fellowships while at Purdue. Finally, many thanks to all my friends who were with me every step of the way on this journey...

Hugo Esquivel

West Lafayette, Indiana

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Uncertainty quantification techniques based on the spectral approach have been studied extensively in the literature to characterize and quantify, at low computational cost, the impact that uncertainties may have on large-scale engineering problems. One such technique is the *generalized polynomial chaos* (gPC) which utilizes a time-independent orthogonal basis to expand a stochastic process in the space of random functions. The method uses a specific Askey-chaos system that is concordant with the measure defined in the probability space in order to ensure exponential convergence to the solution. For nearly two decades, this technique has been used widely by several researchers in the area of uncertainty quantification to solve stochastic problems using the spectral approach. However, a major drawback of the gPC method is that it cannot be used in the resolution of problems that feature strong nonlinear dependencies over the probability space as time progresses. Such downside arises due to the time-independent nature of the random basis, which has the undesirable property to lose unavoidably its optimality as soon as the probability distribution of the system's state starts to evolve dynamically in time.

Another technique is the *time-dependent generalized polynomial chaos* (TD-gPC) which utilizes a time-dependent orthogonal basis to better represent the stochastic part of the solution space (aka random function space or RFS) in time. The development of this technique was motivated by the fact that the probability distribution of the solution changes with time, which in turn requires that the random basis is frequently updated during the simulation to ensure that the mean-square error is kept orthogonal to the discretized RFS. Though this technique works well for problems that feature strong nonlinear dependencies over the probability space, the TD-gPC method possesses a serious issue: it suffers from the curse of dimensionality at the RFS level. This is because in all gPC-based methods the RFS is constructed using a tensor product of vector spaces with each of these representing a single RFS over one of the dimensions of the probability space. As a result, the higher the dimensionality of the probability space, the more vector spaces needed in the construction of a suitable RFS. To reduce the dimensionality of the RFS (and thus, its associated computational cost), gPC-based methods require the use of versatile sparse tensor products within their numerical schemes to

alleviate to some extent the curse of dimensionality at the RFS level. Therefore, this curse of dimensionality in the TD-gPC method alludes to the need of developing a more compelling spectral method that can quantify uncertainties in long-time response of dynamical systems at much lower computational cost.

In this work, a novel numerical method based on the spectral approach is proposed to resolve the curse-of-dimensionality issue mentioned above. The method has been called the *flow-driven spectral chaos* (FSC) because it uses a novel concept called *enriched stochastic flow maps* to track the evolution of a finite-dimensional RFS efficiently in time. The enriched stochastic flow map does not only push the system's state forward in time (as would a traditional stochastic flow map) but also its first few time derivatives. The push is performed this way to allow the random basis to be constructed using the system's enriched state as a germ during the simulation and so as to guarantee exponential convergence to the solution. It is worth noting that this exponential convergence is achieved in the FSC method by using only a few number of random basis vectors, even when the dimensionality of the probability space is considerably high. This is for two reasons: (1) the cardinality of the random basis does not depend upon the dimensionality of the probability space, and (2) the cardinality is bounded from above by $M + n + 1$, where M is the order of the stochastic flow map and n is the order of the governing stochastic ODE. The boundedness of the random basis from above is what makes the FSC method be curse-of-dimensionality free at the RFS level. For instance, for a dynamical system that is governed by a second-order stochastic ODE ($n = 2$) and driven by a stochastic flow map of fourth-order ($M = 4$), the maximum number of random basis vectors to consider within the FSC scheme is just 7, independent whether the dimensionality of the probability space is as low as 1 or as high as 10 000.

With the aim of reducing the complexity of the presentation, this dissertation includes three levels of abstraction for the FSC method, namely: a *specialized version* of the FSC method for dealing with structural dynamical systems subjected to uncertainties (Chapter 2), a *generalized version* of the FSC method for dealing with dynamical systems governed by (nonlinear) stochastic ODEs of arbitrary order (Chapter 3), and a *multi-element version* of the FSC method for dealing with dynamical systems that exhibit discontinuities over the probability space (Chapter 4). This dissertation also includes an implementation of the FSC

method to address the dynamics of large-scale stochastic structural systems more effectively (Chapter 5). The implementation is done via a modal decomposition of the spatial function space as a means to reduce the number of degrees of freedom in the system substantially, and thus, save computational runtime.

1. INTRODUCTION

1.1. A point of departure

Uncertainties are ubiquitous. They arise in a number of different fields such as physics, engineering, economics, sociology, etc. In complex dynamical systems, uncertainties can lead to undesired random variability in the response, which in turn can adversely affect the intended use for which they were designed. In fact, almost every physical quantity in a dynamical system is affected by uncertainty to some extent. In structural engineering, for example, the source of random variability may arise from: material properties, imperfections in geometry, loading scenarios, boundary conditions, etc. This random variability, once it is identified, can be characterized mathematically using random variables, stochastic processes or, more generally, random fields in space and time. Moreover, thanks to the ever-increasing computational capabilities, it has been possible to address—for the first time—high-dimensional problems involving complex physics and random input data through the probabilistic framework, and to even pose new questions by doing so. It should be clear then that the advent of quantum computers will further strengthen the use of the probabilistic framework in the years to come, and the potential to address and solve bigger problems in a reasonable amount of time. This is for instance the case of the light-based quantum computer Jiuzhang, which recently demonstrated that it is capable of outperforming the fastest classical supercomputer in certain tasks [1].

Several methods for solving stochastic differential equations have been proposed to date, including but not limited to: Monte Carlo-based methods [2–4], perturbation-based methods [5–8], operator-based methods [9–12], and spectral-based methods [13–15].

In standard Monte Carlo (aka *Monte Carlo sampling* or MCS), the stochastic problem is typically solved by sampling a large number of realizations from the probability space to estimate the statistics of the output. Although this method is straightforward to implement—as it only requires repetitive executions of deterministic simulations—, it is well known that the statistics converge relatively slow. (For example, the mean converges asymptotically as the square root of N^{-1}, where N is the number of realizations.) Therefore, the need

for a large number of realizations to obtain accurate results often makes the computation of the statistics prohibitive under the MCS framework. As an alternative, the excessive computational burden of the Monte Carlo method can significantly be alleviated if techniques such as the *Latin hypercube sampling* [16, 17] or the *quasi-Monte Carlo sampling* [18–20] are implemented instead. However, when dealing with systems that are already computationally expensive in their own deterministic settings, these techniques might be considered unsuitable to implement. Nonetheless, Monte Carlo-based methods have been successfully applied in the literature to quantify the effects that input uncertainty has on the observables of complex stochastic dynamical systems—and in some cases MCS is perhaps the only feasible way to simulate complex systems featuring high-dimensional probability spaces. This is because MCS is remarkably independent of the dimensionality of the probability space.

In perturbation-based methods, the governing equation of a stochastic dynamical system is generally expressed as an equation involving small variations known as *perturbations*. In this approach, the random variables, or more generally random fields, are expanded via a Taylor series around their mean and truncated up to a specified order. Then it uses perturbation theory to find an approximate solution to the governing equation in hand. In the past, perturbation-based methods were perhaps the most popular of all non-sampling methods used in engineering to solve problems involving random input data, see e.g. [5–8]. However, because high-order perturbations has the inevitable drawback to turn out the governing equation into a complicated system of equations, perturbations were usually limited to second-order variations in practice. This limitation unfolded the need to demand that the magnitude of the uncertainties was not too large in order to ensure a good performance of the method.

Operator-based methods have also been used for uncertainty quantification in the past. They are based on manipulating the stochastic operators in a governing equation to obtain an approximate version of them. They were used in [9–12] to quantify uncertainties in systems not involving time. However, as with perturbation-based methods, this type of methods is limited to problems in which the magnitude of the uncertainties is relatively small.

Finally, and as the name implies, spectral-based methods rely on the application of spectral theory in the resolution of problems involving uncertainties. Since this dissertation is based on the spectral approach, a short overview of this approach is presented below.

1.2. Overview of the spectral approach

1.2.1. Probability space and random space

Because the goal here is to begin with a formal discussion on the spectral approach, a prototype probability space is defined right below to allow the uncertainties in a dynamical system to be modeled properly later on.

Definition 1.1 (Probability space and random space)**.** Let $(\Omega, \boldsymbol{\Omega}, \lambda)$ be a (complete) *probability space*, where Ω is the sample space, $\boldsymbol{\Omega} \subset 2^{\Omega}$ is the σ-algebra on Ω (aka the collection of events), and $\lambda : \boldsymbol{\Omega} \to [0, 1]$ is the probability measure on $\boldsymbol{\Omega}$. Since in practice it is more convenient to relate the probability space with an Euclidean space for computational purposes, let $\xi : (\Omega, \boldsymbol{\Omega}) \to (\mathbb{R}^d, \mathcal{B}_{\mathbb{R}^d})$ be a measurable function (aka random variable[1] in probability theory) given by $\xi = \xi(\omega)$, with $\mathcal{B}_{\mathbb{R}^d}$ denoting the Borel σ-algebra over $\mathbb{R}^d$. In this dissertation, the measure space $(\Xi, \boldsymbol{\Xi}, \mu)$ is called *random space*, where $\Xi = \xi(\Omega) \subset \mathbb{R}^d$ is the set representing the random domain of the system, $\boldsymbol{\Xi} = \mathcal{B}_{\mathbb{R}^d} \cap \Xi$ is the σ-algebra on Ξ, and $\mu : \boldsymbol{\Xi} \to [0, 1]$ is the probability measure on $\boldsymbol{\Xi}$ given by the pushforward of λ by ξ, that is $\mu = \xi_*(\lambda) := \lambda \circ \xi^{-1}$. Note that here d symbolizes the dimensionality of the random space and not necessarily the dimensionality of the probability space—though in this dissertation they are always assumed to be equal.

1.2.2. Problem statement

To illustrate how the spectral approach works with a simple problem, an undamped stochastic single-degree-of-freedom system subjected to free vibration is considered below. The problem can be stated formally as follows.

---

[1]More often than not, ξ also receives the name of 'random vector' when $d > 1$. However, this particular ξ is *not* a (random) vector but a (random) point map from Ω into $\mathbb{R}^d$. Moreover, the structure of a vector space does *not* need to be specified in this $\mathbb{R}^d$ either, and therefore, in this dissertation ξ is never referred to as 'random vector' but always as 'random variable'. As a matter of fact, $\mathbb{R}^d$ is simply a topological space herein, and this space can induce a Borel space, $(\mathbb{R}^d, \mathcal{B}_{\mathbb{R}^d})$, by itself.

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

$$m\ddot{u} + ku = 0 \qquad \text{on } \mathfrak{T} \times \Xi \tag{1.1a}$$

$$\Big\{u(0,\cdot\,) = \boldsymbol{u},\ \dot{u}(0,\cdot\,) = \boldsymbol{v}\Big\} \qquad \text{on } \{0\} \times \Xi, \tag{1.1b}$$

where $\mathfrak{T} = [0, T]$ is a closed interval defining the time interval of interest for running the simulation, $T \in \mathbb{R}^+$ symbolizes the duration of the simulation, $m : \Xi \to \mathbb{R}^+$ is the mass of the system, $k : \Xi \to \mathbb{R}^+$ is the stiffness of the system, and $\boldsymbol{u}, \boldsymbol{v} : \Xi \to \mathbb{R}$ are the initial displacement and initial velocity of the system, respectively. Observe here that $\dot{u} := \partial_t u$ is the velocity of the system and that $\ddot{u} := \partial_t^2 u$ is the acceleration of the system.

1.2.3. Random function space

In the literature, it is customary to assume (right away from the beginning) that $u, \dot{u}, \ddot{u}$ are stochastic processes of second-order, and that $m, k, \boldsymbol{u}, \boldsymbol{v}$ are random variables of second-order as well (even though the first two random variables do not need to be assumed this way for reasons that will be clarified later). In other words, making such an assumption is equivalent to require that $u(t,\cdot\,), \dot{u}(t,\cdot\,), \ddot{u}(t,\cdot\,), m, k, \boldsymbol{u}, \boldsymbol{v}$ are square-integrable functions on $(\Xi, \boldsymbol{\Xi}, \mu)$. This motivates the following definition.

Definition 1.2 (Random function space or RFS)**.** Let $\mathfrak{X} = (L^2(\Xi, \boldsymbol{\Xi}, \mu; \mathbb{R}), \langle\,\cdot\,,\cdot\,\rangle)$ be a Lebesgue square-integrable space equipped with its standard inner product

$$\langle\,\cdot\,,\cdot\,\rangle : L^2(\Xi, \boldsymbol{\Xi}, \mu; \mathbb{R}) \times L^2(\Xi, \boldsymbol{\Xi}, \mu; \mathbb{R}) \to \mathbb{R} \qquad :\Leftrightarrow \qquad \langle f, g\rangle = \int f g \,\mathrm{d}\mu.$$

This *random function space* is the space of all (equivalence classes of) measurable functions $f : (\Xi, \boldsymbol{\Xi}) \to (\mathbb{R}, \mathcal{B}_{\mathbb{R}})$ that are square-integrable with respect to μ, where $\mathcal{B}_{\mathbb{R}}$ stands for the Borel σ-algebra over $\mathbb{R}$. Also, let $\{\Psi_j : (\Xi, \boldsymbol{\Xi}) \to (\mathbb{R}, \mathcal{B}_{\mathbb{R}})\}_{j=0}^{\infty}$ be a complete orthogonal basis in $\mathfrak{X}$, so that $\mathfrak{X} = \operatorname{span}\{\Psi_j\}_{j=0}^{\infty}$ and $\Psi_0(\xi) = 1$ for all $\xi \in \Xi$. Such a basis is always introduced herein for computational purposes only.

Remark 1.1. More precisely, $L^2(\Xi, \boldsymbol{\Xi}, \mu; \mathbb{R})$ is the set of all (equivalence classes of) $\boldsymbol{\Xi}$-measurable $\mathbb{R}$-valued functions that are square μ-integrable on Ξ. That is, $L^2(\Xi, \boldsymbol{\Xi}, \mu; \mathbb{R})$ is the quotient set defined by:

$$\left\{ f : (\Xi, \boldsymbol{\Xi}) \to (\mathbb{R}, \mathcal{B}_{\mathbb{R}}) \;\middle|\; \int f^2 \, \mathrm{d}\mu < \infty \right\} \Big/ \left\{ f : (\Xi, \boldsymbol{\Xi}) \to (\mathbb{R}, \mathcal{B}_{\mathbb{R}}) \;\middle|\; f = 0 \ \mu\text{-a.e.} \right\}.$$

Note that a $\boldsymbol{\Xi}$-measurable $\mathbb{R}$-valued function on Ξ is a function $f : \Xi \to \mathbb{R}$ that satisfies the property

$$f^{-1}(\mathfrak{S}) := \{\xi \in \Xi \mid f(\xi) \in \mathfrak{S}\} \in \boldsymbol{\Xi} \quad \text{for all } \mathfrak{S} \in \mathcal{B}_{\mathbb{R}}.$$

Moreover, it is worth stating that the square μ-integrability property of a function can only be determined using Lebesgue integration theory. See [21, 22] for further details.

Remark 1.2. The space $\mathfrak{Z}$ is known to form a Hilbert space because it is complete under the metric induced by $\langle \cdot, \cdot \rangle$. This gives rise to the following conclusions. *First*, $\mathfrak{Z}$ is a vector space over $\mathbb{R}$. *Second*, $\mathfrak{Z}$ induces a complete normed vector space over $\mathbb{R}$ whose induced standard norm $\| \cdot \| : L^2(\Xi, \boldsymbol{\Xi}, \mu; \mathbb{R}) \to \mathbb{R}_0^+$ is given by $\|f\| = \sqrt{\langle f, f \rangle}$ (and so, $\mathfrak{Z}$ also forms a Banach space with $\| \cdot \|$ as a norm). *Third*, $\mathfrak{Z}$ induces a complete metric space whose induced standard metric $d : L^2(\Xi, \boldsymbol{\Xi}, \mu; \mathbb{R}) \times L^2(\Xi, \boldsymbol{\Xi}, \mu; \mathbb{R}) \to \mathbb{R}_0^+$ is given by $d(f, g) = \|f - g\|$. *Fourth*, $\mathfrak{Z}$ induces a topological space whose induced standard topology $\mathcal{O} = \tau(\mathcal{B})$ is the topology generated by the topological basis $\mathcal{B} = \{B_r(f) \subset L^2(\Xi, \boldsymbol{\Xi}, \mu; \mathbb{R}) \mid r \in \mathbb{R}^+, \, f \in L^2(\Xi, \boldsymbol{\Xi}, \mu; \mathbb{R})\}$, where $B_r(f) := \{g \in L^2(\Xi, \boldsymbol{\Xi}, \mu; \mathbb{R}) \mid d(f, g) < r\}$ is the open ball of radius r centered at f.

Remark 1.3. Suppose $f, g \in \mathfrak{Z}$ and $\{h_n \in \mathfrak{Z}\}_{n=0}^{\infty}$ is a Cauchy sequence. Then, three remarks are in order. *First*, if $f = g$ μ-almost everywhere (μ-a.e.), then $\mu(\mathfrak{N} := \{\xi \in \Xi \mid f(\xi) \neq g(\xi)\}) = 0$, i.e. $f(\xi) = g(\xi)$ for all $\xi \in \Xi \setminus \mathfrak{N}$. *Second*, if $h_n \to h$ as $n \to \infty$, then $h \in \mathfrak{Z}$. *Third*, the product fg is not necessarily in $\mathfrak{Z}$, which means that $\mathfrak{Z}$ does not form an algebra over $\mathbb{R}$.

Remark 1.4. The orthogonality property of the basis $\{\Psi_j \in \mathfrak{Z}\}_{j=0}^{\infty}$ implies that $\langle \Psi_i, \Psi_j \rangle = \langle \Psi_i, \Psi_i \rangle \, \delta_{ij}$, where δ_{ij} is the Kronecker delta.

With this definition of $\mathfrak{Z}$ in place, one can now write any function $f \in \mathfrak{Z}$ using the Fourier series:

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

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

Since $u(t,\cdot), \dot{u}(t,\cdot), \ddot{u}(t,\cdot), m, k, \boldsymbol{u}, \boldsymbol{v}$ are already presumed to be square-integrable functions on $(\Xi, \boldsymbol{\Xi}, \mu)$, then it is evident that all these functions must be elements of $\mathfrak{X}$ for all $t \in \mathfrak{T}$. For instance, if $u(t,\cdot)$ is in $\mathfrak{X}$, then one can write $u(t,\xi)$ in the following manner:

$$u(t,\xi) = \sum_{j=0}^{\infty} u^j(t)\, \Psi_j(\xi) \quad \forall t \in \mathfrak{T}. \tag{1.2}$$

The coefficient $u^j(t)$ is also known as the j-th random mode of u, and as shown, it changes with time.

Remark 1.5. If $u(t,\cdot)$ is in $\mathfrak{X}$, then all its time derivatives $\{\dot{u}(t,\cdot), \ddot{u}(t,\cdot), \ldots\}$ are also in $\mathfrak{X}$. For instance, to check that $\dot{u}(t,\cdot)$ is in $\mathfrak{X}$, we note that:

$$\dot{u}(t,\cdot) := \partial_t u(t,\cdot) = \sum_{j=0}^{\infty} \partial_t u^j(t)\, \Psi_j \equiv \sum_{j=0}^{\infty} \dot{u}^j(t)\, \Psi_j \in \mathfrak{X} \quad \forall t \in \mathfrak{T}.$$

Moreover, the dual space of $\mathfrak{X}$, usually denoted by $\mathfrak{X}'$ in most textbooks, is the space spanned by the continuous linear functionals $\{\Psi^i : \mathfrak{X} \to \mathbb{R}\}_{i=0}^{\infty}$ given by

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

where $[\,\cdot\,,\cdot\,] : \mathfrak{X}' \times \mathfrak{X} \to \mathbb{R}$ is the natural pairing map between $\mathfrak{X}$ and $\mathfrak{X}'$. This space is also known to form a Hilbert space because of the Riesz representation theorem [23].

Therefore, by applying these linear functionals on both sides of (1.2), one can find an exact expression for the j-th random mode of u which is given by:

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

1.2.4. Discretization of random function space

In order to make the solution of (1.1) computationally tractable, $\mathfrak{X}$ needs to be either p-discretized or more generally (h,p)-discretized. In this section, only the former is discussed to simplify the presentation of the spectral discretization.

Let $\mathfrak{X}^{[P]}$ be a finite subspace of $\mathfrak{X}$ so that $\mathfrak{X}^{[P]} = \text{span}\{\Psi_j\}_{j=0}^{P}$ represents a p-discretization of $\mathfrak{X}$ and $P+1 \in \mathbb{N}_1$ defines the dimensionality of $\mathfrak{X}^{[P]}$. Then, an approximation $u^{[P]}(t,\cdot)$ of $u(t,\cdot)$ can be represented in $\mathfrak{X}^{[P]}$ as

$$u(t,\xi) \approx u^{[P]}(t,\xi) = \sum_{j=0}^{P} u^j(t)\,\Psi_j(\xi), \tag{1.3}$$

provided that the random basis is well-graded to allow the approximation of u to be taken this way. However, to reduce notational complexity hereafter, the superscript $^{[P]}$ in u and the summation sign in (1.3) are dropped to obtain the more appealing expression:

$$u(t,\xi) = u^j(t)\,\Psi_j(\xi). \tag{1.4}$$

(Of course with the understanding that $j \in \{0,1,\ldots,P\}$.)

Replacing (1.4) into (1.1) gives

$$m\ddot{u}^j\Psi_j + ku^j\Psi_j = 0 \qquad \text{on } \mathfrak{T}\times\Xi \tag{1.5a}$$

$$\Big\{u^j(0)\,\Psi_j = \boldsymbol{u},\ \dot{u}^j(0)\,\Psi_j = \boldsymbol{v}\Big\} \qquad \text{on } \{0\}\times\Xi. \tag{1.5b}$$

The next step is to project (1.5) onto $\mathfrak{X}^{[P]}$ in order to get a system of $P+1$ ordinary differential equations of second order in the variable t with $2(P+1)$ unknowns (namely, the random modes $u^j = u^j(t)$ and $\dot{u}^j = \dot{u}^j(t)$). This projection is performed here by applying on both sides of each equation the linear functionals $\{\Psi^i \in \mathfrak{X}'\}_{i=0}^{P}$ (one by one) to obtain:

$$m^i{}_j\ddot{u}^j + k^i{}_j u^j = 0 \qquad \text{on } \mathfrak{T} \tag{1.6a}$$

$$\Big\{u^i(0) = \boldsymbol{u}^i,\ \dot{u}^i(0) = \boldsymbol{v}^i\Big\} \qquad \text{on } \{0\}, \tag{1.6b}$$

where $i,j \in \{0,1,\ldots,P\}$ (summation sign implied over repeated index j), and:

$$\begin{aligned} m^i{}_j &= \frac{\langle\Psi_i, m\Psi_j\rangle}{\langle\Psi_i,\Psi_i\rangle} \in \mathbb{R}, \quad k^i{}_j = \frac{\langle\Psi_i, k\Psi_j\rangle}{\langle\Psi_i,\Psi_i\rangle} \in \mathbb{R}, \\ \boldsymbol{u}^i &= \frac{\langle\Psi_i, \boldsymbol{u}\rangle}{\langle\Psi_i,\Psi_i\rangle} \in \mathbb{R} \quad \text{and} \quad \boldsymbol{v}^i = \frac{\langle\Psi_i, \boldsymbol{v}\rangle}{\langle\Psi_i,\Psi_i\rangle} \in \mathbb{R}. \end{aligned} \tag{1.7}$$

The application of these linear functionals to system (1.5) ensures that the mean-square error resulting from the finite representation of u using (1.3) is orthogonal to $\mathfrak{X}^{[P]}$ [24].

Remark 1.6. From (1.7) it follows that m and k do not need to be square-integrable functions on $(\Xi, \boldsymbol{\Xi}, \mu)$ to ensure that $k^i{}_j$ and $m^i{}_j$ are finite, but only the products $m\Psi_j$ and $k\Psi_j$. This is the reason why in Section 1.2.3 an observation in this respect was made to emphasize the fact that m and k do not need to be assumed as random variables of second-order to make the system of equations (1.6) well-defined. However, in practice this assumption is usually enforced to make possible the computation of $\mathrm{Var}[m]$ and $\mathrm{Var}[k]$, and therefore, in this dissertation the same assumption is followed.

1.2.5. Some additional steps regarding the spectral discretization

There are two more steps needed in the spectral discretization of (1.1). Namely:

1. To select a quadrature rule of one's choice to evaluate the inner products shown in (1.7) numerically. Examples of common quadrature rules include: 'grid-based integration' [25–27] and 'Monte Carlo-based integration' [28]. Nonetheless, if the inner products are both given in closed-form and relatively easy to integrate (e.g. like in Section 1.3.2), they can be evaluated exactly using the theory of integration [21, 22].

2. To select a suitable time integration technique to push the state of the system forward in time using small time steps. Examples of common time integration techniques include: the 'Runge-Kutta method' [29] and the 'Newmark-β method' [30, 31].

1.2.6. Computation of probability moments

To clarify now how the probability moments of interest can be computed using the spectral approach, suppose that $z : \mathfrak{T} \times \Xi \to \mathbb{R}$ is a stochastic process of second-order given by $z = z(t, \xi)$; that is, $z(t, \cdot\,) \in \mathfrak{Z}$ for all $t \in \mathfrak{T}$.

Then, it is easy to show that the mean of z, $\mathbf{E}[z] : \mathfrak{T} \to \mathbb{R}$, is given by

$$\mathbf{E}[z](t) := \int z(t, \cdot\,)\, \mathrm{d}\mu = z^0(t), \tag{1.8}$$

and that the variance of z, $\mathrm{Var}[z] : \mathfrak{T} \to \mathbb{R}_0^+$, is given by

$$\mathrm{Var}[z](t) := \int (z(t,\cdot) - \mathbf{E}[z](t))^2 \, \mathrm{d}\mu = \lim_{P\to\infty} \sum_{j=1}^{P} \langle \Psi_j, \Psi_j \rangle \, z^j(t) \, z^j(t). \tag{1.9}$$

A formal derivation of (1.8) and (1.9) can be found in Section 3.2.

For instance, the stochastic process z can be interpreted as either the displacement u, the velocity $\dot{u}$, the acceleration $\ddot{u}$, the kinetic energy $\frac{1}{2}m\dot{u}^2$ (provided of course the variance exists), or any other obtainable response that one may be interested in from the mathematical model of the system.

1.3. Spectral methods

In the literature, there are a number of spectral methods that one can implement in order to solve the problem described in Section 1.2.2 numerically. Namely, the 'polynomial chaos' (PC) [24, 32] (nowadays superseded by the gPC method), the 'generalized polynomial chaos' (gPC) [33], the 'time-dependent gPC' (TD-gPC) [34], the 'modified TD-gPC' (mTD-gPC) [35], the 'gPC-flow map composition' (gPC-FMC) [36], the 'dynamical gPC' (DgPC) [37], and—of course—the corresponding multi-element version of them all.

1.3.1. Illustrative example

For sake of illustration, a case scenario for (1.1) is furnished here. Assume, for example, that the system governed by (1.1) has a mass of $m = 100$ kg, that the stiffness k is an uniformly distributed random variable defined in $[340, 460]$ N/m and given by $k(\xi) = 60\xi + 400$, and that the initial conditions are deterministic and taken as $\boldsymbol{u} = 0.05$ m and $\boldsymbol{v} = 0.20$ m/s. Then, two conclusions can be drawn from this case scenario: (1) the random domain of the system is one-dimensional and given by $\Xi = [-1, 1]$ (for there is only one random variable playing the role as stochastic input in the problem definition), and (2) the probability measure μ is uniform and defined as (using volume-form notation):

$$\mathrm{d}\mu \equiv \mu(\mathrm{d}\xi) = \tfrac{1}{2}\mathrm{d}\xi.$$

Table 1.1. Wiener-Askey scheme for certain types of probability measures [33]

	Probability measure, μ	Wiener-Askey chaos, $\{\Psi_j\}_{j=0}^{\infty}$	Support
Continuous	Gaussian	Hermite-chaos	$\mathbb{R}$
	Gamma	Laguerre-chaos	$\mathbb{R}_0^+$
	Beta	Jacobi-chaos	$[a, b] \subset \mathbb{R}$
	Uniform	Legendre-chaos	$[a, b] \subset \mathbb{R}$
Discrete	Poisson	Charlier-chaos	$\mathbb{N}_0$
	Binomial	Krawtchouk-chaos	$\{0, 1, \ldots, N\}$
	Negative binomial	Meixner-chaos	$\mathbb{N}_0$
	Hypergeometric	Hahn-chaos	$\{0, 1, \ldots, N\}$

1.3.2. The gPC method

To assure exponential convergence to the solution, the gPC method uses a Wiener-Askey chaos system to span the random function space $\mathfrak{Z}$ at early times of the simulation optimally. The chaos system is chosen conditional on the stochastic input's measure, as listed in Table 1.1. Therefore, since the input's measure of this problem is uniform, $\mathfrak{Z}$ is demanded to be spanned according to the Legendre-chaos system, where the first 10 polynomials of the system are known to be given by:

$$\Psi_0(\xi) = 1, \quad \Psi_1(\xi) = \xi, \quad \Psi_2(\xi) = \tfrac{1}{2}(3\xi^2 - 1), \quad \Psi_3(\xi) = \tfrac{1}{2}(5\xi^3 - 3\xi),$$
$$\Psi_4(\xi) = \tfrac{1}{8}(35\xi^4 - 30\xi^2 + 3), \quad \Psi_5(\xi) = \tfrac{1}{8}(63\xi^5 - 70\xi^3 + 15\xi),$$
$$\Psi_6(\xi) = \tfrac{1}{16}(231\xi^6 - 315\xi^4 + 105\xi^2 - 5), \quad \Psi_7(\xi) = \tfrac{1}{16}(429\xi^7 - 693\xi^5 + 315\xi^3 - 35\xi),$$
$$\Psi_8(\xi) = \tfrac{1}{128}(6435\xi^8 - 12012\xi^6 + 6930\xi^4 - 1260\xi^2 + 35) \quad \text{and}$$
$$\Psi_9(\xi) = \tfrac{1}{128}(12155\xi^9 - 25740\xi^7 + 18018\xi^5 - 4620\xi^3 + 315\xi).$$

The Legendre-chaos system forms a complete orthogonal basis in $\mathfrak{Z}$, and as shown above, $\Psi_0 \equiv 1$. Thus, the chaos system in question satisfies the two conditions prescribed in Definition 1.2 for a suitable basis for $\mathfrak{Z}$.

As outlined in Section 1.2.4, the initial value problem that we want to solve is given by (1.6). For instance, when $P = 4$, the initial value problem becomes specifically the 5-by-5 system of second-order ODEs:

$$100\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}\begin{bmatrix} \ddot{u}^0(t) \\ \ddot{u}^1(t) \\ \ddot{u}^2(t) \\ \ddot{u}^3(t) \\ \ddot{u}^4(t) \end{bmatrix} + 400\begin{bmatrix} 1 & \frac{1}{20} & & & \\ \frac{3}{20} & 1 & \frac{3}{50} & & \\ & \frac{1}{10} & 1 & \frac{9}{140} & \\ & & \frac{9}{100} & 1 & \frac{1}{15} \\ & & & \frac{3}{35} & 1 \end{bmatrix}\begin{bmatrix} u^0(t) \\ u^1(t) \\ u^2(t) \\ u^3(t) \\ u^4(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

subject to the initial conditions:

$$\begin{bmatrix} u^0(0) \\ u^1(0) \\ u^2(0) \\ u^3(0) \\ u^4(0) \end{bmatrix} = 0.05 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \text{and} \qquad \begin{bmatrix} \dot{u}^0(0) \\ \dot{u}^1(0) \\ \dot{u}^2(0) \\ \dot{u}^3(0) \\ \dot{u}^4(0) \end{bmatrix} = 0.20 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Remark 1.7. Note that for a Legendre-chaos system, we have:

$$\Big[\langle \Psi_i, \Psi_j \rangle\Big] = \begin{bmatrix} 1 & & & & \\ & \frac{1}{3} & & & \\ & & \frac{1}{5} & & \\ & & & \ddots & \\ & & & & \frac{1}{2P+1} \end{bmatrix}.$$

This probability information is particularly useful when computing, say, the variance response with (1.9).

1.3.3. Numerical results for the gPC method

To conduct the numerical simulation, system (1.6) is integrated over time using the RK4 method (Runge-Kutta method of fourth order [29]). The time-step size used is $\Delta t = 0.005$ s, and it is chosen this small in order to minimize the errors coming from the discretization of the temporal domain. For illustrative purposes, the duration of the simulation is set to last $T = 150$ s. Thence the temporal domain $\mathfrak{T}$ is defined as $[0, 150]$ s for this problem.

Now, for the purpose of quantifying the errors derived from the discretization of $\mathfrak{X}$, two error measures are introduced below. Namely, the local error $\epsilon[f] : \mathfrak{T} \to \mathbb{R}$ and the global error $\epsilon_G[f] : \mathfrak{T} \to \mathbb{R}$; they are given by:

$$\epsilon[f](t) = |f(t) - f_{\text{exact}}(t)|$$
$$\epsilon_G[f](t) = \frac{1}{T}\int_{\mathfrak{T}} \epsilon[f](t)\,\mathrm{d}t \approx \frac{\Delta t}{T}\sum_{i=0}^{\lfloor T/\Delta t \rfloor} |f(t_i) - f_{\text{exact}}(t_i)|.$$

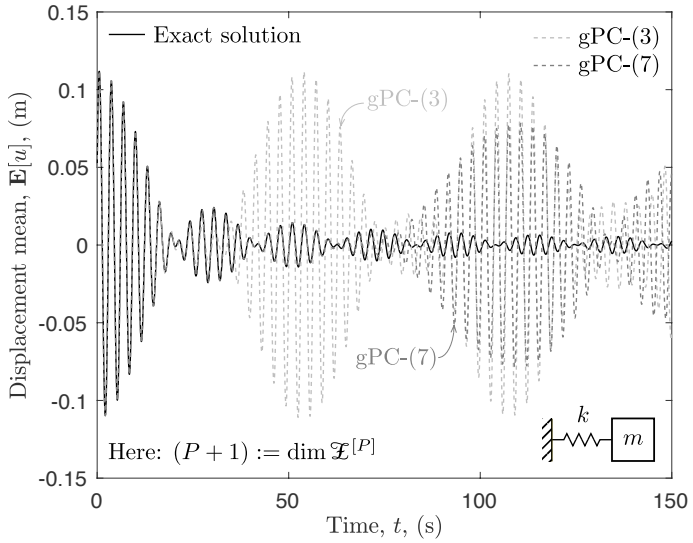
In this section, the function $f : \mathfrak{T} \to \mathbb{R}$ in both these expressions can be interpreted as the mean response, $\mathbf{E}[z]$, or the variance response, $\mathrm{Var}[z]$, as defined in Section 1.2.6.

Figs. 1.1 and 1.2 depict the evolution of the mean and variance of the system's state using both the gPC method and the exact solution[2] for sake of comparison. As observed, even using a 7-dimensional random function space, $\mathfrak{X}^{[6]}$, to represent the stochastic part of the solution space is not sufficient to reproduce the exact solution with high fidelity over time. To further illustrate this lack of convergence when using the gPC method, Figs. 1.3 to 1.5 present the local errors in mean and variance for different p-discretization levels of $\mathfrak{X}$. As shown, the gPC method tends to break down at early times of the simulation and as a function of the parameter P (being the results with $P = 3$ the less accurate among all results and $P = 7$ the more accurate, as expected). This figure clearly shows that after a short while the time-independent nature of the random basis produces a significant reduction in the efficiency of the method in regard to quantifying long-time response effectively during the simulation. Though increasing the dimensionality of the random function space does help delay the long-time integration issue of the method, it is worth noting that the global error for the mean or variance does not decrease considerably as P increases. This conclusion is demonstrated in Fig. 1.6. For a simulation that is set to last $T = 150$ s, it is clear that the global error thrives due to the impossibility of the gPC method to represent the state of the system effectively using a polynomial of degree P over time. Furthermore, increasing the dimensionality of the random function space from 3 to 9 is not even sufficient to reduce the global error by an order of magnitude, showcasing therefore the highly nonlinear nature of the solution at later times of the simulation.

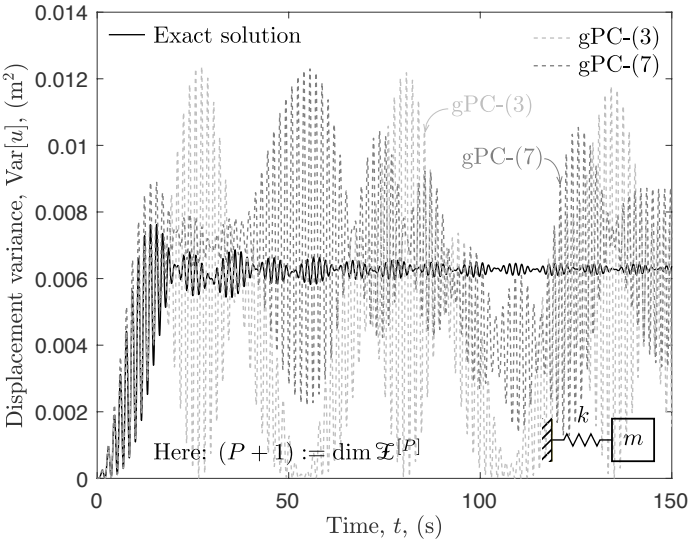
1.3.4. Other gPC-based methods

Essentially, the utilization of a time-independent random basis to span the stochastic part of the solution space is what makes the gPC method be so ineffective at quantifying long-time response of stochastic dynamical systems suitably. In recent years, a number of authors have

[2] As derived in Appendix 2.B.

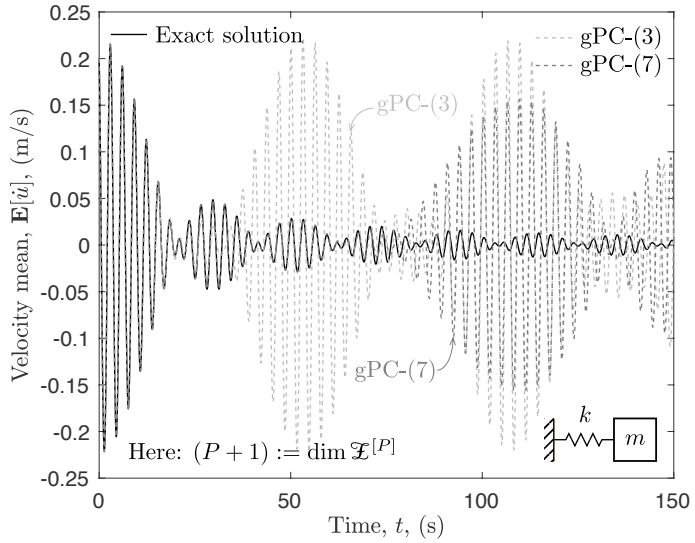


(a) Mean

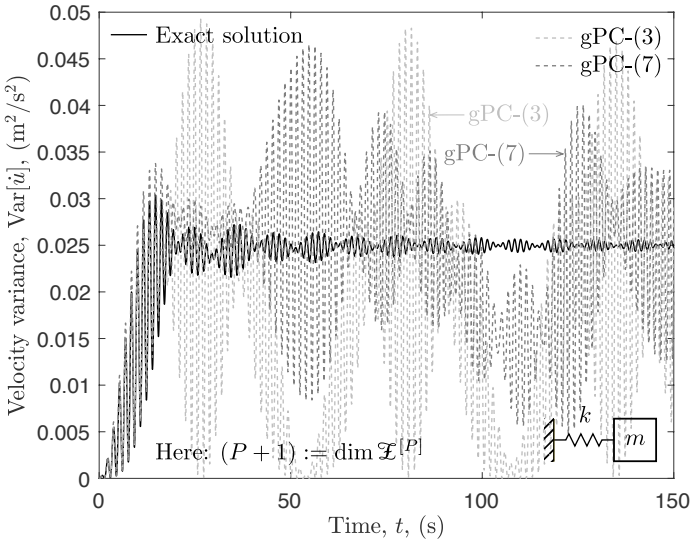


(b) Variance

Figure 1.1. Evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for the case when the p-discretization level of RFS is $\mathfrak{L}^{[2]}$ or $\mathfrak{L}^{[6]}$

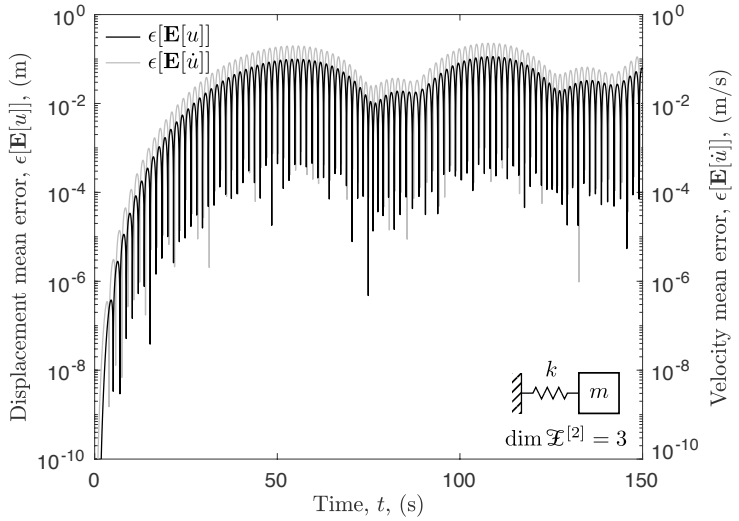


(a) Mean

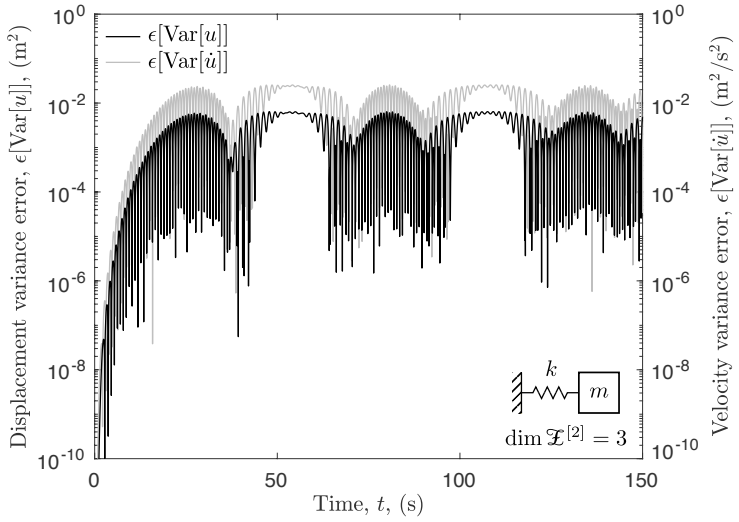


(b) Variance

Figure 1.2. Evolution of $\mathbf{E}[\dot{u}]$ and $\mathrm{Var}[\dot{u}]$ for the case when the p-discretization level of RFS is $\mathfrak{Z}^{[2]}$ or $\mathfrak{Z}^{[6]}$

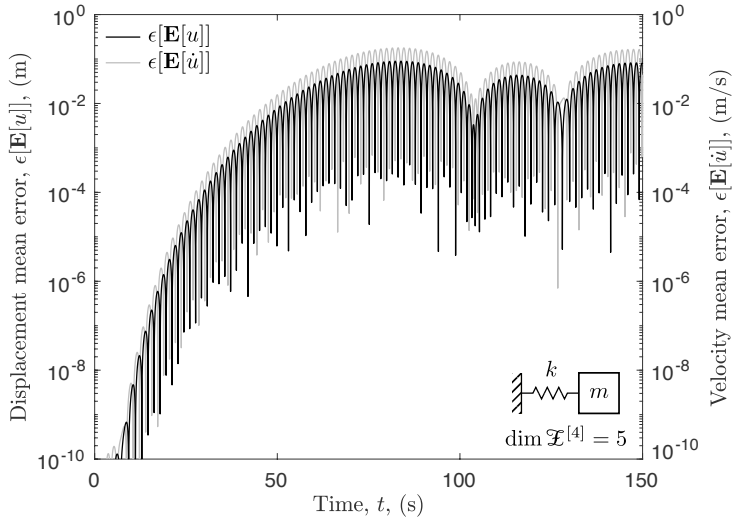


(a) Mean error for $\mathfrak{Z}^{[2]}$

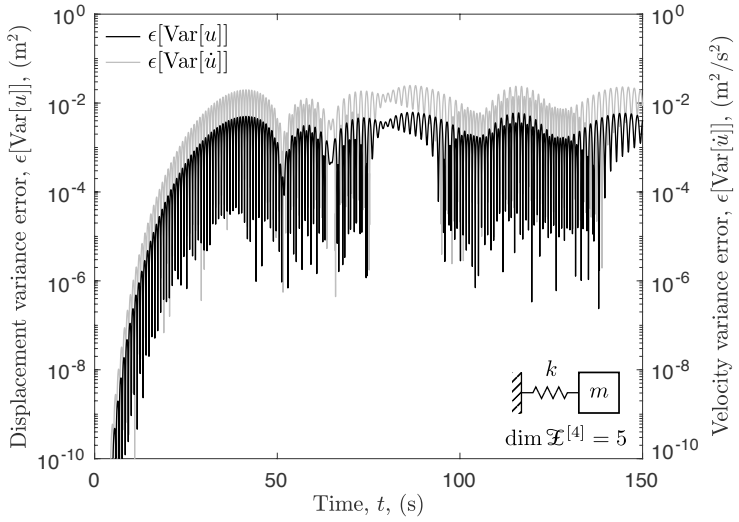


(b) Variance error for $\mathfrak{Z}^{[2]}$

Figure 1.3. Local error evolution of $\mathbf{E}[u]$, $\mathrm{Var}[u]$, $\mathbf{E}[\dot{u}]$ and $\mathrm{Var}[\dot{u}]$ for different p-discretization levels of RFS (Set 1 / 3)

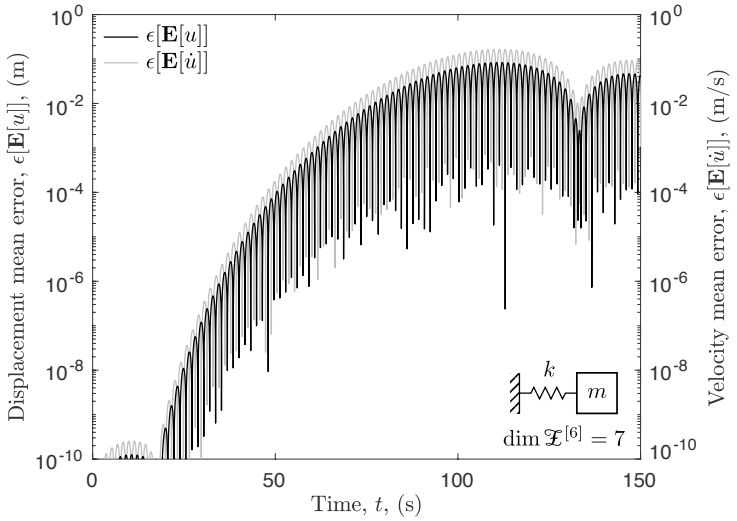


(a) Mean error for $\mathfrak{Z}^{[4]}$

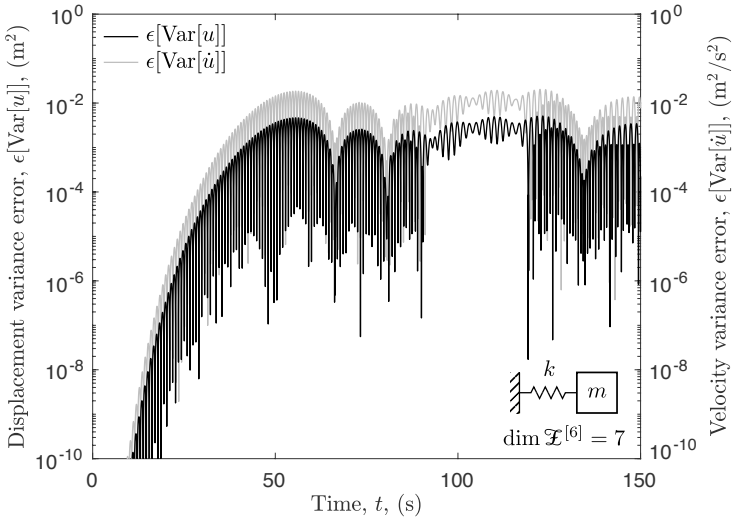


(b) Variance error for $\mathfrak{Z}^{[4]}$

Figure 1.4. Local error evolution of $\mathbf{E}[u]$, $\mathrm{Var}[u]$, $\mathbf{E}[\dot{u}]$ and $\mathrm{Var}[\dot{u}]$ for different p-discretization levels of RFS (Set 2/3)

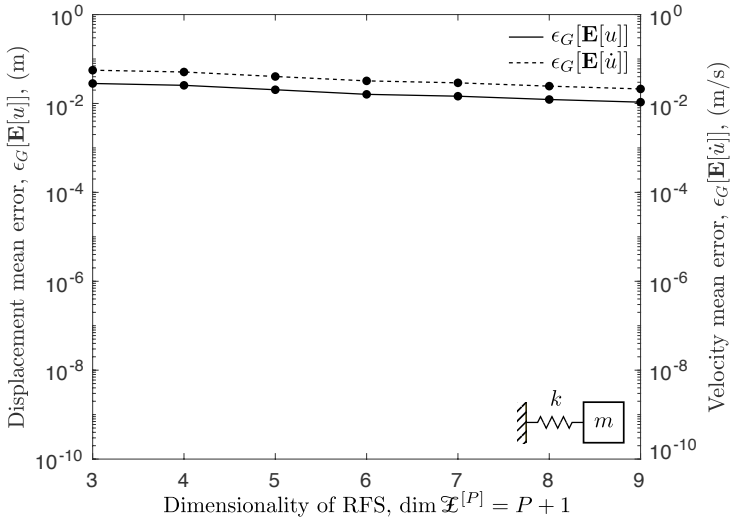


(a) Mean error for $\mathfrak{Z}^{[6]}$

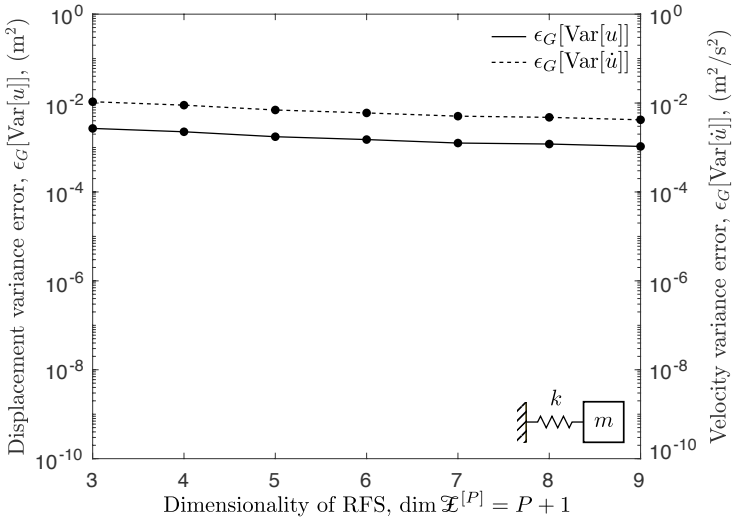


(b) Variance error for $\mathfrak{Z}^{[6]}$

Figure 1.5. Local error evolution of $\mathbf{E}[u]$, $\mathrm{Var}[u]$, $\mathbf{E}[\dot{u}]$ and $\mathrm{Var}[\dot{u}]$ for different p-discretization levels of RFS (Set 3 / 3)



(a) Mean error



(b) Variance error

Figure 1.6. Global error of $\mathbf{E}[u]$, $\mathrm{Var}[u]$, $\mathbf{E}[\dot{u}]$ and $\mathrm{Var}[\dot{u}]$ for different p-discretization levels of RFS

recognized this long-time integration issue in the gPC method, and thus, have proposed new techniques that help alleviate the issue through the implementation of time-dependent random bases in the simulation.

In the opening of Section 1.3, a list of available numerical methods was introduced for sake of reference, including: TD-gPC, mTD-gPC, gPC-FMC and DgPC. These more recent methods use an evolving random basis to better represent the stochastic part of the solution space in time. Though they all can be used to solve the illustrative problem without difficulties, the issue lies in how the random basis is constructed at the reset times or even initially. For high-dimensional probability spaces (or as low as 2-dimensional probability spaces), all gPC-based methods are known to suffer from the curse of dimensionality at the random-function-space level because the discretized version of $\mathfrak{X}$ is constructed using a tensor product of vector spaces—each vector space representing a specific dimension in the probability space and spanned by a set of univariate orthogonal polynomials of up to a certain degree. Consequently, the higher the dimensionality of the probability space, the more vector spaces needed in the construction of a suitable random function space. For instance, for a problem featuring a d-dimensional probability space one can make the following observation regarding the cardinality of a random basis enriched with univariate orthogonal polynomials of degree up to p:

$$\dim \mathfrak{X}^{[P]} := P + 1 = \begin{cases} (p+1)^d & \text{for a full tensor product scheme} \\ \binom{d+p}{p} & \text{for a total-order tensor product scheme.} \end{cases}$$

Thus, for a problem involving $d = 10$ independent random variables and univariate orthogonal polynomials of degree up to $p = 3$, it would be virtually impractical to run a stochastic simulation with any of the two schemes indicated above. Indeed, observe that for a scheme based on full tensor products one would need the dreadful amount of 1 048 576 random vectors to run the simulation, or the much smaller but yet relatively large amount of 286 random vectors for a scheme based on total-order tensor products. This case scenario therefore unveils the limitations of gPC-based methods in dealing with high-dimensional stochastic

problems, alluding thereby the need of developing a more compelling spectral method for the quantification of uncertainties in long-time response of stochastic dynamical systems.

1.4. Motivation and significance of the research

Current building code provisions are moving towards performance-based design (PBD) approaches. This is partly because in current code design procedures there are still open questions on how to estimate the lateral demand and lateral capacity of a structure within certain degree of accuracy. Perhaps the most well-known of them all is how to correctly select or estimate the response modification coefficient of a given structural system using Table 12.2-1 of ASCE 7-16 [38].

Performance-based design is an engineering approach in which the structural design criteria are expressed in terms of achieving a set of performance objectives during the design process of the structure. To reduce costs and ensure safety, different levels of performance objectives are considered, including the prescription of specific performance goals at the operational level, immediate-occupancy level, life-safety level and structural-stability level (e.g. as indicated in the ATC-40 report [39]). Such performance objectives are typically expressed in terms of a structural limit state or the probability of failure against a prescribed probability demand level. The analysis and design methods usually range from linear statics to nonlinear dynamics. In any case, an essential part of the PBD approach lies in how to quantify reliably the uncertainties associated with a given structural response, so that such an information can be utilized subsequently in checking (from a probabilistic standpoint) the structure's compliance against the set of performance objectives initially decided. Therefore, the implementation of stochastic numerical methods that are both accurate and computationally efficient is indispensable in any PBD framework.

Moreover, stochastic numerical methods based on the spectral approach have been utilized extensively in the literature to solve problems featuring random input data. Though this type of methods is particularly efficient when it comes to handling problems with relatively low-dimensional probability spaces, there are still some challenges to be addressed in stochastic problems involving complex physics and long-time response integration. As indicated in

the previous section, gPC-based methods suffer from the curse of dimensionality at the random-function-space level, and sometimes this curse of dimensionality makes the solution computationally intractable to find. Furthermore, it is well recognized that if an evolving random basis is not adopted throughout the simulation, the spectral approach becomes suboptimal over time, thus making the computed solution very inaccurate at later times of the simulation. Consequently, the main goal of this research is to develop a new spectral method that (1) is capable of being implemented in any PBD framework such as those described in the PEER TBI-2.0 [40] or FEMA P-58 [41] reports, that (2) does not bring the curse of dimensionality at the random-function-space level, and that (3) features an evolving random basis within the numerical scheme of the method. In this dissertation, the method that meets such three goal criteria is called the *flow-driven spectral chaos* (or FSC for short), and in the following chapters the method is presented using three levels of abstraction to keep the formulation as simple as possible in the beginning.

1.5. Objectives of the research

The overall goal of this research is to develop a set of spectral-chaos methods that can quantify the uncertainties in long-time response of dynamical systems efficiently. This is achieved herein by studying relatively small, stochastic dynamical systems and comprehending their local and global behaviors under different settings. Earlier studies conducted towards this end showed that in the case of dynamical systems driven by a Taylor-based stochastic flow map, the time derivatives of the solution plays a fundamental role in the description of the system's state over a small interval of time. This observation gave rise to the concept of 'enriched stochastic flow maps', which is used in the FSC method to track the evolution of a finite-dimensional RFS efficiently in time. The concept was demonstrated to be very useful at quantifying uncertainties in long-time response of single-degree-of-freedom systems, and therefore the need for developing it further.

The specific objectives of this research are:

1. To develop a *specialized version* of the FSC method to study the behavior of structural dynamical systems subjected to uncertainties (Chapter 2).

This objective aims at demonstrating that the FSC method is both curse-of-dimensionality free at the random-function-space level and capable of quantifying uncertainties effectively in the long-time response of structural dynamical systems using the spectral approach. The FSC method is therefore devised as an efficient alternative to methods based on the spectral approach for use in the area of stochastic dynamics of structures.

2. To develop a *generalized version* of the FSC method to study the behavior of dynamical systems governed by stochastic ODEs of arbitrary order (Chapter 3).

   This objective aims at generalizing the FSC method proposed in the previous chapter in order to address more general stochastic dynamical systems encountered in a number of different fields, such as: physics, engineering, economics, sociology, etc., including stochastic dynamical systems whose governing differential equation is nonlinear. The FSC method is thereby devised as a versatile tool to quantify uncertainties in long-time response for a wide variety of (nonlinear) stochastic dynamical systems.

3. To develop a *multi-element version* of the FSC method to study the behavior of dynamical systems that exhibit discontinuities over the probability space, and the potential of running the stochastic simulations in an embarrassingly parallel manner (Chapter 4).

   This objective aims at advancing the capabilities of the FSC method to address situations where the solution may be discontinuous over the probability space, and to obtain faster results by taking advantage of the multiple CPU cores available at simulation runtime. This is put all together in order to continue broadening the range of applications of the FSC method and to keep improving the computational flexibility of the method.

4. To develop a *faster version* of the FSC method for the purpose of studying the dynamical behavior of large-scale stochastic structural systems using, for computational efficiency, a relatively small random function space (Chapter 5).

   This objective aims at proposing a more compelling FSC method that can resolve stochastic dynamical systems with a large number of degrees of freedom at a lower computational cost. Here a modal decomposition of the spatial function space is suggested to reduce

considerably the dimensionality of the random function space, and thus, save computational resources during the simulation runtime.

2. FLOW-DRIVEN SPECTRAL CHAOS (FSC) METHOD FOR LONG-TIME INTEGRATION OF STOCHASTIC DYNAMICS OF STRUCTURES

2.1. Introduction

In the past few decades, the area of uncertainty quantification has received increasing attention, primarily in the fields of physics and engineering. This is not surprising since any mathematical description of a real-life physical system is always subject to the effects of input randomness. In structural engineering, for example, the randomness of a system may arise from different sources, such as variability of material properties, imperfections in geometry, loading scenarios, boundary conditions, etc. Though these physical quantities may be random, in most cases their stochastic characteristics can be identified and modeled mathematically using one or more of the numerous distributions available in statistics.

Due to the inherent complexity of real-life systems, closed-form solutions are not always possible. As an alternative, one uses accurate numerical methods that allow one to propagate and quantify the effects of input uncertainties in the system response efficiently. Historically, the polynomial chaos (PC) method developed by Wiener [32] and extended by Ghanem and Spanos [42] in the context of stochastic finite elements, has been used for uncertainty quantification. Xiu and Karniadakis [33] further developed the generalized-PC (gPC) method in which time-independent polynomials are used to decompose a stochastic process into deterministic and non-deterministic parts using the orthogonality property of the basis in the random function space (RFS). The benefit of using such orthogonal polynomials in the RFS is that when the underlying process (i.e. the solution of the ODE) is represented with them, the method leads to exponential convergence to the solution (provided the stochastic part of the solution space is not discretized).

The gPC method has undergone several modifications and extensions to improve its computational efficiency, its effectiveness for long-time integration, and its ability to handle stochastic discontinuities. The *multi-element gPC* and related methods, developed by Karniadakis and others [43–52], adaptively decompose the probability space into elements until a

pre-specified threshold for the relative error in variance is reached. Then a stochastic spectral expansion on each element is used to advance the state of the system one-time step forward, and this process is repeated every time the threshold is exceeded during the simulation. Another approach is the *dynamically orthogonal PC* (DO-PC) method [53–57] where the time rate of change of the spatio-temporal function space is ensured to be orthogonal to itself. This condition, called the dynamically orthogonal (DO) condition, is enforced at every time step as the simulation proceeds. The DO-PC scheme is essentially a generalization of the POD (Proper Orthogonal Decomposition) method [58, 59] and the gPC method in the framework of continuous stochastic dynamical systems.

For long-time dynamical simulations, when the stochasticity of the system has developed significantly, the gPC method fails to capture the probability moments accurately because the probability distribution of the solution changes significantly with time. The time-dependent gPC (TD-gPC) method was proposed by Gerritsma et al. [34] to allow the basis to evolve in time so as to better represent the transient nature of the probability distribution of the solution during the simulation. Heuveline and Schick [35] modified the TD-gPC method (mTD-gPC) to account for systems governed by second-order ODEs and also improved the accuracy of the method. Generally speaking, the mTD-gPC method relies on spanning the stochastic part of the solution space, at certain time steps (aka *reset times*), by taking a full tensor product between an evolving RFS (that depends upon the evolution of the state variables of the system) and the original RFS. However, since a full tensor product is required to be conducted at the reset times, the method suffers from the curse of dimensionality. A *hybrid generalized polynomial chaos* was also developed in [35] to help alleviate the curse of dimensionality of mTD-gPC. This method, however, requires the use of an (h, p)-discretization for the stochastic part of the solution space in contrast to mTD-gPC which solely requires the use of a p-discretization.

More recently, a generalized PC method based on flow map composition was proposed by Luchtenburg et al. [36] to address the long-time uncertainty propagation in dynamical systems more effectively. The method fundamentally uses short-time flow maps based on spectral polynomial bases to account for the stretching and folding effect caused by the evolution of the system's state over time. As with mTD-gPC, this method suffers from the curse of

dimensionality, because a tensor product is needed to construct the spectral polynomial bases during the simulation. Ozen and Bal [37] developed the *dynamical gPC* (DgPC) method to address the long-time behavior of stochastic dynamical systems via a generalization of the PCE (Polynomial Chaos Expansion) framework. They demonstrated that results from DgPC match well with those obtained from other standard methods such as Monte Carlo.

In this paper, a flow-driven spectral chaos (FSC) method is proposed which tracks the evolution of the random basis via an *enriched stochastic flow map* of the state of the system. The enriched flow map consists of the time derivatives of the solution up to a specified order. Unlike any gPC-based method, the number of basis vectors needed in FSC to represent the stochastic part of the solution space does not grow with the dimensionality of the probability space. However, as with any gPC-based method, the FSC method requires the computation of inner products using quadrature points distributed over the entire random domain, and the number of quadrature points can grow significantly with the dimensionality of the probability space. Nevertheless, FSC presents a significant advantage over gPC-based methods as its computational cost, for comparable accuracy, is an order of magnitude lower compared to existing TD-gPC methods. Conversely, for comparable computational cost, FSC is able to achieve an order of magnitude of better accuracy than TD-gPC.

This paper is organized as follows. Section 2.2 introduces the notation and definitions used in this paper and Section 2.3 presents the precise problem statement. The spectral approach for solving this problem is presented in Section 2.4 and Section 2.5 describes the proposed FSC method in detail. Three numerical examples are presented in Section 2.6 to demonstrate and compare the accuracy of FSC to other existing methods, such as mTD-gPC and Monte Carlo. The FSC method is then applied to quantify uncertainties in the structural dynamics of a 3-story building subject to an earthquake excitation in Section 2.7. In Appendix 2.A, we define the random bases that we use to span the random function space of the problem in hand. Finally, in Appendix 2.B, we provide expressions for the mean and variance of the exact response of a single-degree-of-freedom system under free vibration and uniformly-distributed stochastic stiffness, followed by a brief outline of the standard Monte Carlo method in Appendix 2.C.

2.2. Setting and notation

Definition 2.1 (Temporal space)**.** Let $(\mathfrak{T}, \mathcal{O})$ be a topological 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 $\mathbb{R}$. In this paper, $(\mathfrak{T}, \mathcal{O})$ is called the *temporal space* of the system.

Definition 2.2 (Random space)**.** Let $(\Omega, \mathbf{\Omega}, \lambda)$ be a (complete) probability space, where Ω is the sample space, $\mathbf{\Omega} \subset 2^{\Omega}$ is the σ-algebra on Ω (aka the collection of events in probability theory), and $\lambda : \mathbf{\Omega} \to [0, 1]$ is the probability measure on $\mathbf{\Omega}$. Let $\xi : (\Omega, \mathbf{\Omega}) \to (\mathbb{R}^d, \mathcal{B}_{\mathbb{R}^d})$ be a measurable function (aka random variable) given by $\xi = \xi(\omega)$, with $\mathcal{B}_{\mathbb{R}^d}$ denoting the Borel σ-algebra over $\mathbb{R}^d$. In this work, the measure space $(\Xi, \mathbf{\Xi}, \mu)$ is called *random space*, where $\Xi = \xi(\Omega) \subset \mathbb{R}^d$ is a set representing the random domain of the system, d denotes the dimensionality of the random space, $\mathbf{\Xi} = \mathcal{B}_{\mathbb{R}^d} \cap \Xi$ is the σ-algebra on Ξ, and $\mu : \mathbf{\Xi} \to [0, 1]$ is the probability measure on $\mathbf{\Xi}$ defined by the pushforward of λ by ξ, that is $\mu = \xi_*(\lambda)$.

Note that more structure can be added to these spaces whenever they are needed in the analysis. However, in order to keep the above definitions as elementary as possible, we singled out those mathematical objects that did not play a crucial role in the development of this work, such as the definition of a metric, a norm or an inner product for the underlying set.

Definition 2.3 (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 $f : (\mathfrak{T}, \mathcal{O}) \to (\mathbb{R}, \mathcal{O}_{\mathbb{R}})$ that have continuous first n derivatives on $(\mathfrak{T}, \mathcal{O})$.

Definition 2.4 (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

$$\langle \cdot, \cdot \rangle : L^2(\Xi, \mathbf{\Xi}, \mu; \mathbb{R}) \times L^2(\Xi, \mathbf{\Xi}, \mu; \mathbb{R}) \to \mathbb{R} \qquad :\Leftrightarrow \qquad \langle f, g \rangle = \int f g \, \mathrm{d}\mu.$$

This *random function space* (aka RFS in this paper) is the space of all measurable functions $f : (\Xi, \mathbf{\Xi}) \to (\mathbb{R}, \mathcal{B}_{\mathbb{R}})$ that are square-integrable with respect to μ. (By f we actually mean an equivalence class of square-integrable functions that are equal μ-almost everywhere; usually

denoted by $[f]$ in the literature.) This inner product space is known to form a Hilbert space because it is complete under the metric induced by the inner product. Furthermore, we define $\{\Psi_j : (\Xi, \boldsymbol{\Xi}) \to (\mathbb{R}, \mathcal{B}_{\mathbb{R}})\}_{j=0}^{\infty}$ to be a complete orthogonal basis in $\mathfrak{Z}$, such that $\Psi_0(\xi) = 1$ for all $\xi \in \Xi$. It is worth noting that such a basis does not necessarily need to consist of d-variate polynomials as in Refs. [13, 33], but may also consist of more general functions (including non-elementary functions such as wavelets).

Therefore, any function $f \in \mathfrak{Z}$ 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 symbolizing an exponentiation.

Moreover, the dual space of $\mathfrak{Z}$, which we denote by $\mathfrak{Z}'$, is simply the space spanned by the continuous linear functionals $\{\Psi^i : \mathfrak{Z} \to \mathbb{R}\}_{i=0}^{\infty}$ defined by:

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

where $[\,\cdot\,,\cdot\,] : \mathfrak{Z}' \times \mathfrak{Z} \to \mathbb{R}$ is the natural pairing map between $\mathfrak{Z}$ and $\mathfrak{Z}'$. This continuous dual space is also known to form a Hilbert space, thanks to the Riesz representation theorem [23].

We recall that the orthogonality property of the basis $\{\Psi_j \in \mathfrak{Z}\}_{j=0}^{\infty}$ implies that:

$$\langle \Psi_i, \Psi_j \rangle := \int \Psi_i \Psi_j \, \mathrm{d}\mu = \langle \Psi_i, \Psi_i \rangle \, \delta_{ij},$$

where δ_{ij} is the Kronecker delta.

Definition 2.5 (Solution space and root space)**.** Let $\mathcal{U} = \mathcal{T}(2) \otimes \mathfrak{Z}$ and $\mathcal{V} = \mathcal{T}(0) \otimes \mathfrak{Z}$ be, respectively, the *solution space* and the *root space* of the system.

In what follows, we assume that the components of the d-tuple random variable $\xi = (\xi^1, \ldots, \xi^d)$ are mutually independent, and as sketched in Fig. 2.1 that the random domain Ξ is a hypercube of d dimensions obtained by performing a d-fold Cartesian product of intervals $\bar{\Xi}_i := \xi^i(\Omega)$. It is for this reason that we define the probability measure in $\mathfrak{Z}$ hereafter as

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

where $\mu^i(\mathrm{d}\xi^i) =: \mathrm{d}\mu^i$ represents the probability measure of $\mathrm{d}\xi^i$ in the vicinity of $\xi^i \in \bar{\Xi}_i$.

2.3. Problem statement

In this work, we are interested in solving the non-autonomous, second-order ODE described below.

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

$$m\ddot{u} + \mathcal{F}[u, \dot{u}] = p \qquad \text{on } \mathfrak{T} \times \Xi \tag{2.1a}$$

$$\Big\{u(0, \cdot\,) = \mathscr{u},\ \dot{u}(0, \cdot\,) = \mathscr{v}\Big\} \qquad \text{on } \{0\} \times \Xi, \tag{2.1b}$$

where $m, \mathcal{F}[u, \dot{u}], p : \mathfrak{T} \times \Xi \to \mathbb{R}$ are elements of $\mathscr{V}$ such that $m(t, \xi) \neq 0$ for all $(t, \xi) \in \mathfrak{T} \times \Xi$, and $\mathscr{u}, \mathscr{v} : \Xi \to \mathbb{R}$ are elements of $\mathscr{X}$. Note that in (2.1), $\dot{u} := \partial_t u$ and $\ddot{u} := \partial_t^2 u$ denote, respectively, the first and second partial derivatives of u with respect to time. Thus, $\dot{u} : \mathfrak{T} \times \Xi \to \mathbb{R}$ is an element of $\mathscr{T}(1) \otimes \mathscr{X}$, and $\ddot{u} : \mathfrak{T} \times \Xi \to \mathbb{R}$ is an element of $\mathscr{T}(0) \otimes \mathscr{X} \equiv \mathscr{V}$ (the root space).

When (2.1) is specialized to study the (nonlinear) behavior of a single-degree-of-freedom system, $m\ddot{u}$ represents the inertial force of the system with $m : \mathfrak{T} \times \Xi \to \mathbb{R}^+$ symbolizing the mass of the system, $\mathcal{F}[u, \dot{u}]$ is the damping and resisting force of the system, and p is the external force acting on the system. Furthermore, in this case u, $\dot{u}$ and $\ddot{u}$ denote the displacement, the velocity and the acceleration response of the system, respectively.

When (2.1) is written in modeling notation, it becomes

$$y = \boldsymbol{\mathcal{M}}[u][x] \quad \text{subject to initial condition} \quad \boldsymbol{\mathcal{I}}[u], \tag{2.1*}$$

where $\boldsymbol{\mathcal{M}}[u] : \mathscr{V}^3 \to \mathscr{V}^s$ represents the mathematical model of the system defined by (2.1a), $x = (x_1, x_2, x_3) : \mathfrak{T} \times \Xi \to \mathbb{R}^3$ is the 3-tuple input of $\boldsymbol{\mathcal{M}}[u]$, and $y = (y_1, \ldots, y_s) : \mathfrak{T} \times \Xi \to \mathbb{R}^s$ is the s-tuple output of $\boldsymbol{\mathcal{M}}[u]$ (aka the s-tuple observable in physics or the s-tuple response in engineering). In addition, $\boldsymbol{\mathcal{I}}[u]$ represents the initial condition for $\boldsymbol{\mathcal{M}}[u]$ which is given by (2.1b). Thus, by comparing (2.1) to (2.1*), the components of x can be identified in the following way: $x_1 = m$, $x_2 = \mathcal{F}[u, \dot{u}]$ and $x_3 = p$. The objective of this mathematical model is to propagate and quantify the effects of input uncertainty x on system's output y. Therefore,

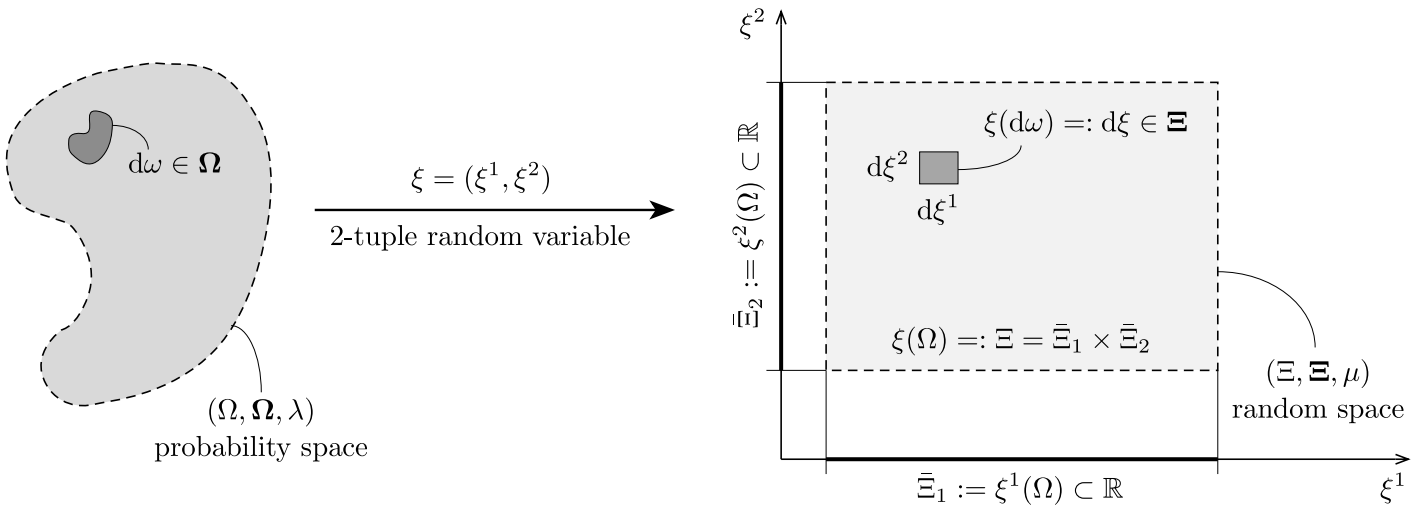


Figure 2.1. Relationship between probability space and random space for $d = 2$

besides seeking u in $\mathcal{U}$ as mentioned earlier, it is also important to compute the probability moments of y as time progresses.

2.4. Solution based on the spectral approach

2.4.1. Discretization of random function space

Since by hypothesis u is an element of $\mathcal{U}$, then it can be represented by the Fourier series

$$u(t,\xi) = \sum_{j=0}^{\infty} u^j(t)\,\Psi_j(\xi), \tag{2.2}$$

where u^j is a temporal function in $\mathcal{T}(2)$ denoting the j-th random mode of u. This series, usually referred to as *stochastic spectral expansion* in the literature [14,15], has the remarkable property that when u is sufficiently smooth in the solution space (and, of course, provided that the basis is orthogonal with respect to the probability measure defined in $\mathfrak{X}$), it leads to exponential convergence to the solution [33,60].

For the purpose of this manuscript, let us simply consider a p-discretization of the random function space as follows. Let $\mathfrak{X}^{[P]} = \text{span}\{\Psi_j\}_{j=0}^{P}$ be a finite subspace of $\mathfrak{X}$ with $P+1 \in \mathbb{N}_1$ denoting the dimensionality of the subspace. If we let $u^{[P]}(t,\cdot\,)$ be an element of $\mathfrak{X}^{[P]}$, then it is evident from (2.2) that[1]:

$$u(t,\xi) \approx u^{[P]}(t,\xi) = \sum_{j=0}^{P} u^j(t)\,\Psi_j(\xi) \equiv u^j(t)\,\Psi_j(\xi), \tag{2.3}$$

where for notational convenience we have omitted the summation sign in the last equality (aka Einstein summation convention). Therefore, unless otherwise noted hereinafter, a summation sign will always be implied over the repeated index $j \in \{0,1,\ldots,P\}$.

Substituting (2.3) into (2.1) gives

$$m\ddot{u}^j\Psi_j + \mathcal{F}[u^j\Psi_j, \dot{u}^j\Psi_j] = p \qquad \text{on } \mathfrak{T}\times\Xi \tag{2.4a}$$

$$\left\{u^j(0)\,\Psi_j = \mathfrak{u},\ \dot{u}^j(0)\,\Psi_j = \mathfrak{v}\right\} \qquad \text{on } \{0\}\times\Xi. \tag{2.4b}$$

[1] As long as we assume that $\{\Psi_j\}_{j=0}^{\infty}$ is well-graded to carry out the approximation of u this way.

Projecting (2.4) onto $\mathfrak{X}^{[P]}$ yields a system of $P+1$ ordinary differential equations of second order in the variable t, where the unknowns are the random modes $u^j = u^j(t)$ and $\dot{u}^j = \dot{u}^j(t)$:

$$\langle \Psi_i, m\Psi_j \rangle \, \ddot{u}^j + \langle \Psi_i, \mathcal{F}[u^j\Psi_j, \dot{u}^j\Psi_j] \rangle = \langle \Psi_i, p \rangle \qquad \text{on } \mathfrak{T} \tag{2.5a}$$

$$\Big\{ u^i(0) = \langle \Psi_i, \boldsymbol{u} \rangle / \langle \Psi_i, \Psi_i \rangle, \;\; \dot{u}^i(0) = \langle \Psi_i, \boldsymbol{v} \rangle / \langle \Psi_i, \Psi_i \rangle \Big\} \qquad \text{on } \{0\} \tag{2.5b}$$

with $i, j \in \{0, 1, \ldots, P\}$. Note that in order to get (2.5), we simply applied on both sides of each equation the linear functionals $\{\Psi^i \in \mathfrak{X}'\}_{i=0}^{P}$ one by one, and then we simplified the resulting expressions. It is also worth noting that because the randomness of the stochastic system has effectively been absorbed by the application of the aforementioned functionals, the system of equations that we are dealing with at this point is no longer 'stochastic' but 'deterministic'. In other words, the system now depends merely on the time variable t rather than on the tuple (t, ξ).

System (2.5) can also be restated using multilinear and tensor algebra notation as follows:

$$m^i{}_j \ddot{u}^j + \mathcal{F}^i[u^j, \dot{u}^j] = p^i \qquad \text{on } \mathfrak{T} \tag{2.6a}$$

$$\Big\{ u^i(0) = \boldsymbol{u}^i, \; \dot{u}^i(0) = \boldsymbol{v}^i \Big\} \qquad \text{on } \{0\}, \tag{2.6b}$$

where $i, j \in \{0, 1, \ldots, P\}$ (summation sign implied over repeated index j), and:

$$m^i{}_j(t) = \langle \Psi_i, m(t, \cdot)\, \Psi_j \rangle / \langle \Psi_i, \Psi_i \rangle, \quad \mathcal{F}^i[u^j, \dot{u}^j](t) = \langle \Psi_i, \mathcal{F}[u^j\Psi_j, \dot{u}^j\Psi_j](t, \cdot) \rangle / \langle \Psi_i, \Psi_i \rangle,$$
$$p^i(t) = \langle \Psi_i, p(t, \cdot) \rangle / \langle \Psi_i, \Psi_i \rangle, \quad \boldsymbol{u}^i = \langle \Psi_i, \boldsymbol{u} \rangle / \langle \Psi_i, \Psi_i \rangle \quad \text{and} \quad \boldsymbol{v}^i = \langle \Psi_i, \boldsymbol{v} \rangle / \langle \Psi_i, \Psi_i \rangle,$$

whence $m^i{}_j, \mathcal{F}^i[u^j, \dot{u}^j], p^i \in \mathcal{T}(0)$ and $\boldsymbol{u}^i, \boldsymbol{v}^i \in \mathbb{R}$. To simplify notation, we have taken $\mathcal{F}^i[u^j, \dot{u}^j]$ as the short notation for $\mathcal{F}^i[u^0, \ldots, u^j, \ldots, u^P, \dot{u}^0, \ldots, \dot{u}^j, \ldots, \dot{u}^P]$.

To evaluate the inner products approximately, any integration technique of one's choice can be used, including those addressed in [25–28]. If, for instance, a Gaussian quadrature rule is adopted, the inner products are computed with:

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

where $w_i \in \mathbb{R}^+$ denotes the quadrature weight associated with the Gaussian quadrature point $\xi_i \in \Xi$ (w.r.t. μ), and $Q \in \mathbb{N}_1$ represents the number of quadrature points involved in approximating the evaluation of the inner product.

2.4.2. Discretization of temporal function space

Once $\mathfrak{X}$ has been discretized, the temporal function space can be discretized using an (h, p)-discretization for $\mathcal{T}(2)$. In the literature, there exists an extensive list of time integration techniques that one can employ in order to solve the ODE system given by (2.6) numerically. For example, the Runge-Kutta method [29] of fourth-order (aka RK4 method) or the Newmark-β method [30, 31].

2.4.3. Computation of probability moments

In this manuscript, the probability moments of interest are the mean and the variance of the system's response. For this reason, we define these objects below.

Suppose that $z := y_k$ is the k-th component of output $y = \boldsymbol{\mathcal{M}}[u][x]$. If $z \in \mathcal{V}$, then it can approximately be expanded with a truncated Fourier series similar to the one set forth in (2.3) to obtain:

$$z(t, \xi) \approx z^{[P]}(t, \xi) = \sum_{j=0}^{P} z^j(t)\, \Psi_j(\xi) \equiv z^j(t)\, \Psi_j(\xi), \tag{2.7}$$

where the j-th random mode of z is given by

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

(Note that P in expression (2.7) does not necessarily need to be the same as in (2.3).)

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

$$\mathbf{E}[z](t) := \int z(t, \cdot)\, \mathrm{d}\mu = z^0(t), \tag{2.8}$$

whereas the variance of z, $\mathrm{Var}[z] : \mathfrak{T} \to \mathbb{R}_0^+$, is defined by the partial sum:

$$\mathrm{Var}[z](t) := \int (z(t,\cdot) - \mathbf{E}[z](t))^2 \, \mathrm{d}\mu = \sum_{j=1}^{P} \langle \Psi_j, \Psi_j \rangle \, z^j(t) \, z^j(t). \tag{2.9}$$

2.5. Flow-driven spectral chaos (FSC) method

2.5.1. Stochastic flow map

Observe that the stochastic system given by (2.1) can also be expressed as:

$$\partial_t^2 u(t,\xi) := \boldsymbol{\ell}(t,\xi,s(t,\xi)) = \Big(p(t,\xi) - \mathcal{F}[u,\dot{u}](t,\xi)\Big)/m(t,\xi) \qquad \text{on } \mathfrak{T}\times\Xi \tag{2.10a}$$

$$\Big\{u(0,\xi) = \boldsymbol{u}(\xi),\, \dot{u}(0,\xi) = \boldsymbol{v}(\xi)\Big\} \qquad \text{on } \{0\}\times\Xi, \tag{2.10b}$$

where $s = (u,\dot{u}) \in \prod_{j=1}^{2} \mathcal{T}(3-j) \otimes \mathfrak{X}$ is the configuration state of the system over $\mathfrak{T}\times\Xi$, and $\boldsymbol{\ell} : \mathfrak{T}\times\Xi\times\mathbb{R}^2 \to \mathbb{R}$ is a noisy, non-autonomous function defining the response $\ddot{u} = \partial_t^2 u$.

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

$$u(t_i + h, \xi) = \sum_{j=0}^{\infty} \frac{h^j}{j!} \partial_t^j u(t_i,\xi) = \sum_{j=0}^{M} \frac{h^j}{j!} \partial_t^j u(t_i,\xi) + O(h^{M+1})(\xi), \tag{2.11}$$

where $h := t - t_i$ is the time-step size used for the simulation around t_i (once t is fixed), $t_i \in \mathfrak{T}$ is the time instant of the simulation, and $M \in \mathbb{N}_1$ is the order of the flow map we are interested to implement.

For this system, the *stochastic flow map* of order M, $\varphi(M) : \mathbb{R}\times\mathfrak{X}^2 \to \mathfrak{X}^2$, can be defined as a random map given by:

$$\begin{aligned}\varphi(M)(h, s(t_i,\cdot)) =: s(t_i+h,\cdot) &= \Big(u(t_i+h,\cdot), \dot{u}(t_i+h,\cdot)\Big) - O(h^{M+1}) \\ &= \left(\sum_{j=0}^{M} \frac{h^j}{j!} \partial_t^j u(t_i,\cdot), \sum_{j=0}^{M} \frac{h^j}{j!} \partial_t^{j+1} u(t_i,\cdot)\right),\end{aligned} \tag{2.12}$$

where this $s(t_i+h,\cdot)$ is the same as in (2.10) if $M \to \infty$. However, to avoid unnecessary complexity in notation, no distinction between these definitions will be made in this work. That is, from now on we will assume that there is an equivalence relation $\sim$ between

$s(t_i + h, \cdot) - O(h^{M+1})$ and the $s(t_i + h, \cdot)$ defined in (2.12). Notice that (2.12) now requires that $s = (u, \dot{u}) \in \prod_{j=1}^{2} \mathcal{T}(M - j + 2) \otimes \mathfrak{E}$.

For the sake of illustration, suppose $M = 4$. Differentiating (2.10a) with respect to time three times (i.e. $M - 1$ times) gives

$$\partial_t^3 u := \mathrm{D}_t \boldsymbol{\ell} = \partial_t \boldsymbol{\ell} + \partial_u \boldsymbol{\ell}\, \partial_t u + \partial_{\dot{u}} \boldsymbol{\ell}\, \partial_t^2 u \tag{2.13a}$$

$$\partial_t^4 u := \mathrm{D}_t^2 \boldsymbol{\ell} = \partial_t^2 \boldsymbol{\ell} + 2\, \partial_{tu}^2 \boldsymbol{\ell}\, \partial_t u + (2\, \partial_{t\dot{u}}^2 \boldsymbol{\ell} + \partial_u \boldsymbol{\ell})\, \partial_t^2 u + \partial_{\dot{u}} \boldsymbol{\ell}\, \partial_t^3 u \tag{2.13b}$$

$$\begin{aligned} \partial_t^5 u := \mathrm{D}_t^3 \boldsymbol{\ell} = \partial_t^3 \boldsymbol{\ell} + 3\, \partial_{ttu}^3 \boldsymbol{\ell}\, \partial_t u + 3\, (\partial_{tt\dot{u}}^3 \boldsymbol{\ell} + \partial_{tu}^2 \boldsymbol{\ell})\, \partial_t^2 u \\ + (3\, \partial_{t\dot{u}}^2 \boldsymbol{\ell} + \partial_u \boldsymbol{\ell})\, \partial_t^3 u + \partial_{\dot{u}} \boldsymbol{\ell}\, \partial_t^4 u. \end{aligned} \tag{2.13c}$$

Hence, when $M = 4$, the stochastic flow map of the system, $\varphi(4) = (\varphi^1(4), \varphi^2(4))$, is given by:

$$\begin{aligned} \varphi(4)(h, s(t_i, \cdot)) =: s(t_i + h, \cdot) &= \Big(u(t_i + h, \cdot), \dot{u}(t_i + h, \cdot)\Big) - O(h^5) \\ &= \left(\sum_{j=0}^{4} \frac{h^j}{j!} \partial_t^j u(t_i, \cdot), \sum_{j=0}^{4} \frac{h^j}{j!} \partial_t^{j+1} u(t_i, \cdot) \right), \end{aligned} \tag{2.14}$$

where the second and higher time derivatives of u at $t = t_i$ are computed with (2.10a) and (2.13), respectively.

Note that if (2.10) is an autonomous ODE, the expressions prescribed by (2.13) reduce to:

$$\partial_t^3 u = \partial_u \boldsymbol{\ell}\, \partial_t u + \partial_{\dot{u}} \boldsymbol{\ell}\, \partial_t^2 u, \quad \partial_t^4 u = \partial_u \boldsymbol{\ell}\, \partial_t^2 u + \partial_{\dot{u}} \boldsymbol{\ell}\, \partial_t^3 u \quad \text{and} \quad \partial_t^5 u = \partial_u \boldsymbol{\ell}\, \partial_t^3 u + \partial_{\dot{u}} \boldsymbol{\ell}\, \partial_t^4 u. \tag{2.13*}$$

2.5.2. Enriched stochastic flow map

In this work, we define the *enriched stochastic flow map* of order M, $\hat{\varphi}(M) : \mathbb{R} \times \mathfrak{E}^{M+2} \to \mathfrak{E}^{M+2}$, such that its k-th component is given by:

$$\hat{\varphi}^k(M)(h, \hat{s}(t_i, \cdot)) =: \hat{s}^k(t_i + h, \cdot) = \begin{cases} \varphi^k(M)(h, s(t_i, \cdot)) & \text{for } k \in \{1, 2\} \\ \mathrm{D}_t^{k-3} \boldsymbol{\ell}(t_i + h, \cdot, s(t_i + h, \cdot)) & \text{otherwise,} \end{cases} \tag{2.15}$$

where $\hat{s} = (u, \dot{u}, \ldots, \partial_t^{M+1} u) \in \prod_{j=1}^{M+2} \mathcal{T}(M - j + 2) \otimes \mathfrak{E}$ is called the *enriched configuration state* of the system over $\mathfrak{T} \times \Xi$, and $k \in \{1, 2, \ldots, M + 2\}$.

For instance, when $M = 4$, the components of the enriched stochastic flow map, $\hat{\varphi}(4)$, are

$$\hat{\varphi}^1(4)(h, s(t_i, \cdot)) := \varphi^1(4)(h, s(t_i, \cdot)) = s^1(t_i + h, \cdot) = u(t_i + h, \cdot) - O(h^5), \tag{2.16a}$$

$$\hat{\varphi}^2(4)(h, s(t_i, \cdot)) := \varphi^2(4)(h, s(t_i, \cdot)) = s^2(t_i + h, \cdot) = \dot{u}(t_i + h, \cdot) - O(h^5), \tag{2.16b}$$

$$\hat{\varphi}^3(4) := \not{\ell} = \partial_t^2 u, \quad \hat{\varphi}^4(4) := \mathrm{D}_t \not{\ell} = \partial_t^3 u, \quad \hat{\varphi}^5(4) := \mathrm{D}_t^2 \not{\ell} = \partial_t^4 u$$
$$\text{and} \quad \hat{\varphi}^6(4) := \mathrm{D}_t^3 \not{\ell} = \partial_t^5 u, \tag{2.16c}$$

where $\hat{\varphi}^1(4)$ and $\hat{\varphi}^2(4)$ are computed with (2.14), $\hat{\varphi}^3(4)$ with (2.10a), and $\{\hat{\varphi}^k(4)\}_{k=4}^6$ with (2.13).

2.5.3. Derivation of the FSC method

According to Section 2.5.1, the state of a system driven by a stochastic flow map of order M is:

$$u(t, \xi) = \sum_{j=0}^{M} \frac{(t - t_i)^j}{j!} \partial_t^j u(t_i, \xi) \quad \text{and} \quad \dot{u}(t, \xi) = \sum_{j=0}^{M} \frac{(t - t_i)^j}{j!} \partial_t^{j+1} u(t_i, \xi), \tag{2.17}$$

with the provision that the stochastic process u is analytic on the temporal domain.

From these two expressions, it can be seen that the state of the system has been decomposed effectively into deterministic and non-deterministic parts. That is, the deterministic part $(t - t_i)^j / j!$ consisting of a temporal function in $\mathscr{T}$, and the non-deterministic part $\partial_t^j u(t_i, \xi)$ consisting of a random function in $\mathscr{Z}$. If the set of functions associated with the non-deterministic part, i.e. $\{\partial_t^j u(t_i, \cdot)\}_{j=0}^{M+1}$, is orthogonalized with respect to the measure in $\mathscr{Z}$, then (2.17) can also be written in the following way:

$$u(t, \xi) = \sum_{j=1}^{M+2} u^j(t)\, \Psi_j(\xi) \quad \text{and} \quad \dot{u}(t, \xi) = \sum_{j=1}^{M+2} \dot{u}^j(t)\, \Psi_j(\xi). \tag{2.18}$$

Hence, if the space associated with the stochastic part of the solution space were to be spanned with $\{\Psi_j\}_{j=1}^{M+2}$, then $u(t, \cdot)$ and $\dot{u}(t, \cdot)$ would be elements of that space around $t = t_i$.

However, since one cannot always guarantee that the constant functions are in such a space construction, we prefer to write (2.18) in the following final form instead:

$$u(t,\xi) = \sum_{j=0}^{M+2} u^j(t)\,\Psi_j(\xi) \quad \text{and} \quad \dot{u}(t,\xi) = \sum_{j=0}^{M+2} \dot{u}^j(t)\,\Psi_j(\xi), \tag{2.19}$$

where $\Psi_0 \equiv 1$ is the identically-equal-to-one function as per Definition 2.4.

Therefore, for a system driven by a stochastic flow map of order M, the maximum number of basis vectors to use in a simulation with FSC is bounded from above by $M+3$. Hence, regardless of the dimensionality of the random space, the probability information of the system's state can be completely captured in $\mathfrak{L}^{[P]}$ if $P = M+2$. It is for this reason that our FSC scheme is superior in terms of efficiency in comparison to mTD-gPC which uses a combination of full and total-order tensor products to construct a suitable basis for $\mathfrak{L}^{[P]}$ around $t = t_i$. We emphasize, however, that the FSC scheme does not address by itself the curse of dimensionality at the random-space level, since we still have the issue that the bigger the random space is ($d \gg 1$), the more difficult it is to compute the inner products accurately in (2.6) and (2.9). This is still an open area of research and there are several approaches available in the literature for dealing with this issue [25–28].

Moreover, to reduce the computational cost associated with orthogonalizing $M+2$ basis vectors, it is sometimes convenient to start the FSC analysis with the smallest value for M (i.e. $M = 1$), and then to progressively increment it if more accurate results are desired for the simulation. Therefore, the minimum number of basis vectors to use in a simulation with FSC is bounded from below by 4.

2.5.4. FSC scheme

Suppose that a stochastic system such as (2.1) has been given. 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 $s_{.i} = s|_{\mathrm{cl}(\mathfrak{T}_i)\times\Xi}$ to be the restriction of s to $\mathfrak{R}_i := \mathrm{cl}(\mathfrak{T}_i) \times \Xi$. (Recall that $s = (u, \dot{u})$ 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 (Fig. 2.2), proceed as follows:

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

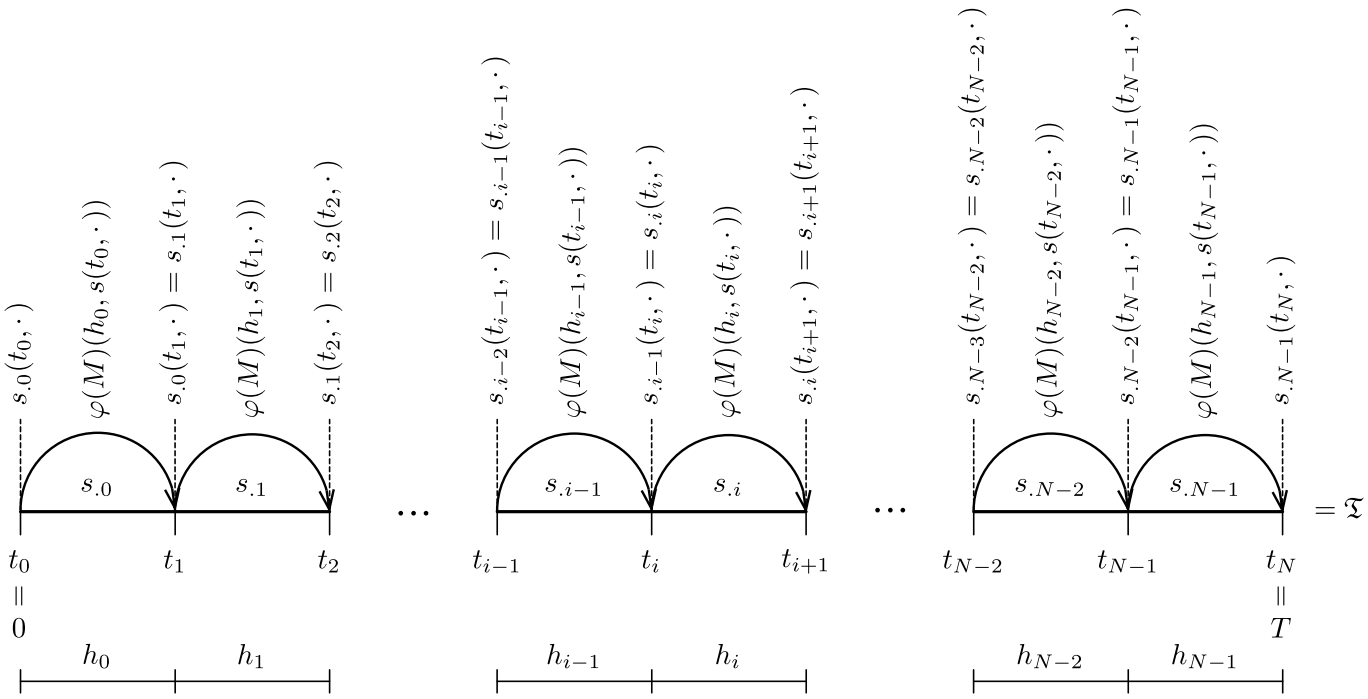


Figure 2.2. Evolution of a dynamical system via a stochastic flow map of order M, provided that h_i is taken sufficiently small and M sufficiently large so as to have $s(t_N,\cdot) \approx s_{\cdot N-1}(t_N,\cdot)$ at the end of the simulation. Then, as a means to avoid complexity in notation, we take $s(t_i+h,\cdot) \sim s_{\cdot i}(t_i+h,\cdot)$ for $h \geq 0$, and $s(t_i+h,\cdot) \sim s_{\cdot i-1}(t_i+h,\cdot)$ for $h \leq 0$

(a) Define a solution representation for the configuration state $s_{.i}$ in the following way.

- Take $\{\Phi_{j.i} := \hat{\varphi}^j(M)(0, \hat{s}(t_i, \cdot))\}_{j=1}^P$ to be an ordered set of linearly independent functions in $\mathfrak{Z}$ with $3 \leq P \leq M + 2$, and define $\Phi_{0.i} \equiv 1$. 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 are deterministic or linearly dependent, please see Remark 2.1.)
- Orthogonalize the set $\{\Phi_{j.i}\}_{j=0}^P$ using the Gram-Schmidt process [61], so that the resulting set $\{\Psi_{j.i}\}_{j=0}^P$ is an orthogonal basis in $\mathfrak{Z}$. That is, for $j \in \{0, 1, \ldots, P\}$:

$$\Psi_{j.i} := \Phi_{j.i} - \sum_{k=0}^{j-1} \frac{\langle \Phi_{j.i}, \Psi_{k.i} \rangle}{\langle \Psi_{k.i}, \Psi_{k.i} \rangle} \Psi_{k.i}. \tag{2.20}$$

- Define $\mathfrak{Z}_i^{[P]} = \text{span}\{\Psi_{j.i}\}_{j=0}^P$ to be a p-discretization of $\mathfrak{Z}$ over the region $\mathfrak{R}_i$. Since $\mathfrak{Z}_i^{[P]}$ is an evolving function space, expansion (2.3) is now to be read as:

$$u_{.i}(t, \xi) \approx u_{.i}^{[P]}(t, \xi) = \sum_{j=0}^{P} u^j{}_{.i}(t)\, \Psi_{j.i}(\xi) \equiv u^j{}_{.i}(t)\, \Psi_{j.i}(\xi). \tag{2.3*}$$

  Hence, $\dot{u}_{.i} = \partial_t u_{.i}$.

(b) Transfer the random modes of $s_{.i-1} = (u_{.i-1}, \dot{u}_{.i-1})$ to $s_{.i} = (u_{.i}, \dot{u}_{.i})$ at $t = t_i$, given that $i \geq 1$.

One way to achieve this is to ensure that the probability information of the system's state is transferred in the mean-square sense. Put differently, we wish to make sure that the equalities shown below hold in the mean-square sense (summation sign implied only over repeated index k):

$$u_{.i}(t_i, \xi) = u_{.i-1}(t_i, \xi) \quad \Longleftrightarrow \quad u^k{}_{.i}(t_i)\, \Psi_{k.i}(\xi) = u^k{}_{.i-1}(t_i)\, \Psi_{k.i-1}(\xi) \tag{2.21a}$$

$$\dot{u}_{.i}(t_i, \xi) = \dot{u}_{.i-1}(t_i, \xi) \quad \Longleftrightarrow \quad \dot{u}^k{}_{.i}(t_i)\, \Psi_{k.i}(\xi) = \dot{u}^k{}_{.i-1}(t_i)\, \Psi_{k.i-1}(\xi). \tag{2.21b}$$

Projecting (2.21) onto $\mathfrak{Z}_i^{[P]}$ gives:

$$u^k{}_{.i}(t_i)\, \langle \Psi_{j.i}, \Psi_{k.i} \rangle = u^k{}_{.i-1}(t_i)\, \langle \Psi_{j.i}, \Psi_{k.i-1} \rangle \tag{2.22a}$$

$$\dot{u}^k{}_{.i}(t_i)\, \langle \Psi_{j.i}, \Psi_{k.i} \rangle = \dot{u}^k{}_{.i-1}(t_i)\, \langle \Psi_{j.i}, \Psi_{k.i-1} \rangle. \tag{2.22b}$$

Thus, upon simplification yields the random modes of $s_{\cdot i} = (u_{\cdot i}, \dot{u}_{\cdot i})$ at $t = t_i$:

$$u^j{}_{\cdot i}(t_i) = \frac{\langle \Psi_{j\cdot i}, \Psi_{k\cdot i-1} \rangle}{\langle \Psi_{j\cdot i}, \Psi_{j\cdot i} \rangle} u^k{}_{\cdot i-1}(t_i) \quad \text{and} \quad \dot{u}^j{}_{\cdot i}(t_i) = \frac{\langle \Psi_{j\cdot i}, \Psi_{k\cdot i-1} \rangle}{\langle \Psi_{j\cdot i}, \Psi_{j\cdot i} \rangle} \dot{u}^k{}_{\cdot i-1}(t_i), \qquad (2.6b^*)$$

where $j \in \{0, 1, \ldots, P\}$ (summation sign implied over repeated index k). These are to be interpreted as the initial conditions of the system over the region $\mathfrak{R}_i$.

If $i = 0$, the initial conditions are computed with (2.6b) directly.

(c) Substitute (2.3*) into (2.1) to obtain (2.4).

(d) Project (2.4a) onto $\mathfrak{L}_i^{[P]}$ to obtain (2.6a) subject to (2.6b*). Note that if $i = 0$, (2.6a) is subject to (2.6b).

(e) Integrate (2.6) 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 $\{u^j{}_{\cdot i}(t_{i+1})\}_{j=0}^P$ and $\{\dot{u}^j{}_{\cdot i}(t_{i+1})\}_{j=0}^P$.

(f) Compute both the mean and the variance of each of the components of output $y = \boldsymbol{\mathcal{M}}[u][x]$ over $\mathfrak{R}_i$, by recurring to the formulas prescribed by (2.8) and (2.9).

2. Post-process results.

Remark 2.1. Any of the following two approaches can be carried out at the start of the simulation ($i = 0$) to address the case when the initial conditions are deterministic or, more generally, linearly dependent:

- When the initial conditions are deterministic, the first two vectors are required to be removed from the set $\{\Phi_{j\cdot i} := \hat{\varphi}^j(M)(0, \hat{s}(t_i, \cdot))\}_{j=1}^P$ for they are constant, and when the initial conditions are stochastic but linearly dependent, only one of them needs to be removed from the set.

- When the initial conditions are deterministic or linearly dependent, the gPC method [13,33] can be employed instead to advance the state of the system one-time step forward; that is, from $s(t_0 = 0, \cdot)$ to $s(t_1, \cdot)$. After this, the gPC method can be switched over FSC to continue pushing the system's state forward in time.

Remark 2.2. Compared to the standard TD-gPC by Gerritsma et al. [34], in our FSC scheme we update the stochastic part of the solution space at every time step to minimize the error

propagation over time. We do this without loss of generality since the scheme can conveniently be modified to incorporate a stopping criterion of one's choice and update the random basis only when the criterion is met.

2.6. Numerical results

We demonstrate and compare the performance of the FSC scheme to the mTD-gPC scheme using two numerical examples for the dynamical system described in Section 2.3. We also define the local error, $\epsilon : \mathcal{T} \to \mathcal{T}$, and the global error, $\epsilon_G : \mathcal{T} \to \mathbb{R}$, with the following expressions:

$$\epsilon[f](t) = |f(t) - f_{\text{exact}}(t)| \tag{2.23a}$$

$$\epsilon_G[f] = \frac{1}{T}\int_{\mathfrak{T}} |f(t) - f_{\text{exact}}(t)|\,\mathrm{d}t \approx \frac{\Delta t}{T}\sum_{i=0}^{N} |f(t_i) - f_{\text{exact}}(t_i)|, \tag{2.23b}$$

where Δt 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 $t_N = N\,\Delta t = T$).

2.6.1. Single-degree-of-freedom system under free vibration

We consider an undamped single-degree-of-freedom system governed by $m\ddot{u} + ku = 0$ with mass $m = 100$ kg and stochastic stiffness $k(\xi) = \xi$ subjected to free vibration. Three different cases of stochasticity are considered as listed in Table 2.1. The system has an initial displacement of $u(0,\cdot) \equiv 0.05$ m, and an initial velocity of $\dot{u}(0,\cdot) \equiv 0.20$ m/s. The duration of the simulation is set to $T = 150$ s. To minimize the errors coming from the discretization of $\mathcal{T}(2)$, the time-step size is taken as $\Delta t = 0.005$ s, meaning that a total of $N = 30\,000$ time steps are employed in the simulation. To integrate (2.6) over time[2], we use the RK4 method, and as described in Section 2.5.4, the stochastic part of the solution space is updated at every time step in order to obtain accurate results. Lastly, because the initial conditions of the

[2]In this problem we have taken $\mathcal{F}[u,\dot{u}] = ku$, and thus $\mathcal{F}^i[u^j,\dot{u}^j](t) = k^i{}_j\,u^j(t)$ with $k^i{}_j = \langle\Psi_i, k\Psi_j\rangle/\langle\Psi_i,\Psi_i\rangle$.

system are deterministic, we opt to employ the gPC method (with $P = 6$) [33] for the first 5 seconds of the simulation, in an effort to ensure that the stochasticity of the system's state is well developed for the analysis with FSC or mTD-gPC.

Figs. 2.3 and 2.4 show the evolution of the mean and variance of the system's state. From these figures, it is observed that the response obtained with FSC using only 7 basis vectors has the ability to reproduce the exact response (from Appendix 2.B) with high fidelity. This is the reason why the two plots appear to be indistinguishable from each other. The figures also show the limit values for each response computed using the exact expressions given by (2.30) and (2.32).

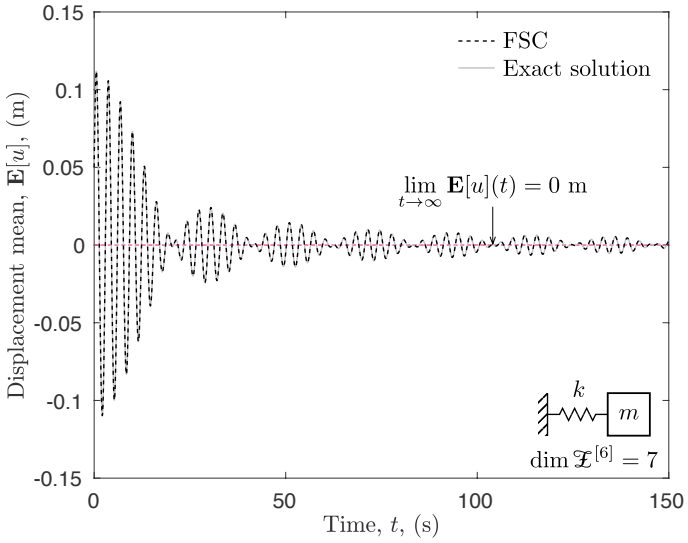
Figs. 2.5 to 2.7 present the local errors in mean and variance of the system's state using different choices of number of basis vectors ranging from 3 to 7. Note that even though the FSC scheme requires P to be greater than or equal to 3, here we also study the case when $P = 2$ for sake of comparison. Figs. 2.5 to 2.7 show that as the number of basis vectors increases, so does the accuracy of the results. In particular, by increasing the number of basis vectors from 3 to 5, the accuracy of the results improves significantly by an order of magnitude of 6 (approximately from 10^{-1} to 10^{-7} for the mean), whereas when the number of basis vectors is increased from 5 to 7, the improvement in error is more moderate (to approximately 10^{-8} for the mean). However, for the 7-basis-vector case we do see an improvement in the accuracy of the solution as time progresses due to the increase in the number of basis vectors used. Fig. 2.8 presents the convergence of global errors as a function of the number of basis vectors and the different distributions used to define the stochasticity of k. The FSC scheme achieves exponential convergence when 3, 4 and 5 basis vectors are used, but adding more basis vectors does not improve the accuracy of the response. The primary reason for this slow-down in convergence is that the accuracy of the solution is limited by machine precision and the fact that the probability information is being transferred in the mean-square sense at every time step (for P is finite). In fact, these plots indicate that there is no reason to implement more than 5 basis vectors into the simulation as it does not improve the accuracy of the results significantly. It is also apparent from this figure that when k is assumed gamma-distributed, the results are not as accurate as those obtained from the uniform and beta distributions. The

Table 2.1. Case studies considered for a single-degree-of-freedom system under free vibration and stochastic stiffness

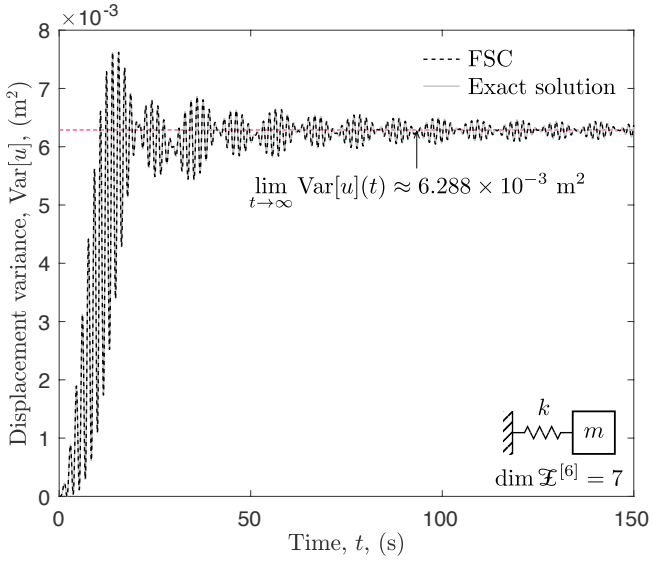
Case	Probability distribution*	Probability moments	Quadrature rule
1	Uniform $\sim \xi \in \Xi = [a, b]$	$\mathbf{E}[\xi] = 400$ N/m	Gauss-Legendre (100 points)
	$a = 340$ N/m, $b = 460$ N/m	$\mathrm{Var}[\xi] = 1\,200$ N^2/m^2	
2	Beta$(\alpha, \beta) \sim \xi \in \Xi = [a, b]$	$\mathbf{E}[\xi] \approx 374.3$ N/m	Gauss-Jacobi (95 points)
	$\alpha = 2$, $\beta = 5$	$\mathrm{Var}[\xi] \approx 367.3$ N^2/m^2	$\alpha_J := \beta - 1 = 4$
	$a = 340$ N/m, $b = 460$ N/m		$\beta_J := \alpha - 1 = 1$
3	Gamma$(\alpha, \beta) \sim \xi \in \Xi = [a, \infty)$	$\mathbf{E}[\xi] = 440$ N/m	Gauss-Laguerre (135 points)
	$\alpha = 10$, $\beta = 1/10$	$\mathrm{Var}[\xi] = 1\,000$ N^2/m^2	$\alpha_L := \alpha - 1 = 9$
	$a = 340$ N/m		

*Probability density function, $f : \Xi \to \mathbb{R}_0^+$:

$$\text{Uniform} \sim f(\xi) = \frac{1}{b-a}, \quad \text{Beta} \sim f(\xi) = \frac{(\xi - a)^{\alpha-1}\,(b - \xi)^{\beta-1}}{(b-a)^{\alpha+\beta-1}\,\mathrm{B}(\alpha, \beta)} \quad \text{and} \quad \text{Gamma} \sim f(\xi) = \frac{\beta^\alpha}{\Gamma(\alpha)}(\xi - a)^{\alpha-1}\exp(-\beta\,(\xi - a)).$$

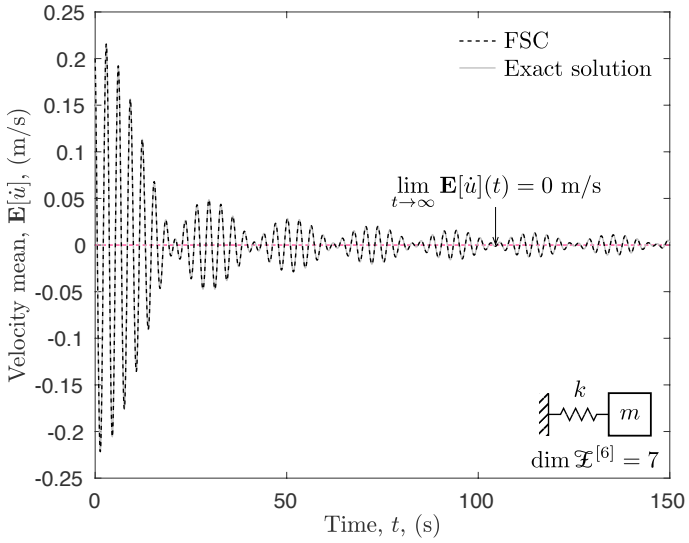


(a) Mean

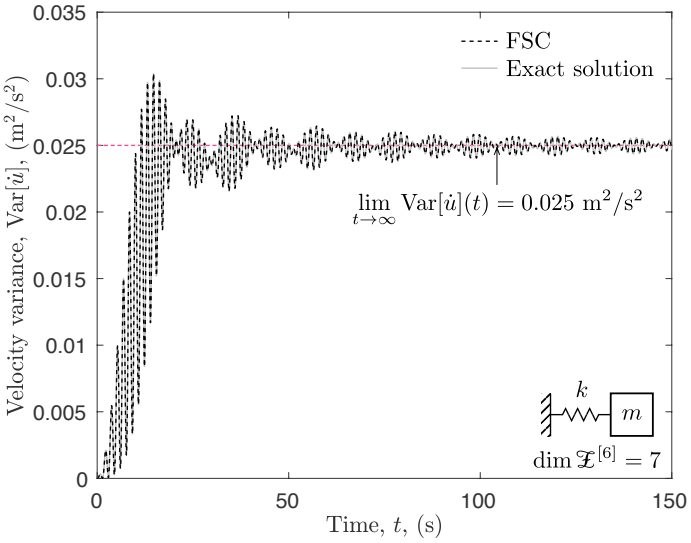


(b) Variance

Figure 2.3. Evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for the case when the p-discretization level of RFS is $\mathfrak{Z}^{[6]}$ and $\mu \sim$ Uniform

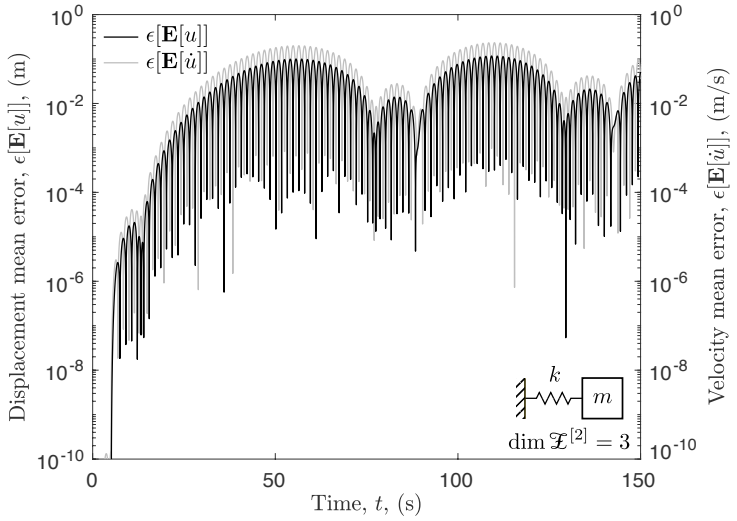


(a) Mean

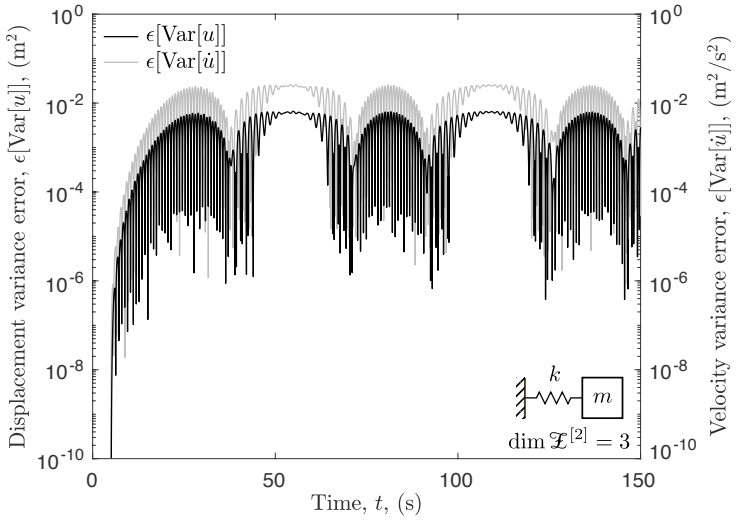


(b) Variance

Figure 2.4. Evolution of $\mathbf{E}[\dot{u}]$ and $\mathrm{Var}[\dot{u}]$ for the case when the p-discretization level of RFS is $\mathfrak{Z}^{[6]}$ and $\mu \sim$ Uniform

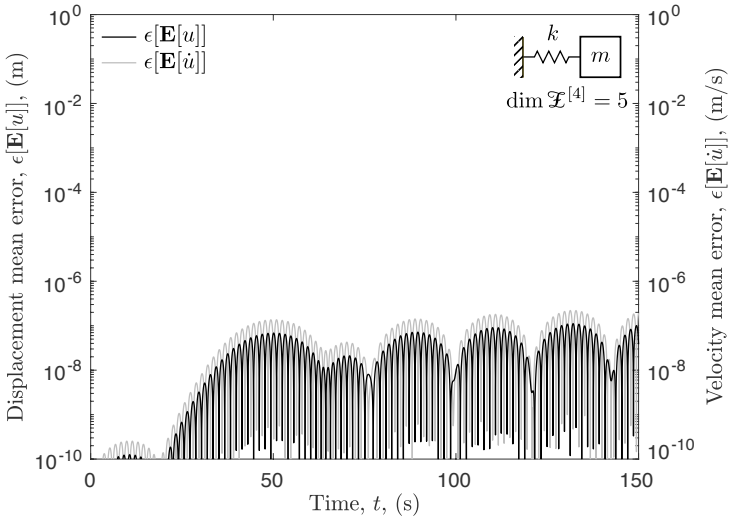


(a) Mean error for $\mathfrak{Z}^{[2]}$

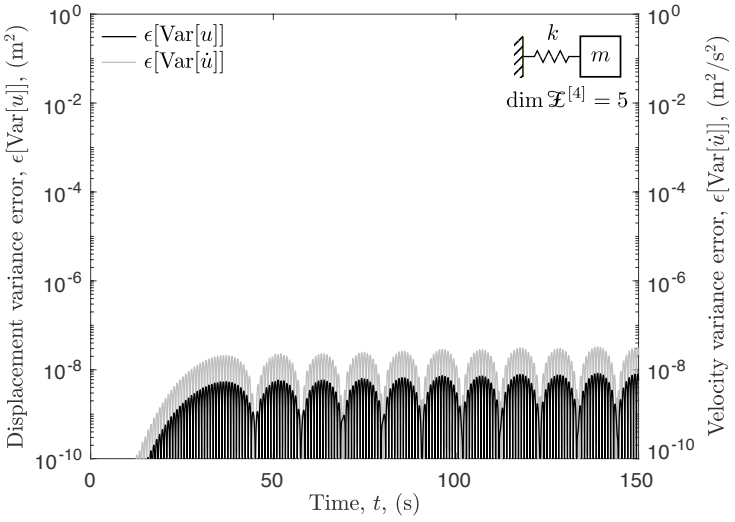


(b) Variance error for $\mathfrak{Z}^{[2]}$

Figure 2.5. Local error evolution of $\mathbf{E}[u]$, $\mathrm{Var}[u]$, $\mathbf{E}[\dot{u}]$ and $\mathrm{Var}[\dot{u}]$ for different p-discretization levels of RFS and for $\mu \sim$ Uniform (SET 1/3)

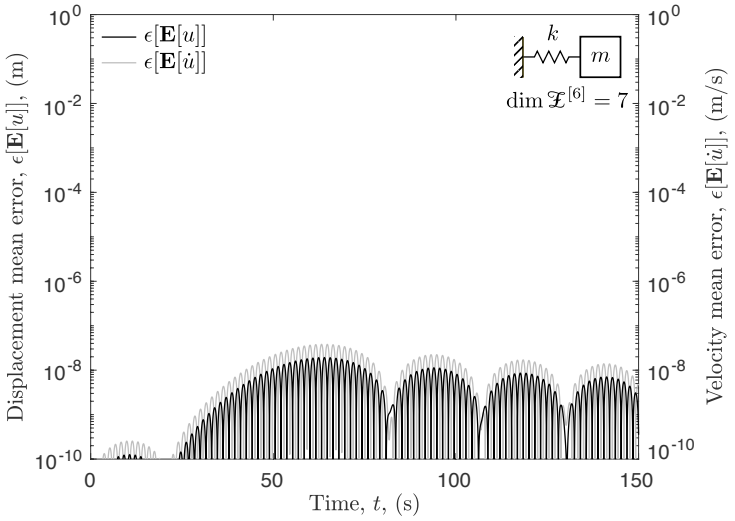


(a) Mean error for $\mathfrak{Z}^{[4]}$

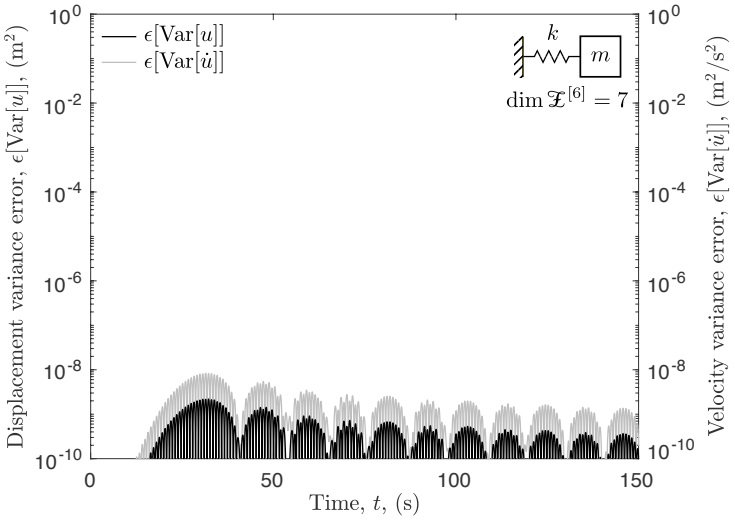


(b) Variance error for $\mathfrak{Z}^{[4]}$

Figure 2.6. Local error evolution of $\mathbf{E}[u]$, $\mathrm{Var}[u]$, $\mathbf{E}[\dot{u}]$ and $\mathrm{Var}[\dot{u}]$ for different p-discretization levels of RFS and for $\mu \sim$ Uniform (SET 2 / 3)

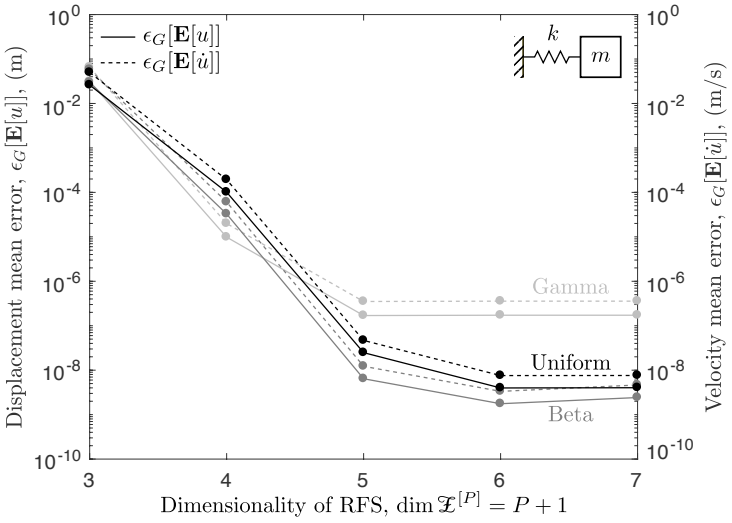


(a) Mean error for $\mathfrak{Z}^{[6]}$

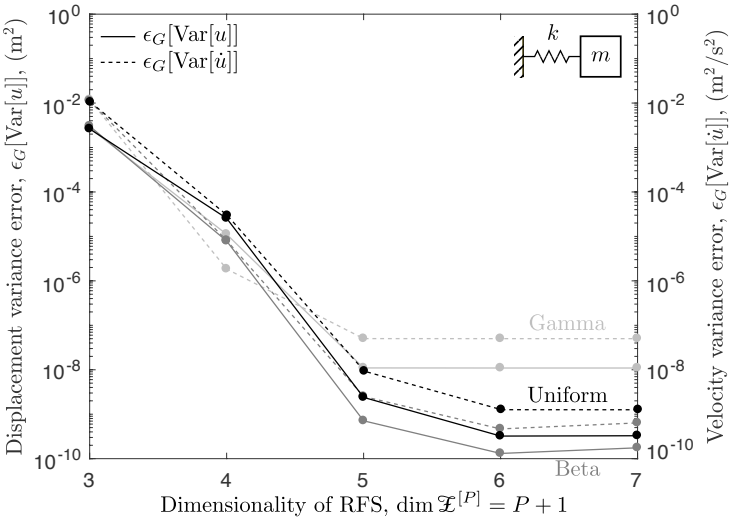


(b) Variance error for $\mathfrak{Z}^{[6]}$

Figure 2.7. Local error evolution of $\mathbf{E}[u]$, $\mathrm{Var}[u]$, $\mathbf{E}[\dot{u}]$ and $\mathrm{Var}[\dot{u}]$ for different p-discretization levels of RFS and for $\mu \sim$ Uniform (SET 3 / 3)

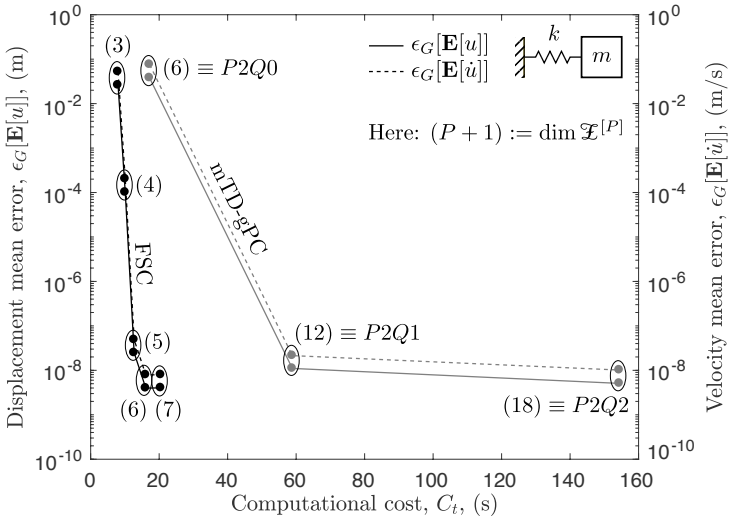


(a) Mean error

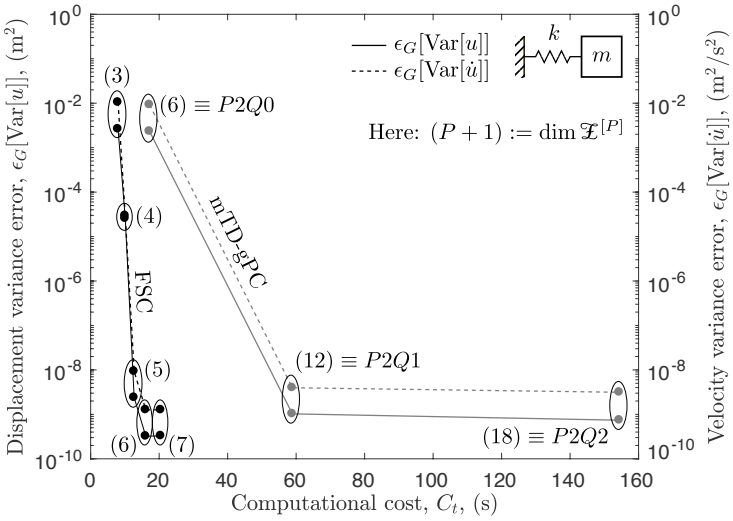


(b) Variance error

Figure 2.8. Global error of $\mathbf{E}[u]$, $\mathrm{Var}[u]$, $\mathbf{E}[\dot{u}]$ and $\mathrm{Var}[\dot{u}]$ for different p-discretization levels of RFS



(a) Mean error



(b) Variance error

Figure 2.9. Global error versus computational cost for $\mu \sim$ Uniform

reason behind this outcome is that when k is gamma-distributed its support is unbounded, which from a numerical viewpoint leads to the dreaded case of unbounded basis vectors.

Fig. 2.9 plots the global errors as a function of computational cost[3] of the FSC and mTD-gPC schemes expressed in terms of the wall-clock time taken to complete the computation. The implementation of both schemes was optimized as much as reasonably possible, and the labels $P2Q0$, $P2Q1$ and $P2Q2$ are defined in Ref. [35] (Pg. 45). The comparison of computational cost is shown here only for Case 1 of Table 2.1 (for sake of brevity), but similar trends are observed for cases 2 and 3 as well. Note that FSC is much faster in comparison to mTD-gPC for a similar level of error. For instance, in order to attain a global error of approximately 10^{-8}, FSC runs about 3.5 times faster than mTD-gPC. This is because, in general, FSC requires much fewer basis vectors than mTD-gPC—to achieve an error of about 10^{-8}, FSC requires only 6 basis vectors in comparison to 12 for mTD-gPC. Another reason for the superior efficiency of FSC is that, for mTD-gPC, the orthogonalization process needs to be conducted three times when the random basis is demanded to be updated during the simulation (one time for the monomials of u, another time for the monomials of $\dot{u}$, and one more time after performing the tensor product)[4]. This also explains why using 6 basis vectors in both methods, FSC runs slightly faster than mTD-gPC. Furthermore, we see that the probability information is better encoded in FSC because it uses a fewer number of basis vectors to achieve the same level of accuracy. Fig. 2.9 also reveals that increasing the number of basis vectors from 6 to 7 for FSC and 12 to 18 for mTD-gPC does not improve the accuracy of the results significantly. This, again, is because of the limited precision of the machine and the fact that the probability information is being transferred in the mean-square sense.

---

[3]All problems in this work were run in MATLAB R2016b [62] 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).

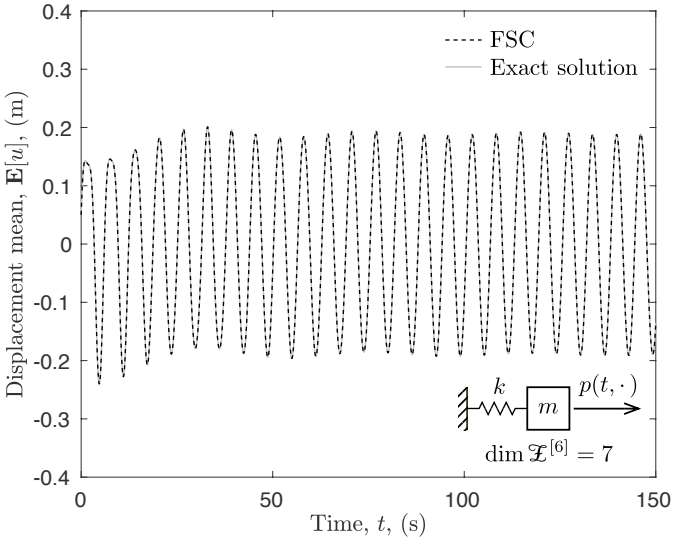
[4]We tested the method by orthogonalizing only once—namely, after performing the tensor product between the monomials of u and $\dot{u}$—, and found that the accuracy of the results degrades noticeably.

2.6.2. Single-degree-of-freedom system under forced vibration

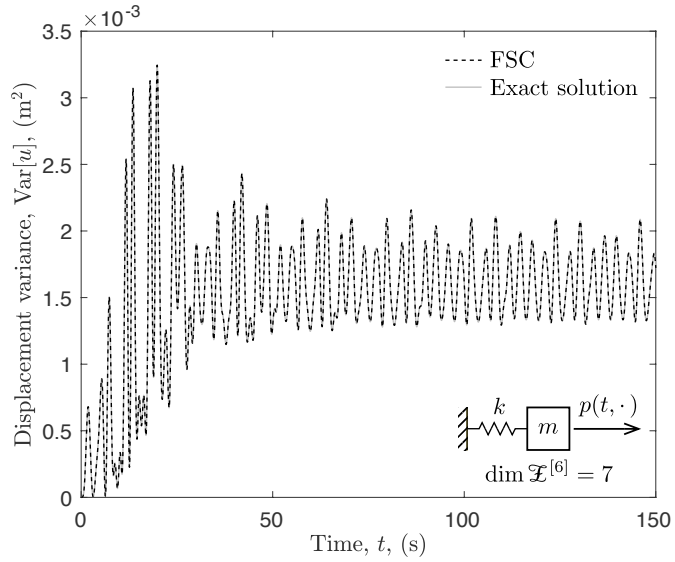
In this example, we show that the number of basis vectors needed in the simulation does not increase when the dimensionality of the random space increases. For this, we consider the same system described in the Example 2.6.1 (including the same deterministic initial conditions), with the only difference being that the system is subjected to a stochastic external force given by $p(t,\cdot) = q\sin(t)$. That is, the system is now governed by $m\ddot{u}+ku = p$. Here the stiffness $k(\xi) = \xi^1$ is taken to be the same as Case 1 of Table 2.1 with $\bar{\Xi}_1 = [340, 460]$ N/m. The amplitude of the external force $q(\xi) = \xi^2$ is assumed beta-distributed with parameters $\alpha = 2$ and $\beta = 5$ in $\bar{\Xi}_2 = [51, 69]$ N, giving thereby probability moments: $\mathbf{E}[\xi^2] \approx 56.14$ N and $\mathrm{Var}[\xi^2] \approx 8.265$ N^2. Because two random variables are present in the mathematical model, the random domain of the system is 2-dimensional, and thus, it is defined by $\Xi = \bar{\Xi}_1 \times \bar{\Xi}_2$ with $\mu \sim \text{Uniform} \otimes \text{Beta}$. For this example, the inner products are computed using a quadrature rule constructed by performing a cartesian product between 100 Gauss-Legendre points distributed along the ξ^1-axis and 95 Gauss-Jacobi points distributed along the ξ^2-axis. Finally, the gPC method (with $P = 8$) is used for the first 0.5 seconds of the simulation to allow the stochasticity of the system to develop sufficiently before using the FSC scheme.

Figs. 2.10 and 2.11 show the evolution of the mean and variance of the system's state. As in the previous example, the numerical solution obtained using FSC with 7 basis vectors is indistinguishable from the exact response[5]. Figs. 2.12 to 2.15 depict the local and global errors in mean and variance of the system's state. Here we also notice the same trend found in Figs. 2.5 to 2.8. That is, as the number of basis vectors increases, so does the accuracy of the results. Moreover, when the number of basis vectors increases from 3 to 5, the error for the mean drops down from approximately 10^{-1} to 10^{-7}, whereas increasing the number of basis vectors from 5 to 7 does not result in a noticeable improvement in the computation of the probability moments. Note that using the same number of basis vectors as in Example 2.6.1 led to similar levels of error, even though the dimensionality of the random space in

[5]To obtain the 'exact' solution for $\mathbf{E}[u]$, $\mathrm{Var}[u]$, $\mathbf{E}[\dot{u}]$ and $\mathrm{Var}[\dot{u}]$, the corresponding values for $\mathbf{E}[u(t_i,\cdot)]$, $\mathrm{Var}[u(t_i,\cdot)]$, $\mathbf{E}[\dot{u}(t_i,\cdot)]$ and $\mathrm{Var}[\dot{u}(t_i,\cdot)]$ were computed at each instant of time $t_i \in \mathfrak{T}$ using the `vpaintegral` command (provided in the MATLAB's Symbolic Math Toolbox [62]) with `RelTol` set equal to 10^{-14}. The exact displacement response, u, is well known and can be found in any structural dynamics textbook, e.g. [31, 63].

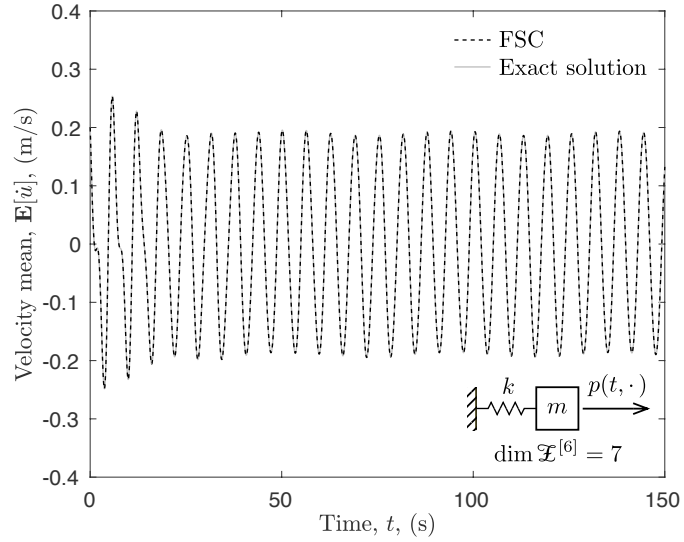


(a) Mean

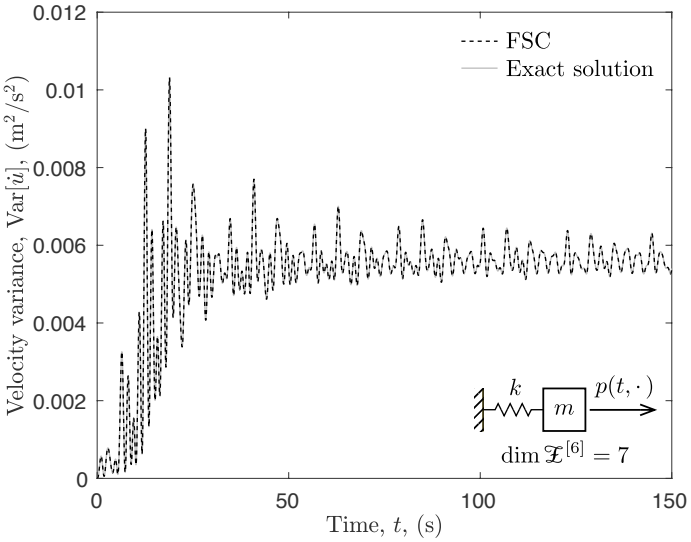


(b) Variance

Figure 2.10. Evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for the case when the p-discretization level of RFS is $\mathfrak{Z}^{[6]}$ and $\mu \sim \text{Uniform} \otimes \text{Beta}$

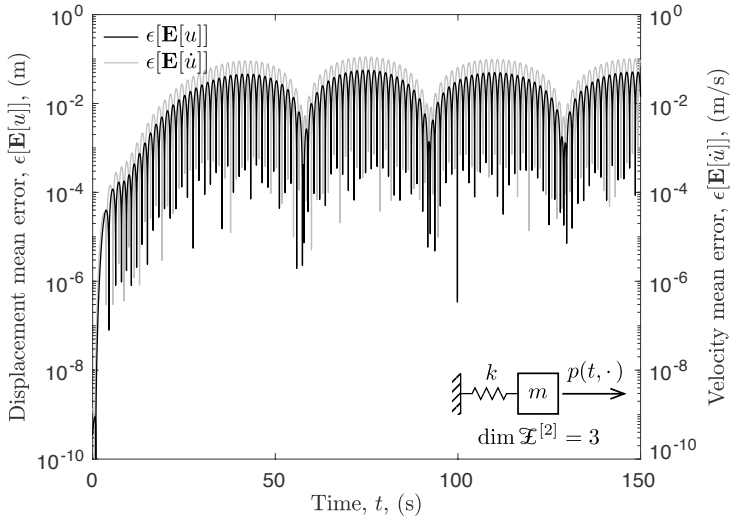


(a) Mean

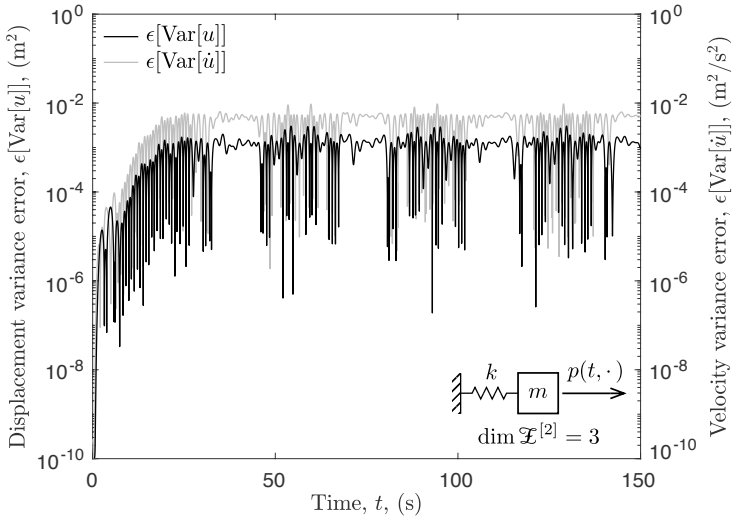


(b) Variance

Figure 2.11. Evolution of $\mathbf{E}[\dot{u}]$ and $\mathrm{Var}[\dot{u}]$ for the case when the p-discretization level of RFS is $\mathfrak{Z}^{[6]}$ and $\mu \sim \text{Uniform} \otimes \text{Beta}$

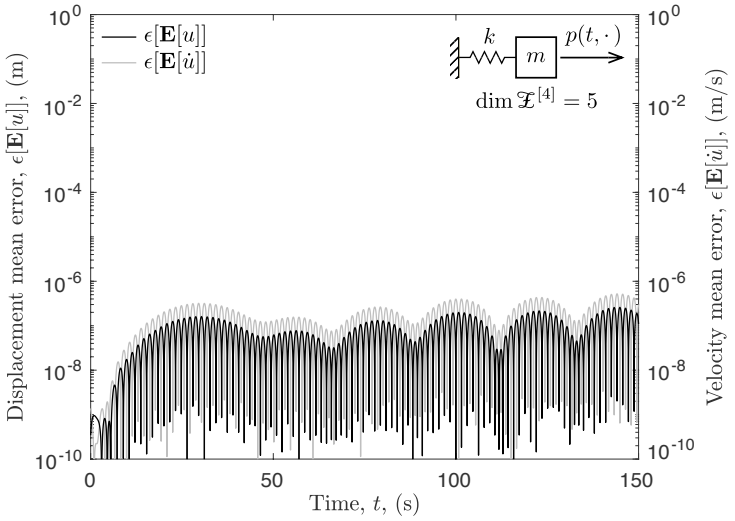


(a) Mean error for $\mathfrak{Z}^{[2]}$

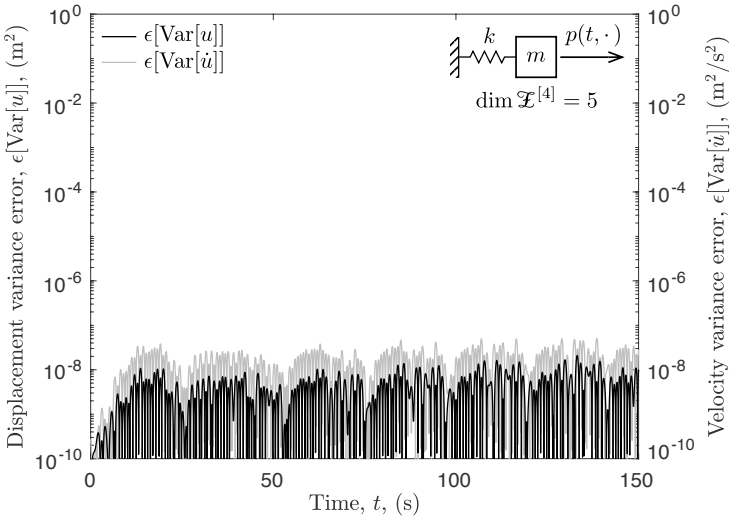


(b) Variance error for $\mathfrak{Z}^{[2]}$

Figure 2.12. Local error evolution of $\mathbf{E}[u]$, $\mathrm{Var}[u]$, $\mathbf{E}[\dot{u}]$ and $\mathrm{Var}[\dot{u}]$ for different p-discretization levels of RFS and for $\mu \sim \mathrm{Uniform} \otimes \mathrm{Beta}$ (SET 1/3)

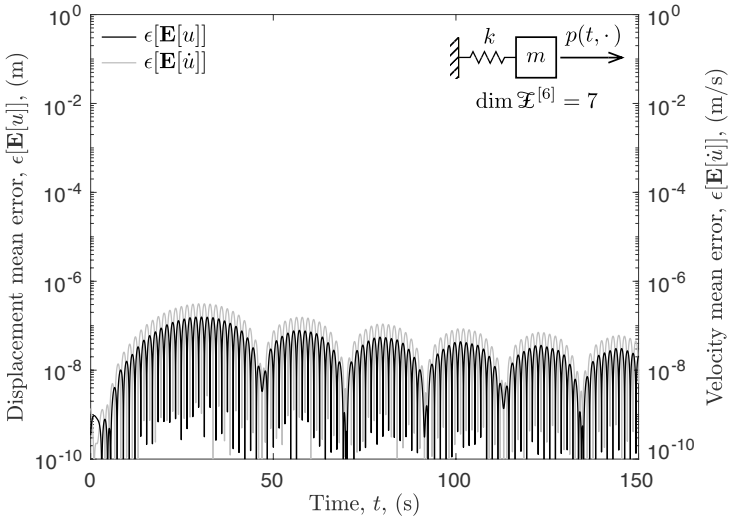


(a) Mean error for $\mathfrak{Z}^{[4]}$

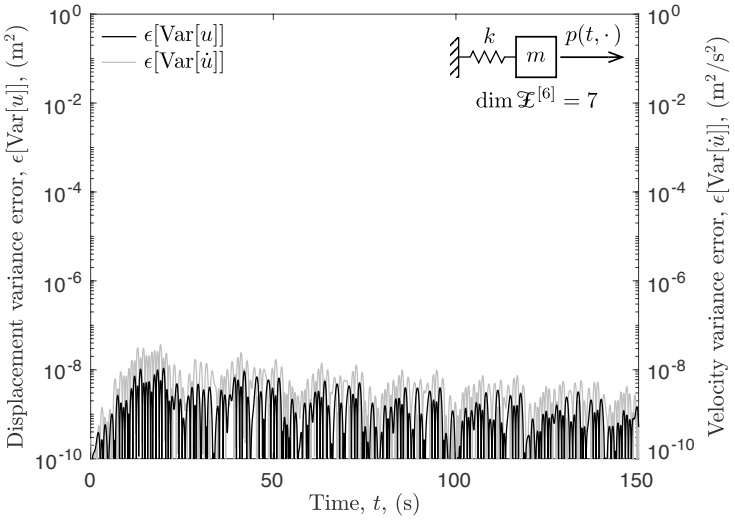


(b) Variance error for $\mathfrak{Z}^{[4]}$

Figure 2.13. Local error evolution of $\mathbf{E}[u]$, $\mathrm{Var}[u]$, $\mathbf{E}[\dot{u}]$ and $\mathrm{Var}[\dot{u}]$ for different p-discretization levels of RFS and for $\mu \sim \mathrm{Uniform} \otimes \mathrm{Beta}$ (SET 2/3)

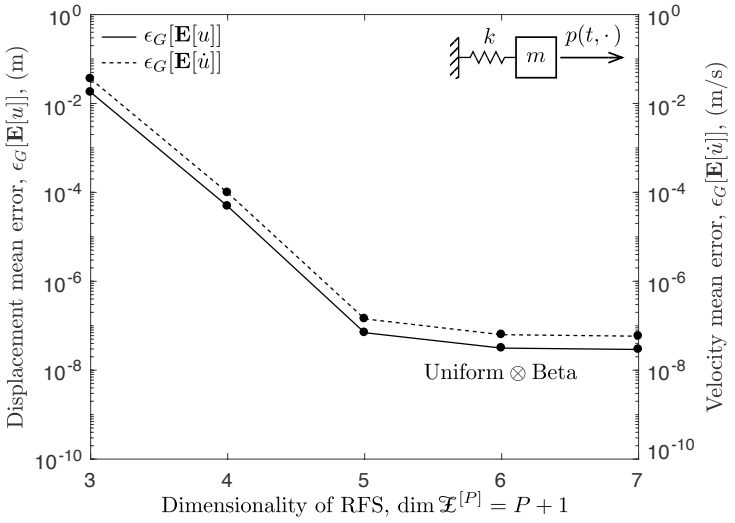


(a) Mean error for $\mathfrak{Z}^{[6]}$

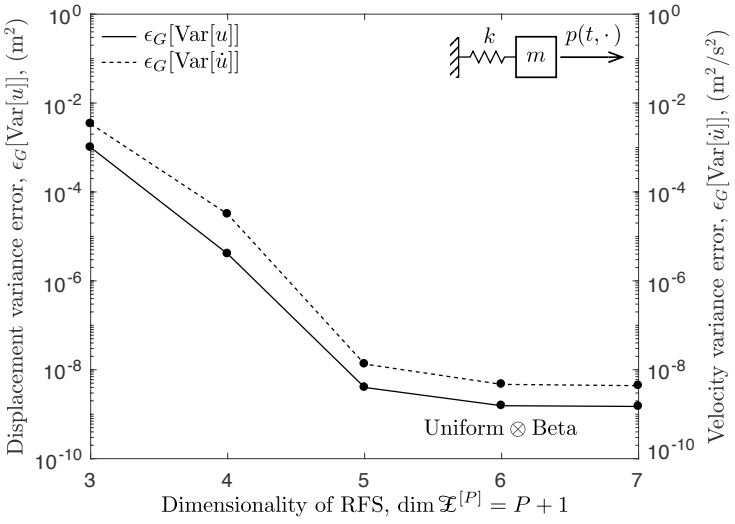


(b) Variance error for $\mathfrak{Z}^{[6]}$

Figure 2.14. Local error evolution of $\mathbf{E}[u]$, $\mathrm{Var}[u]$, $\mathbf{E}[\dot{u}]$ and $\mathrm{Var}[\dot{u}]$ for different p-discretization levels of RFS and for $\mu \sim \mathrm{Uniform} \otimes \mathrm{Beta}$ (SET 3/3)

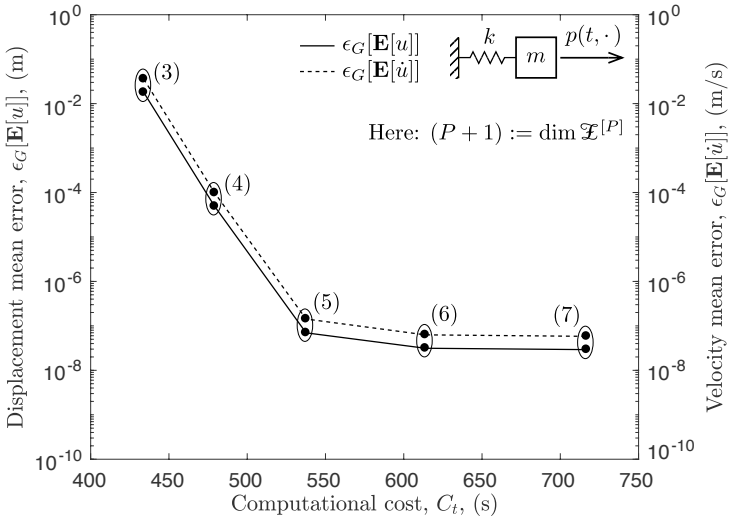


(a) Mean error

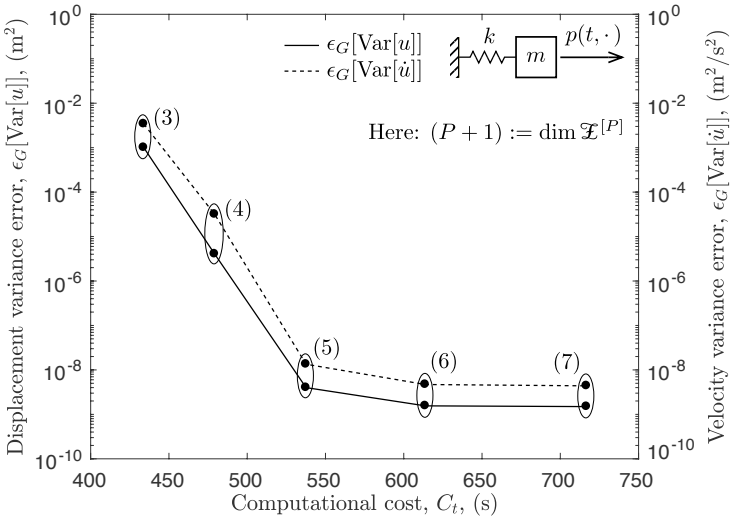


(b) Variance error

Figure 2.15. Global error of $\mathbf{E}[u]$, $\mathrm{Var}[u]$, $\mathbf{E}[\dot{u}]$ and $\mathrm{Var}[\dot{u}]$ for different p-discretization levels of RFS



(a) Mean error



(b) Variance error

Figure 2.16. Global error versus computational cost for $\mu \sim \text{Uniform} \otimes \text{Beta}$

this example is twice that of Example 2.6.1. However, despite the number of basis vectors being the same, the computational cost of this example is much higher than that of Example 2.6.1 (as per Fig. 2.16) because of the increase in the number of quadrature points needed to compute the inner products accurately.

2.6.3. Nonlinear single-degree-of-freedom system under free vibration

In this last example, we explore the nonlinear behavior of a single-degree-of-freedom system in order to test the ability of the FSC method to solve nonlinear problems. The governing differential equation for this system is set to be given by

$$m\ddot{u} + (1 + \rho u^2)ku = 0, \tag{2.24}$$

where $m = 100$ kg is the mass of the system, $k(\xi) = \xi^1$ is a beta-distributed random variable representing the strength of the stiffness which is given by Case 2 of Table 2.1 with $\bar{\Xi}_1 = [340, 460]$ N/m, and $\rho(\xi) = \xi^2$ is a uniformly-distributed random variable denoting the contributing factor to the nonlinearity of the system in $\bar{\Xi}_2 = [-20, -30]$ m^{-2}. The probability moments for ξ^2 are thus: $\mathbf{E}[\xi^2] = -25$ m^{-2} and $\mathrm{Var}[\xi^2] \approx 8.333$ m^{-4}. The system has an initial displacement of $u(0, \cdot) \equiv 0.05$ m and an initial velocity of $\dot{u}(0, \cdot) \equiv 0.20$ m/s. As in the previous example, the random domain of the system is two-dimensional and defined by $\Xi = \bar{\Xi}_1 \times \bar{\Xi}_2$ with $\mu \sim \text{Beta} \otimes \text{Uniform}$. The inner products are again computed with a Gaussian quadrature rule using the same number of points indicated in the previous section, and the gPC method (with $P = 8$) is used for the first second of the simulation. The system is integrated over time using the RK4 method with a time-step size of $\Delta t = 0.005$ s, and the simulation is set to last $T = 150$ s.

Remark 2.3. According to (2.6), the temporal function $\mathcal{F}^i[u^j, \dot{u}^j]$ associated with $\mathcal{F}[u, \dot{u}] = (1 + \rho u^2)ku$ is given by:

$$\mathcal{F}^i[u^j, \dot{u}^j](t) = \frac{\langle \Psi_i, k\Psi_j \rangle}{\langle \Psi_i, \Psi_i \rangle} u^j(t) + \frac{\langle \Psi_i, \rho k \Psi_j \Psi_k \Psi_l \rangle}{\langle \Psi_i, \Psi_i \rangle} u^j(t)\, u^k(t)\, u^l(t).$$

Note that a summation sign is implied over every repeated index j, k and l.

Since a closed-form solution for (2.24) does not exist, we use the standard Monte Carlo method described in Appendix 2.C to compare the accuracy of the FSC results against it. To this end, one million realizations are randomly sampled from the random domain to conduct the Monte Carlo simulation. The evolution of the mean and variance of the system's displacement is depicted in Figs. 2.17 and 2.18. These figures show that by using only 5 basis vectors, the FSC results can effectively reproduce the Monte Carlo results. In fact, Figs. 2.19 and 2.20 further indicate that when $P = 4, 5$, the FSC results are 4-order-of-magnitude accurate for the mean and about 5-order-of-magnitude accurate for the variance. However, this is not the case for $P = 3$, which overall is one order of magnitude less accurate and displays a nearly linear drift after 50 s. Moreover, it can be seen that the FSC results with $P = 4$ and $P = 5$ are indistinguishable from each other, chiefly because the one-million Monte Carlo simulation used as the reference solution is an approximate version of the solution. This explains why the accuracy of the results did not increase when P was set equal to 5. Therefore, comparable results are achievable for this nonlinear problem if FSC is run with $P = 4$.

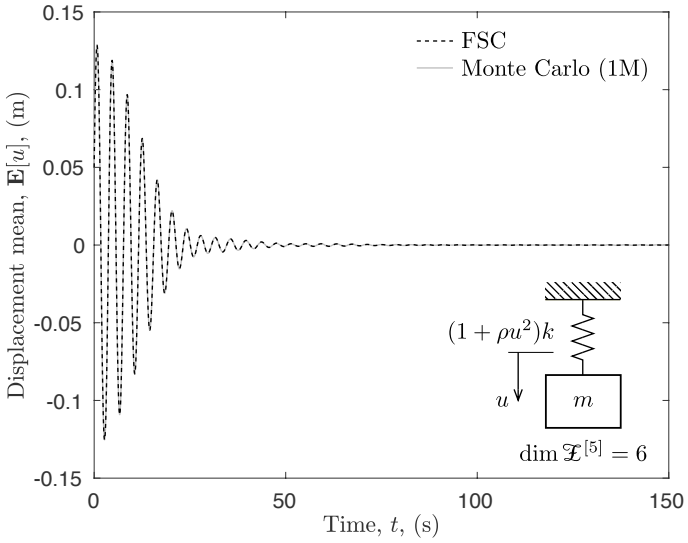
2.7. Application to structural dynamics

In structural dynamics, real-life systems are commonly modeled as multiple-degree-of-freedom systems. In order to demonstrate how FSC can be utilized in a more general setting, in this section we quantify the response uncertainties of a 3-story building (Fig. 2.21a) excited by the effects of a ground motion. The ground motion is taken here to be one of the ground accelerations recorded from the *1940 El Centro Earthquake*[6] event. A plot of this ground motion is depicted in Fig. 2.21b for sake of reference.

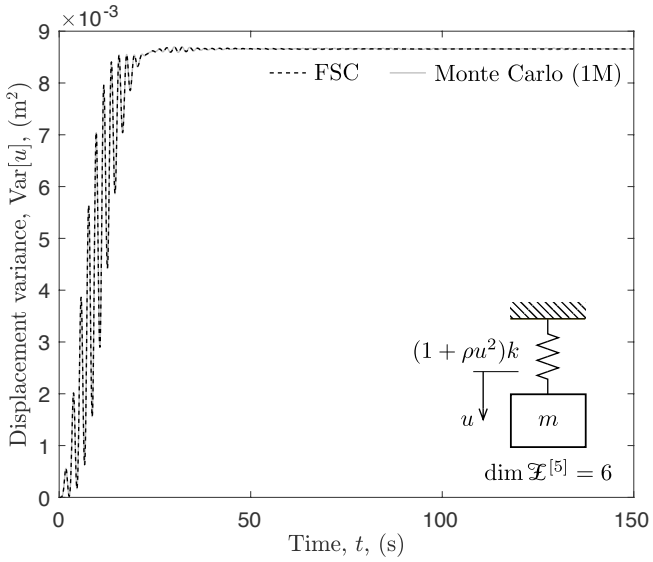
The governing differential equation of motion for this system is:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = -\mathbf{M}\boldsymbol{\iota}\ddot{u}_g \quad (=: \mathbf{p}), \tag{2.25}$$

[6]This ground acceleration was obtained from the *PEER Ground Motion Database* [64]. Website: `https://ngawest2.berkeley.edu`. Event's name: `Imperial Valley-02`. Station's name: `El Centro Array #9`. File's name: `RSN6_IMPVALL.I_I-ELC180.AT2`.

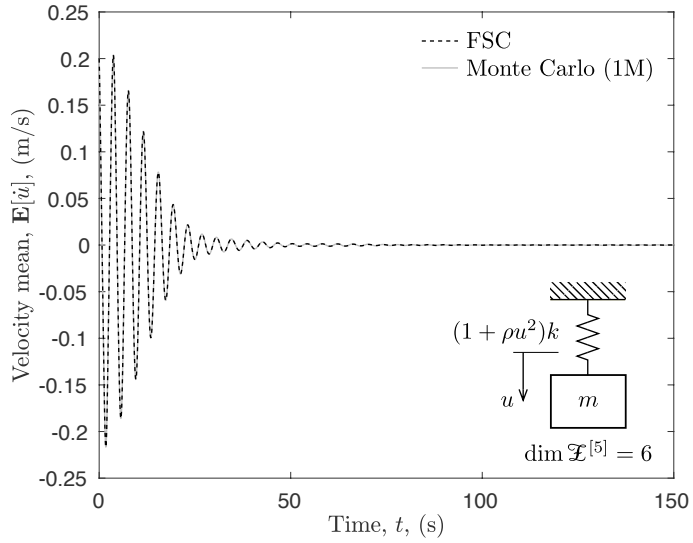


(a) Mean

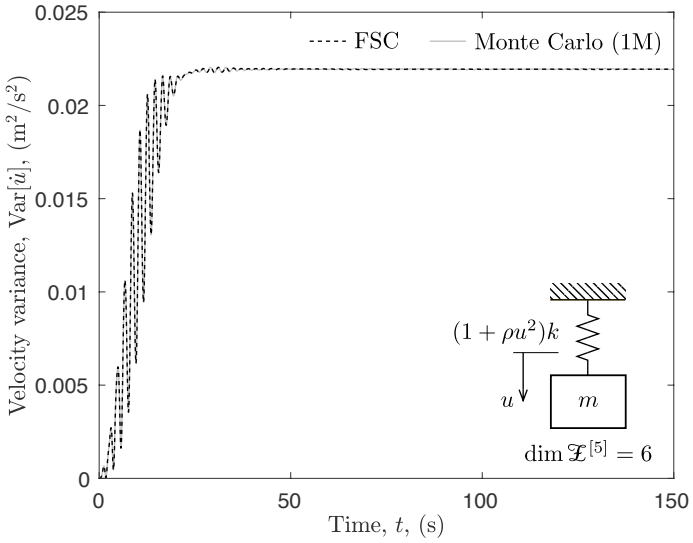


(b) Variance

Figure 2.17. Evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for the case when the p-discretization level of RFS is $\mathfrak{Z}^{[5]}$ and $\mu \sim \mathrm{Beta} \otimes \mathrm{Uniform}$

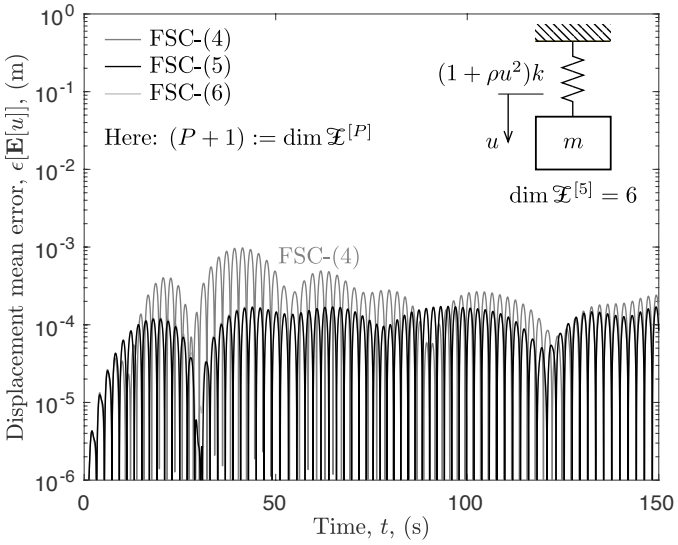


(a) Mean

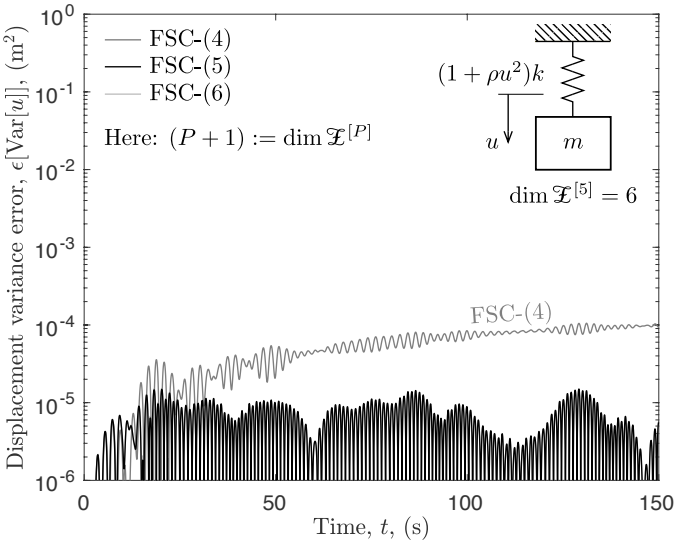


(b) Variance

Figure 2.18. Evolution of $\mathbf{E}[\dot{u}]$ and $\mathrm{Var}[\dot{u}]$ for the case when the p-discretization level of RFS is $\mathfrak{X}^{[5]}$ and $\mu \sim \text{Beta} \otimes \text{Uniform}$

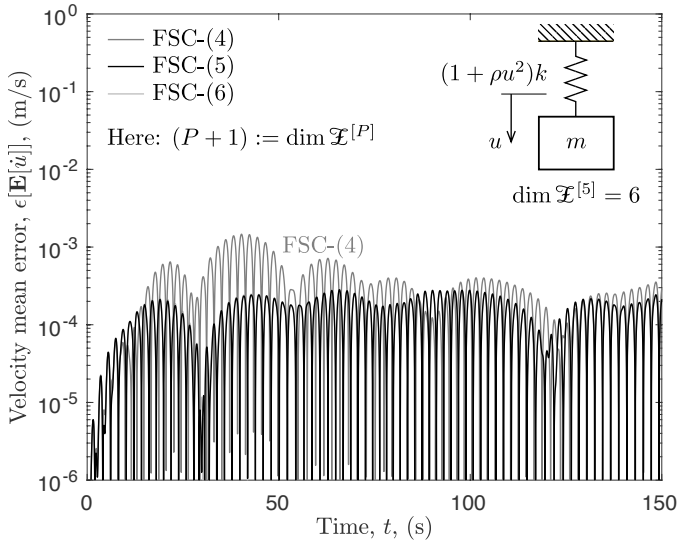


(a) Mean error

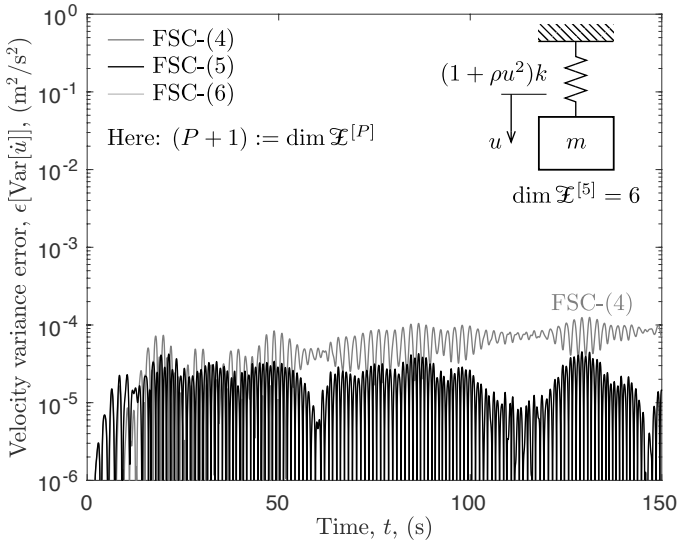


(b) Variance error

Figure 2.19. Local error evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different p-discretization levels of RFS with respect to the 1-million Monte Carlo simulation ($\mu \sim \mathrm{Beta} \otimes \mathrm{Uniform}$)

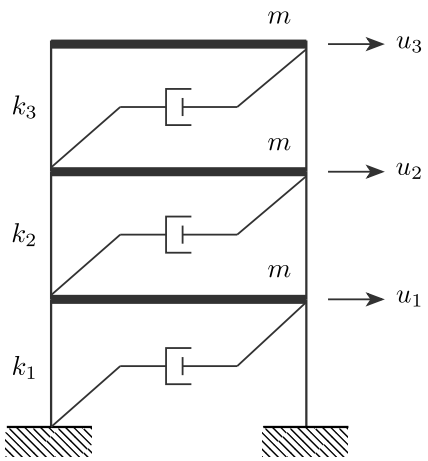


(a) Mean error

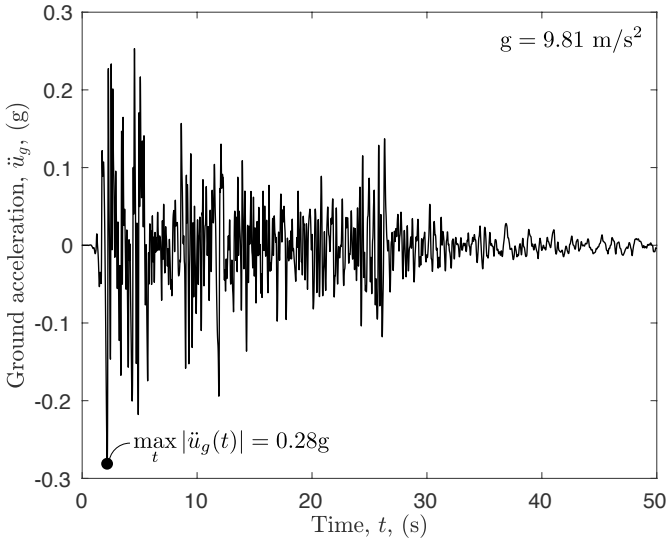


(b) Variance error

Figure 2.20. Local error evolution of $\mathbf{E}[\dot{u}]$ and $\mathrm{Var}[\dot{u}]$ for different p-discretization levels of RFS with respect to the 1-million Monte Carlo simulation ($\mu \sim \text{Beta} \otimes \text{Uniform}$)



(a) Surrogate model of a 3-story building for lateral-load analysis in one direction



(b) Ground acceleration of *1940 El Centro Earthquake*

Figure 2.21. Surrogate model of a 3-story building to investigate its lateral behavior under an earthquake scenario

where $\mathbf{M} \in \mathrm{L}(\mathbb{R}^3, \mathbb{R}^3)$ is the mass matrix, $\mathbf{C} : \Xi \to \mathrm{L}(\mathbb{R}^3, \mathbb{R}^3)$ is the damping matrix, $\mathbf{K} : \Xi \to \mathrm{L}(\mathbb{R}^3, \mathbb{R}^3)$ is the stiffness matrix, $\boldsymbol{\iota} \in \mathbb{R}^3$ is the influence vector, and $\ddot{u}_g : \mathfrak{T} \to \mathbb{R}$ is the ground acceleration characterized by a real-valued function of time. The vectors $\mathbf{u}, \dot{\mathbf{u}} := \partial_t \mathbf{u}, \ddot{\mathbf{u}} := \partial_t^2 \mathbf{u} : \mathfrak{T} \times \Xi \to \mathbb{R}^3$ represent, respectively, the displacement, the velocity and the acceleration of the system, where $\mathbf{u}^T = [u_1, u_2, u_3]$ is the unknown vector sought, and u_3 denotes the roof displacement of the 3-story building (our response of interest here).

The parameters of this system are defined as

$$\mathbf{M} = m \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}, \quad \mathbf{K}(\xi) = \begin{bmatrix} k_1(\xi) + k_2(\xi) & -k_2(\xi) & \\ -k_2(\xi) & k_2(\xi) + k_3(\xi) & -k_3(\xi) \\ & -k_3(\xi) & k_3(\xi) \end{bmatrix},$$

and $\mathbf{C}(\xi) = \alpha(\xi)\,\mathbf{M} + \beta(\xi)\,\mathbf{K}(\xi)$, where $m = 500$ Mg, $k_1(\xi) = \xi^1 \sim \mathrm{Beta}(2,5)$ in $[850 \times 10^3, 1\,150 \times 10^3]$ kN/m, $k_2(\xi) = \xi^2 \sim \mathrm{Beta}(2,5)$ in $[680 \times 10^3, 920 \times 10^3]$ kN/m, $k_3(\xi) = \xi^3 \sim \mathrm{Beta}(2,5)$ in $[680 \times 10^3, 920 \times 10^3]$ kN/m, $\alpha(\xi) = \xi^4 \sim$ Uniform in $[0.4, 0.7]$ s^{-1}, and $\beta(\xi) = \xi^5 \sim$ Uniform in $[0.4 \times 10^{-3}, 0.7 \times 10^{-3}]$ s. Furthermore, $\boldsymbol{\iota}^T = [1,1,1]$, and $\ddot{u}_g$ is defined according to Fig. 2.21b. The random domain for this system is thus 5-dimensional:

$$\Xi = \prod_{i=1}^{5} \bar{\Xi}_i \equiv [850 \times 10^3, 1\,150 \times 10^3] \times [680 \times 10^3, 920 \times 10^3]^2 \times [0.4, 0.7] \times [0.4 \times 10^{-3}, 0.7 \times 10^{-3}],$$

and we assume that the initial state of the system is at rest, i.e. $\mathbf{u}(0) = \dot{\mathbf{u}}(0) = \mathbf{0}$. Note that when the expected values of k_1, k_2 and k_3 are utilized to define $\mathbf{K}$, the fundamental period of the system is approximately 0.33 s, which is consistent with a typical 3-story building found in practice featuring a damping ratio of about 2% for the first two modal frequencies.

Following the formulation presented in Section 2.4, the system of equations (2.6) takes the form:

$$\mathbf{M}^i{}_j \ddot{\mathbf{u}}^j + \mathbf{C}^i{}_j \dot{\mathbf{u}}^j + \mathbf{K}^i{}_j \mathbf{u}^j = \mathbf{p}^i \qquad \text{on } \mathfrak{T} = [0, 50] \text{ s} \tag{2.26a}$$

$$\left\{ \mathbf{u}^i(0) = \mathbf{0},\ \dot{\mathbf{u}}^i(0) = \mathbf{0} \right\} \qquad \text{on } \{0\}, \tag{2.26b}$$

where a summation sign is implied over the repeated index j, $\mathbf{M}^i{}_j = \mathbf{M}\delta^i{}_j$, $\mathbf{p}^i(t) = -\mathbf{M}\boldsymbol{\iota}\,\ddot{u}_g(t)\,\delta^i{}_0$,

$$\mathbf{C}^i{}_j = m\begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}\frac{\langle\Psi_i, \alpha\Psi_j\rangle}{\langle\Psi_i, \Psi_i\rangle} + \begin{bmatrix} 1 & & \\ & 0 & \\ & & 0 \end{bmatrix}\frac{\langle\Psi_i, \beta k_1\Psi_j\rangle}{\langle\Psi_i, \Psi_i\rangle}$$
$$+ \begin{bmatrix} 1 & -1 & \\ -1 & 1 & \\ & & 0 \end{bmatrix}\frac{\langle\Psi_i, \beta k_2\Psi_j\rangle}{\langle\Psi_i, \Psi_i\rangle} + \begin{bmatrix} 0 & & \\ & 1 & -1 \\ & -1 & 1 \end{bmatrix}\frac{\langle\Psi_i, \beta k_3\Psi_j\rangle}{\langle\Psi_i, \Psi_i\rangle},$$

and

$$\mathbf{K}^i{}_j = \begin{bmatrix} 1 & & \\ & 0 & \\ & & 0 \end{bmatrix}\frac{\langle\Psi_i, k_1\Psi_j\rangle}{\langle\Psi_i, \Psi_i\rangle} + \begin{bmatrix} 1 & -1 & \\ -1 & 1 & \\ & & 0 \end{bmatrix}\frac{\langle\Psi_i, k_2\Psi_j\rangle}{\langle\Psi_i, \Psi_i\rangle}$$
$$+ \begin{bmatrix} 0 & & \\ & 1 & -1 \\ & -1 & 1 \end{bmatrix}\frac{\langle\Psi_i, k_3\Psi_j\rangle}{\langle\Psi_i, \Psi_i\rangle}$$

with $i, j \in \{0, 1, \ldots, P\}$, and $\delta^i{}_j$ denoting the Kronecker delta.

For this problem, we take $P = 9$ for the set of linearly independent functions $\{\Phi_{j.i} := \hat{\varphi}^j(0, \hat{\mathbf{s}}(t_i, \cdot))\}_{j=1}^P$, where $\hat{\varphi} : \mathbb{R} \times \mathfrak{Z}^9 \to \mathfrak{Z}^9$ is given by

$$\hat{\varphi}(h, \hat{\mathbf{s}}(t_i, \cdot)) =: \hat{\mathbf{s}}(t_i + h, \cdot) = (\mathbf{u}(t_i + h, \cdot), \dot{\mathbf{u}}(t_i + h, \cdot), \ddot{\mathbf{u}}(t_i + h, \cdot)),$$

and $\hat{\mathbf{s}} = (\mathbf{u}, \dot{\mathbf{u}}, \ddot{\mathbf{u}}) \equiv (u_1, u_2, u_3, \dot{u}_1, \dot{u}_2, \dot{u}_3, \ddot{u}_1, \ddot{u}_2, \ddot{u}_3)$.

Remark 2.4. Notice that this P is the smallest value we can choose for a problem featuring three degrees of freedom and whose governing stochastic differential equation is of second-order in time. This is because for each degree of freedom, the smallest RFS that one can construct using the FSC method is one whose P is equal to 3. However, we emphasize that this issue is not particular to the FSC method. It is well recognized that the more degrees of freedom a dynamical system has, the more basis vectors needed to construct a suitable random function space for the system's state at any given time. For example, in TD-gPC-based methods, this

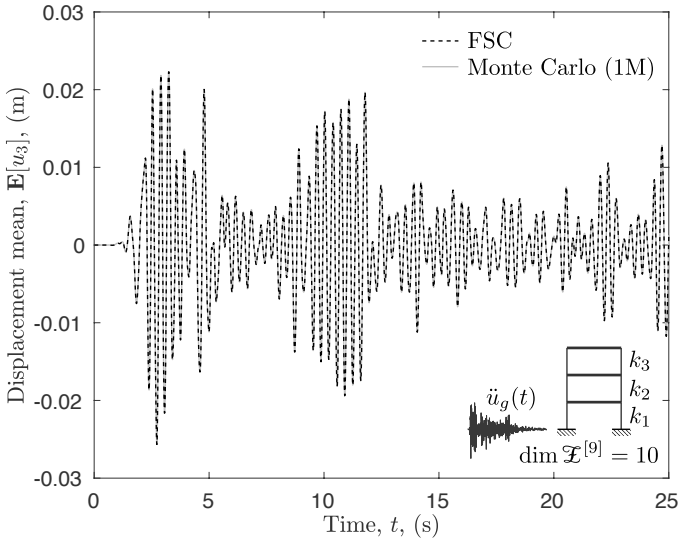
would be equivalent to perform a full tensor product between all the RFS's generated at each degree of freedom.

To integrate (2.26) numerically, we employ the RK4 method with a time-step size of $\Delta t = 0.01$ s (which is concordant with the sample frequency of the ground acceleration record). To evaluate the inner products approximately, we use 15 Gaussian quadrature points on each random axis, resulting in $15^5 = 759\,375$ quadrature points distributed across the entire random domain.

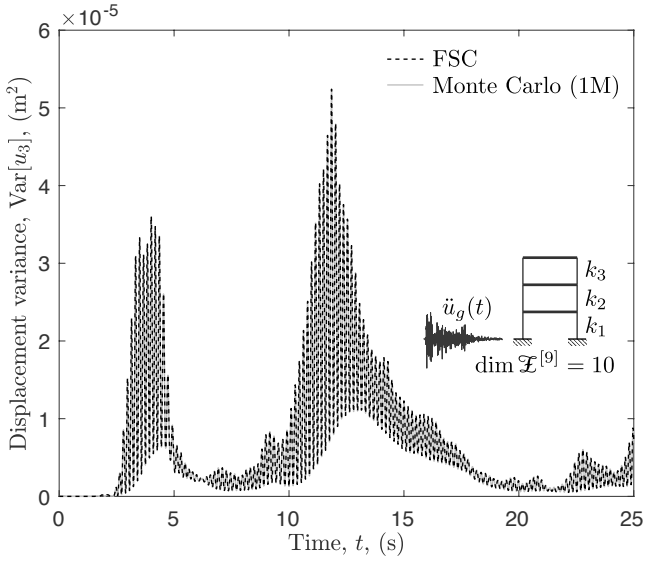
The results in Figs. 2.22 and 2.23 depict the solutions obtained from employing FSC and the standard Monte Carlo method (as described in Appendix 2.C) to quantify the uncertainties of the response. One million realizations were randomly sampled from the random domain to conduct the Monte Carlo simulation. Once again, it is apparent that FSC is able to capture the system's uncertainties with high fidelity and its solution is indistinguishable from that of the Monte Carlo method. For clarity, only the first 25 s of the solution are presented in Figs. 2.22 and 2.23, however, the conclusion drawn above applies to the last 25 s of the solution as well.

2.8. Conclusion

A novel numerical method, called the *flow-driven spectral chaos* (FSC) method, is presented for capturing uncertainties in structural dynamics using the spectral approach. The FSC method uses the concept of enriched stochastic flow maps to track the evolution of the system's state efficiently in an augmented random phase space. The method is not only computationally more efficient than the TD-gPC approach but also easy to implement, since the flow map that we use in the scheme is nothing but the time derivatives of the solution up to a specific order. Moreover, since the random basis is defined with these time derivatives, the number of basis vectors required to characterize the stochastic part of the solution space does not depend upon the dimensionality of the probability space. This remarkable property opens up the possibility of investigating systems with high-dimensional probability spaces at low computational cost—an issue that has plagued the spectral approach since the introduction of the PC method.

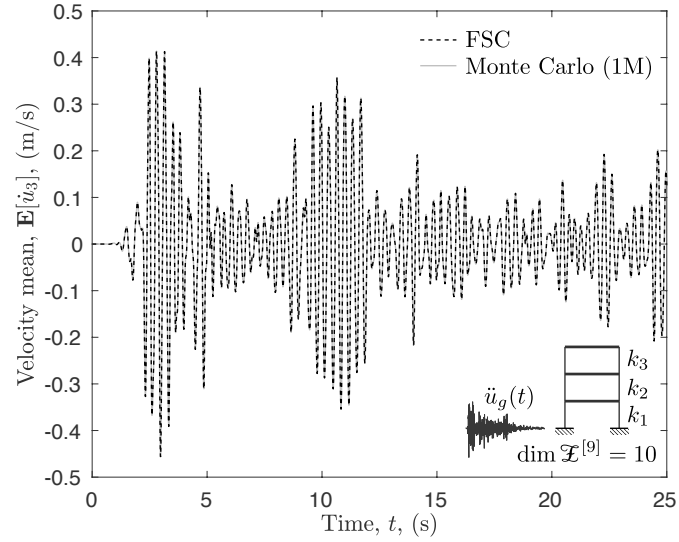


(a) Mean

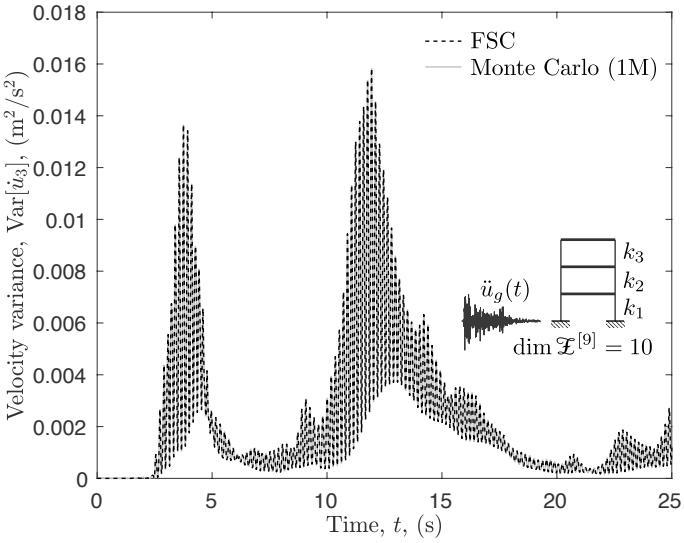


(b) Variance

Figure 2.22. Evolution of $\mathbf{E}[u_3]$ and $\mathrm{Var}[u_3]$ for the case when the p-discretization level of RFS is $\mathfrak{X}^{[9]}$ and $\mu \sim \mathrm{Beta}^{\otimes 3} \otimes \mathrm{Uniform}^{\otimes 2}$



(a) Mean



(b) Variance

Figure 2.23. Evolution of $\mathbf{E}[\dot{u}_3]$ and $\mathrm{Var}[\dot{u}_3]$ for the case when the p-discretization level of RFS is $\mathfrak{X}^{[9]}$ and $\mu \sim \mathrm{Beta}^{\otimes 3} \otimes \mathrm{Uniform}^{\otimes 2}$

The three numerical examples presented in Section 2.6 show that the FSC scheme is able to capture the response of the system with high accuracy using a small number of basis vectors and at a relatively low computational cost. The illustrative problem described in Section 2.7 also shows that the FSC method can be readily applied to real-world structures involving multiple degrees of freedom. As a result, the FSC method has the potential to be used in the context of large-scale structural engineering problems to quantify the uncertainties of long-time response with high fidelity.

2.A. Random bases for illustrative examples

Tables 2.2 and 2.3 present the non-orthogonalized version of the random bases that we use in this manuscript to solve the examples described in Section 2.6.

2.B. Undamped single-degree-of-freedom system under free vibration

The objective of this section is to provide the exact response expressions of an undamped single-degree-of-freedom system subjected to free vibration for the case when the stiffness is assumed uniformly distributed with parameters k_a and k_b. The problem is stated formally as follows.

Problem statement Consider a stochastic, undamped single-degree-of-freedom system with mass $m \in \mathbb{R}^+$, and stiffness $k : \Xi \to \mathbb{R}^+$ given by $k(\xi) = \xi \sim$ Uniform in $[k_a, k_b]$, subjected to free vibration. Note that: $k_b > k_a > 0$.

The *first problem* is to find the displacement of the system $u : \mathfrak{T} \times \Xi \to \mathbb{R}$ in $\mathcal{U}$, such that:

$$m\ddot{u} + ku = 0 \qquad \text{on } \mathfrak{T} \times \Xi \tag{2.27a}$$

$$\Big\{u(0,\cdot) = \boldsymbol{u},\, \dot{u}(0,\cdot) = \boldsymbol{v}\Big\} \qquad \text{on } \{0\} \times \Xi \tag{2.27b}$$

where $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}$, and $\dot{u} := \partial_t u$ and $\ddot{u} := \partial_t^2 u$ are the velocity and acceleration of the system, respectively.

The *second problem* is to find the expectation and variance of u, $\dot{u}$ and $\ddot{u}$ as a function of time.

Exact solution The solution of (2.27) is well-known [31, 63], and it is given by

$$u(t,\xi) = \boldsymbol{u}\cos\Big((\omega \circ k)(\xi)\, t\Big) + \frac{\boldsymbol{v}}{(\omega \circ k)(\xi)}\sin\Big((\omega \circ k)(\xi)\, t\Big), \tag{2.28}$$

where the natural circular frequency of the system, $\omega : \mathbb{R}^+ \to \mathbb{R}^+$, is defined by

$$\omega(k) = \sqrt{\frac{k}{m}}.$$

Table 2.2. Non-orthogonalized version of the random bases used in Section 2.6 (SET 1 / 2)

Single-degree-of-freedom system under **free vibration** (Section 2.6.1)*:

$$\Phi_{0.i}(\xi) := 1$$

$$\Phi_{1.i}(\xi) := \hat{\varphi}^1(M)(0, \hat{s}(t_i, \xi)) = u_{.i}(t_i, \xi) = u_{.i-1}(t_i, \xi)$$

$$\Phi_{2.i}(\xi) := \hat{\varphi}^2(M)(0, \hat{s}(t_i, \xi)) = \dot{u}_{.i}(t_i, \xi) = \dot{u}_{.i-1}(t_i, \xi)$$

$$\Phi_{3.i}(\xi) := \hat{\varphi}^3(M)(0, \hat{s}(t_i, \xi)) = \partial_t^2 u_{.i}(t_i, \xi) = -\frac{k(\xi)}{m}\, u_{.i}(t_i, \xi)$$

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

$$\Phi_{5.i}(\xi) := \hat{\varphi}^5(M)(0, \hat{s}(t_i, \xi)) = \partial_t^4 u_{.i}(t_i, \xi) = -\frac{k(\xi)}{m}\, \partial_t^2 u_{.i}(t_i, \xi)$$

$$\Phi_{6.i}(\xi) := \hat{\varphi}^6(M)(0, \hat{s}(t_i, \xi)) = \partial_t^5 u_{.i}(t_i, \xi) = -\frac{k(\xi)}{m}\, \partial_t^3 u_{.i}(t_i, \xi)$$

$$\vdots \quad \text{(until } P = M + 2 \text{ if needed)}$$

Single-degree-of-freedom system under **forced vibration** (Section 2.6.2)*:

$$\Phi_{0.i}(\xi) := 1$$

$$\Phi_{1.i}(\xi) := \hat{\varphi}^1(M)(0, \hat{s}(t_i, \xi)) = u_{.i}(t_i, \xi) = u_{.i-1}(t_i, \xi)$$

$$\Phi_{2.i}(\xi) := \hat{\varphi}^2(M)(0, \hat{s}(t_i, \xi)) = \dot{u}_{.i}(t_i, \xi) = \dot{u}_{.i-1}(t_i, \xi)$$

$$\Phi_{3.i}(\xi) := \hat{\varphi}^3(M)(0, \hat{s}(t_i, \xi)) = \partial_t^2 u_{.i}(t_i, \xi) = \frac{1}{m}\big(q(\xi)\, \sin(t_i) - k(\xi)\, u_{.i}(t_i, \xi)\big)$$

$$\Phi_{4.i}(\xi) := \hat{\varphi}^4(M)(0, \hat{s}(t_i, \xi)) = \partial_t^3 u_{.i}(t_i, \xi) = \frac{1}{m}\big(q(\xi)\, \cos(t_i) - k(\xi)\, \dot{u}_{.i}(t_i, \xi)\big)$$

$$\Phi_{5.i}(\xi) := \hat{\varphi}^5(M)(0, \hat{s}(t_i, \xi)) = \partial_t^4 u_{.i}(t_i, \xi) = \frac{1}{m}\big(-q(\xi)\, \sin(t_i) - k(\xi)\, \partial_t^2 u_{.i}(t_i, \xi)\big)$$

$$\Phi_{6.i}(\xi) := \hat{\varphi}^6(M)(0, \hat{s}(t_i, \xi)) = \partial_t^5 u_{.i}(t_i, \xi) = \frac{1}{m}\big(-q(\xi)\, \cos(t_i) - k(\xi)\, \partial_t^3 u_{.i}(t_i, \xi)\big)$$

$$\vdots \quad \text{(until } P = M + 2 \text{ if needed)}$$

*The random basis is defined over the region $\mathfrak{R}_i = \mathrm{cl}(\mathfrak{T}_i) \times \Xi$.

Table 2.3. Non-orthogonalized version of the random bases used in Section 2.6 (SET 2/2)

Nonlinear single-degree-of-freedom system under **free vibration** (Section 2.6.3)*:

$$\Phi_{0.i}(\xi) := 1$$

$$\Phi_{1.i}(\xi) := \hat{\varphi}^1(M)(0, \hat{s}(t_i, \xi)) = u_{.i}(t_i, \xi) = u_{.i-1}(t_i, \xi)$$

$$\Phi_{2.i}(\xi) := \hat{\varphi}^2(M)(0, \hat{s}(t_i, \xi)) = \dot{u}_{.i}(t_i, \xi) = \dot{u}_{.i-1}(t_i, \xi)$$

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

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

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

$$\vdots \quad \text{(until } P = M + 2 \text{ if needed)}$$

*The random basis is defined over the region $\mathfrak{R}_i = \mathrm{cl}(\mathfrak{T}_i) \times \Xi$.

Then, the following exact expressions can be derived for the expectation and variance of u, $\dot{u}$ and $\ddot{u}$.

Exact expectation The expectation of u, $\dot{u}$ and $\ddot{u}$ are given by:

$$\mathbf{E}[u](t) = \kappa(t)\left(\tau_u(t,k_b) - \tau_u(t,k_a)\right) \tag{2.29a}$$

$$\mathbf{E}[\dot{u}](t) = \kappa(t)\left(\tau_v(t,k_b) - \tau_v(t,k_a)\right) \tag{2.29b}$$

$$\mathbf{E}[\ddot{u}](t) = \kappa(t)\left(\tau_a(t,k_b) - \tau_a(t,k_a)\right), \tag{2.29c}$$

where $\kappa : \mathfrak{T} \to \mathbb{R}^+$ is given by $\kappa(t) = 2m/((k_b - k_a)t^2)$, and $\tau_u, \tau_v, \tau_a : \mathfrak{T} \times \mathbb{R}^+ \to \mathbb{R}$ are defined in Table 2.4.

A closer look at the above expressions indicates that in the long term the absolute mean of the response is dominated by $\kappa = \kappa(t)$ which is a function that tends to zero as time goes to infinity. For this reason,

$$\mathbf{E}[u], \mathbf{E}[\dot{u}], \mathbf{E}[\ddot{u}] \to 0 \quad \text{as} \quad t \to \infty. \tag{2.30}$$

Exact variance The variance of u, $\dot{u}$ and $\ddot{u}$ are given by:

$$\mathrm{Var}[u](t) = \kappa(t)\Big(\varrho_u(t,k_b) - \varrho_u(t,k_a)\Big) - \mathbf{E}[u]^2(t) \tag{2.31a}$$

$$\mathrm{Var}[\dot{u}](t) = \kappa(t)\Big(\varrho_v(t,k_b) - \varrho_v(t,k_a)\Big) - \mathbf{E}[\dot{u}]^2(t) \tag{2.31b}$$

$$\mathrm{Var}[\ddot{u}](t) = \kappa(t)\Big(\varrho_a(t,k_b) - \varrho_a(t,k_a)\Big) - \mathbf{E}[\ddot{u}]^2(t), \tag{2.31c}$$

where $\varrho_u, \varrho_v, \varrho_a : \mathfrak{T} \times \mathbb{R}^+ \to \mathbb{R}$ are defined in Table 2.5. In the expression for ϱ_u, the function $\mathrm{Ci} : \mathbb{R}^+ \to \mathbb{R}$ is the cosine integral given by

$$\mathrm{Ci}(x) = -\int_x^\infty \frac{\cos y}{y}\,\mathrm{d}y.$$

Table 2.4. Definition of τ-functions for a single-degree-of-freedom system subjected to free vibration with $k \sim$ Uniform in $[k_a, k_b]$

$\tau_u(t,k) = \{\omega(k)\,t\sin(\omega(k)\,t) + \cos(\omega(k)\,t)\}\,u - \cos(\omega(k)\,t)\,v\,t$
$\tau_v(t,k) = -\{2\,\omega(k)\,t\sin(\omega(k)\,t) + (2-\omega^2(k)\,t^2)\cos(\omega(k)\,t)\}\,u t^{-1} + \{\omega(k)\,t\sin(\omega(k)\,t) + \cos(\omega(k)\,t)\}\,v$
$\tau_a(t,k) = -\{(\omega^3(k)\,t^3 - 6\,\omega(k)\,t)\sin(\omega(k)\,t) + 3(\omega^2(k)\,t^2-2)\cos(\omega(k)\,t)\}\,u t^{-2} - \{2\,\omega(k)\,t\sin(\omega(k)\,t) + (2-\omega^2(k)\,t^2)\cos(\omega(k)\,t)\}\,v t^{-1}$

Table 2.5. Definition of ϱ-functions for a single-degree-of-freedom system subjected to free vibration with $k \sim$ Uniform in $[k_a, k_b]$

$\varrho_u(t,k) = -\frac{1}{4}\{\sin^2(\omega(k)\,t) - \omega(k)\,t\sin(2\,\omega(k)\,t) - \omega^2(k)\,t^2\}\,u^2 - \frac{1}{2}\cos(2\,\omega(k)\,t)\,u v t + \frac{1}{4}\{\ln(k) - 2\,\mathrm{Ci}(2\,\omega(k)\,t)\}\,v^2 t^2$
$\varrho_v(t,k) = \frac{1}{16}\{2\,(3\,\omega(k)\,t - 2\,\omega^3(k)\,t^3)\sin(2\,\omega(k)\,t) + 3\,(1-2\,\omega^2(k)\,t^2)\cos(2\,\omega(k)\,t) + 2\,\omega^4(k)\,t^4\}\,u^2 t^{-2} - \frac{1}{4}\{2\,\omega(k)\,t\sin(2\,\omega(k)\,t) + (1-2\,\omega^2(k)\,t^2)\cos(2\,\omega(k)\,t)\}\,u v t^{-1} - \frac{1}{4}\{\sin^2(\omega(k)\,t) - \omega(k)\,t\sin(2\,\omega(k)\,t) - \omega^2(k)\,t^2\}\,v^2$
$\varrho_a(t,k) = \frac{1}{48}\{6\,(2\,\omega^5(k)\,t^5 - 10\,\omega^3(k)\,t^3 + 15\,\omega(k)\,t)\sin(2\,\omega(k)\,t) + 15\,(2\,\omega^4(k)\,t^4 - 6\,\omega^2(k)\,t^2 + 3)\cos(2\,\omega(k)\,t) + 4\,\omega^6(k)\,t^6\}\,u^2 t^{-4} + \frac{1}{8}\{4\,(2\,\omega^3(k)\,t^3 - 3\,\omega(k)\,t)\sin(2\,\omega(k)\,t) - 2\,(2\,\omega^4(k)\,t^4 - 6\,\omega^2(k)\,t^2 + 3)\cos(2\,\omega(k)\,t)\}\,u v t^{-3} - \frac{1}{16}\{2\,(2\,\omega^3(k)\,t^3 - 3\,\omega(k)\,t)\sin(2\,\omega(k)\,t) + 3\,(2\,\omega^2(k)\,t^2 - 1)\cos(2\,\omega(k)\,t) - 2\,\omega^4(k)\,t^4\}\,v^2 t^{-2}$

We see that not only the variance of the response is bounded for all $t \in \mathfrak{T}$, but also as t goes to infinity:

$$\lim_{t\to\infty} \mathrm{Var}[u](t) = \frac{1}{2}\boldsymbol{u}^2 + \frac{1}{2}\ln\left(\frac{k_b}{k_a}\right)\left(\frac{m}{k_b - k_a}\right)\boldsymbol{v}^2 \tag{2.32a}$$

$$\lim_{t\to\infty} \mathrm{Var}[\dot{u}](t) = \frac{1}{4}\left(\frac{k_a + k_b}{m}\right)\boldsymbol{u}^2 + \frac{1}{2}\boldsymbol{v}^2 \tag{2.32b}$$

$$\lim_{t\to\infty} \mathrm{Var}[\ddot{u}](t) = \frac{1}{6}\left(\frac{k_a^2 + k_a k_b + k_b^2}{m^2}\right)\boldsymbol{u}^2 + \frac{1}{4}\left(\frac{k_a + k_b}{m}\right)\boldsymbol{v}^2. \tag{2.32c}$$

Therefore, the underlying process u is naturally of second-order.

2.C. Overview of standard Monte Carlo method

The Monte Carlo method is the most popular numerical technique used in stochastic modeling to quantify the effects of input uncertainty on system's outputs. It is basically a 'brute-force' method of attack that typically involves sampling a large number of realizations from the random space to estimate the statistics of the output. It is well-known that when N realizations are considered, the mean converges asymptotically as the square root of N^{-1}, and thus, it is remarkably independent of the dimensionality of the random space [3]. In this paper we use *standard Monte Carlo* to validate the FSC method in Sections 2.6.3 and 2.7.

The general procedure for conducting a standard Monte Carlo simulation is simple. Consider the stochastic system given by (2.1*):

$$y(t,\xi) = \boldsymbol{\mathcal{M}}[u][x](t,\xi) \quad \text{subject to initial condition} \quad \boldsymbol{\mathcal{I}}[u](\xi). \tag{2.1**}$$

Then:

1. Generate N realizations of the d-tuple random variable ξ in order to obtain the random set $\{\xi_i\}_{i=1}^N$. These N realizations are based on randomly sampling N points from the random domain Ξ according to the cumulative distribution function $F : \Xi \to [0,1]$ given by

$$F(\xi) = \prod_{j=1}^{d} \mu^j\big((-\infty, \xi^j]\big), \quad \text{or equivalently,} \quad F(\xi) = \int_{-\infty}^{\xi^1} \cdots \int_{-\infty}^{\xi^d} \mathrm{d}\mu^1 \cdots \mathrm{d}\mu^d. \tag{2.33}$$

   This way we can also obtain the input set $\{x(t,\xi_i)\}_{i=1}^N$ for reference purposes.

2. Solve (2.1**) for each random point ξ_i to obtain the output set $\{y(t,\xi_i)\}_{i=1}^N$.

3. Aggregate results to estimate the statistics of output y as a function of time. For instance, if we let z denote the k-th component of $y = (y_1, \ldots, y_s)$, then its statistical mean $\mathbf{E}^*[z] : \mathfrak{T} \to \mathbb{R}$ and statistical variance $\mathrm{Var}^*[z] : \mathfrak{T} \to \mathbb{R}_0^+$ are given by:

$$\mathbf{E}^*[z](t) = \frac{1}{N}\sum_{i=1}^{N} z(t,\xi_i) \quad \text{and} \quad \mathrm{Var}^*[z](t) = \frac{1}{N-1}\sum_{i=1}^{N}\Big(z(t,\xi_i) - \mathbf{E}^*[z](t)\Big)^2. \tag{2.34}$$

3. FLOW-DRIVEN SPECTRAL CHAOS (FSC) METHOD FOR SIMULATING LONG-TIME DYNAMICS OF ARBITRARY-ORDER NON-LINEAR STOCHASTIC DYNAMICAL SYSTEMS

3.1. 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 [2–4], collocation-based methods [46, 50, 65–67], perturbation-based methods [5–8], operator-based methods [9–12], and spectral-based methods [13–15]. 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 [32] and then further extended in [68, 69], 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 [70] 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 [71]. 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. [72] demonstrated that, under Wiener's framework of Hermite functionals, the convergence rate of systems subjected to Gaussian input is exponential but substantially slower otherwise [33, 73].

Nonetheless, steady progress was made between the 1950s and 1980s towards generalizing Wiener's ideas for systems with non-Gaussian inputs (e.g. [74–78]).

In the early 1990s, Ghanem and Spanos [24, 42] 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. [79–82]).

In 2002, Xiu and Karniadakis [33] 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. [83–86]). 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. [44, 46, 48, 50, 52, 87]) have been proved to be capable of solving numerous problems in engineering and sciences (e.g. [45, 47, 49, 51, 88, 89]).

The dynamically orthogonal PC (DO-PC) is another approach used for uncertainty quantification. It was formulated by Sapsis and Lermusiaux [53] 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 [58, 59] and the gPC method. Since its inception, the DO-PC has undergone further modifications and extensions to broaden its range of applications (e.g. [54–57]). An error analysis for the DO-PC method can be found in [90].

In 2010, the time-dependent gPC (TD-gPC) method was proposed by Gerritsma et al. [34] 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 [35] 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 [35] 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. [36] 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 [37] 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 3.9, 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 3.2 introduces the setting and notation used in this manuscript, and then a quick overview of the standard gPC method is provided. Section 3.3 discusses the Gram-Schmidt process for random function spaces and also outlines a *new theorem* that has been developed for orthogonalizing a sequence of independent random functions. This theorem is then utilized in the FSC scheme (Section 3.6) to transfer the probability information of the system's state *exactly* at the current time of the simulation. Section 3.4 reviews the concept of stochastic flow map, followed by the definition of *enriched* stochastic flow map in Section 3.5. In Section 3.6 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 3.7, followed by a discussion of the numerical results in Section 3.8. In Section 3.9 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 3.A and 3.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 3.9 using the spectral approach. Finally, Appendix 3.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.

3.2. Setting and notation

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

Definition 3.1 (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 3.1. 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 3.3 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 3.2 (Random space)**.** Let $(\Omega, \mathbf{\Omega}, \lambda)$ be a (complete) *probability space*, where Ω is the sample space, $\mathbf{\Omega} \subset 2^{\Omega}$ is the σ-algebra on Ω (aka the collection of events), and $\lambda : \mathbf{\Omega} \to [0, 1]$ is the probability measure on $\mathbf{\Omega}$. Let $\xi : (\Omega, \mathbf{\Omega}) \to (\mathbb{R}^d, \mathcal{B}_{\mathbb{R}^d})$ be a measurable function (aka random variable) given by $\xi = \xi(\omega)$, with $\mathcal{B}_{\mathbb{R}^d}$ denoting the Borel σ-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 σ-algebra on Ξ, and $\mu : \mathbf{\Xi} \to [0, 1]$ is the probability measure on $\mathbf{\Xi}$ defined by the pushforward of λ by ξ, that is $\mu = \xi_*(\lambda)$. Here d symbolizes the dimensionality of the random space.

Remark 3.2. In addition to the standard definition of a 'probability space', we define a 'random space' in Definition 3.2 to address cases where the probability space may be abstract. The random variable ξ 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 3.3 (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 $f : (\mathfrak{T}, \mathcal{O}) \to (\mathbb{R}, \mathcal{O}_{\mathbb{R}})$ that have continuous first n derivatives on $(\mathfrak{T}, \mathcal{O})$.

Remark 3.3. The temporal function space is thus defined to indicate the level of differentiability that some temporal functions need to possess in Sections 3.4 and 3.5—especially those concerning with the state of the dynamical system under consideration.

Definition 3.4 (Random function space)**.** Let $\mathfrak{L} = (L^2(\Xi, \boldsymbol{\Xi}, \mu; \mathbb{R}), \langle \cdot, \cdot \rangle)$ be a Lebesgue square-integrable space equipped with its standard inner product $\langle \cdot, \cdot \rangle : L^2(\Xi, \boldsymbol{\Xi}, \mu; \mathbb{R}) \times L^2(\Xi, \boldsymbol{\Xi}, \mu; \mathbb{R}) \to \mathbb{R}$ given by $\langle f, g \rangle = \int fg \, \mathrm{d}\mu$. This *random function space* (aka RFS in this manuscript) is the space of all (equivalence classes of) measurable functions $f : (\Xi, \boldsymbol{\Xi}) \to (\mathbb{R}, \mathcal{B}_{\mathbb{R}})$ that are square-integrable with respect to μ. This space is known to form a Hilbert space because it is complete under the metric induced by the inner product. In addition, let $\{\Psi_j : (\Xi, \boldsymbol{\Xi}) \to (\mathbb{R}, \mathcal{B}_{\mathbb{R}})\}_{j=0}^{\infty}$ be a complete orthogonal basis in $\mathfrak{L}$, such that $\Psi_0(\xi) = 1$ for all $\xi \in \Xi$.

Remark 3.4. 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 $\mathfrak{L}$ 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 $\mathfrak{L}$. 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. [25–28]. In Section 3.9, 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 3.4 it follows that any function $f \in \mathfrak{L}$ 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 $\mathfrak{Z}$. 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:

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

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

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

Here $[\,\cdot\,,\cdot\,] : \mathfrak{Z}' \times \mathfrak{Z} \to \mathbb{R}$ represents the dual pairing between $\mathfrak{Z}$ and $\mathfrak{Z}'$ 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 [23], which means that the map $\Psi^i \mapsto \Psi_i / \langle \Psi_i, \Psi_i \rangle$ is an isometric isomorphism between $\mathfrak{Z}'$ and $\mathfrak{Z}$.

Definition 3.5 (Solution space and root space)**.** Let $\mathscr{U} = \mathscr{T}(n) \otimes \mathfrak{Z}$ and $\mathscr{V} = \mathscr{T}(0) \otimes \mathfrak{Z}$ be the *solution space* and the *root space* of the system, respectively. Then, as a result of these definitions, we have: $\mathfrak{Z} \subset \mathscr{U} \subset \mathscr{V}$.

Remark 3.5. 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 $\xi = (\xi^1, \ldots, \xi^d)$ are mutually independent and that the random domain Ξ is a hypercube of d dimensions obtained by performing a d-fold Cartesian product of intervals $\bar{\Xi}_i := \xi^i(\Omega)$.

Letting $\mu^i(\mathrm{d}\xi^i) =: \mathrm{d}\mu^i$ denote the probability measure of $\mathrm{d}\xi^i$ around $\xi^i \in \bar{\Xi}_i$, one can then define the measure in $\mathfrak{X}$ by

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

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

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

$$\mathcal{L}[u] = f \qquad \text{on } \mathfrak{T} \times \Xi \tag{3.1a}$$

$$\left\{\mathcal{B}_k[u](0,\cdot) = b_k\right\}_{k=1}^{n} \qquad \text{on } \{0\} \times \Xi, \tag{3.1b}$$

where $\mathcal{L} : \mathcal{U} \to \mathcal{V}$ is a partial differential operator of order $(n, 0)$, $\mathcal{B}_k[\,\cdot\,](0,\cdot) : \mathcal{U} \to \mathfrak{X}$ is a partial differential operator of order $(n-1, 0)$ that upon differentiation evaluates the resulting function at $t = 0$, $f : \mathfrak{T} \times \Xi \to \mathbb{R}$ is a function in $\mathcal{V}$ given by $f = f(t, \xi)$, and $b_k : \Xi \to \mathbb{R}$ is a function in $\mathfrak{X}$ 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 $b_{10}, b_{11}, b_{20}, b_{21} \in \mathfrak{X}$ 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 (3.1), 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 $u(t,\cdot) \in \mathfrak{L}$ for all $t \in \mathfrak{T}$.

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

$$u(t,\xi) = \sum_{j=0}^{\infty} u^j(t)\, \Psi_j(\xi), \tag{3.2}$$

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 [14,15], 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 $\mathfrak{L}$, the expansion given by (3.2) will lead to exponential convergence to the solution, since $\{\Psi_j\}_{j=0}^{\infty}$ is an orthogonal basis with respect to the probability measure μ in $\mathfrak{L}$. 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 [61].

A system governed by (3.1) can also be expressed in modeling notation as

$$y = \boldsymbol{\mathcal{M}}[u][x] \quad \text{subject to initial condition} \quad \boldsymbol{\mathcal{I}}[u], \tag{3.1*}$$

where $\boldsymbol{\mathcal{M}}[u] : \mathcal{V}^r \to \mathcal{V}^s$ represents the mathematical model of the system defined by (3.1a), $x = (x_1, \ldots, x_r) : \mathfrak{T} \times \Xi \to \mathbb{R}^r$ is the r-tuple input of $\boldsymbol{\mathcal{M}}[u]$, and $y = (y_1, \ldots, y_s) : \mathfrak{T} \times \Xi \to \mathbb{R}^s$ is the s-tuple output of $\boldsymbol{\mathcal{M}}[u]$ (aka the s-tuple observable in physics or the s-tuple response in engineering). In addition, $\boldsymbol{\mathcal{I}}[u]$ represents the initial condition for $\boldsymbol{\mathcal{M}}[u]$ which is given by (3.1b). 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 (3.1) 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 $\mathfrak{L}$ as follows. Let $\mathfrak{L}^{[P]} = \operatorname{span}\{\Psi_j\}_{j=0}^{P}$ be a finite subspace of $\mathfrak{L}$ with $P+1 \in \mathbb{N}_1$ denoting the dimensionality of the subspace, and let $u^{[P]}(t,\cdot)$ be an element of $\mathfrak{L}^{[P]}$.

Then, from (3.2) it follows that:

$$u(t,\xi) \approx u^{[P]}(t,\xi) = \sum_{j=0}^{P} u^j(t)\,\Psi_j(\xi), \tag{3.3}$$

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 (3.3) can be determined as

$$P+1 = \binom{d+p}{p} = \frac{(d+p)!}{d!p!}. \tag{3.4}$$

This expression shows that the total number of terms used in (3.3) 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 [15]. 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 3.6. An orthogonal basis in $\mathfrak{Z}^{[P]}$ can be constructed as products of univariate orthogonal polynomials in the following way. Let $\{\Psi_j^{(i)} : \bar{\Xi}_i \to \mathbb{R}\}_{j=0}^{\infty}$ be an orthogonal basis with respect to μ^i, where $i \in \{1,2,\ldots,d\}$. 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 $\xi^i \in \bar{\Xi}_i$. Then, one defines:

$$\Psi_j \equiv \Psi_{\pi(k)} := \bigotimes_{i=1}^{d} \Psi_{k_i}^{(i)} = \Psi_{k_1}^{(1)} \otimes \cdots \otimes \Psi_{k_d}^{(d)}, \tag{3.5}$$

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

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

In these expressions, $\pi : \mathbb{N}_0^d \to \mathbb{N}_0$ 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 (3.5) to make p (which is always taken less than $\max |k|$) be the maximal order polynomial in $\{\Psi_j\}_{j=0}^P$.

Remark 3.7. In Section 3.6 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 3.6, but by recurring instead to the time derivatives of the solution itself. This is in contrast to the standard TD-gPC method [34], 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 (3.3) will simply be written hereafter as

$$u(t, \xi) = u^j(t)\, \Psi_j(\xi), \tag{3.6}$$

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

Substituting (3.6) into (3.1) gives

$$\mathcal{L}[u^j \Psi_j] = f \qquad \text{on } \mathfrak{T} \times \Xi \tag{3.7a}$$

$$\Big\{\mathcal{B}_k[u^j \Psi_j](0, \cdot\,) = b_k\Big\}_{k=1}^n \qquad \text{on } \{0\} \times \Xi. \tag{3.7b}$$

Projecting (3.7) onto $\mathfrak{Z}^{[P]}$ 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:

$$\Psi^i\Big[\mathcal{L}[u^j \Psi_j]\Big] = \Psi^i[f] \qquad \text{on } \mathfrak{T} \tag{3.8a}$$

$$\Big\{\Psi^i\Big[\mathcal{B}_k[u^j \Psi_j](0, \cdot\,)\Big] = \Psi^i[b_k]\Big\}_{k=1}^n \qquad \text{on } \{0\} \tag{3.8b}$$

with $i, j \in \{0, 1, \ldots, P\}$. This is the so-called *orthogonal projection* of (3.7) onto $\mathfrak{Z}^{[P]}$, and it ensures that the mean-square error resulting from the finite representation of u using (3.3) is orthogonal to $\mathfrak{Z}^{[P]}$ [24].

A closer look at (3.8) 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 (3.8) does not depend on the tuple (t, ξ) but only on t at this stage of the analysis.

Discretization of temporal function space Because (3.8) 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, $\mathbf{E} : \mathfrak{X} \to \mathbb{R}$, is a linear map that outputs the expected value of a real-valued random variable, and it is given by:

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

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

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

These two maps can be used as the building block to construct other maps, such as the variance of f which is defined as $\mathrm{Var}[f] = \mathrm{Cov}[f, f]$. 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 $y = \boldsymbol{\mathcal{M}}[u][x]$ (from (3.1*)). If $z \in \mathscr{V}$, then it can be expanded with a polynomial chaos similar to the one set forth in (3.3) 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 (3.3), 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, $\mathbf{E}[z] : \mathfrak{T} \to \mathbb{R}$, is easy to compute and it is given by the first random mode of z:

$$\mathbf{E}[z](t) := \int z(t,\cdot)\,\mathrm{d}\mu = z^j(t)\int \Psi_j\,\mathrm{d}\mu = z^j(t)\,\langle \Psi_j, \Psi_0\rangle = z^0(t). \tag{3.9}$$

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

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

and thus, upon further simplification we get

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

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

$$\mathrm{Var}[z](t) := \mathrm{Cov}[z,z](t,t) = \sum_{j=1}^{P} \Upsilon_{jj}\,z^j(t)\,z^j(t). \tag{3.10}$$

Probability distributions In this paper, we employ four different probability distributions to characterize the stochasticity in the systems defined in Sections 3.7 and 3.9, 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, $f : \Xi \to \mathbb{R}_0^+$, given by:

$$\text{Uniform} \sim f(\xi) = \frac{1}{b-a} \quad \text{on } \Xi = [a, b],$$

$$\text{Beta}(\alpha, \beta) \sim f(\xi) = \frac{(\xi - a)^{\alpha-1} (b - \xi)^{\beta-1}}{(b-a)^{\alpha+\beta-1} \, \text{B}(\alpha, \beta)} \quad \text{on } \Xi = [a, b]$$

$$\text{Gamma}(\alpha, \beta) \sim f(\xi) = \frac{\beta^\alpha}{\Gamma(\alpha)} (\xi - a)^{\alpha-1} \exp(-\beta \, (\xi - a)) \quad \text{on } \Xi = [a, \infty), \quad \text{and}$$

$$\text{Normal}(\mu, \sigma^2) \sim f(\xi) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2} \left(\frac{\xi - \mu}{\sigma} \right)^2 \right] \quad \text{on } \Xi = \mathbb{R}.$$

Numerical integration The numerical integration of an inner product can be carried out at least in two different ways: using *grid-based integration* [25–27] or *Monte Carlo-based integration* [28]. 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 [91] 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 3.7, 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:

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

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 μ.

Remark 3.8. We note that when fg is a sufficiently smooth integrand, we have:

$$\mathcal{Q}^{[Q]}[fg] \to \int fg \, \mathrm{d}\mu \quad \text{as} \quad Q \to \infty,$$

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 3.9. In this case we choose Monte Carlo integration because the dimensionality of the random space is up to 10.

3.3. The Gram-Schmidt process for random function spaces

Suppose that we have a non-orthogonal basis in $\mathfrak{L}$ 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:

$$\Psi_j := \Phi_j - \sum_{k=0}^{j-1} \frac{\langle \Phi_j, \Psi_k \rangle}{\langle \Psi_k, \Psi_k \rangle} \Psi_k \quad \forall j \in \mathbb{N}_0. \tag{3.11}$$

Recall that in Section 3.2 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.[1] 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 3.1.

Theorem 3.1. *Let $\mathfrak{X}$ be a random function space, and let $\{\Phi_j\}_{j=1}^{\infty}$ be an ordered set of linearly independent functions in $\mathfrak{X}$ such that the constant functions are not in the set. Then, $\{\Psi_j\}_{j=0}^{\infty}$ is an orthogonal basis in $\mathfrak{X}$ given by:*

$$\Psi_j := \Phi_j - \mathbf{E}[\Phi_j]\,\Psi_0 - \sum_{k=1}^{j-1} \frac{\det \triangle_k(j)}{\det \square_k} \Psi_k \quad \textit{with} \quad \Psi_0 \equiv 1, \tag{3.12}$$

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

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

and $\triangle_k : \{k+1, k+2, \ldots\} \to \mathscr{M}(k \times k, \mathbb{R})$ is a map defined by

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

with $\triangle_1(j) = \mathrm{Cov}[\Phi_j, \Phi_1]$ and

$$\triangle_2(j) = \begin{bmatrix} \mathrm{Cov}[\Phi_1, \Phi_1] & \mathrm{Cov}[\Phi_1, \Phi_2] \\ \mathrm{Cov}[\Phi_j, \Phi_1] & \mathrm{Cov}[\Phi_j, \Phi_2] \end{bmatrix}.$$

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

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

[1] Please see Appendix 3.C for a time-complexity analysis for Theorem 3.1 and the traditional Gram-Schmidt process.

(3.12) 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 (3.11) yields

$$\Psi_1 = \Phi_1 - \frac{\langle \Phi_1, \Psi_0 \rangle}{\langle \Psi_0, \Psi_0 \rangle} \Psi_0 = \Phi_1 - \mathbf{E}[\Phi_1]\, \Psi_0, \tag{3.13}$$

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

❑ **When $j = 2$.** From (3.11) we get

$$\Psi_2 = \Phi_2 - \frac{\langle \Phi_2, \Psi_0 \rangle}{\langle \Psi_0, \Psi_0 \rangle} \Psi_0 - \frac{\langle \Phi_2, \Psi_1 \rangle}{\langle \Psi_1, \Psi_1 \rangle} \Psi_1. \tag{3.14}$$

Replacing (3.13) into (3.14) gives

$$\Psi_2 = \Phi_2 - \mathbf{E}[\Phi_2]\, \Psi_0 - \frac{\mathrm{Cov}[\Phi_1, \Phi_2]}{\mathrm{Cov}[\Phi_1, \Phi_1]} \Psi_1 = \Phi_2 - \mathbf{E}[\Phi_2]\, \Psi_0 - \frac{\det \triangle_1(2)}{\det \square_1} \Psi_1,$$

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

❑ **When $j = 3$.** In a similar fashion, (3.11) yields

$$\begin{aligned} \Psi_3 = \Phi_3 - \mathbf{E}[\Phi_3]\, \Psi_0 &- \frac{\mathrm{Cov}[\Phi_1, \Phi_3]}{\mathrm{Cov}[\Phi_1, \Phi_1]} \Psi_1 \\ &- \frac{\mathrm{Cov}[\Phi_1, \Phi_1]\, \mathrm{Cov}[\Phi_3, \Phi_2] - \mathrm{Cov}[\Phi_1, \Phi_2]\, \mathrm{Cov}[\Phi_3, \Phi_1]}{\mathrm{Cov}[\Phi_1, \Phi_1]\, \mathrm{Cov}[\Phi_2, \Phi_2] - \mathrm{Cov}[\Phi_1, \Phi_2]\, \mathrm{Cov}[\Phi_2, \Phi_1]} \Psi_2 \\ &= \Phi_3 - \mathbf{E}[\Phi_3]\, \Psi_0 - \frac{\det \triangle_1(3)}{\det \square_1} \Psi_1 - \frac{\det \triangle_2(3)}{\det \square_2} \Psi_2. \end{aligned}$$

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 $k = 2$:

$$\langle \Psi_2, \Psi_2 \rangle = \mathbf{E}[(\Psi_2)^2] = \mathbf{E}\left[\left(\Phi_2 - \mathbf{E}[\Phi_2]\, \Psi_0 - \frac{\mathrm{Cov}[\Phi_1, \Phi_2]}{\mathrm{Cov}[\Phi_1, \Phi_1]} \Psi_1 \right)^2 \right]. \tag{3.15}$$

Substituting (3.13) into (3.15) yields

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

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

$$\langle \Psi_k, \Psi_k \rangle = \frac{\det \square_k}{\det \square_{k-1}} \quad \forall k \in \mathbb{N}_1, \tag{3.16}$$

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:

$$\langle \Phi_j, \Psi_2 \rangle = \left\langle \Phi_j, \Phi_2 - \mathbf{E}[\Phi_2]\,\Psi_0 - \frac{\mathrm{Cov}[\Phi_1, \Phi_2]}{\mathrm{Cov}[\Phi_1, \Phi_1]}\Psi_1 \right\rangle. \tag{3.17}$$

Replacing (3.13) into (3.17) gives

$$\langle \Phi_j, \Psi_2 \rangle = \frac{\mathrm{Cov}[\Phi_1, \Phi_1]\,\mathrm{Cov}[\Phi_j, \Phi_2] - \mathrm{Cov}[\Phi_1, \Phi_2]\,\mathrm{Cov}[\Phi_j, \Phi_1]}{\mathrm{Cov}[\Phi_1, \Phi_1]} = \frac{\det \triangle_2(j)}{\det \square_1}.$$

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

$$\langle \Phi_j, \Psi_k \rangle = \frac{\det \triangle_k(j)}{\det \square_{k-1}} \quad \forall k \in \mathbb{N}_1, \tag{3.18}$$

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

Therefore, substituting (3.16) and (3.18) into (3.11) yields expression (3.12). □

This theorem will be implemented in Section 3.6 to transfer the probability information exactly at the current time of the simulation.

Example 3.1. Let us consider a one-dimensional random space with domain $\Xi = [-1, 1]$ and measure μ given by $\mathrm{d}\mu(\xi) = \frac{3}{2}\xi^2\,\mathrm{d}\xi$, and suppose that we are interested in finding an orthogonal basis in one of the infinitely many $\mathfrak{L}^{[5]}$ spaces that we can construct by applying Theorem 3.1 over a set of 5 monomials in the variable ξ.

To this end, let $\{\Phi_j(\xi) = \xi^j\}_{j=1}^5$ be this set. Then, its expectation vector and covariance matrix are:

$$\Big[\mathbf{E}[\Phi_j]\Big] = \begin{bmatrix} 0 \\ \frac{3}{5} \\ 0 \\ \frac{3}{7} \\ 0 \end{bmatrix},$$

and

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

From (3.12) it follows that an orthogonal basis in the chosen $\mathscr{Z}^{[5]}$ space is given by:

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

By way of illustration, the determinant ratios that appear in (3.12) were computed with expressions such as this:

$$\frac{\det \triangle_2(4)}{\det \square_2} = \frac{\begin{vmatrix} \mathrm{Cov}[\Phi_1, \Phi_1] & \mathrm{Cov}[\Phi_1, \Phi_2] \\ \mathrm{Cov}[\Phi_4, \Phi_1] & \mathrm{Cov}[\Phi_4, \Phi_2] \end{vmatrix}}{\begin{vmatrix} \mathrm{Cov}[\Phi_1, \Phi_1] & \mathrm{Cov}[\Phi_1, \Phi_2] \\ \mathrm{Cov}[\Phi_2, \Phi_1] & \mathrm{Cov}[\Phi_2, \Phi_2] \end{vmatrix}} = \frac{\begin{vmatrix} \frac{3}{5} & 0 \\ 0 & \frac{8}{105} \end{vmatrix}}{\begin{vmatrix} \frac{3}{5} & 0 \\ 0 & \frac{12}{175} \end{vmatrix}} = \frac{10}{9}.$$

3.4. Stochastic flow map

Provided sufficient regularity, a stochastic system governed by (3.1) can be expressed in explicit form as

$$\partial_t^n u(t,\xi) = \pounds(t,\xi,s(t,\xi)) \qquad \text{on } \mathfrak{T}\times\Xi \tag{3.19a}$$

$$\left\{\partial_t^{k-1} u(0,\xi) = c_k(\xi)\right\}_{k=1}^{n} \qquad \text{on } \{0\}\times\Xi, \tag{3.19b}$$

where $s = (u, \partial_t u, \ldots, \partial_t^{n-1} u) \in \prod_{j=1}^{n} \mathcal{T}(n-j+1) \otimes \mathfrak{X}$ is the configuration state of the system over $\mathfrak{T}\times\Xi$, $\pounds : \mathfrak{T}\times\Xi\times\mathbb{R}^n \to \mathbb{R}$ is a noisy, non-autonomous function (which can also be regarded as a function in $\mathcal{V}$) concordant with (3.1a), and $c_k : \Xi \to \mathbb{R}$ is a function in $\mathfrak{X}$ concordant with (3.1b).

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 $u \in \mathcal{T}(n+M-1) \otimes \mathfrak{X} \subset \mathcal{U}$, 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 (3.19) for a first-order ODE ($n = 1$) yields

$$\partial_t u(t,\xi) = \pounds(t,\xi,u(t,\xi)) \qquad \text{on } \mathfrak{T}\times\Xi \tag{3.20a}$$

$$u(0,\xi) = c(\xi) \qquad \text{on } \{0\}\times\Xi. \tag{3.20b}$$

Differentiating (3.20a) with respect to time three times gives:

$$\partial_t^2 u := \mathrm{D}_t \mathscr{f} = \partial_t \mathscr{f} + \partial_u \mathscr{f}\, \partial_t u \tag{3.21a}$$

$$\partial_t^3 u := \mathrm{D}_t^2 \mathscr{f} = \partial_t^2 \mathscr{f} + 2\, \partial_{tu}^2 \mathscr{f}\, \partial_t u + \partial_u \mathscr{f}\, \partial_t^2 u + \mathscr{g}_3 \tag{3.21b}$$

$$\partial_t^4 u := \mathrm{D}_t^3 \mathscr{f} = \partial_t^3 \mathscr{f} + 3\, \partial_{ttu}^3 \mathscr{f}\, \partial_t u + 3\, \partial_{tu}^2 \mathscr{f}\, \partial_t^2 u + \partial_u \mathscr{f}\, \partial_t^3 u + \mathscr{g}_4, \tag{3.21c}$$

where $\mathscr{g}_3 = \partial_u^2 \mathscr{f} \left(\partial_t u\right)^2$, and $\mathscr{g}_4 = 3\, \partial_{tuu}^3 \mathscr{f} \left(\partial_t u\right)^2 + \partial_u^3 \mathscr{f} \left(\partial_t u\right)^3 + 3\, \partial_u^2 \mathscr{f}\, \partial_t^2 u\, \partial_t u$.

A stochastic flow map of order 4, $\varphi(4) : \mathbb{R} \times \mathfrak{X} \to \mathfrak{X}$, 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 (3.20a) and (3.21).

Second-order ODE Likewise, specializing (3.19) for a second-order ODE ($n = 2$) yields

$$\partial_t^2 u(t, \xi) = \mathscr{f}(t, \xi, u(t, \xi), \dot{u}(t, \xi)) \qquad \text{on } \mathfrak{T} \times \Xi \tag{3.22a}$$

$$\left\{u(0, \xi) = c_1(\xi),\ \dot{u}(0, \xi) = c_2(\xi)\right\} \qquad \text{on } \{0\} \times \Xi, \tag{3.22b}$$

where $\dot{u} := \partial_t u$. Differentiating (3.22a) with respect to time three times gives:

$$\partial_t^3 u := \mathrm{D}_t \mathscr{f} = \partial_t \mathscr{f} + \partial_u \mathscr{f}\, \partial_t u + \partial_{\dot{u}} \mathscr{f}\, \partial_t^2 u \tag{3.23a}$$

$$\partial_t^4 u := \mathrm{D}_t^2 \mathscr{f} = \partial_t^2 \mathscr{f} + 2\, \partial_{tu}^2 \mathscr{f}\, \partial_t u + \left(2\, \partial_{t\dot{u}}^2 \mathscr{f} + \partial_u \mathscr{f}\right) \partial_t^2 u + \partial_{\dot{u}} \mathscr{f}\, \partial_t^3 u + \mathscr{h}_4 \tag{3.23b}$$

$$\begin{aligned} \partial_t^5 u := \mathrm{D}_t^3 \mathscr{f} = \partial_t^3 \mathscr{f} + 3\, \partial_{ttu}^3 \mathscr{f}\, \partial_t u + 3 \left(\partial_{tt\dot{u}}^3 \mathscr{f} + \partial_{tu}^2 \mathscr{f}\right) \partial_t^2 u \\ + \left(3\, \partial_{t\dot{u}}^2 \mathscr{f} + \partial_u \mathscr{f}\right) \partial_t^3 u + \partial_{\dot{u}} \mathscr{f}\, \partial_t^4 u + \mathscr{h}_5, \end{aligned} \tag{3.23c}$$

where $\mathscr{h}_4 = \partial_u^2 \mathscr{f} \left(\partial_t u\right)^2 + 2\, \partial_{u\dot{u}}^2 \mathscr{f}\, \partial_t u\, \partial_t^2 u + \partial_{\dot{u}}^2 \mathscr{f} \left(\partial_t^2 u\right)^2$, and

$$\begin{aligned} \mathscr{h}_5 = 3\, \partial_{tuu}^3 \mathscr{f} \left(\partial_t u\right)^2 + 3 \left(\partial_{t\dot{u}\dot{u}}^3 \mathscr{f} + \partial_{u\dot{u}}^2 \mathscr{f}\right) \left(\partial_t^2 u\right)^2 + \partial_u^3 \mathscr{f} \left(\partial_t u\right)^3 + \partial_{\dot{u}}^3 \mathscr{f} \left(\partial_t^2 u\right)^3 \\ + 3 \left(2\, \partial_{tu\dot{u}}^3 \mathscr{f} + \partial_u^2 \mathscr{f}\right) \partial_t u\, \partial_t^2 u + 3\, \partial_{u\dot{u}}^2 \mathscr{f}\, \partial_t u\, \partial_t^3 u + 3\, \partial_{\dot{u}}^2 \mathscr{f}\, \partial_t^2 u\, \partial_t^3 u \\ + 3\, \partial_{uu\dot{u}}^3 \mathscr{f} \left(\partial_t u\right)^2 \partial_t^2 u + 3\, \partial_{uu\dot{u}}^3 \mathscr{f}\, \partial_t u \left(\partial_t^2 u\right)^2. \end{aligned}$$

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

$$\varphi(4)(h, s(t_i, \cdot)) := \Big(u(t_i + h, \cdot), \dot{u}(t_i + h, \cdot)\Big) - O(h^5) \\ = \left(\sum_{j=0}^{4} \frac{h^j}{j!} \partial_t^j u(t_i, \cdot), \sum_{j=0}^{4} \frac{h^j}{j!} \partial_t^{j+1} u(t_i, \cdot) \right), \quad (3.24)$$

where the second and higher time derivatives of u at $t = t_i$ are computed with (3.22a) and (3.23).

Remark 3.9. In these expressions we observe that if (3.19) is a linear ODE, then $\mathscr{g}_3 = \mathscr{g}_4 = \mathscr{h}_4 = \mathscr{h}_5 \equiv 0$. If in addition it is autonomous, expressions (3.21) and (3.23) reduce, respectively, to:

$$\partial_t^2 u = \partial_u \mathscr{f}\, \partial_t u, \quad \partial_t^3 u = \partial_u \mathscr{f}\, \partial_t^2 u \quad \text{and} \quad \partial_t^4 u = \partial_u \mathscr{f}\, \partial_t^3 u. \quad (3.21^*)$$

$$\partial_t^3 u = \partial_u \mathscr{f}\, \partial_t u + \partial_{\dot{u}} \mathscr{f}\, \partial_t^2 u, \quad \partial_t^4 u = \partial_u \mathscr{f}\, \partial_t^2 u + \partial_{\dot{u}} \mathscr{f}\, \partial_t^3 u \\ \text{and} \quad \partial_t^5 u = \partial_u \mathscr{f}\, \partial_t^3 u + \partial_{\dot{u}} \mathscr{f}\, \partial_t^4 u. \quad (3.23^*)$$

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

$$\varphi(M)(h, s(t_i, \cdot)) := \Big(u(t_i + h, \cdot), \ldots, \partial_t^{k-1} u(t_i + h, \cdot), \ldots, \partial_t^{n-1} u(t_i + h, \cdot)\Big) \\ - O(h^{M+1}), \quad (3.25)$$

where its k-th component, $\varphi^k(M) : \mathbb{R} \times \mathfrak{X}^n \to \mathfrak{X}$, 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 $k \in \{1, 2, \ldots, n\}$.

In expression (3.25), the n-th time derivative of u at $t = t_i$ is computed with (3.19a), and if $M = 4$, then the next time derivatives of u are given by:

$$\partial_t^{n+1} u := \frac{\mathrm{d}\mathscr{f}}{\mathrm{d}t} = \frac{\partial \mathscr{f}}{\partial t} + \frac{\partial \mathscr{f}}{\partial s^k} \partial_t^k u, \quad (3.26a)$$

$$\partial_t^{n+2} u := \frac{\mathrm{d}^2 \mathscr{f}}{\mathrm{d}t^2} = \frac{\partial^2 \mathscr{f}}{\partial t^2} + 2 \frac{\partial^2 \mathscr{f}}{\partial t \partial s^k} \partial_t^k u + \frac{\partial \mathscr{f}}{\partial s^k} \partial_t^{k+1} u + \frac{\partial^2 \mathscr{f}}{\partial s^k \partial s^l} \partial_t^k u\, \partial_t^l u, \quad (3.26b)$$

and

$$\partial_t^{n+3}u := \frac{\mathrm{d}^3 \boldsymbol{\ell}}{\mathrm{d}t^3} = \frac{\partial^3 \boldsymbol{\ell}}{\partial t^3} + 3\frac{\partial^3 \boldsymbol{\ell}}{\partial t^2 \partial s^k}\partial_t^k u + 3\frac{\partial^2 \boldsymbol{\ell}}{\partial t \partial s^k}\partial_t^{k+1}u + \frac{\partial \boldsymbol{\ell}}{\partial s^k}\partial_t^{k+2}u + 3\frac{\partial^3 \boldsymbol{\ell}}{\partial t \partial s^k \partial s^l}\partial_t^k u\,\partial_t^l u + 3\frac{\partial^2 \boldsymbol{\ell}}{\partial s^k \partial s^l}\partial_t^{k+1}u\,\partial_t^l u + \frac{\partial^3 \boldsymbol{\ell}}{\partial s^k \partial s^l \partial s^m}\partial_t^k u\,\partial_t^l u\,\partial_t^m u, \quad (3.26c)$$

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 $k, l, m \in \{1, 2, \ldots, n\}$.

For an autonomous, n-th-order linear ODE, the first $M-1$ time derivatives of (3.19a) reduce to:

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

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 $\mathfrak{X}^n$ (i.e. in the *random phase space* of the system[2]), provided that the time-step size used is greater than zero. For example, Fig. 3.1 depicts the case of a system governed by a second-order ODE whose state motion in $\mathfrak{X}^2$ 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 $\mathfrak{X}^2$ if its map were known beforehand.

Remark 3.10. In practice, it is easier to compute the time derivatives of $\boldsymbol{\ell}$ 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 $\boldsymbol{\ell}$ at $t = t_i$, we get:

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

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

---

[2]Note that $\mathfrak{X}^n$ is the n-fold cartesian product of $\mathfrak{X}$, where n denotes the order of the system's governing ODE with respect to time.

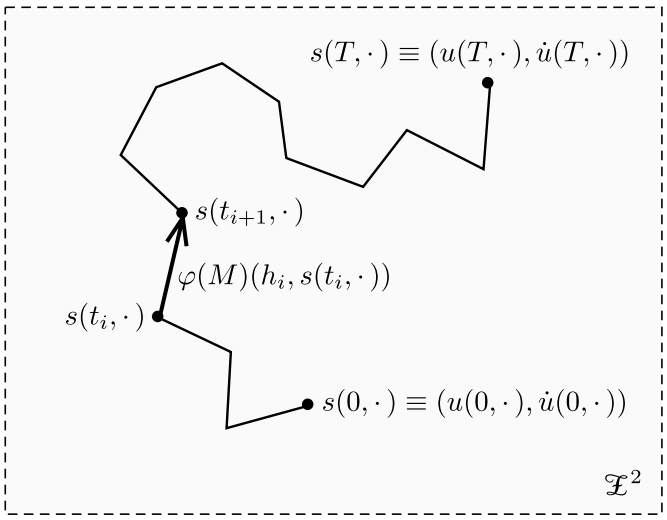


Figure 3.1. Evolution of a second-order dynamical system via a stochastic flow map of order M (with $h_i > 0$)

Example 3.2. Consider a specialization of the stochastic system given by (3.1) 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:

$$a_2(t,\xi)\,\ddot{u}(t,\xi) + a_1(t,\xi)\,\dot{u}(t,\xi) + a_0(t,\xi)\,u(t,\xi) = f(t,\xi) \qquad \text{on } \mathfrak{T}\times\Xi \tag{3.27a}$$

$$\left.\begin{aligned} b_{11}(\xi)\,\dot{u}(0,\xi) + b_{10}(\xi)\,u(0,\xi) &= b_1(\xi) \\ b_{21}(\xi)\,\dot{u}(0,\xi) + b_{20}(\xi)\,u(0,\xi) &= b_2(\xi) \end{aligned}\right\} \qquad \text{on } \{0\}\times\Xi, \tag{3.27b}$$

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

Provided sufficient regularity, an explicit form of (3.27) can be established by:

$$\begin{aligned} \partial_t^2 u(t,\xi) &:= \pmb{\ell}(t,\xi,u(t,\xi),\dot{u}(t,\xi)) \\ &= \bar{f}(t,\xi) + \bar{a}_0(t,\xi)\,u(t,\xi) + \bar{a}_1(t,\xi)\,\dot{u}(t,\xi) \qquad \text{on } \mathfrak{T}\times\Xi \end{aligned} \tag{3.28a}$$

$$\Big\{u(0,\xi) = \bar{b}_1(\xi),\ \dot{u}(0,\xi) = \bar{b}_2(\xi)\Big\} \qquad \text{on } \{0\}\times\Xi, \tag{3.28b}$$

where $\bar{f} = f/a_2$, $\bar{a}_0 = -a_0/a_2$ and $\bar{a}_1 = -a_1/a_2$ are elements of $\mathscr{V}$, 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 $\mathfrak{L}$.

This is, of course, a non-autonomous stochastic system because $\pmb{\ell}$ depends explicitly on the time variable t, and what is more, it features a time-dependent stochastic input $x = (x_1, x_2, x_3) \in \mathscr{V}^3$ 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 (3.24) to produce:

$$\begin{aligned} \varphi^1(4)(h, s(t_i,\cdot)) &= u(t_i,\cdot) + h\,\dot{u}(t_i,\cdot) + \tfrac{1}{2}h^2\,\partial_t^2 u(t_i,\cdot) \\ &\quad + \tfrac{1}{6}h^3\,\partial_t^3 u(t_i,\cdot) + \tfrac{1}{24}h^4\,\partial_t^4 u(t_i,\cdot) \end{aligned} \tag{3.29a}$$

$$\begin{aligned} \varphi^2(4)(h, s(t_i,\cdot)) &= \dot{u}(t_i,\cdot) + h\,\partial_t^2 u(t_i,\cdot) + \tfrac{1}{2}h^2\,\partial_t^3 u(t_i,\cdot) \\ &\quad + \tfrac{1}{6}h^3\,\partial_t^4 u(t_i,\cdot) + \tfrac{1}{24}h^4\,\partial_t^5 u(t_i,\cdot), \end{aligned} \tag{3.29b}$$

where the second time derivative $\partial_t^2 u(t_i, \cdot\,)$ is computed with (3.28a), and the next time derivatives $\{\partial_t^j u(t_i, \cdot\,)\}_{j=3}^5$ with (3.23) from where one obtains: $\hbar_4 = \hbar_5 \equiv 0$ and

$$\partial_t \ell = \partial_t \bar{f} + \partial_t \bar{a}_0\, u + \partial_t \bar{a}_1\, \dot{u}, \quad \partial_t^2 \ell = \partial_t^2 \bar{f} + \partial_t^2 \bar{a}_0\, u + \partial_t^2 \bar{a}_1\, \dot{u},$$
$$\partial_t^3 \ell = \partial_t^3 \bar{f} + \partial_t^3 \bar{a}_0\, u + \partial_t^3 \bar{a}_1\, \dot{u}, \quad \text{(3.30a)}$$

$$\partial_u \ell = \bar{a}_0, \quad \partial_{tu}^2 \ell = \partial_t \bar{a}_0, \quad \partial_{ttu}^3 \ell = \partial_t^2 \bar{a}_0, \quad \text{(3.30b)}$$

$$\partial_{\dot{u}} \ell = \bar{a}_1, \quad \partial_{t\dot{u}}^2 \ell = \partial_t \bar{a}_1, \quad \partial_{tt\dot{u}}^3 \ell = \partial_t^2 \bar{a}_1. \quad \text{(3.30c)}$$

3.5. 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 $\mathfrak{X}^n$, but also ℓ (as displayed in Eq. (3.19a)) and its first $M-1$ time derivatives. Because of this, we define the *enriched stochastic flow map* of order M, $\hat{\varphi}(M) : \mathbb{R} \times \mathfrak{X}^{n+M} \to \mathfrak{X}^{n+M}$, such that its k-th component is given by:

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

with $k \in \{1, 2, \ldots, n+M\}$. Here $\hat{s} = (u, \partial_t u, \ldots, \partial_t^{n+M-1} u) \in \prod_{j=1}^{n+M} \mathcal{T}(n+M-j) \otimes \mathfrak{X}$ is called the *enriched configuration state* of the system over $\mathfrak{T} \times \Xi$. Observe that $\hat{s}^{n+1} := \partial_t^n u = \ell$ is defined by (3.19a), and that if $M = 4$, then $\{\hat{s}^k := \partial_t^{k-1} u = \mathrm{D}_t^{k-n-1} \ell\}_{k=n+2}^{n+4}$ is given by the expressions prescribed in (3.26).

Example 3.3. Consider the stochastic system presented in Example 3.2. The associated enriched stochastic flow map for that system, $\hat{\varphi}(4)$, is found to be:

$$\hat{\varphi}^1(4)(h, s(t_i, \cdot\,)) := \varphi^1(4)(h, s(t_i, \cdot\,)) = s^1(t_i + h, \cdot\,) = u(t_i + h, \cdot\,) - O(h^5), \quad \text{(3.31a)}$$

$$\hat{\varphi}^2(4)(h, s(t_i, \cdot\,)) := \varphi^2(4)(h, s(t_i, \cdot\,)) = s^2(t_i + h, \cdot\,) = \dot{u}(t_i + h, \cdot\,) - O(h^5), \quad \text{(3.31b)}$$

$$\hat{\varphi}^3(4) := \ell = \partial_t^2 u, \quad \hat{\varphi}^4(4) := \mathrm{D}_t \ell = \partial_t^3 u, \quad \hat{\varphi}^5(4) := \mathrm{D}_t^2 \ell = \partial_t^4 u$$
$$\text{and} \quad \hat{\varphi}^6(4) := \mathrm{D}_t^3 \ell = \partial_t^5 u, \quad \text{(3.31c)}$$

where $\hat{\varphi}^1(4)$ and $\hat{\varphi}^2(4)$ are computed with (3.29), $\hat{\varphi}^3(4)$ with (3.28a), and $\{\hat{\varphi}^k(4)\}_{k=4}^6$ with (3.23). In Example 3.2 we found that from (3.23) one obtains $\hbar_4 = \hbar_5 \equiv 0$ and (3.30).

3.6. 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 3.4, we assumed that the solution of the system governed by (3.19) was analytic on the temporal domain. This assumption implies that u can be represented by a Taylor series centered at $t = t_i \in \mathfrak{T}$ for all $\xi \in \Xi$:

$$u(t,\xi) = \sum_{j=0}^{\infty} \frac{(t-t_i)^j}{j!} \partial_t^j u(t_i,\xi), \tag{3.32}$$

where $(t-t_i)^j/j!$ is nothing but a temporal function in $\mathscr{T}$, and $\partial_t^j u(t_i,\xi)$ is a random function in $\mathfrak{X}$. From this it follows that if $\{\partial_t^j u(t_i,\xi)\}_{j=0}^{\infty}$ is orthogonalized with respect to the measure in $\mathfrak{X}$, one can write expression (3.32) in the following way:

$$u(t,\xi) = \sum_{j=0}^{\infty} u^j(t)\, \Psi_j(\xi). \tag{3.2}$$

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 $\mathfrak{Z}^{[P]}$ 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.

Remark 3.11. 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 $\mathfrak{Z}$ is such that $P = n + M$. However, as we will see in Section 3.8, accurate results are also achievable for a lower discretization level of $\mathfrak{Z}$, 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 (3.1). 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 $s_{\cdot i} = s|_{\mathrm{cl}(\mathfrak{T}_i)\times\Xi}$ to be the restriction of s to $\mathfrak{R}_i := \mathrm{cl}(\mathfrak{T}_i) \times \Xi$. (Please see Fig. 3.2, 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 $s_{\cdot i}$ in the following way.

   - Take $\Phi_{0\cdot i} \equiv 1$ and $\{\Phi_{j\cdot i} := \hat{\varphi}^j(M)(0, \hat{s}(t_i, \cdot))\}_{j=1}^{P}$ to be an ordered set of linearly independent functions in $\mathfrak{Z}$ with $n + 1 \leq P \leq n + M$. Observe that $\hat{\varphi}(M)(0, \hat{s}(t_i, \cdot)) \equiv \hat{s}_{\cdot i}(t_i, \cdot) = \hat{s}_{\cdot i-1}(t_i, \cdot)$ 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 3.12.) It is worth mentioning that, for computational efficiency, we have chosen $h_i = 0$ for the definition of $\{\Phi_{j\cdot i}\}_{j=1}^{P}$, 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 $\{\Phi_{j\cdot i}\}_{j=0}^{P}$ using the Gram-Schmidt process [61], so that the resulting set $\{\Psi_{j\cdot i}\}_{j=0}^{P}$ is an orthogonal basis in $\mathfrak{Z}$. That is, for $j \in \{0, 1, \ldots, P\}$:

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

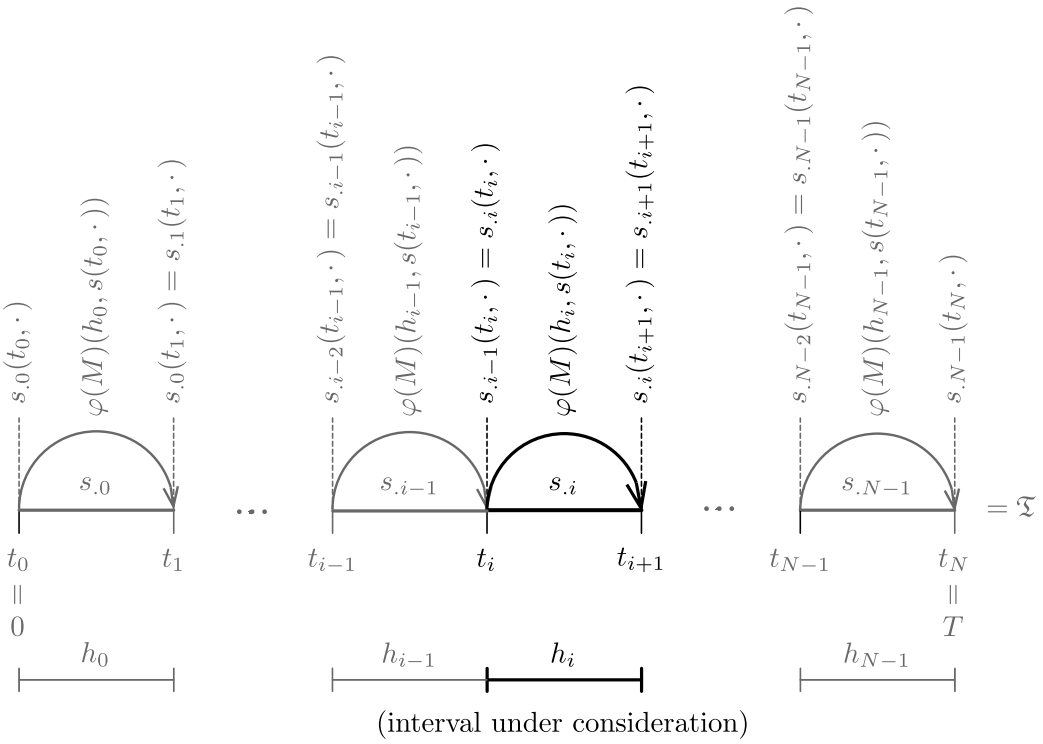


Figure 3.2. Evolution of a dynamical system via a stochastic flow map of order M (with $h_i > 0$)

(Equivalently, Theorem 3.1 can be employed in this step.)

- Define $\mathfrak{X}_i^{[P]} = \text{span}\{\Psi_{j.i}\}_{j=0}^{P}$ to be a p-discretization of $\mathfrak{X}$ over the region $\mathfrak{R}_i$. Since $\mathfrak{X}_i^{[P]}$ is an evolving function space, expansion (3.3) is now to be read as:

$$u_{.i}(t,\xi) \approx u_{.i}^{[P]}(t,\xi) = \sum_{j=0}^{P} u^j{}_{.i}(t)\,\Psi_{j.i}(\xi) \equiv u^j{}_{.i}(t)\,\Psi_{j.i}(\xi). \tag{3.3*}$$

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

(b) Transfer the random modes of

$$s_{.i-1} = (u_{.i-1}, \partial_t u_{.i-1}, \ldots, \partial_t^{n-1} u_{.i-1}) \quad \text{to} \quad s_{.i} = (u_{.i}, \partial_t u_{.i}, \ldots, \partial_t^{n-1} u_{.i})$$

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):

$$s^l{}_{.i}(t_i,\xi) = s^l{}_{.i-1}(t_i,\xi) \quad \Longleftrightarrow \quad (s^l)^k{}_{.i}(t_i)\,\Psi_{k.i}(\xi) = (s^l)^k{}_{.i-1}(t_i)\,\Psi_{k.i-1}(\xi). \tag{3.33}$$

Projecting (3.33) onto $\mathfrak{X}_i^{[P]}$ gives the random modes of each of the components of $s_{.i}$ at $t = t_i$:

$$(s^l)^j{}_{.i}(t_i) = \sum_{k=0}^{P} \frac{\langle \Psi_{j.i}, \Psi_{k.i-1} \rangle}{\langle \Psi_{j.i}, \Psi_{j.i} \rangle} (s^l)^k{}_{.i-1}(t_i), \tag{3.8b*}$$

where $l \in \{1, 2, \ldots, n\}$ and $j \in \{0, 1, \ldots, P\}$. This approach was first introduced by Heuveline and Schick in [35].

FSC-2. This approach transfers the probability information exactly. In order to do so, Theorem 3.1 is implemented to obtain the random modes of each of the components of $s_{.i}$ at $t = t_i$. Thus, from (3.12) it follows that:

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

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

$$(s^l)^j{}_{.i}(t_i) = \begin{cases} \mathbf{E}[s^l{}_{.i-1}(t_i, \cdot)] & \text{for } j = 0 \\ \dfrac{\det \triangle_j(l)}{\det \square_j} & \text{for } 0 < j < l \\ 1 & \text{for } j = l \\ 0 & \text{otherwise,} \end{cases} \tag{3.8b*}$$

where $l \in \{1, 2, \ldots, n\}$ and $j \in \{0, 1, \ldots, P\}$. This approach was first introduced by Gerritsma et al. in [34], but it had not been generalized for high-order stochastic ODEs until now.

If $i = 0$, the initial conditions are computed with (3.8b) directly.

(c) Substitute (3.3*) into (3.1) to obtain (3.7).

(d) Project (3.7a) onto $\mathfrak{L}_i^{[P]}$ to obtain (3.8a) subject to (3.8b*). Note that if $i = 0$, (3.8a) is subject to (3.8b).

(e) Integrate (3.8) 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 $y = \boldsymbol{\mathcal{M}}[u][x]$ over $\mathfrak{R}_i$, by recurring to the formulas prescribed by (3.9) and (3.10).

2. Post-process results.

Remark 3.12. 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 $A : \mathfrak{X}^{n+1} \to \mathfrak{X}^{r+1}$ is a linear map given by:

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

  For example, if $n = 4$ and $r = 2$, the map $A : \mathfrak{X}^5 \to \mathfrak{X}^3$ 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 3.13. 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. [34] 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 3.4. Consider Example 3.2 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 $s_{.i} = (u_{.i}, \dot{u}_{.i})$:

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

with $j \in \{0, 1, \ldots, P\}$. 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:

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

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

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

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

3.7. 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.

3.7.1. 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 $v : \mathfrak{T} \times \Xi \to \mathbb{R}$ in $\mathcal{U}$, such that (μ-a.e.):

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

where $m = 4$ kg is the mass of the falling body, $k : \Xi \to \mathbb{R}^+$ is the air resistance given by $k(\xi) = \xi$, $g = 9.81$ m/s^2 is the gravity acceleration, $\mathfrak{v} = 50$ 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*, Uniform $\sim \xi \in \Xi = [a, b]$, and the second distribution is a *beta distribution*, Beta$(\alpha, \beta) \sim \xi \in \Xi = [a, b]$. The parameters for both distributions are: $(a, b) = (1, 2)$ kg/s and $(\alpha, \beta) = (2, 5)$.

3.7.2. 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 $u : \mathfrak{T} \times \Xi \to \mathbb{R}$ in $\mathcal{U}$, such that (μ-a.e.):

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

where $m = 100$ kg is the mass of the system, $k : \Xi \to \mathbb{R}^+$ is the stiffness of the system given by $k(\xi) = \xi$, $\mathfrak{u} = 0.05$ m and $\mathfrak{v} = 0.20$ 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*, Uniform $\sim \xi \in \Xi = [a, b]$, the second distribution is a *beta distribution*, Beta$(\alpha_1, \beta_1) \sim \xi \in \Xi = [a, b]$, and the third distribution is a *gamma distribution*, 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 $(\alpha_1, \beta_1, \alpha_2, \beta_2) = (2, 5, 10, 0.1)$.

3.7.3. 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, $m : \Xi \to \mathbb{R}^+$, and the stiffness of the system, $k : \Xi \to \mathbb{R}^+$, 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, Ξ 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 $(\alpha, \beta) = (2, 5)$.

3.7.4. 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 $u : \mathfrak{T} \times \Xi \to \mathbb{R}$ in $\mathcal{U}$, such that (μ-a.e.):

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

where $k : \Xi \to \mathbb{R}$ is a mechanical parameter given by $k(\xi) = \xi$; $\mathfrak{u} = 1$ m, $\mathfrak{v} = -1$ m/s and $\mathfrak{a} = 2$ m/s^2 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 $(\alpha, \beta) = (2, 5)$.

3.7.5. 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 $u : \mathfrak{T} \times \Xi \to \mathbb{R}$ in $\mathcal{U}$, such that (μ-a.e.):

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

where $k : \Xi \to \mathbb{R}$ is a mechanical parameter given by $k(\xi) = \xi$; $\mathcal{u} = 1$ m, $\mathcal{v} = -1$ m/s, $\mathcal{a} = 2$ m/s^2 and $\mathcal{j} = -3$ m/s^3 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 $(\alpha, \beta) = (2, 5)$.

3.7.6. 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 3.A. The problem can be stated as follows.

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

$$m\ddot{u} - (1 - \rho u^2)\, c\dot{u} + ku = 0 \qquad \text{on } \mathfrak{T} \times \Xi \tag{3.34a}$$

$$\left\{u(0,\cdot) = \mathcal{u},\, \dot{u}(0,\cdot) = \mathcal{v}\right\} \qquad \text{on } \{0\} \times \Xi, \tag{3.34b}$$

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^{-2} is the contributing factor to the nonlinearity of the oscillator, $k = 400$ N/m is the stiffness of the oscillator, $\mathcal{u} \sim \text{Beta}(2, 5)$ is a *beta-distributed* random variable defined in $[0.05, 0.25]$ m, $\mathcal{v} = 2\mathcal{u} - 0.10$ 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.

3.8. Discussion on numerical results

In this section, we demonstrate and compare the performance of the FSC scheme using the two approaches developed in Section 3.6 for the transfer of the probability information, namely FSC-1 and FSC-2. We do this for each of the problems described in Section 3.7, followed by a computational-cost comparison between FSC and mTD-gPC [35] for Problem 2 of Section 3.7.2.

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

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

where Δt 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 [29] 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[3]. 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 3.13, 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.

[3]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.

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

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

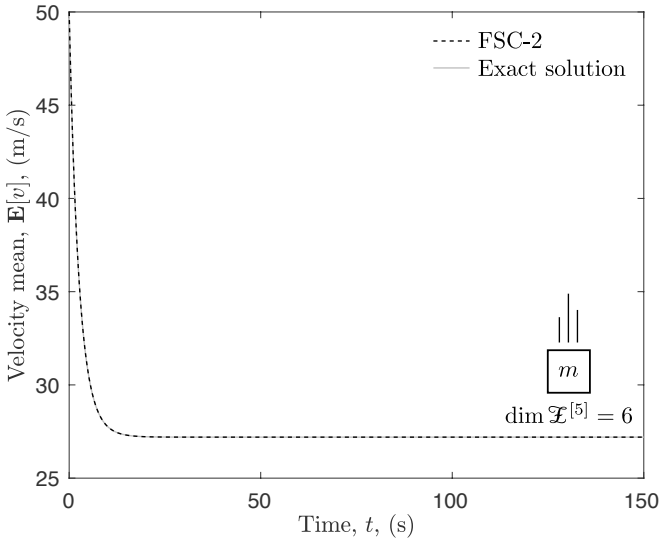
All problems are run in MATLAB R2016b [62] on a 2017 MacBook Pro with quad-core *3.1 GHz Intel Core i7* processor (hyper-threading technology enabled), *16 GB 2133 MHz LPDDR3* memory and *1 TB PCI-Express SSD* storage (APFS-formatted), running macOS Mojave (version 10.14.6).

3.8.1. Numerical results for the five linear systems

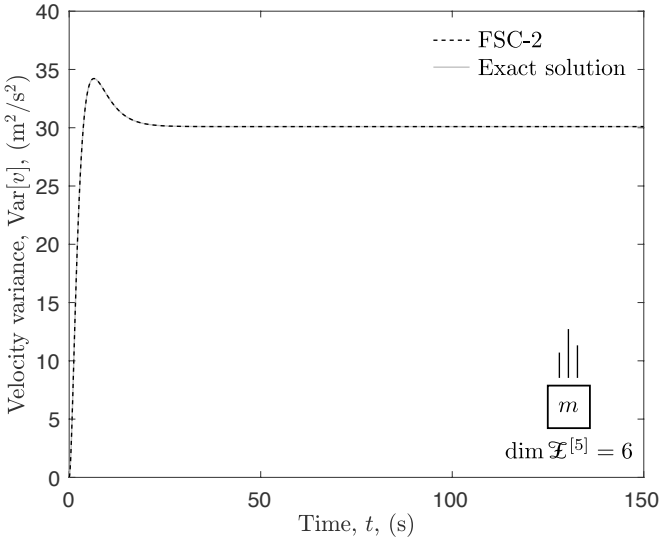
Figs. 3.3 to 3.7 show the evolution of the mean and the variance of one of the system's responses using FSC-2 and the exact solution[4] for sake of comparison. In particular, Fig. 3.3 shows the evolution of the system's velocity for Problem 1, Figs. 3.4 to 3.6 the evolution of the system's displacement for Problems 2 to 4, and Fig. 3.7 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. 3.8 to 3.16 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

[4]To obtain both the 'exact' mean and the 'exact' variance, MATLAB's Symbolic Math Toolbox [62] 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.

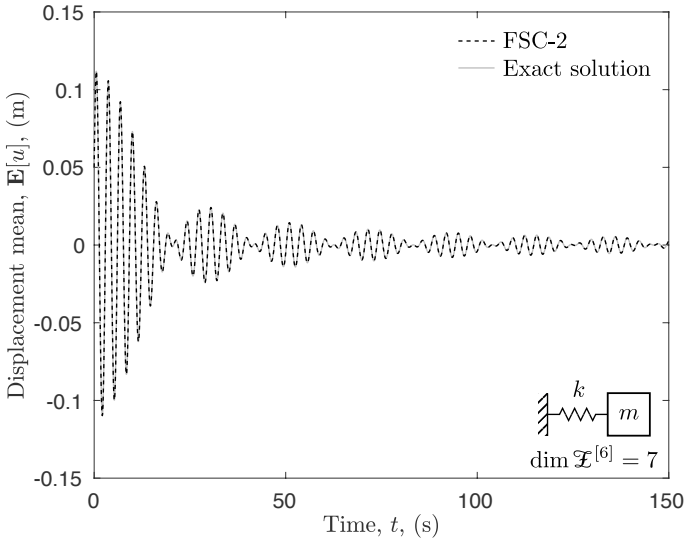


(a) Mean

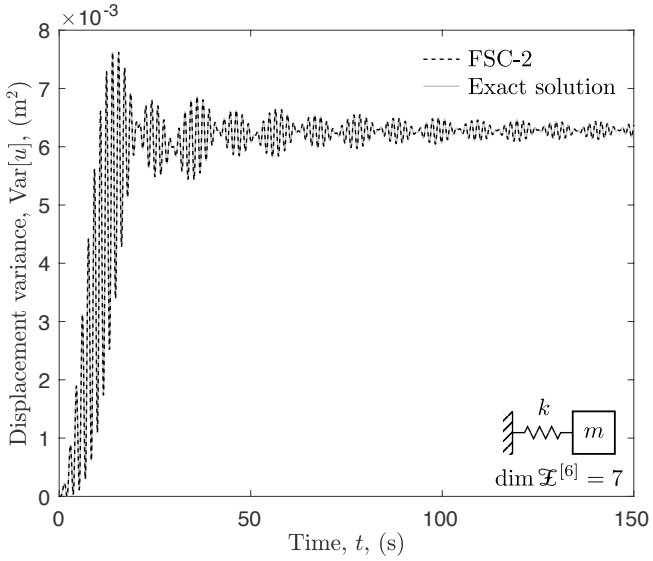


(b) Variance

Figure 3.3. *Problem 1* — Evolution of $\mathbf{E}[v]$ and $\mathrm{Var}[v]$ for the case when the p-discretization level of RFS is $\mathscr{Z}^{[5]}$ and $\mu \sim$ Uniform

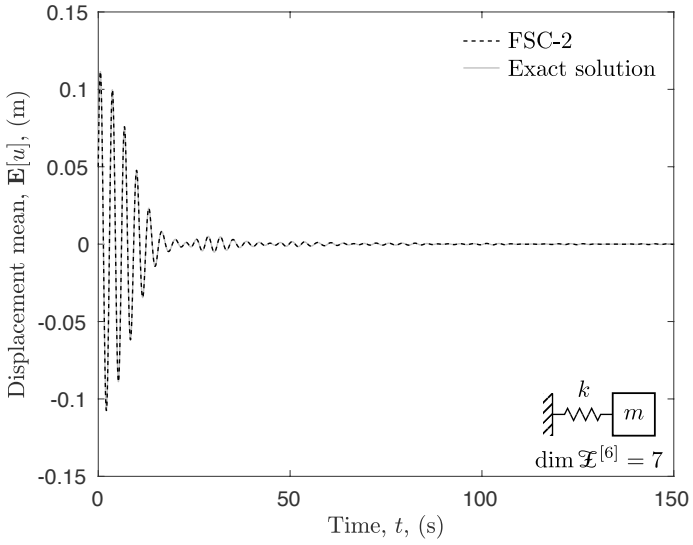


(a) Mean

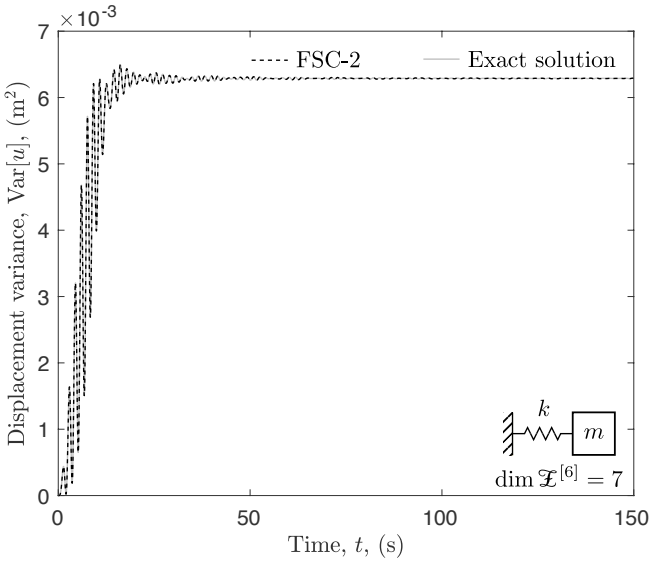


(b) Variance

Figure 3.4. *Problem 2* — Evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for the case when the p-discretization level of RFS is $\mathscr{Z}^{[6]}$ and $\mu \sim$ Uniform

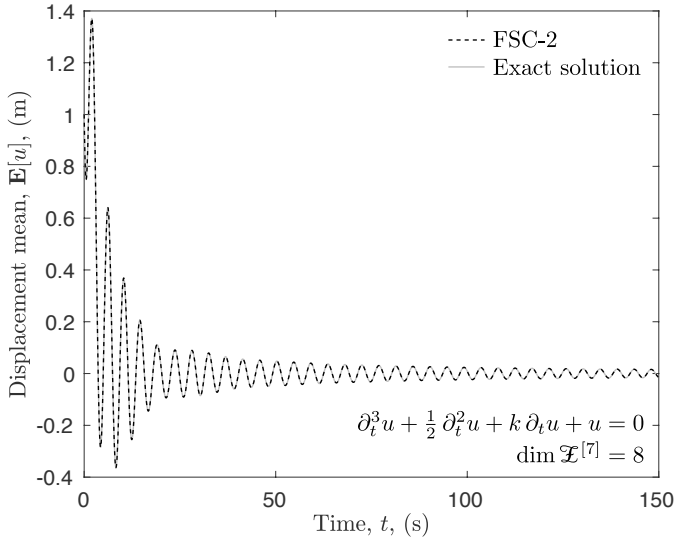


(a) Mean

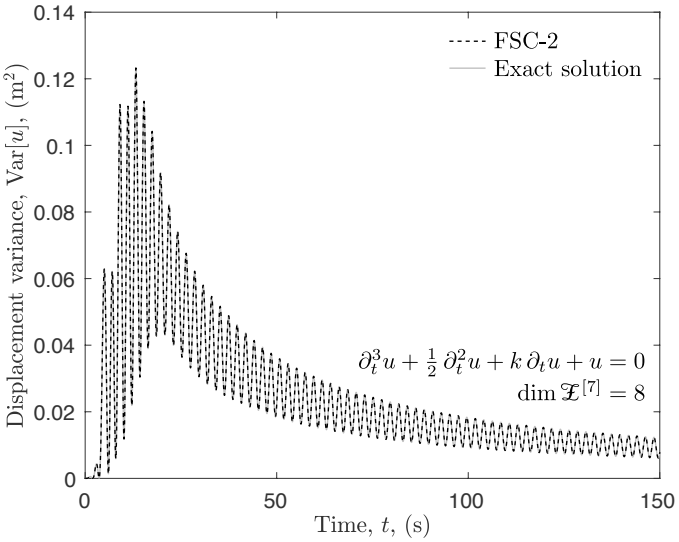


(b) Variance

Figure 3.5. *Problem 3* — Evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for the case when the p-discretization level of RFS is $\mathscr{Z}^{[6]}$ and $\mu \sim \text{Uniform} \otimes \text{Uniform}$

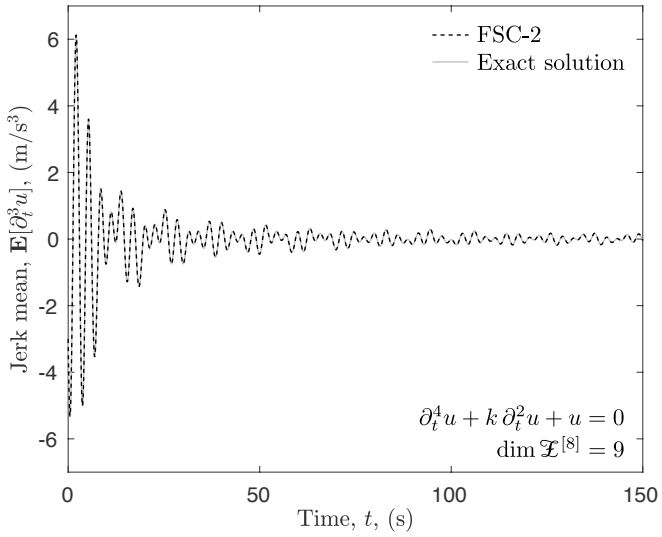


(a) Mean

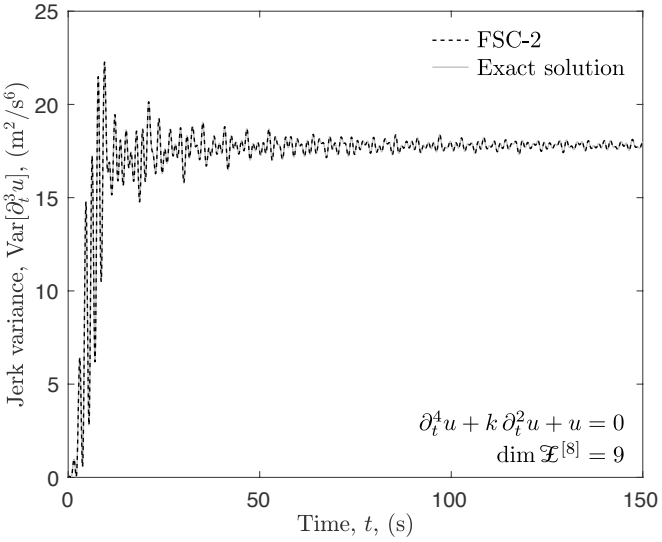


(b) Variance

Figure 3.6. *Problem 4* — Evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for the case when the p-discretization level of RFS is $\mathfrak{Z}^{[7]}$ and $\mu \sim$ Uniform



(a) Mean

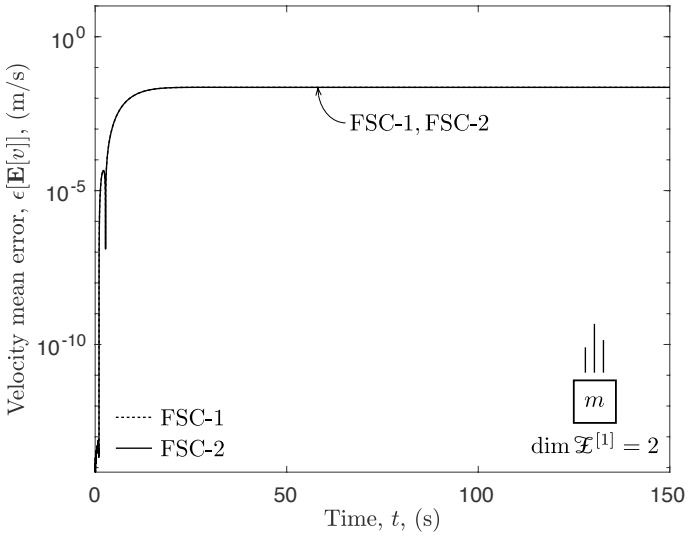


(b) Variance

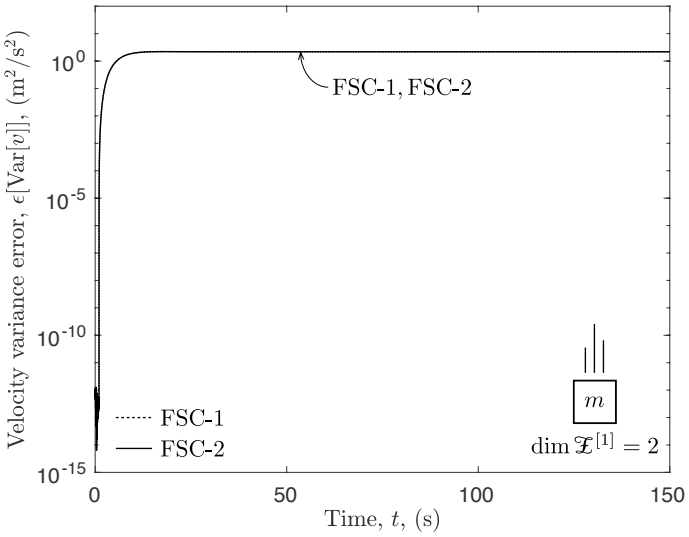
Figure 3.7. *Problem 5* — Evolution of $\mathbf{E}[\partial_t^3 u]$ and $\mathrm{Var}[\partial_t^3 u]$ for the case when the p-discretization level of RFS is $\mathfrak{X}^{[8]}$ and $\mu \sim$ Uniform

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 Figs. 3.14 to 3.16). 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.

Figs. 3.17 to 3.21 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 3.7.1 to 3.7.5. 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. 3.17 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. 3.18 and 3.19 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. 3.19 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

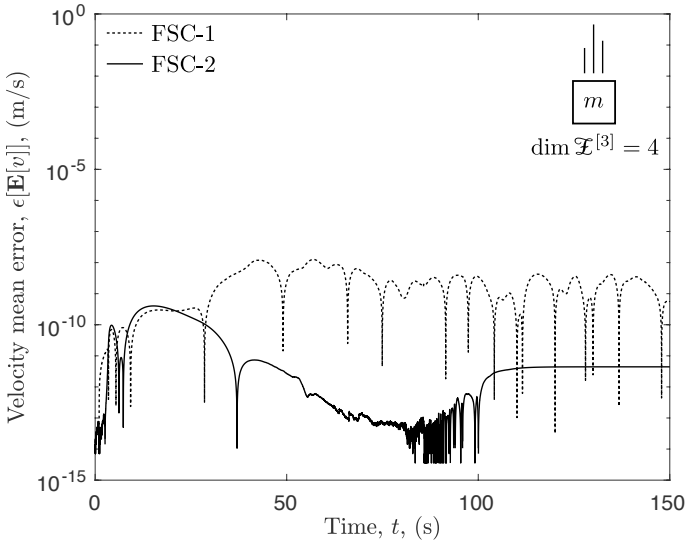


(a) Mean error for $\mathfrak{Z}^{[1]}$

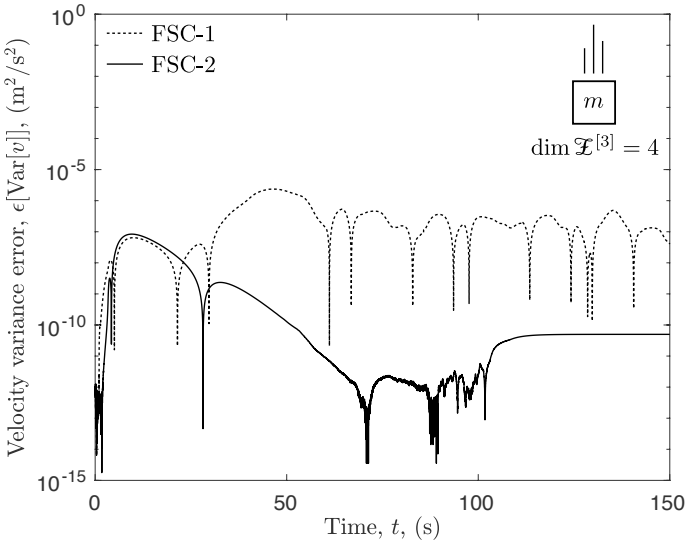


(b) Variance error for $\mathfrak{Z}^{[1]}$

Figure 3.8. *Problem 1* — Local error evolution of $\mathbf{E}[v]$ and $\mathrm{Var}[v]$ for different p-discretization levels of RFS and for $\mu \sim$ Uniform (SET 1/3)

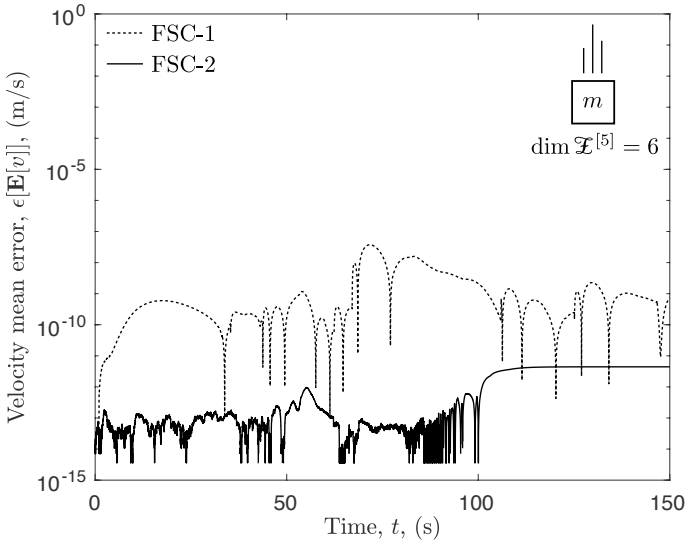


(a) Mean error for $\mathfrak{Z}^{[3]}$

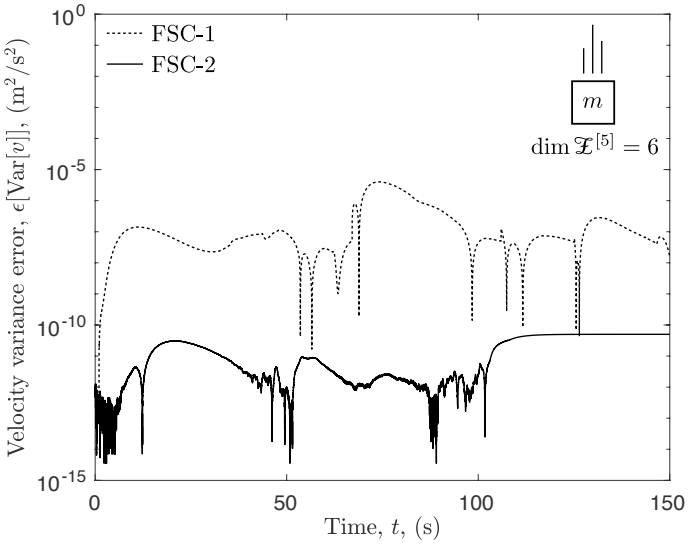


(b) Variance error for $\mathfrak{Z}^{[3]}$

Figure 3.9. *Problem 1* — Local error evolution of $\mathbf{E}[v]$ and $\mathrm{Var}[v]$ for different p-discretization levels of RFS and for $\mu \sim$ Uniform (SET 2/3)

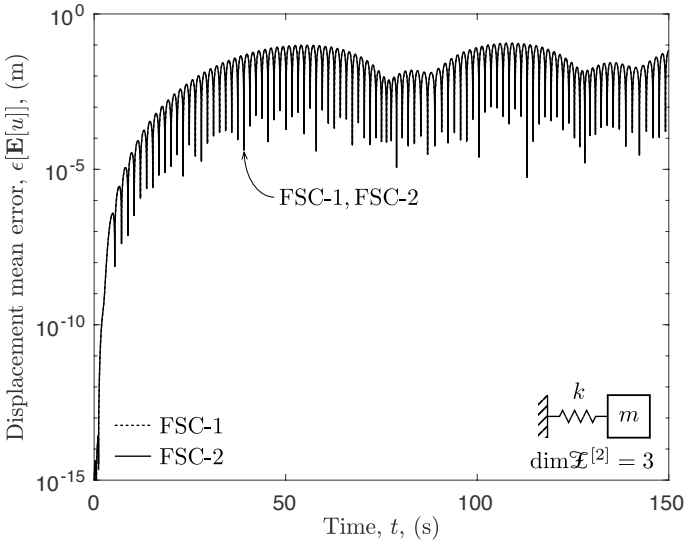


(a) Mean error for $\mathfrak{X}^{[5]}$

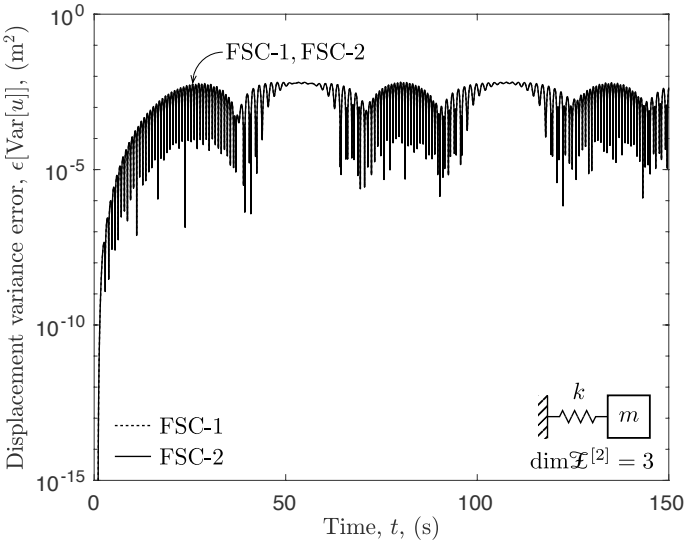


(b) Variance error for $\mathfrak{X}^{[5]}$

Figure 3.10. *Problem 1* — Local error evolution of $\mathbf{E}[v]$ and $\mathrm{Var}[v]$ for different p-discretization levels of RFS and for $\mu \sim$ Uniform (SET 3/3)

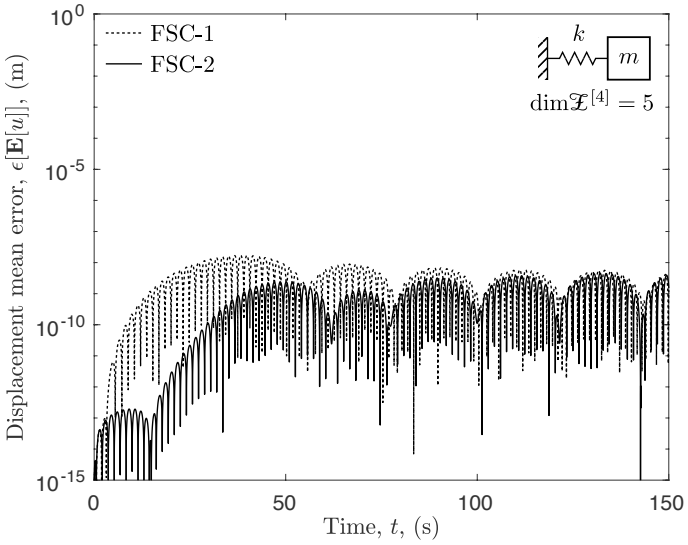


(a) Mean error for $\mathscr{Z}^{[2]}$

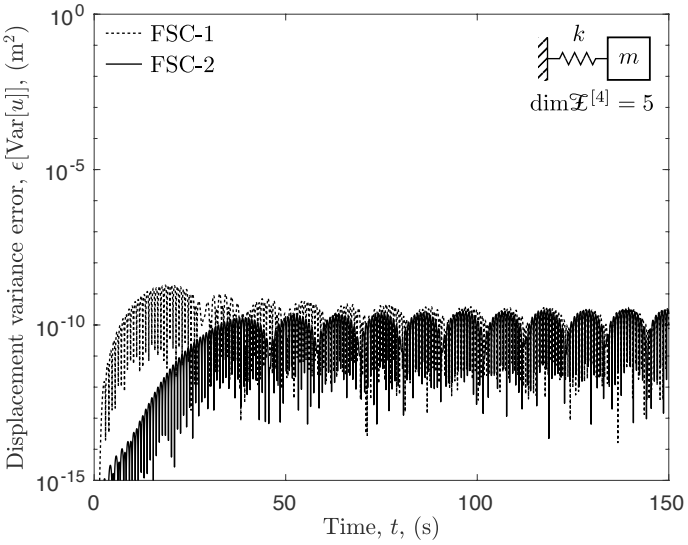


(b) Variance error for $\mathscr{Z}^{[2]}$

Figure 3.11. *Problem 2* — Local error evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different p-discretization levels of RFS and for $\mu \sim$ Uniform (SET 1/3)

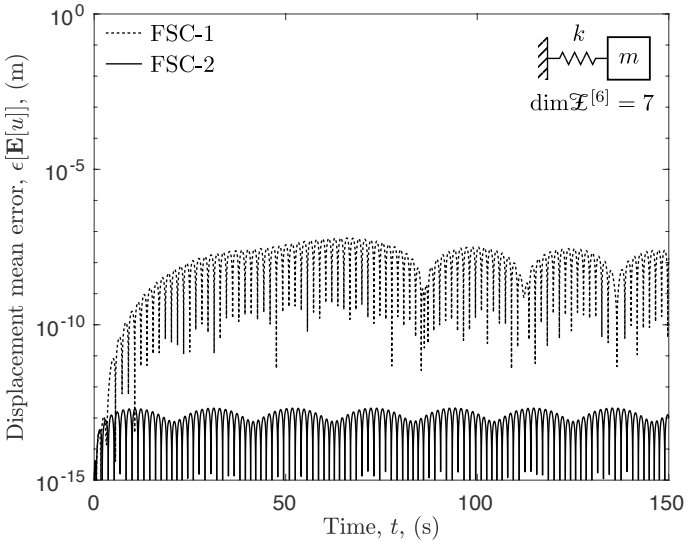


(a) Mean error for $\mathscr{Z}^{[4]}$

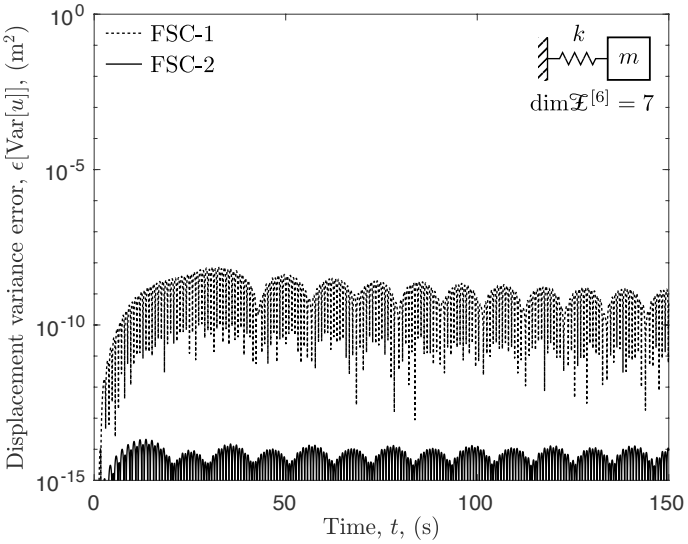


(b) Variance error for $\mathscr{Z}^{[4]}$

Figure 3.12. *Problem 2* — Local error evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different p-discretization levels of RFS and for $\mu \sim$ Uniform (SET 2/3)

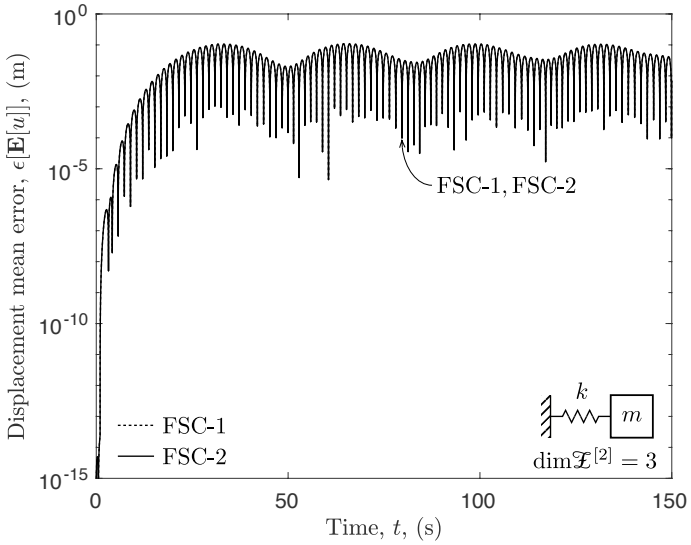


(a) Mean error for $\mathfrak{Z}^{[6]}$

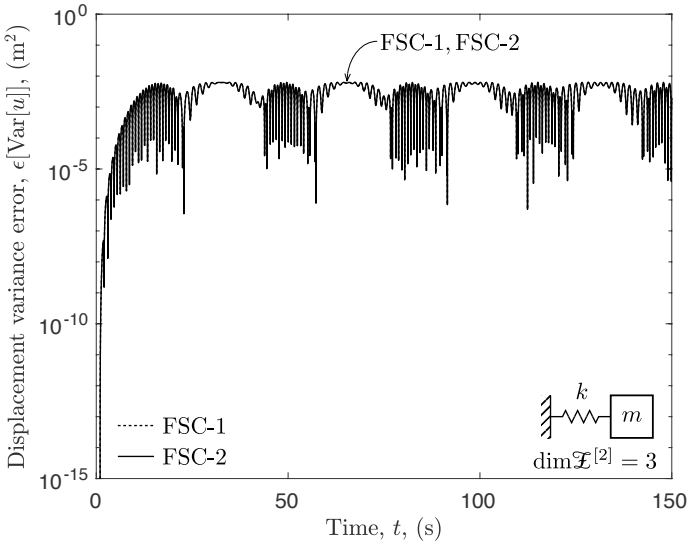


(b) Variance error for $\mathfrak{Z}^{[6]}$

Figure 3.13. *Problem 2* — Local error evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different p-discretization levels of RFS and for $\mu \sim$ Uniform (SET 3/3)

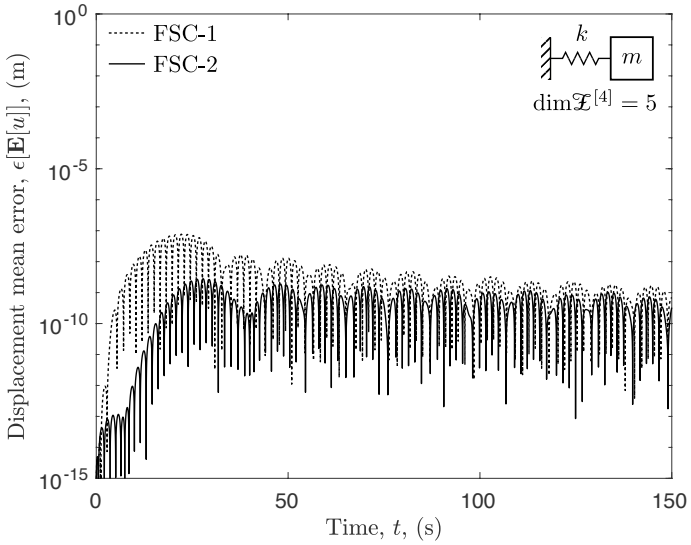


(a) Mean error for $\mathscr{Z}^{[2]}$

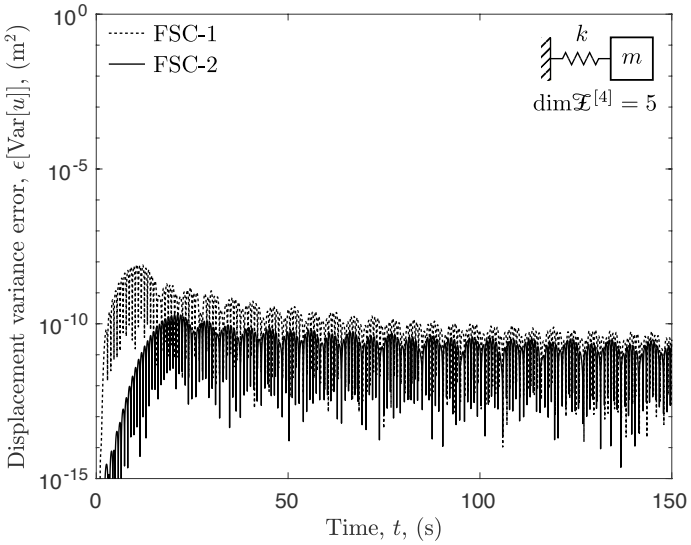


(b) Variance error for $\mathscr{Z}^{[2]}$

Figure 3.14. *Problem 3* — Local error evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different p-discretization levels of RFS and for $\mu \sim \text{Uniform} \otimes \text{Uniform}$ (SET 1/3)

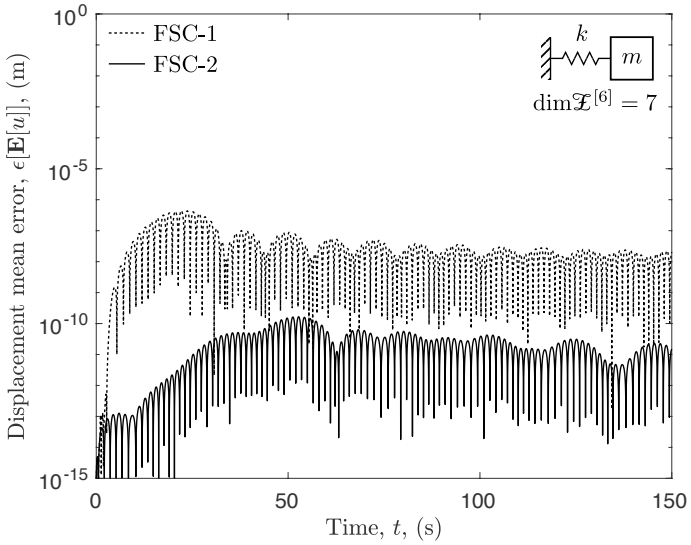


(a) Mean error for $\mathfrak{Z}^{[4]}$

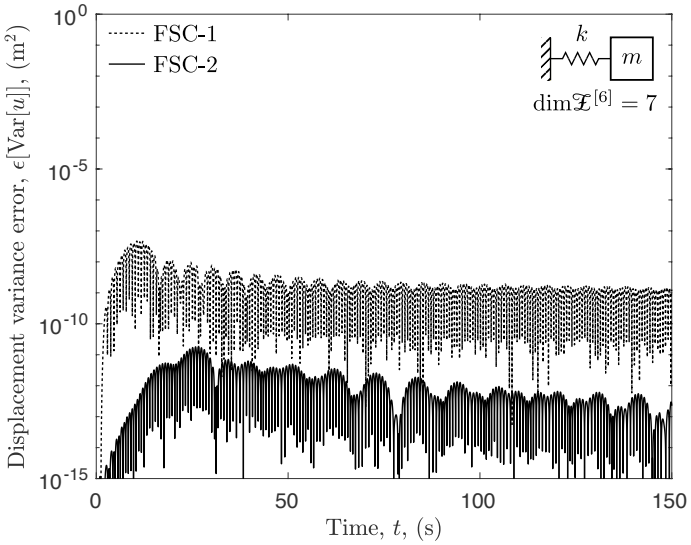


(b) Variance error for $\mathfrak{Z}^{[4]}$

Figure 3.15. *Problem 3* — Local error evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different p-discretization levels of RFS and for $\mu \sim \mathrm{Uniform} \otimes \mathrm{Uniform}$ (SET 2/3)



(a) Mean error for $\mathfrak{Z}^{[6]}$



(b) Variance error for $\mathfrak{Z}^{[6]}$

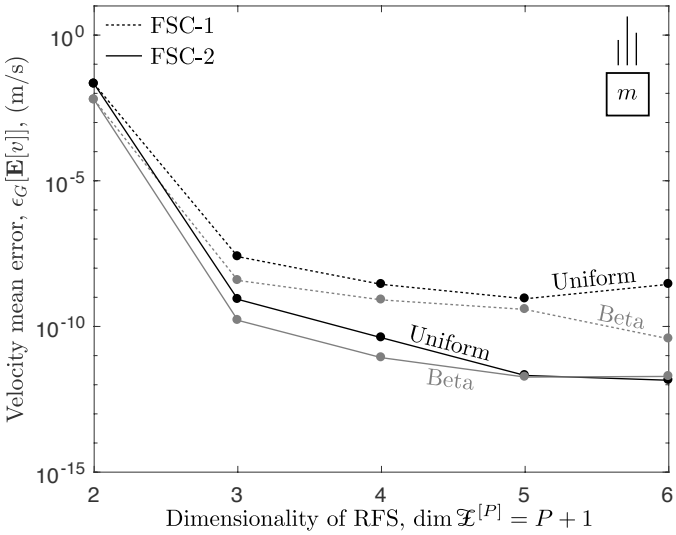
Figure 3.16. *Problem 3* — Local error evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different p-discretization levels of RFS and for $\mu \sim \text{Uniform} \otimes \text{Uniform}$ (SET 3/3)

already produces a global error of approximately 10^{-10}. Finally, Fig. 3.21 indicates that when the response is unbounded (e.g. when ξ is assumed normally distributed), FSC-2 has the ability to control better the error propagation of the solution as the simulation proceeds.

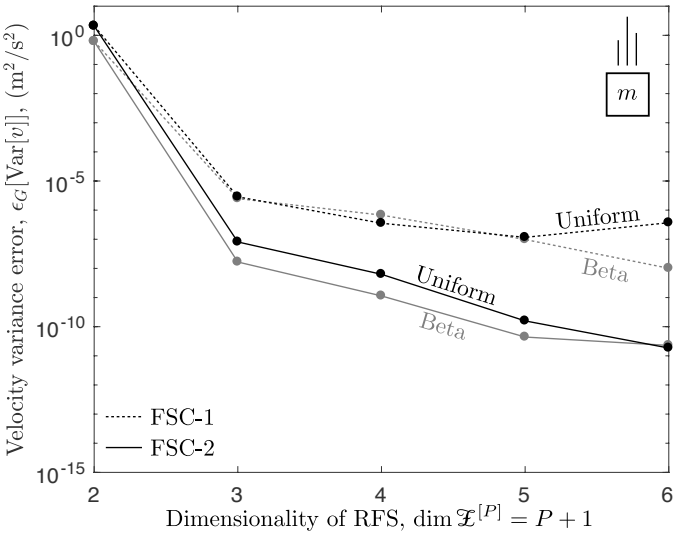
Fig. 3.22 presents the convergence of global errors for Problem 2 as a function of the number of basis vectors and for different time-step sizes. The goal of this figure is to show the implications of increasing the time-step size used by default ($\Delta t = 0.001$ s) in regard to the accuracy of the results. Though here we only depict the case when ξ 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.

Fig. 3.23 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. 3.24 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. [35]

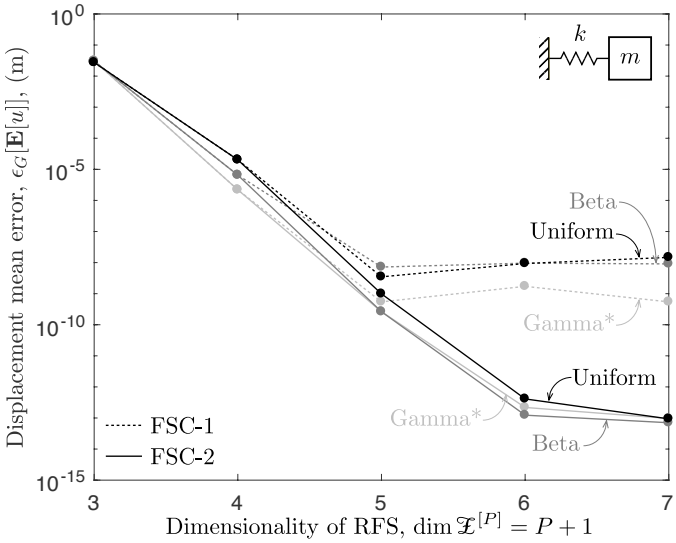


(a) Mean error

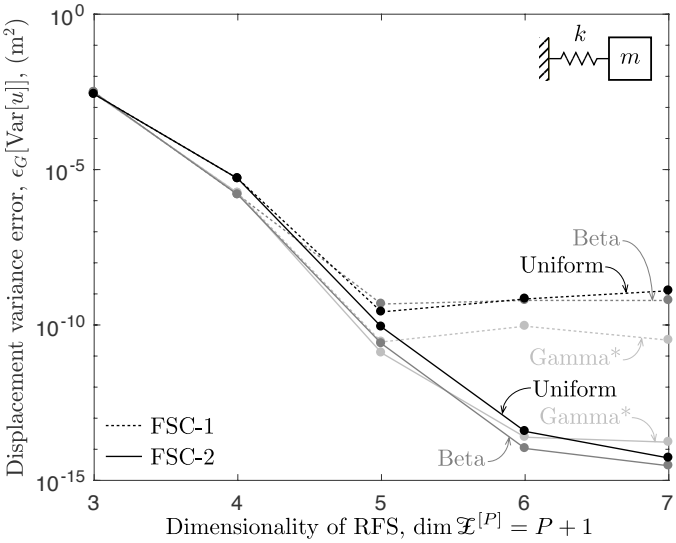


(b) Variance error

Figure 3.17. *Problem 1* — Global error of $\mathbf{E}[v]$ and $\mathrm{Var}[v]$ for different p-discretization levels of RFS

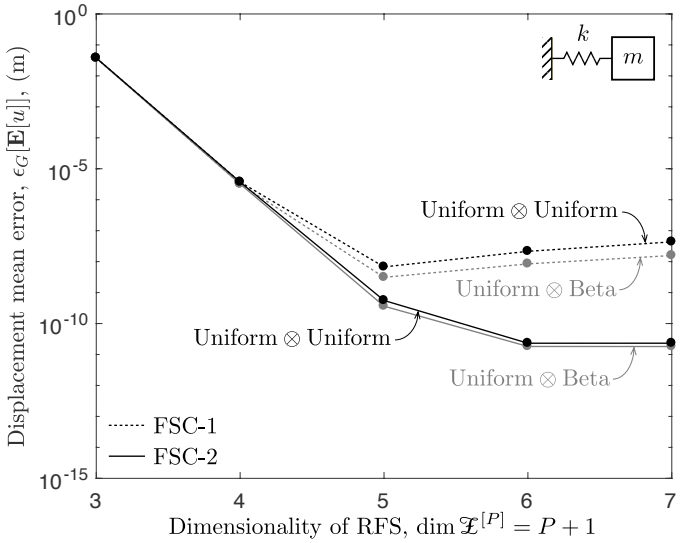


(a) Mean error

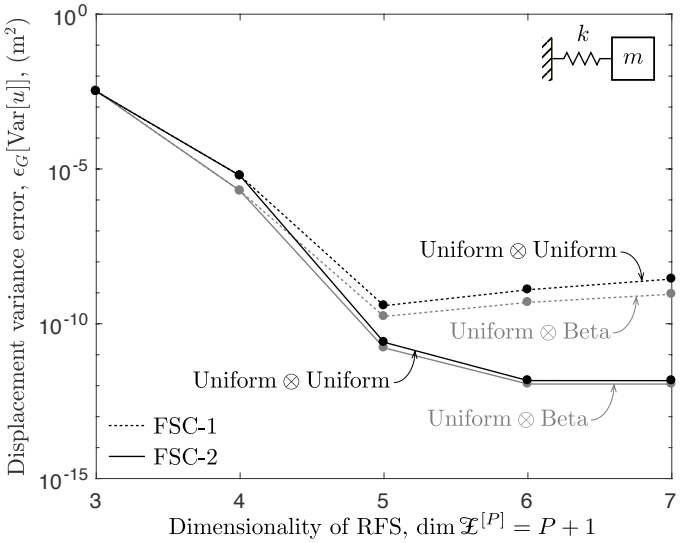


(b) Variance error

Figure 3.18. *Problem 2* — Global error of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different p-discretization levels of RFS

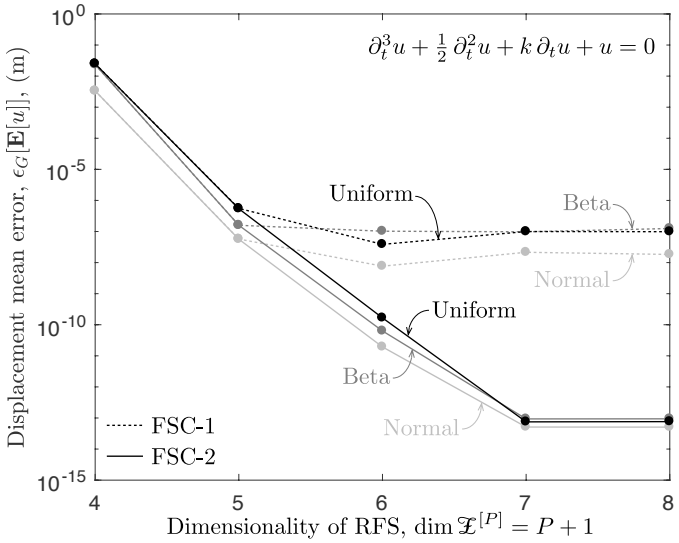


(a) Mean error

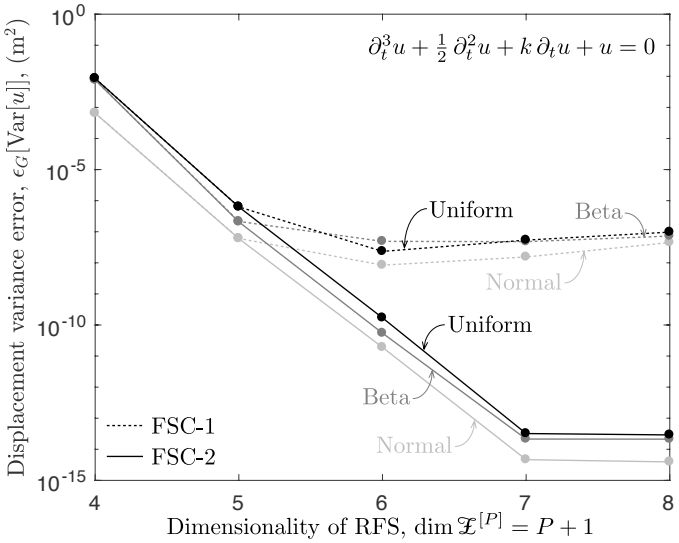


(b) Variance error

Figure 3.19. *Problem 3* — Global error of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different p-discretization levels of RFS

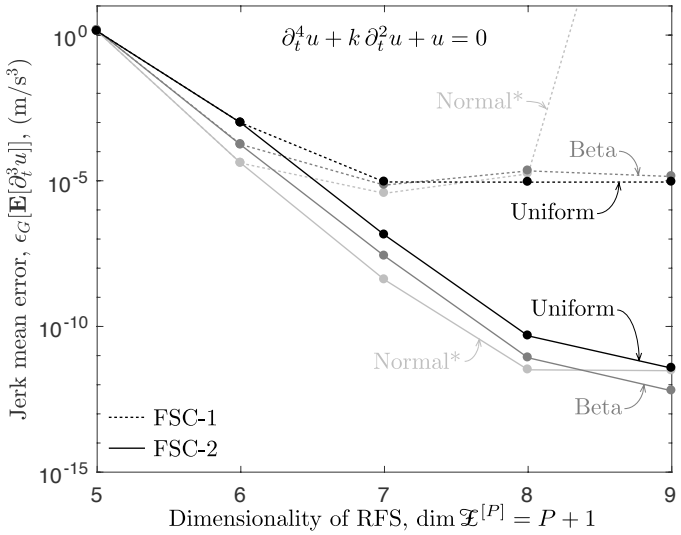


(a) Mean error

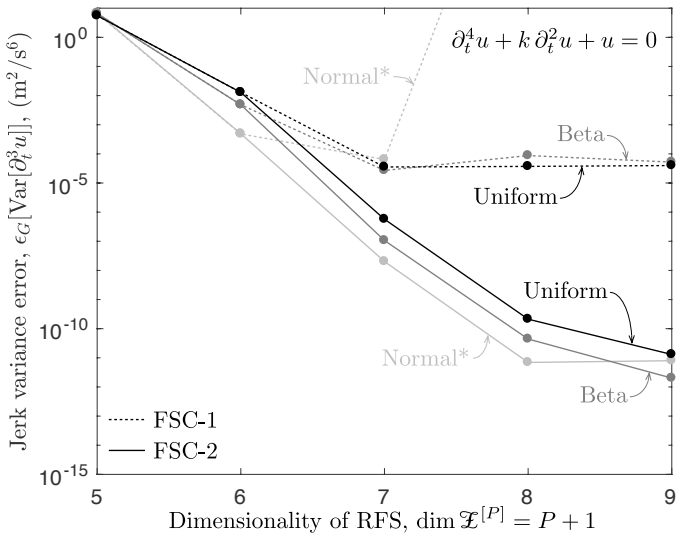


(b) Variance error

Figure 3.20. *Problem 4* — Global error of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different p-discretization levels of RFS

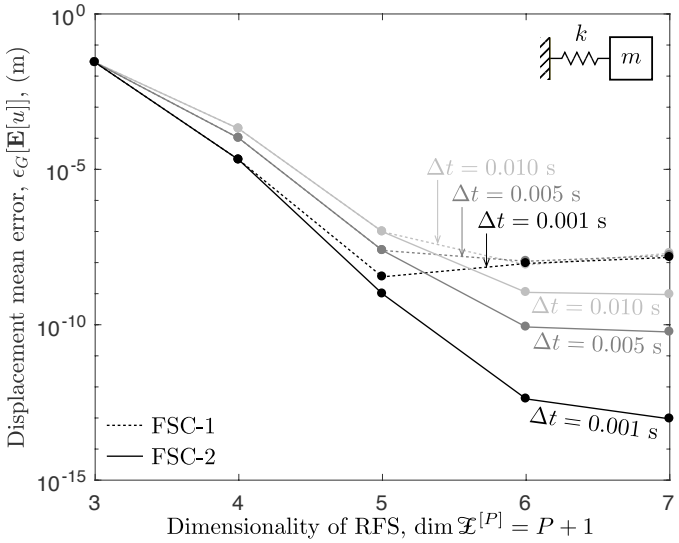


(a) Mean error

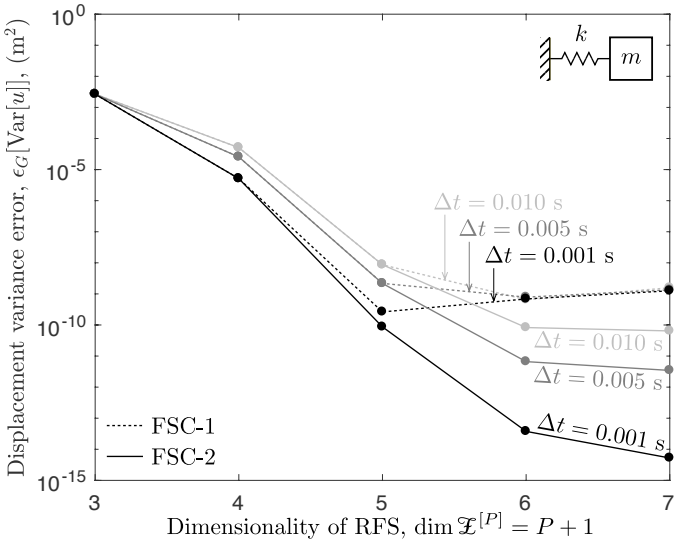


(b) Variance error

Figure 3.21. *Problem 5* — Global error of $\mathbf{E}[\partial_t^3 u]$ and $\mathrm{Var}[\partial_t^3 u]$ for different p-discretization levels of RFS

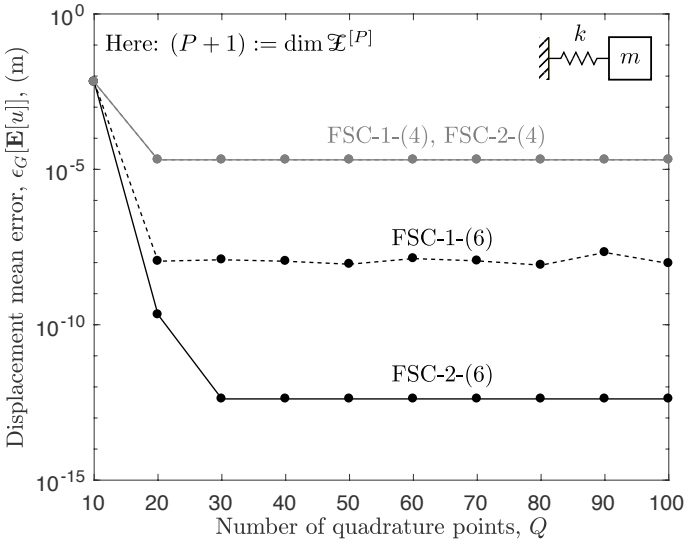


(a) Mean error

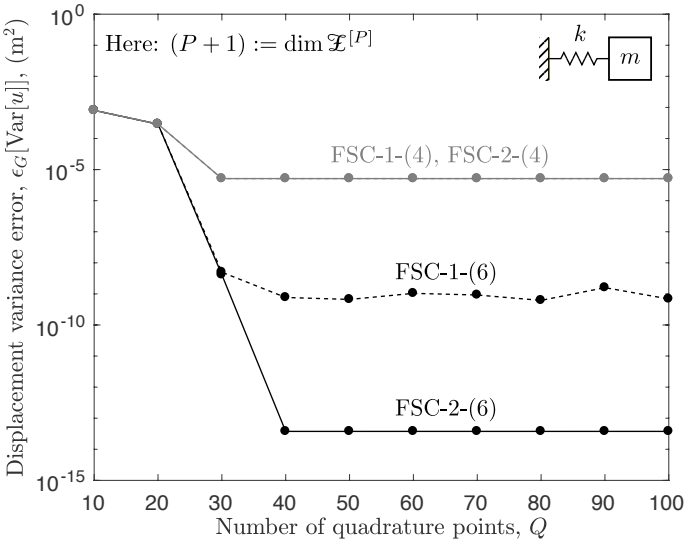


(b) Variance error

Figure 3.22. *Problem 2* — Global error of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different p-discretization levels of RFS and time-step sizes ($\mu \sim$ Uniform)



(a) Mean error



(b) Variance error

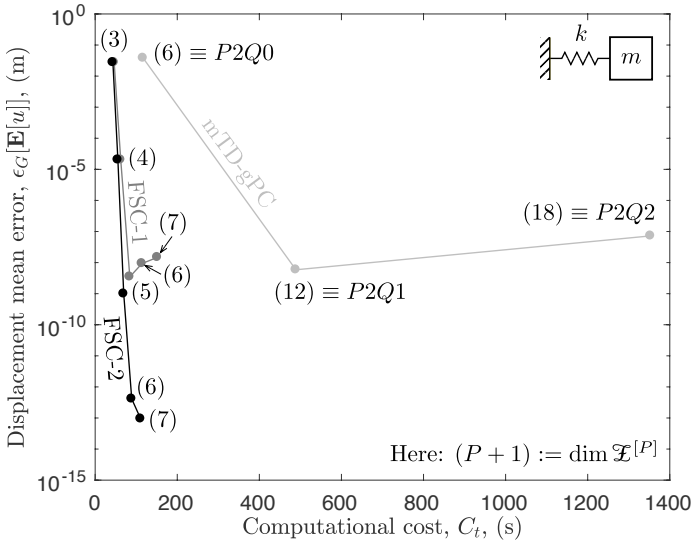
Figure 3.23. *Problem 2* — Global error of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ as a function of the number of quadrature points employed to estimate the inner products ($\mu \sim$ Uniform)

(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. 3.24 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.

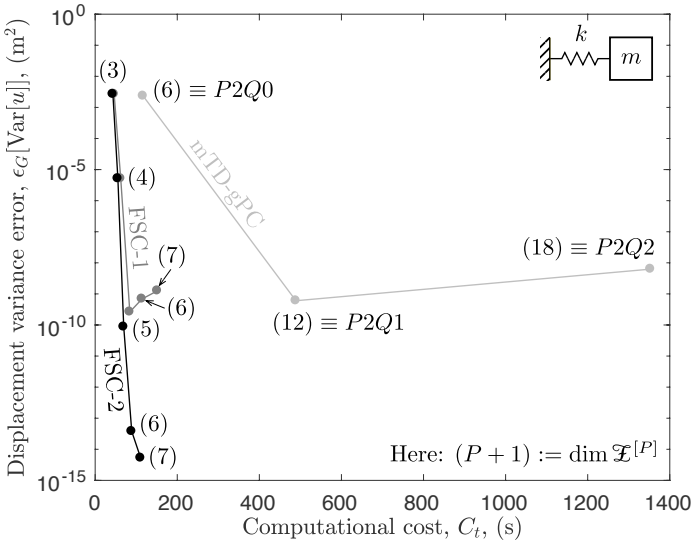
3.8.2. Numerical results for the nonlinear system

For this section we use the RK4 method with $\Delta t = 0.005$ s to speed up the simulations. Fig. 3.25 depicts the evolution of the mean and variance of the oscillator's displacement using FSC-2 and a Monte Carlo 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. 3.26 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

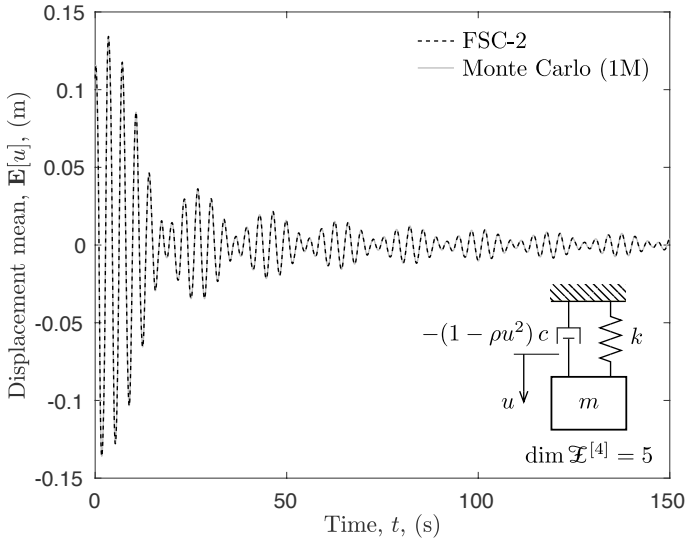


(a) Mean error

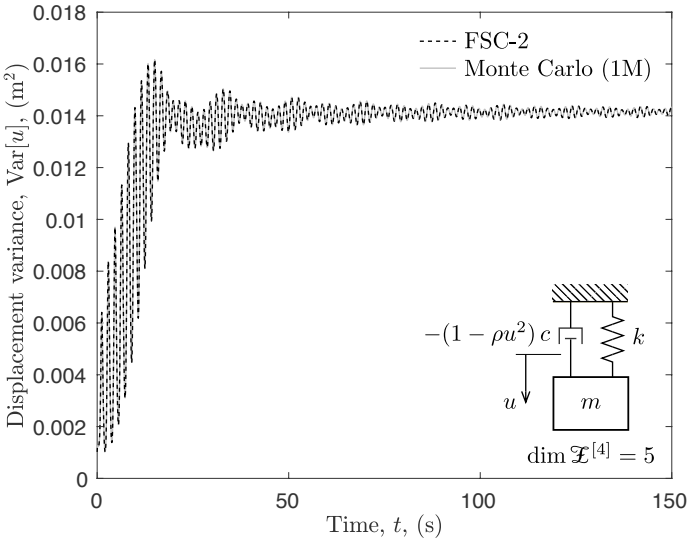


(b) Variance error

Figure 3.24. *Problem 2* — Global error of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ versus computational cost for different p-discretization levels of RFS and for $\mu \sim$ Uniform

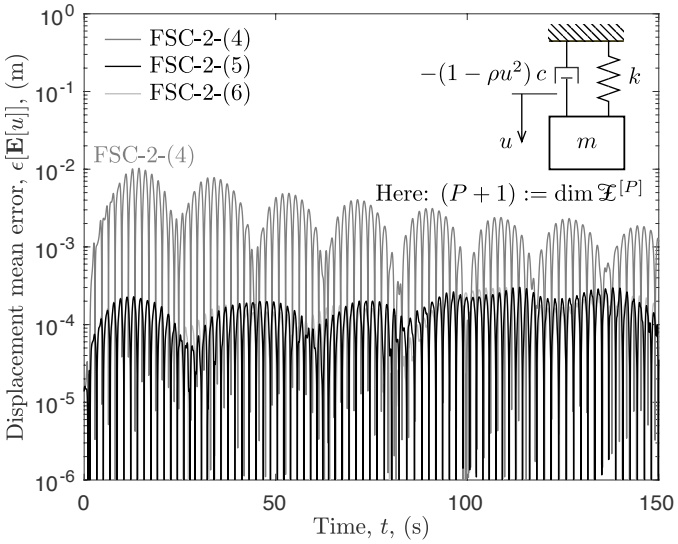


(a) Mean

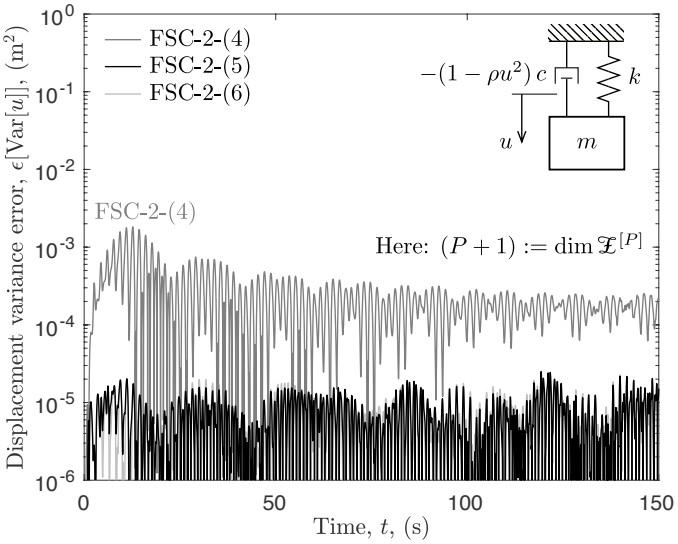


(b) Variance

Figure 3.25. *Problem 6 (the Van-der-Pol oscillator)* — Evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for the case when the p-discretization level of RFS is $\mathfrak{X}^{[4]}$ and $\mu \sim \text{Uniform} \otimes \text{Beta}$



(a) Mean error



(b) Variance error

Figure 3.26. *Problem 6 (the Van-der-Pol oscillator)* — Local error evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different p-discretization levels of RFS with respect to the 1-million Monte Carlo simulation ($\mu \sim \text{Uniform} \otimes \text{Beta}$)

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.

3.9. 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.

3.9.1. Problem statement

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

$$\ddot{u} + ku = f \qquad \text{on } \mathfrak{T} \times \Xi \tag{3.35a}$$

$$\Big\{u(0,\cdot) = \boldsymbol{u},\ \dot{u}(0,\cdot) = \boldsymbol{v}\Big\} \qquad \text{on } \{0\} \times \Xi, \tag{3.35b}$$

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

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

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 $\xi = (\xi^1, \ldots, \xi^d)$

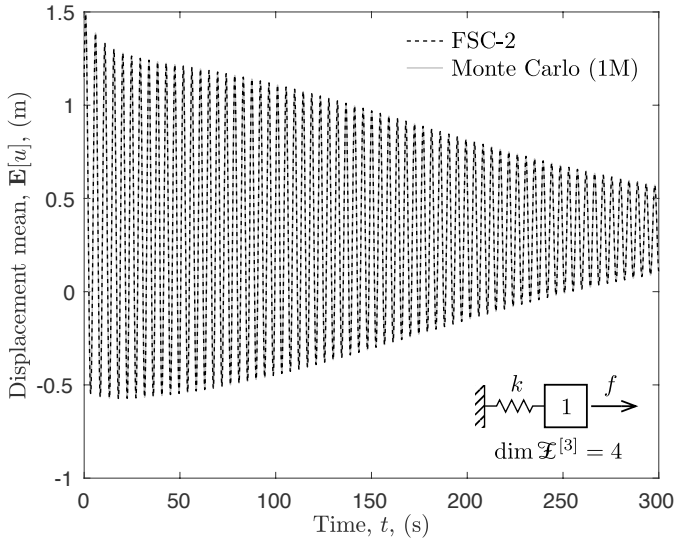
for a d-dimensional space. The spectral discretization of this problem is derived in detail in Appendix 3.B.

3.9.2. Discussion on numerical results

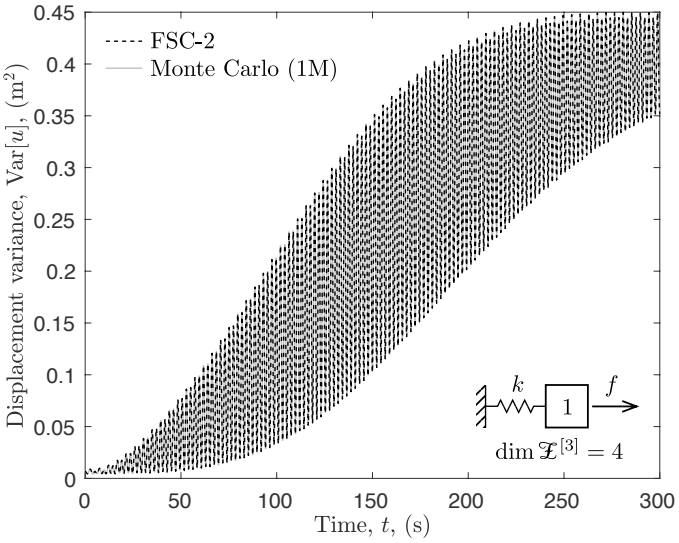
To run all the simulations, we employ the RK4 method with a time-step size of 0.01 s. In Table 3.2 we have provided the non-orthogonalized version of the random basis for use within the FSC scheme. Because the random space is in this case high-dimensional, the inner products are computed with a Monte 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 Figs. 3.27 to 3.29 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. $\mathfrak{L}^{[3]}$ 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 (3.4) would have indicated the need of using 66 basis vectors to enrich the random basis with quadratic functions. However, as we see in Figs. 3.27 to 3.29, 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

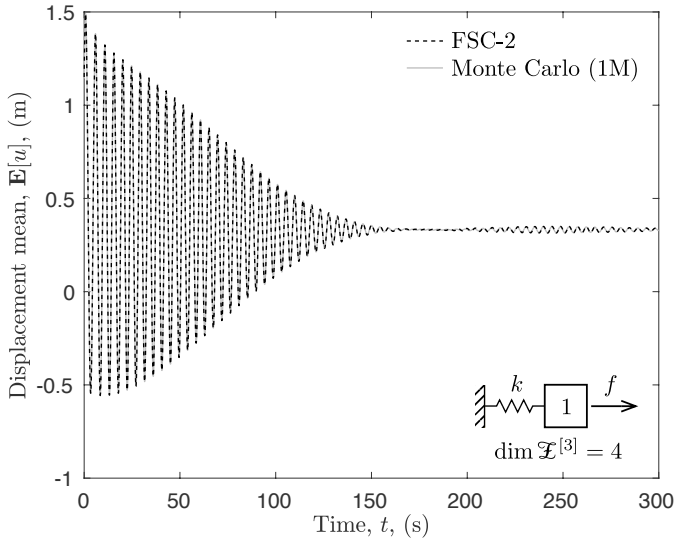


(a) Mean for $d = 5$

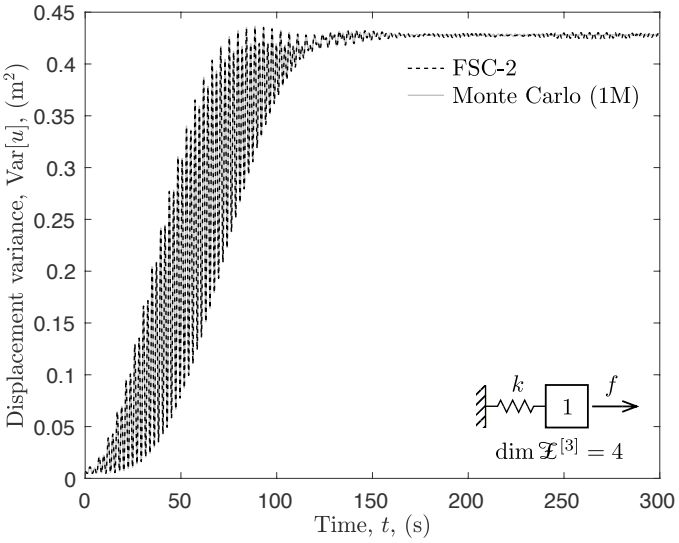


(b) Variance for $d = 5$

Figure 3.27. Evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for the case when the p-discretization level of RFS is $\mathfrak{X}^{[3]}$ and $\mu \sim \mathrm{Beta}^{\otimes 3} \otimes \mathrm{Uniform}^{\otimes d-3}$ (SET 1/3)

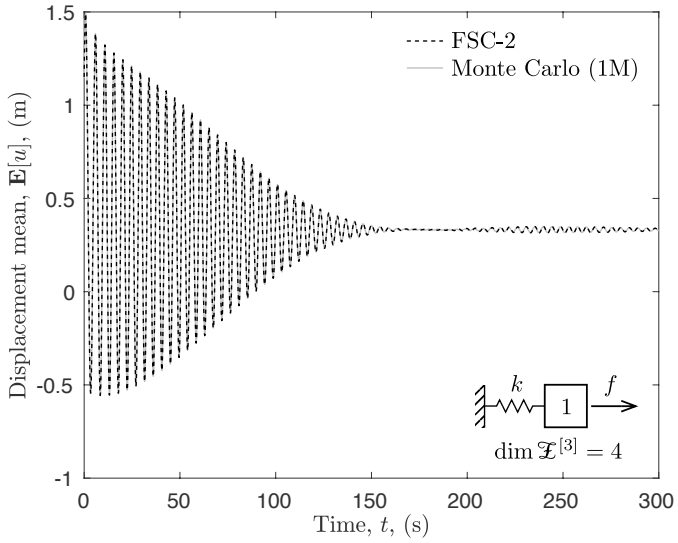


(a) Mean for $d = 7$

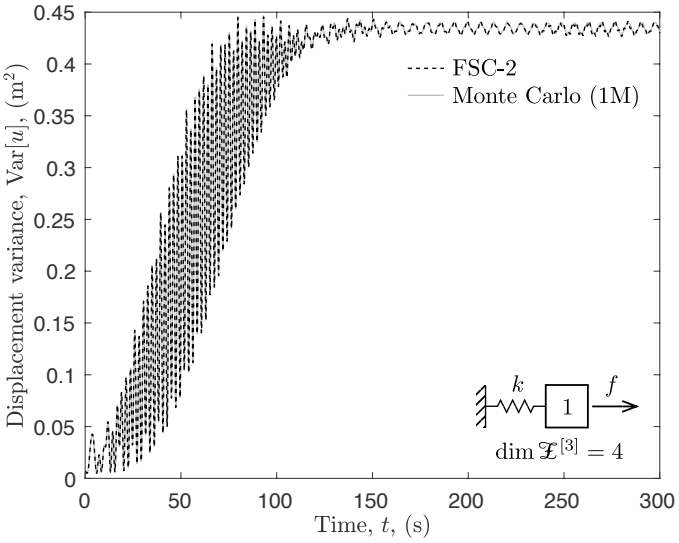


(b) Variance for $d = 7$

Figure 3.28. Evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for the case when the p-discretization level of RFS is $\mathfrak{X}^{[3]}$ and $\mu \sim \mathrm{Beta}^{\otimes 3} \otimes \mathrm{Uniform}^{\otimes d-3}$ (SET 2/3)



(a) Mean for $d = 10$



(b) Variance for $d = 10$

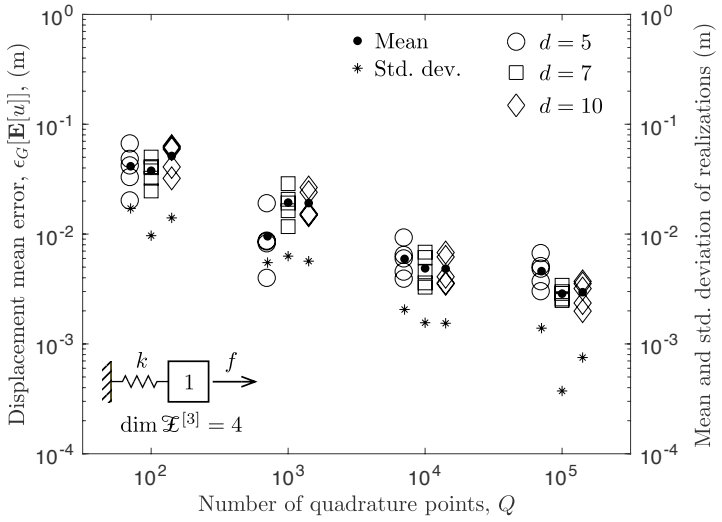
Figure 3.29. Evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for the case when the p-discretization level of RFS is $\mathfrak{X}^{[3]}$ and $\mu \sim \mathrm{Beta}^{\otimes 3} \otimes \mathrm{Uniform}^{\otimes d-3}$ (SET 3/3)

a Monte Carlo simulation with one million realizations. In Fig. 3.30 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.

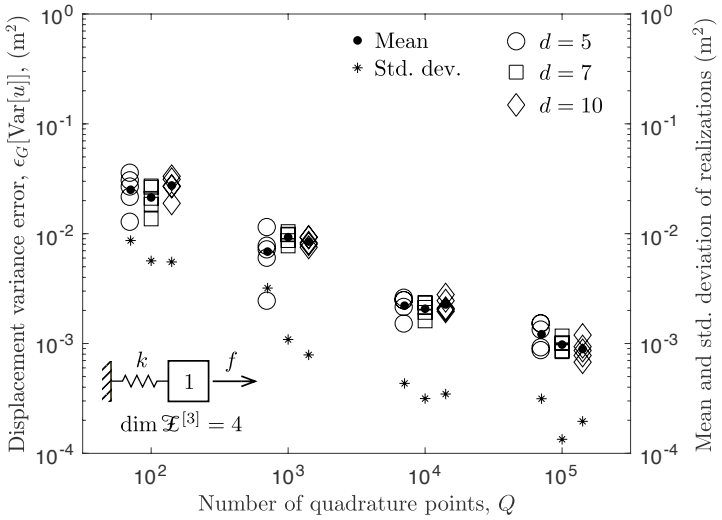
From this section it follows that if we combine the FSC method with a Monte Carlo integration technique, the curse of dimensionality can be fully annihilated at both random levels, namely: at the random-space level and at the random-function-space level. This makes the FSC method more suitable for solving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

3.10. 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.) 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.



(a) Mean error



(b) Variance error

Figure 3.30. Global error of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ as a function of the number of quadrature points employed to estimate the inner products with a Monte Carlo integration ($\mu \sim \mathrm{Beta}^{\otimes 3} \otimes \mathrm{Uniform}^{\otimes d-3}$)

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 3.8, 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 [33] and TD-gPC [34], 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 3.9 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.

3.A. The Van-der-Pol oscillator

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

The random function space for the Van-der-Pol oscillator described in Section 3.7.6 is discretized here for sake of reference. Let $\mathfrak{X}^{[P]}$ be a finite subspace of $\mathfrak{X}$. Then, it is clear that the oscillator's displacement can be represented in $\mathfrak{X}^{[P]}$ with:

$$u(t,\xi) \approx u^{[P]}(t,\xi) = \sum_{j=0}^{P} u^j(t)\,\Psi_j(\xi) \equiv u^j(t)\,\Psi_j(\xi). \tag{3.36}$$

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

$$c(\xi) = \sum_{m=0}^{2} c^m\,\tilde{\Psi}_m(\xi) \equiv c^m\,\tilde{\Psi}_m(\xi) \quad \text{and} \quad \mathcal{u}(\xi) = \sum_{m=0}^{2} \mathcal{u}^m\,\tilde{\Psi}_m(\xi) \equiv \mathcal{u}^m\,\tilde{\Psi}_m(\xi), \tag{3.37}$$

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

Remark 3.14. In this problem, $\dim \tilde{\mathfrak{X}} = 3$ because c and $\mathcal{u}$ are the only independent random variables in the model's input. In a more general setting, $\dim \tilde{\mathfrak{X}} = \tilde{N} + 1$, where $\tilde{N}$ represents the number of independent random variables specified in input x of mathematical model $\boldsymbol{\mathcal{M}}[u]$.

Replacing (3.36) and (3.37) into (3.34), and then projecting onto $\mathfrak{X}^{[P]}$, yields a system of $P+1$ second-order ordinary differential equations in the variable t (with initial conditions):

$$m\,\mathbb{M}^i{}_{j000}\ddot{u}^j - \mathbb{M}^i{}_{j00m}\dot{u}^j c^m + \rho\,\mathbb{M}^i{}_{jklm}\dot{u}^j u^k u^l c^m + k\,\mathbb{M}^i{}_{j000}u^j = 0 \qquad \text{on } \mathfrak{T} \tag{3.38a}$$

$$\left\{\mathbb{M}^i{}_{j000}\,u^j(0) = \mathbb{M}^i{}_{000m}\mathcal{u}^m,\ \mathbb{M}^i{}_{j000}\,\dot{u}^j(0) = 2\,\mathbb{M}^i{}_{000m}\mathcal{u}^m - 0.10\,\mathbb{M}^i{}_{0000}\right\} \qquad \text{on } \{0\}, \tag{3.38b}$$

where $\mathbb{M} = \mathbb{M}^{i}{}_{jklm}\Psi_i \otimes \Psi^j \otimes \Psi^k \otimes \Psi^l \otimes \tilde{\Psi}^m : \mathfrak{Z}^{[P]\prime} \times \mathfrak{Z}^{[P]} \times \mathfrak{Z}^{[P]} \times \mathfrak{Z}^{[P]} \times \tilde{\mathfrak{Z}} \to \mathbb{R}$ is the discretized *random tensor*[5] for the whole dynamical system (assumed to exist) given by (a summation sign is implied over every repeated index)

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

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

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

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

However, since $\mathbb{M}^{i}{}_{j000}$ is nothing but $\delta^{i}_{\ j}$ (the Kronecker delta), system (3.38) can be restated as follows:

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

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

Table 3.1 presents the non-orthogonalized version of the random basis that we use in the simulations with FSC over the region $\mathfrak{R}_i = \mathrm{cl}(\mathfrak{T}_i) \times \Xi$.

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

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

The random function space for the parametric, high-dimensional problem described in Section 3.9 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 3.A) to represent the entire stochasticity of the system at hand is not feasible.

[5] This tensor is also known as *multiplication tensor* in the literature, see e.g. [14, 15].

Table 3.1. Non-orthogonalized version of the random basis used for the Van-der-Pol oscillator (Section 3.7.6)

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

Instead, we derive below a collection of random tensors for each of the random variables that appear in (3.35) separately, namely: $\boldsymbol{k}(t)$, $\boldsymbol{f}(t)$, $\boldsymbol{u}$ and $\boldsymbol{v}$.

As in Appendix 3.A, we take $\mathfrak{Z}^{[P]}$ to be a finite subspace of $\mathfrak{Z}$, and the solution representation for (3.35) as that stipulated by (3.36). Therefore, replacing (3.36) into (3.35), and then projecting onto $\mathfrak{Z}^{[P]}$ gives

$$\ddot{u}^i + k^i{}_j u^j = f^i \qquad \text{on } \mathfrak{T} \tag{3.39a}$$

$$\left\{u^i(0) = u^i,\, \dot{u}^i(0) = v^i\right\} \qquad \text{on } \{0\}, \tag{3.39b}$$

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

$$k^i{}_j(t) = \langle \Psi_i, k(t,\cdot)\, \Psi_j \rangle / \langle \Psi_i, \Psi_i \rangle, \quad f^i(t) = \langle \Psi_i, f(t,\cdot) \rangle / \langle \Psi_i, \Psi_i \rangle,$$
$$u^i = \langle \Psi_i, u \rangle / \langle \Psi_i, \Psi_i \rangle \quad \text{and} \quad v^i = \langle \Psi_i, v \rangle / \langle \Psi_i, \Psi_i \rangle.$$

Remark 3.15. Note that the discretized random tensor associated with k is of type $(1,1)$, i.e. $\boldsymbol{k}(t) = k^i{}_j(t)\, \Psi_i \otimes \Psi^j : \mathfrak{Z}^{[P]\prime} \times \mathfrak{Z}^{[P]} \to \mathbb{R}$ (assumed to exist), whereas those associated with f, u and v are of type $(1,0)$, i.e. $\boldsymbol{f}(t) = f^i(t)\, \Psi_i, \boldsymbol{u} = u^i\, \Psi_i, \boldsymbol{v} = v^i\, \Psi_i : \mathfrak{Z}^{[P]\prime} \to \mathbb{R}$. Because of this, the last three random tensors can also be regarded as random vectors in $\mathfrak{Z}^{[P]}$, and so we have the following identification: $\boldsymbol{f}(t) \mapsto f(t,\cdot)$, $\boldsymbol{u} \mapsto u$ and $\boldsymbol{v} \mapsto v$.

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

Table 3.2 presents the non-orthogonalized version of the random basis that we use in the simulations with FSC over the region $\mathfrak{R}_i = \mathrm{cl}(\mathfrak{T}_i) \times \Xi$.

3.C. Time-complexity analysis for Theorem 3.1 and the traditional Gram-Schmidt process

In this last section we derive the number of elementary operations[6] that we need to perform in order to obtain an orthogonal basis from a set of linearly independent random

[6]By elementary operations we mean: addition, subtraction, multiplication and division on the real numbers.

Table 3.2. Non-orthogonalized version of the random basis used for the d-dimensional stochastic problem (Section 3.9)

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

functions. We do this for Theorem 3.1 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 3.1 In Table 3.3 we have disaggregated the number of elementary operations into the different steps involved in Theorem 3.1. If we assume that both the expectation vector and the covariance matrix are known beforehand, the number of operations needed in Theorem 3.1 is

$$N_{\text{op}} = \sum_{i=3}^{5} N_{\text{op},i} = \begin{cases} 0 & \text{for } P = 0 \\ P(P+1)Q + \frac{1}{30}P^5 + \frac{1}{6}P^4 - \frac{1}{3}P^3 + \frac{1}{3}P^2 - \frac{6}{5}P + 1 & \text{for } P \geq 1. \end{cases} \tag{3.40}$$

Otherwise, the number of operations needed is

$$N_{\text{op}} = \sum_{i=1}^{5} N_{\text{op},i} = \begin{cases} 0 & \text{for } P = 0 \\ \frac{7}{2}P(P + \frac{11}{7})Q + \frac{1}{30}P^5 + \frac{1}{6}P^4 - \frac{1}{3}P^3 - \frac{1}{6}P^2 - \frac{27}{10}P + 1 & \text{for } P \geq 1. \end{cases} \tag{3.41}$$

Traditional Gram-Schmidt process As with Theorem 3.1, in Table 3.4 we have disaggregated into steps the number of elementary operations needed in the traditional Gram-Schmidt process (from (3.11)), yielding

$$N'_{\text{op}} = \sum_{i=1}^{3} N'_{\text{op},i} = \begin{cases} 0 & \text{for } P = 0 \\ \frac{5}{2}(P^2 + \frac{9}{5}P - \frac{6}{5})Q - 2P + 1 & \text{for } P \geq 1. \end{cases} \tag{3.42}$$

Performance comparison Combining the result obtained in (3.40) with that in (3.42) produces the plot depicted in Fig. 3.31. From this figure we can see that as the number of quadrature points increases, the time complexity of Theorem 3.1 decreases relative to that of the traditional Gram-Schmidt process. In other words, when the number of quadrature points

Table 3.3. Number of elementary operations involved in each step of Theorem 3.1 to get $\{\Phi_j\}_{j=1}^{P} \mapsto \{\Psi_j\}_{j=0}^{P}$

Theorem 1:

1. Compute mean vector

   $N_{\text{op},1} := \#\big(\{\mathbf{E}[\Phi_j]\}_{j=1}^{P}\big) = P(2Q-1) \quad \text{for } P \geq 1$

2. Compute covariance matrix

   $N_{\text{op},2} := \#\big(\{\text{Cov}[\Phi_i, \Phi_j]\}_{i,j=1}^{P}\big) = \frac{1}{2}P(P+1)(5Q-1) \quad \text{for } P \geq 1$

3. Compute the numerators of determinant ratio

   $N_{\text{op},3} := \#\big(\{\det \triangle_k(j)\}_{j=2,k=1}^{P,j-1}\big) = \frac{1}{30}P(P-1)(P-2)(P^2+3P+7) \quad \text{for } P \geq 2$

4. Compute the denominators of determinant ratio

   $N_{\text{op},4} := \#\big(\{\det \square_k\}_{k=1}^{P-1}\big) = \frac{1}{6}(P-1)(P-2)(P^2+P+3) \quad \text{for } P \geq 2$

5. Compute orthogonal basis

   $N_{\text{op},5} := \#\big(\{\Psi_j\}_{j=0}^{P}\big) = \frac{1}{2}P\big((P+1)(2Q+1)-2\big) \quad \text{for } P \geq 1$

Remark:

Here: $\mathbf{A} \in \mathscr{M}(n \times n, \mathbb{R})$, and $\mathbf{LU}$ is the LU-decomposition of $\mathbf{A}$

- Computation of mean

  $$\#\left(\mathbf{E}[f] \approx \sum_{i=1}^{Q} f(\xi_i)\, w_i\right) = 2Q-1$$

- Computation of covariance

  $$\#\left(\text{Cov}[f,g] \approx \sum_{i=1}^{Q} \big(f(\xi_i) - \mathbf{E}[f]\big)\big(g(\xi_i) - \mathbf{E}[g]\big)\, w_i\right) = 5Q-1$$

- Computation of determinant

  $$\#\big(\det \mathbf{A} = \det(\mathbf{LU}) = \det \mathbf{U}\big) = 2\sum_{i=1}^{n}(n-i)(n-i+1) + n - 1 = \tfrac{1}{3}(2n^3+n-3)$$

Table 3.4. Number of elementary operations involved in each step of the traditional Gram-Schmidt process to get $\{\Phi_j\}_{j=1}^{P} \mapsto \{\Psi_j\}_{j=0}^{P}$

Traditional Gram-Schmidt process:

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

is sufficiently large, Theorem 3.1 surpasses in efficiency the traditional Gram-Schmidt process. In particular, if $P = 3, 4$, Theorem 3.1 is up to 2.75 times faster. However, if the expectation vector or the covariance matrix are not known beforehand, Theorem 3.1 is slower than the traditional Gram-Schmidt process, because the Q-coefficient in (3.41) is always larger than that in (3.42) for all values of P.

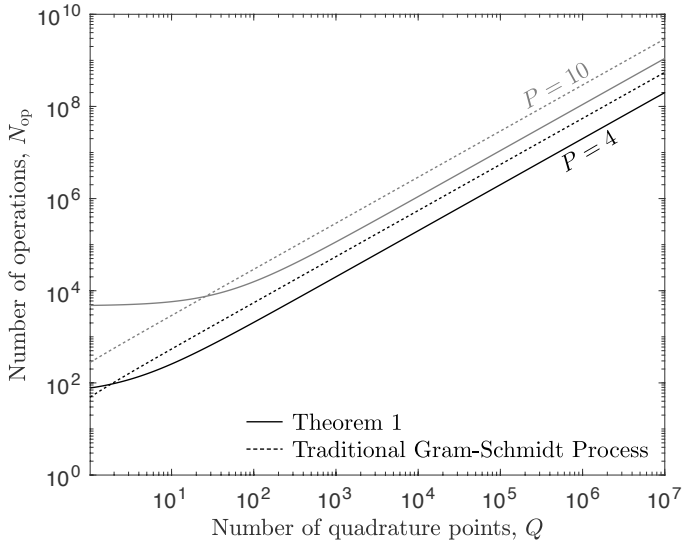


Figure 3.31. Time-complexity analysis for Theorem 3.1 and the traditional Gram-Schmidt process

4. MULTI-ELEMENT FLOW-DRIVEN SPECTRAL CHAOS (ME-FSC) METHOD FOR UNCERTAINTY QUANTIFICATION OF DYNAMICAL SYSTEMS

4.1. Introduction

In the past few decades, the spectral approach has gained increasing popularity among researchers as a powerful tool to solve stochastic problems at low computational cost. This assertion is especially true when it comes to solving problems where the dimensionality of the probability space is relatively low. However, for problems where the dimensionality of the probability space is relatively high, such an assertion may not always hold, especially in cases where the curse of dimensionality at the random-function-space (RFS) level cannot be alleviated notably within the numerical scheme of the method. This is, for instance, the case of gPC-based methods (e.g. [33–37, 43]) where the curse of dimensionality at the RFS level appears to arise naturally within the numerical scheme of the method. That is, when the random input consists of mutually independent random variables, the stochastic part of the solution space (aka random function space or RFS) is spanned via a tensor product of vector spaces with each of these vector spaces representing a space spanned by a finite set of univariate orthogonal polynomials[1]. However, such a construction of the RFS can only ensure exponential convergence to the solution at early times of the simulation if the RFS is not kept updated frequently during the simulation.

To further illustrate the curse of dimensionality in gPC-based methods, let us consider the gPC method [33] and the solution of a stochastic dynamical system featuring a random input with d mutually independent random variables. In gPC, the RFS is constructed via a tensor product of vector spaces. Each vector space is spanned using a finite set of univariate orthogonal polynomials which are taken from the Askey family and are concordant with the probability measure. The cardinality of each of these random bases is determined by the highest polynomial degree p in the set, giving thereby $p+1$ basis vectors in each vector space.

[1]These polynomials are made orthogonal with respect to the measure defined in the probability space for computational efficiency.

In this sense, if a full tensor product were to be adopted in the gPC scheme, $(p+1)^d$ basis vectors would be needed, and thus, in a numerial scheme based on full tensor products the cardinality of the random basis grows exponentially fast as a function of d and p. Moreover, if a total-order tensor product were to be adopted, $\binom{d+p}{p}$ basis vectors would be needed instead. In this case, the cardinality of the random basis grows combinatorially fast as a function of d and p, alleviating the curse of dimensionality at the RFS level but not solving it completely.

The flow-driven spectral chaos (FSC) method [92] is a new numerical technique recently developed by the authors with the goal of tackling the long-time integration issue found in the gPC method using the spectral approach. The method uses the concept of *enriched* stochastic flow maps to track the evolution of a finite dimensional RFS efficiently in time. In FSC, the dimensionality of the random phase space is deliberately increased to allow both the system's state and its first few time derivatives to be pushed forward over time. Then, the enriched state of the system is used as a germ to construct a suitable RFS for use within the current time step of the simulation. It is worth mentioning that for dynamical systems with an n-tuple state and driven by a stochastic flow map of order M, the cardinality of the random basis is bounded from above by $n+M+1$. This boundedness from above is what makes the FSC method be curse-of-dimensionality free at the RFS level, even when the probability space is high-dimensional.

In this paper, we present a multi-element version of the FSC method to deal with stochastic discontinuities and long-time integration of stochastic dynamical systems more efficiently. This new technique is called the *multi-element flow-driven spectral chaos* (ME-FSC) method. In ME-FSC, the random domain is partitioned into several elements, and then the problem is solved separately on each random element using the FSC method. Then, the results are aggregated to compute the mean and variance of the response with the law of total probability. This approach is similar to the multi-element gPC (ME-gPC) method [43], with the only difference being that here the gPC method is not employed on each random element in order to annihilate completely the curse of dimensionality at the RFS level. The benefit of using ME-FSC is threefold. First, the simulation can be run simultaneously on machines with multiple CPU cores (or if needed on separate machines) to reduce the excessive computational burden associated with the simulation. Second, if the random input is discontinuous over the

probability space, the random domain can be partitioned into several elements to assure that at most the discontinuity will only appear on regions of measure zero. Third, if an adaptive criterium is introduced within the ME-FSC scheme (so as to allow the elements to get smaller on-the-fly whenever a threshold value is exceeded), the errors can be kept to a minimum during the simulation.

This paper has been structured in the following way. The setting and notation used in this manuscript is formally introduced in Section 4.2, and then the problem statement we are interested in is presented in Section 4.3. A quick overview of the concept associated with 'enriched stochastic flow maps' is provided in Section 4.4 to allow the construction of an optimal, low-dimensional RFS during the simulation. Subsequently, in Section 4.5 the ME-FSC method is presented in great detail along with an outline of the numerical scheme used in this work. Finally, in Section 4.6 four numerical examples are explored to test the performance of the ME-FSC method and document the findings.

4.2. Setting and notation

4.2.1. Basic spaces needed in the ME-FSC method

In this paper, we consider Definitions 1 to 5 presented in a previous work by the authors [92]. Because of their importance in the development of the ME-FSC method, they are also presented here.

Definition 4.1 (Temporal space)**.** Let $(\mathfrak{T}, \mathcal{O})$ be a topological space, where $\mathfrak{T} = [0, T]$ is a closed interval representing the temporal domain of the system, $\mathcal{O} = \mathcal{O}_{\mathbb{R}} \cap \mathfrak{T}$ is the topology on $\mathfrak{T}$ with $\mathcal{O}_{\mathbb{R}}$ denoting the standard topology associated with $\mathbb{R}$, and $T \in \mathbb{R}^+$ symbolizes the duration of the simulation. In this paper, such a space is called *temporal space* of the system.

Definition 4.2 (Random space)**.** Let $(\Omega, \boldsymbol{\Omega}, \lambda)$ be a (complete) probability space, where Ω is the sample space, $\boldsymbol{\Omega} \subset 2^{\Omega}$ is the σ-algebra on Ω, and $\lambda : \boldsymbol{\Omega} \to [0, 1]$ is the probability measure on $\boldsymbol{\Omega}$. Let $\xi = (\xi^1, \ldots, \xi^d) : (\Omega, \boldsymbol{\Omega}) \to (\mathbb{R}^d, \mathcal{B}_{\mathbb{R}^d})$ be a measurable function given by $\xi = \xi(\omega)$, with $\mathcal{B}_{\mathbb{R}^d}$ standing for the Borel σ-algebra associated with $\mathbb{R}^d$. In this work, the measure space $(\Xi, \boldsymbol{\Xi}, \mu)$ is termed *random space*, where $\Xi = \xi(\Omega) \subset \mathbb{R}^d$ is a set representing

the random domain of the system, $\boldsymbol{\Xi} = \mathcal{B}_{\mathbb{R}^d} \cap \Xi$ is the σ-algebra on Ξ, $\mu : \boldsymbol{\Xi} \to [0, 1]$ is the probability measure on $\boldsymbol{\Xi}$ given by $\mu = \xi_*(\lambda) := \lambda \circ \xi^{-1}$ (i.e. the pushforward of λ by ξ), and d symbolizes the dimensionality of the random space.

Fig. 4.1 depicts the relationship between probability space and random space for $d = 2$, and a case scenario in which the random space has been partitioned into 9 random elements for sake of illustration. As shown, we will always assume that the components of $\xi = (\xi^1, \ldots, \xi^d)$ are mutually independent and that the random domain Ξ is a hypercube of d dimensions obtained by performing a d-fold Cartesian product of intervals $\bar{\Xi}_i := \xi^i(\Omega)$.

Definition 4.3 (Temporal function space)**.** Let $\mathcal{T}(n) = C^n(\mathfrak{T}; \mathbb{R})$ be a continuous n-differentiable function space. This is the space of all function $f : (\mathfrak{T}, \mathcal{O}) \to (\mathbb{R}, \mathcal{O}_{\mathbb{R}})$ that have continuous first n derivatives on $(\mathfrak{T}, \mathcal{O})$. In this paper, such a space is called *temporal function space*.

Definition 4.4 (Random function space)**.** Let $\mathfrak{X} = (L^2(\Xi, \boldsymbol{\Xi}, \mu; \mathbb{R}), \langle \cdot, \cdot \rangle)$ be a Lebesgue square-integrable space equipped with its standard inner product

$$\langle \cdot, \cdot \rangle : L^2(\Xi, \boldsymbol{\Xi}, \mu; \mathbb{R}) \times L^2(\Xi, \boldsymbol{\Xi}, \mu; \mathbb{R}) \to \mathbb{R} \qquad :\Leftrightarrow \qquad \langle f, g \rangle = \int fg \, \mathrm{d}\mu.$$

This is the space of all (equivalence classes of) measurable functions $f : (\Xi, \boldsymbol{\Xi}) \to (\mathbb{R}, \mathcal{B}_{\mathbb{R}})$ that are square μ-integrable on Ξ. In this paper, such a space receives the name of *random function space* or RFS for short. Also, we define $\{\Psi_j : (\Xi, \boldsymbol{\Xi}) \to (\mathbb{R}, \mathcal{B}_{\mathbb{R}})\}_{j=0}^{\infty}$ to be a complete orthogonal basis in $\mathfrak{X}$, such that $\Psi_0(\xi) = 1$ for all $\xi \in \Xi$.

Remark 4.1. The following are four remarks regarding Definition 4.4.

1. It is well-known that $\mathfrak{X}$ forms a Hilbert space because it is complete under the metric induced by $\langle \cdot, \cdot \rangle$.

2. If $f \in \mathfrak{X}$, then it can be represented by the Fourier series:

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

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

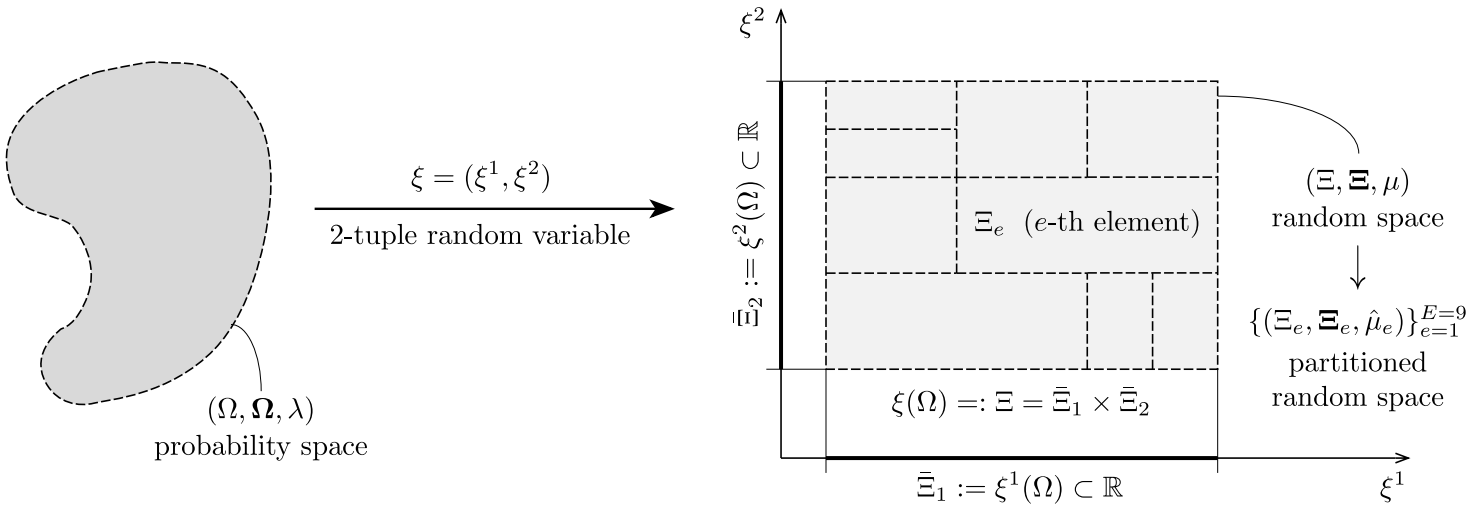


Figure 4.1. Relationship between probability space and random space for $d = 2$ (along with a case partition of the random space)

3. The orthogonality property of the basis in $\mathfrak{Z}$ means that

$$\langle \Psi_i, \Psi_j \rangle := \int \Psi_i \Psi_j \, \mathrm{d}\mu = \langle \Psi_i, \Psi_i \rangle \, \delta_{ij},$$

where δ_{ij} is the Kronecker delta.

4. The dual space of $\mathfrak{Z}$, which we denote by $\mathfrak{Z}'$ herein, is the space spanned by the continuous linear functionals $\{\Psi^i : \mathfrak{Z} \to \mathbb{R}\}_{i=0}^{\infty}$ defined by

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

Definition 4.5 (Solution space and root space)**.** Let $\mathscr{U} = \mathscr{T}(n) \otimes \mathfrak{Z}$ and $\mathscr{V} = \mathscr{T}(0) \otimes \mathfrak{Z}$ denote the *solution space* and the *root space*, respectively. In this sense, $\mathscr{U}$ and $\mathscr{V}$ are two spaces constructed via a tensor product of vector spaces.

4.2.2. Additional spaces needed in the ME-FSC method

We now present some extra definitions needed in the development of the ME-FSC method.

Definition 4.6 (Partitioned random space)**.** Let $\{\Xi_e\}_{e=1}^{E}$ be a partition of the random domain, where $\Xi_e \neq \emptyset$ represents the e-th element of the partition, and $E \in \mathbb{N}_2$ is the number of random elements employed in the partition. Then, we define $\boldsymbol{\Xi}_e = \boldsymbol{\Xi} \cap \Xi_e$ to be the σ-algebra on Ξ_e, and $\mu_e = \mu|_{\boldsymbol{\Xi}_e}$ to be the restriction of μ to $\boldsymbol{\Xi}_e$. Also, we define $\hat{\mu}_e : \boldsymbol{\Xi}_e \to [0, 1]$ to be the probability measure on $\boldsymbol{\Xi}_e$ given by

$$\hat{\mu}_e = \frac{\mu_e}{\mu(\Xi_e)}, \quad \text{or equivalently,} \quad \mathrm{d}\hat{\mu}_e = \frac{\mathrm{d}\mu_e}{\mu(\Xi_e)}.$$

Therefore, in this work the *partitioned random space* is defined as a finite sequence of disjoint random spaces $\{(\Xi_e, \boldsymbol{\Xi}_e, \hat{\mu}_e)\}_{e=1}^{E}$. In Fig. 4.1 we show a prototype depiction of such a partition.

Definition 4.7 (Partitioned random function space)**.** If $\{(\Xi_e, \boldsymbol{\Xi}_e, \hat{\mu}_e)\}_{e=1}^{E}$ is a partition of the random space, then let $\{\mathfrak{Z}_e\}_{e=1}^{E}$ be its associated *partitioned random function space*, where $\mathfrak{Z}_e$

represents the e-th RFS for the random element Ξ_e. That is, $\mathfrak{Z}_e = (L^2(\Xi_e, \boldsymbol{\Xi}_e, \hat{\mu}_e; \mathbb{R}); \langle \cdot, \cdot \rangle_e)$, where

$$\langle \cdot, \cdot \rangle_e : L^2(\Xi_e, \boldsymbol{\Xi}_e, \hat{\mu}_e; \mathbb{R}) \times L^2(\Xi_e, \boldsymbol{\Xi}_e, \hat{\mu}_e; \mathbb{R}) \to \mathbb{R} \qquad :\Leftrightarrow \qquad \langle f, g \rangle_e = \int f g \, \mathrm{d}\hat{\mu}_e.$$

This is the space of all (equivalence classes of) measurable functions $f : (\Xi_e, \boldsymbol{\Xi}_e) \to (\mathbb{R}, \mathcal{B}_{\mathbb{R}})$ that are square $\hat{\mu}_e$-integrable on Ξ_e. As with Definition 4.4, the complete orthogonal basis in $\mathfrak{Z}_e$, $\{\Psi_{j.e} : (\Xi_e, \boldsymbol{\Xi}_e) \to (\mathbb{R}, \mathcal{B}_{\mathbb{R}})\}_{j=0}^{\infty}$, is defined such that $\Psi_{0.e}(\xi) = 1$ for all $\xi \in \Xi_e$.

Remark 4.2. Note that Remark 4.1 is also applicable for each of the $\mathfrak{Z}_e$'s mentioned above, provided that the following changes are made:

$$\mathfrak{Z} \mapsto \mathfrak{Z}_e, \quad \mu \mapsto \hat{\mu}_e, \quad \langle \cdot, \cdot \rangle \mapsto \langle \cdot, \cdot \rangle_e, \quad f \mapsto f_{.e}, \quad f^k \mapsto f^k{}_{.e},$$
$$\Psi_k \mapsto \Psi_{k.e} \quad \text{and} \quad \Psi^k \mapsto \Psi^k{}_{.e}$$

with k symbolizing i or j.

4.3. Problem statement

In this work, we consider the same stochastic problem described in [92]. Namely:

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

$$\mathcal{L}[u] = f \qquad \text{on } \mathfrak{T} \times \Xi \tag{4.1a}$$
$$\left\{\mathcal{B}_k[u](0, \cdot) = b_k\right\}_{k=1}^{n} \qquad \text{on } \{0\} \times \Xi, \tag{4.1b}$$

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

However, for illustration purposes, we very often consider the much simpler case[2] when the mathematical model of the system can be represented (μ-a.e.) by an undamped oscillator under the action of an external force:

$$\mathcal{L}[u] = f \qquad :\Leftrightarrow \qquad m\ddot{u} + ku = p \tag{4.1a*}$$

$$\Big\{\mathcal{B}_k[u](0,\cdot\,) = b_k\Big\}_{k=1}^{n=2} \qquad :\Leftrightarrow \qquad \Big\{u(0,\cdot\,) = \boldsymbol{u},\ \dot{u}(0,\cdot\,) = \boldsymbol{v}\Big\}, \tag{4.1b*}$$

where $m : \Xi \to \mathbb{R}^+$ is the mass of the system, $k : \Xi \to \mathbb{R}^+$ is the stiffness of the system, $p : \mathfrak{T} \times \Xi \to \mathbb{R}$ is the external force applied to the system, and $\boldsymbol{u}, \boldsymbol{v} : \Xi \to \mathbb{R}$ are the prescribed initial conditions of the system. Observe here that $\dot{u} := \partial_t u$ and $\ddot{u} := \partial_t^2 u$ are the first and second partial derivatives of u with respect to time. We assume that $m, k, \boldsymbol{u}, \boldsymbol{v} \in \mathfrak{L}$, $p \in \mathscr{V}$ and of course that $u \in \mathscr{U}$.

4.4. (Enriched) stochastic flow map

We now turn our attention to the two flow maps considered in this manuscript. For a generalization of these two flow maps, please refer to [92].

4.4.1. Stochastic flow map

If we assume that the stochastic system given by (4.1*) is sufficiently regular over $\mathfrak{T} \times \Xi$, we get:

$$\partial_t^2 u(t,\xi) := \boldsymbol{\ell}(t,\xi,s(t,\xi)) = \bar{p}(t,\xi) - \bar{k}(\xi)\, u(t,\xi) \qquad \text{on } \mathfrak{T} \times \Xi \tag{4.2a}$$

$$\Big\{u(0,\xi) = \boldsymbol{u}(\xi),\ \dot{u}(0,\xi) = \boldsymbol{v}(\xi)\Big\} \qquad \text{on } \{0\} \times \Xi, \tag{4.2b}$$

where $s = (u, \dot{u}) \in \prod_{j=1}^{2} \mathscr{T}(3-j) \otimes \mathfrak{L}$ is the configuration state of the system over $\mathfrak{T} \times \Xi$, $\boldsymbol{\ell} : \mathfrak{T} \times \Xi \times \mathbb{R}^2 \to \mathbb{R}$ is a noisy, non-autonomous function (which can also be regarded as a function in $\mathscr{V}$) defining the response $\ddot{u} = \partial_t^2 u$, $\bar{p} = p/m : \mathfrak{T} \times \Xi \to \mathbb{R}$ is the external force applied to the system per unit mass, and $\bar{k} = k/m : \Xi \to \mathbb{R}^+$ is the stiffness of the system per

[2]Nonetheless, we emphasize that this work has been generalized to deal with problems as complicated as (4.1) in Section 4.5.

unit mass. In the context of (4.1*), by 'sufficiently regular over $\mathfrak{T} \times \Xi$' we mean that $\bar{p} \in \mathscr{V}$ and $\bar{k} \in \mathscr{Z}$ (and thus, $\boldsymbol{\ell} \in \mathscr{V}$).

If we further assume that the solution is analytic on $\mathfrak{T}$ for all $\xi \in \Xi$, then one can write u using the Taylor series:

$$u(t_i + h, \xi) = \sum_{j=0}^{\infty} \frac{h^j}{j!} \partial_t^j u(t_i, \xi) = \sum_{j=0}^{M} \frac{h^j}{j!} \partial_t^j u(t_i, \xi) + O(h^{M+1})(\xi),$$

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.

Hence, if each of the assumptions mentioned above holds, a (local) stochastic flow map of order M, $\varphi(M) : \mathbb{R} \times \mathscr{Z}^2 \to \mathscr{Z}^2$, can be provided for (4.2) to be:

$$\varphi(M)(h, s(t_i, \cdot)) := \Big(u(t_i + h, \cdot), \dot{u}(t_i + h, \cdot)\Big) - O(h^{M+1}), \tag{4.3}$$

where $\varphi^1(M), \varphi^2(M) : \mathbb{R} \times \mathscr{Z}^2 \to \mathscr{Z}$ are given by:

$$\begin{aligned}
\varphi^1(M)(h, s(t_i, \cdot)) &:= u(t_i, \cdot) + h\, \dot{u}(t_i, \cdot) + \tfrac{1}{2} h^2\, \partial_t^2 u(t_i, \cdot) + \cdots + \tfrac{1}{M!} h^M\, \partial_t^M u(t_i, \cdot) \\
\varphi^2(M)(h, s(t_i, \cdot)) &:= \dot{u}(t_i, \cdot) + h\, \partial_t^2 u(t_i, \cdot) + \tfrac{1}{2} h^2\, \partial_t^3 u(t_i, \cdot) + \cdots + \tfrac{1}{M!} h^M\, \partial_t^{M+1} u(t_i, \cdot).
\end{aligned}$$

It is worth mentioning that the second time derivative $\partial_t^2 u(t_i, \cdot)$ is computed using (4.2a), and that the next time derivatives $\{\partial_t^{j+2} u(t_i, \cdot)\}_{j=1}^{M-1}$ are computed using the recursive expression:

$$\partial_t^{j+2} u(t, \xi) := \mathrm{D}_t^j \boldsymbol{\ell}(t, \xi, s(t, \xi)) = \partial_t^j \bar{p}(t, \xi) - \bar{k}(\xi)\, \partial_t^j u(t, \xi) \qquad \forall j \in \{1, 2, \ldots, M-1\}. \tag{4.4}$$

In simple terms, the goal of $\varphi(M)$ is to push the system's state $s(t_i, \cdot) = (u(t_i, \cdot), \dot{u}(t_i, \cdot))$ over time. Note that the push is *forward* if $h > 0$, it is *backward* if $h < 0$, and it is *still* if $h = 0$.

4.4.2. Enriched stochastic flow map

The goal of the associated enriched flow map, $\hat{\varphi}(M) : \mathbb{R} \times \mathscr{Z}^{M+2} \to \mathscr{Z}^{M+2}$, is to push not only the system's state $s(t_i, \cdot) = (u(t_i, \cdot), \dot{u}(t_i, \cdot))$ over time, but also $\boldsymbol{\ell}(t_i, \cdot, s(t_i, \cdot))$ and its

first $M-1$ time derivatives at $t=t_i$. This is the reason why the k-th component of $\hat{\varphi}(M)$ is given by:

$$\hat{\varphi}^k(M)(h,\hat{s}(t_i,\cdot)) =: \hat{s}^k(t_i+h,\cdot) = \begin{cases} \varphi^k(M)(h,s(t_i,\cdot)) & \text{for } k\in\{1,2\} \\ \mathrm{D}_t^{k-3}\boldsymbol{\ell}(t_i+h,\cdot,s(t_i+h,\cdot)) & \text{for } k\in\{3,4,\ldots,M+2\} \end{cases}$$

where $\hat{s}=(u,\partial_t u,\ldots,\partial_t^{M+1}u)\in\prod_{j=1}^{M+2}\mathcal{T}(M-j+2)\otimes\mathfrak{X}$ is the enriched configuration state of the system over $\mathfrak{T}\times\Xi$. Note here that $\mathrm{D}_t^0\boldsymbol{\ell}:=\boldsymbol{\ell}$ is nothing but the function given by (4.2a), and that $\{\mathrm{D}_t^{k-3}\boldsymbol{\ell}\}_{k=4}^{M+2}$ is the set of functions given by (4.4) with $j=k-3$.

4.5. Multi-element flow-driven spectral chaos (ME-FSC) method

This section is devoted to presenting the multi-element version of the FSC method in detail. As mentioned in the introduction, the procedure is similar to the ME-gPC method [43], but with the only difference being that here we do not use the gPC method on each random element but the FSC method for computational efficiency. We first begin by briefly reviewing the multi-element concept behind the ME-FSC method, and then we explain how the probability moments can be computed using the local information available on each random element. The section is then concluded with an outline of the ME-FSC scheme used in this work.

4.5.1. Overview of the ME-FSC method

Because $u\in\mathcal{U}$ by assumption, we can represent such a function using the Fourier series:

$$u(t,\xi)=\sum_{j=0}^{\infty}u^j(t)\,\Psi_j(\xi)\quad\text{on }\mathfrak{T}\times\Xi, \tag{4.5}$$

where u^j is a temporal function in $\mathcal{T}(2)$ symbolizing the j-th random mode of u.

Now, let $\{(\Xi_e,\boldsymbol{\Xi}_e,\hat{\mu}_e)\}_{e=1}^E$ be a partitioned random space, and let $\{\mathfrak{X}_e\}_{e=1}^E$ be its associated partitioned RFS. Then, a p-discretization for each of these $\mathfrak{X}_e$'s can be stipulated by letting $\mathfrak{X}_e^{[P_e]}=\mathrm{span}\{\Psi_{j.e}\}_{j=0}^{P_e}$ be a finite subspace of $\mathfrak{X}_e$ with $P_e+1\in\mathbb{N}_1$ denoting the dimensionality of $\mathfrak{X}_e^{[P_e]}$.

Moreover, if we define $u_{.e}(t,\cdot) = u(t,\cdot)|_{\Xi_e}$ to be the restriction of $u(t,\cdot)$ to Ξ_e for all $t \in \mathfrak{T}$, then an approximation $u_{.e}^{[P_e]}(t,\cdot)$ of $u_{.e}(t,\cdot)$ can be represented in $\mathfrak{Z}_e^{[P_e]}$ by

$$u_{.e}(t,\xi) \approx u_{.e}^{[P_e]}(t,\xi) = \sum_{j=0}^{P_e} u^j{}_{.e}(t)\,\Psi_{j.e}(\xi) \equiv u^j{}_{.e}(t)\,\Psi_{j.e}(\xi) \quad \text{on } \mathfrak{T}\times\Xi_e, \tag{4.6}$$

where we have omitted the summation sign in the last equality for notational simplicity, and the summation index is always taken as $j \in \{0,1,\dots,P_e\}$ unless indicated otherwise. Similarly as before, $u^j{}_{.e}$ represents a temporal function in $\mathcal{T}(2)$ symbolizing the j-th random mode of $u_{.e}$.

Remark 4.3. In writing expression (4.6), we assumed that $\{\Psi_{j.e}\}_{j=0}^{\infty}$ was a well-graded orthogonal basis so that the approximation of $u_{.e}$ could be carried out the way shown. In the scheme provided in Section 4.5.3, such an assumption does not represent an issue at all since all the random bases utilized in the ME-FSC scheme are well-graded by construction.

The problem now reduces to find E independent solutions to (4.1) by looping across the random domain from $e = 1$ to $e = E$ and using the following procedure.

Substituting (4.6) into (4.1) gives

$$\mathcal{L}[u^j{}_{.e}\Psi_{j.e}] = f \qquad \text{on } \mathfrak{T}\times\Xi_e \tag{4.7a}$$

$$\Big\{\mathcal{B}_k[u^j{}_{.e}\Psi_{j.e}](0,\cdot) = b_k\Big\}_{k=1}^{n} \qquad \text{on } \{0\}\times\Xi_e. \tag{4.7b}$$

Projecting (4.7) onto $\mathfrak{Z}_e^{[P_e]}$ yields a system of P_e+1 ordinary differential equations of second order in the variable t, where the unknowns are the random modes $u^j{}_{.e} = u^j{}_{.e}(t)$ and their first $n-1$ time derivatives:

$$\Psi^i{}_{.e}\Big[\mathcal{L}[u^j{}_{.e}\Psi_{j.e}]\Big] = \Psi^i{}_{.e}[f] \qquad \text{on } \mathfrak{T} \tag{4.8a}$$

$$\Big\{\Psi^i{}_{.e}\Big[\mathcal{B}_k[u^j{}_{.e}\Psi_{j.e}](0,\cdot)\Big] = \Psi^i{}_{.e}[b_k]\Big\}_{k=1}^{n} \qquad \text{on } \{0\} \tag{4.8b}$$

with $i,j \in \{0,1,\dots,P_e\}$. Notice that this projection is done here by applying on both sides of each equation the linear functionals $\{\Psi^i{}_{.e} \in \mathfrak{Z}_e'\}_{i=0}^{P_e}$ one by one.

For the specific case of a system given by (4.1*), it is clear that (4.8) would reduce to

$$m^i{}_{j.e}\ddot{u}^j{}_{.e} + k^i{}_{j.e}u^j{}_{.e} = p^i{}_{.e} \qquad \text{on } \mathfrak{T} \tag{4.8a*}$$

$$\left\{u^i{}_{.e}(0) = \boldsymbol{u}^i{}_{.e},\ \dot{u}^i{}_{.e}(0) = \boldsymbol{v}^i{}_{.e}\right\} \qquad \text{on } \{0\}, \tag{4.8b*}$$

where $i, j \in \{0, 1, \ldots, P_e\}$ (summation sign implied only over repeated index j), and:

$$m^i{}_{j.e} = \langle \Psi_{i.e}, m\Psi_{j.e}\rangle_e / \langle \Psi_{i.e}, \Psi_{i.e}\rangle_e, \quad k^i{}_{j.e} = \langle \Psi_{i.e}, k\Psi_{j.e}\rangle_e / \langle \Psi_{i.e}, \Psi_{i.e}\rangle_e,$$

$$p^i{}_{.e}(t) = \langle \Psi_{i.e}, p(t, \cdot\,)\rangle_e / \langle \Psi_{i.e}, \Psi_{i.e}\rangle_e,$$

$$u^i{}_{.e} = \langle \Psi_{i.e}, \boldsymbol{u}\rangle_e / \langle \Psi_{i.e}, \Psi_{i.e}\rangle_e \quad \text{and} \quad v^i{}_{.e} = \langle \Psi_{i.e}, \boldsymbol{v}\rangle_e / \langle \Psi_{i.e}, \Psi_{i.e}\rangle_e.$$

4.5.2. Computation of probability moments

As a means to simplify the notation henceforth, we note that a system such as (4.1) can also be rewritten using modeling notation:

$$y = \boldsymbol{\mathcal{M}}[u][x] \quad \text{subject to initial condition} \quad \boldsymbol{\mathcal{I}}[u],$$

where $\boldsymbol{\mathcal{M}}[u] : \mathcal{V}^r \to \mathcal{V}^s$ is the mathematical model of the system defined by (4.1a), $x = (x_1, \ldots, x_r) : \mathfrak{T} \times \Xi \to \mathbb{R}^r$ is the r-tuple input of $\boldsymbol{\mathcal{M}}[u]$, $y = (y_1, \ldots, y_s) : \mathfrak{T} \times \Xi \to \mathbb{R}^s$ is the s-tuple output of $\boldsymbol{\mathcal{M}}[u]$ (aka the s-tuple observable in physics or the s-tuple response in engineering), and $\boldsymbol{\mathcal{I}}[u]$ is the initial condition of the system defined by (4.1b).

Now, let $z = y_k$ be the k-th component of output $y = \boldsymbol{\mathcal{M}}[u][x]$. If $z \in \mathcal{V}$, then it is clear that such a function can be represented approximately over the e-th random element using the following expansion:

$$z_{.e}(t, \xi) \approx z_{.e}^{[P_e]}(t, \xi) = \sum_{j=0}^{P_e} z^j{}_{.e}(t)\, \Psi_{j.e}(\xi) \equiv z^j{}_{.e}(t)\, \Psi_{j.e}(\xi) \quad \text{on } \mathfrak{T} \times \Xi_e,$$

where $z_{.e}(t, \cdot\,) = z(t, \cdot\,)|_{\Xi_e}$ denotes the restriction of $z(t, \cdot\,)$ to Ξ_e for all $t \in \mathfrak{T}$.

In this section, we are interested in computing the mean and variance of z over $\mathfrak{T} \times \Xi$ using the local information available on each random element. To do so, we first make the following observation regarding the (local) mean and (local) variance of z given that $\xi \in \Xi_e$.

As it should be easy to verify, the (local) mean of z given that $\xi \in \Xi_e$, $\mathbf{E}_e[z] : \mathfrak{T} \to \mathbb{R}$, is nothing but

$$\mathbf{E}_e[z](t) \equiv \mathbf{E}[z \mid \Xi_e](t) := \int z(t,\cdot)\,\mathrm{d}\hat{\mu}_e = \int z_{.e}(t,\cdot)\,\mathrm{d}\hat{\mu}_e = z^0{}_{.e}(t), \tag{4.10}$$

and that the (local) variance of z given that $\xi \in \Xi_e$, $\mathrm{Var}_e[z] : \mathfrak{T} \to \mathbb{R}_0^+$, is

$$\begin{aligned}\mathrm{Var}_e[z](t) \equiv \mathrm{Var}[z \mid \Xi_e](t) &:= \int \Big(z(t,\cdot) - \mathbf{E}_e[z](t)\Big)^2\,\mathrm{d}\hat{\mu}_e = \int z^2(t,\cdot)\,\mathrm{d}\hat{\mu}_e - \mathbf{E}_e[z]^2(t) \\ &= \int z_{.e}{}^2(t,\cdot)\,\mathrm{d}\hat{\mu}_e - \mathbf{E}_e[z]^2(t) = \sum_{j=1}^{P} \langle \Psi_{j.e}, \Psi_{j.e} \rangle_e\, z^j{}_{.e}(t)\, z^j{}_{.e}(t). \end{aligned} \tag{4.11}$$

Therefore, the (global) mean of z, $\mathbf{E}[z] : \mathfrak{T} \to \mathbb{R}$, which is given by

$$\mathbf{E}[z](t) := \int z(t,\cdot)\,\mathrm{d}\mu = \sum_{e=1}^{E} \int_{\Xi_e} z(t,\cdot)\,\mathrm{d}\mu = \sum_{e=1}^{E} \int z(t,\cdot)\,\mathrm{d}\mu_e,$$

can be further simplified to:

$$\mathbf{E}[z](t) = \sum_{e=1}^{E} \mu(\Xi_e) \int z(t,\cdot)\,\mathrm{d}\hat{\mu}_e = \sum_{e=1}^{E} \mu(\Xi_e)\, \mathbf{E}_e[z](t). \tag{4.12}$$

This final form of $\mathbf{E}[z]$ is also known as the *law of total expectation*.

Moreover, we note that the (global) variance of z, $\mathrm{Var}[z] : \mathfrak{T} \to \mathbb{R}_0^+$, which is given by

$$\begin{aligned}\mathrm{Var}[z](t) := \int \Big(z(t,\cdot) - \mathbf{E}[z](t)\Big)^2\,\mathrm{d}\mu &= \int z^2(t,\cdot)\,\mathrm{d}\mu - \mathbf{E}[z]^2(t) \\ &= \sum_{e=1}^{E} \int_{\Xi_e} z^2(t,\cdot)\,\mathrm{d}\mu - \mathbf{E}[z]^2(t), \end{aligned} \tag{4.13}$$

can be further simplified if we recognize that *first*:

$$\begin{aligned}\int_{\Xi_e} z^2(t,\cdot)\,\mathrm{d}\mu = \mu(\Xi_e) \int z^2(t,\cdot)\,\mathrm{d}\hat{\mu}_e = \mu(\Xi_e) \int z_{.e}{}^2(t,\cdot)\,\mathrm{d}\hat{\mu}_e \\ = \mu(\Xi_e) \Big(\mathrm{Var}_e[z](t) + \mathbf{E}_e[z]^2(t)\Big)\end{aligned}$$

and that *second*:

$$\mathbf{E}[z]^2(t) = \sum_{e_1=1}^{E}\sum_{e_2=1}^{E} \mu(\Xi_{e_1})\,\mu(\Xi_{e_2})\,\mathbf{E}_{e_1}[z](t)\,\mathbf{E}_{e_2}[z](t)$$
$$= \sum_{e=1}^{E} \mu(\Xi_e)^2\,\mathbf{E}_e[z]^2(t) + 2\sum_{e_1=2}^{E}\sum_{e_2=1}^{e_1-1} \mu(\Xi_{e_1})\,\mu(\Xi_{e_2})\,\mathbf{E}_{e_1}[z](t)\,\mathbf{E}_{e_2}[z](t).$$

Hence, replacing these two expressions into (4.13) and simplifying, one gets

$$\mathrm{Var}[z](t) = \sum_{e=1}^{E} \mu(\Xi_e)\,\mathrm{Var}_e[z](t) + \sum_{e=1}^{E} \mu(\Xi_e)\,(1-\mu(\Xi_e))\,\mathbf{E}_e[z]^2(t)$$
$$- 2\sum_{e_1=2}^{E}\sum_{e_2=1}^{e_1-1} \mu(\Xi_{e_1})\,\mu(\Xi_{e_2})\,\mathbf{E}_{e_1}[z](t)\,\mathbf{E}_{e_2}[z](t). \quad (4.14)$$

This final form of $\mathrm{Var}[z]$ is also known as the *law of total variance*.

4.5.3. ME-FSC scheme

Consider the stochastic dynamical system given by (4.1). Let $\{\mathfrak{T}_i \times \Xi_e\}_{i=0,e=1}^{N-1,E}$ be a partition of the temporal-random domain $\mathfrak{T}\times\Xi$ (aka system's domain), where $\mathfrak{T}_i \neq \emptyset$ is the i-th element of the temporal domain, $\Xi_e \neq \emptyset$ is the e-th element of the random domain, and $\mathfrak{T}_i \times \Xi_e$ is the (i,e)-th element of the system's domain. It is worth mentioning that this partition gives rise to the partitioned random space $\{(\Xi_e, \boldsymbol{\Xi}_e, \hat{\mu}_e)\}_{e=1}^{E}$ and associated partitioned RFS $\{\mathfrak{Z}_e\}_{e=1}^{E}$, as defined in Section 4.2. Moreover, for notational convenience, we define $s_{.ie} = s|_{\mathrm{cl}(\mathfrak{T}_i)\times\Xi_e}$ to be the restriction of s to $\mathfrak{R}_{ie} := \mathrm{cl}(\mathfrak{T}_i) \times \Xi_e$, and $s_{..e} = s|_{\mathfrak{T}\times\Xi_e}$ to be the restriction of s to $\mathfrak{T} \times \Xi_e$. For illustration purposes, Fig. 4.2 depicts the evolution of a dynamical system by applying successively a stochastic flow map of order M over a partitioned random domain with E elements.

In this sense, if we assume that the system is driven by a stochastic flow map of order M, we can proceed as follows. (Recall that n denotes the order of the governing ODE with respect to time, as displayed in (4.1).)

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

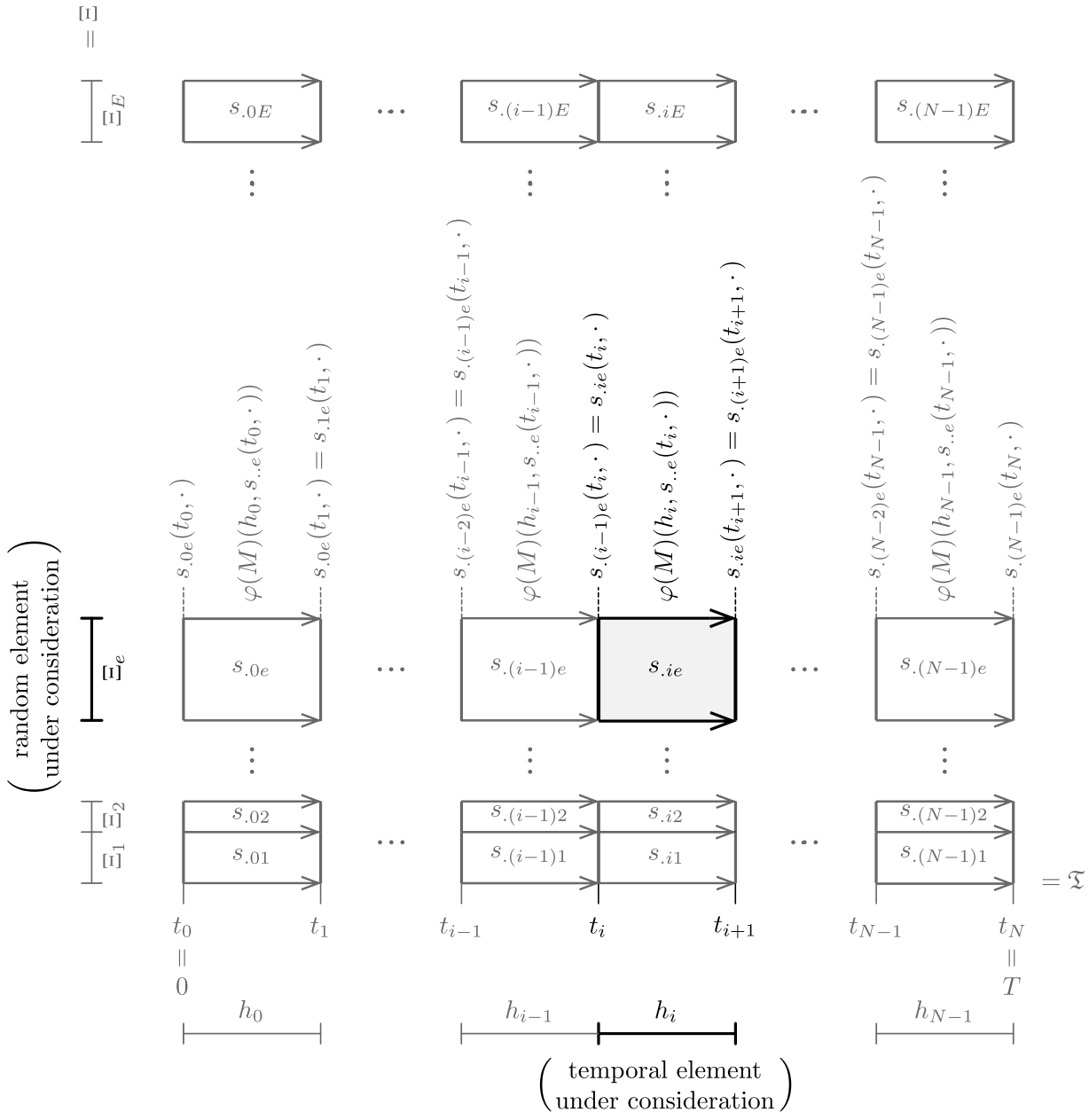


Figure 4.2. Evolution of a dynamical system via a stochastic flow map of order M (with $h_i > 0$) over a partitioned random domain

i. Loop across the random domain from $e = 1$ to $e = E$. Note that this loop can be parallelized from a computational standpoint since each iteration is independent of the others.

(a) Define a solution representation for the configuration state $s_{.ie}$ in the following way.

- Take $\Phi_{0.ie} \equiv 1$ and $\{\Phi_{j.ie} := \hat{\varphi}^j(M)(0, \hat{s}_{..e}(t_i, \cdot))\}_{j=1}^{P_e}$ to be an ordered set of linearly independent functions in $\mathfrak{Z}_e$ with $n + 1 \leq P \leq n + M$. Note that $\hat{\varphi}(M)(0, \hat{s}_{..e}(t_i, \cdot)) \equiv \hat{s}_{.ie}(t_i, \cdot) = \hat{s}_{.(i-1)e}(t_i, \cdot)$ for $i \geq 1$. However, if $i = 0$, then $\hat{\varphi}(M)(0, \hat{s}_{..e}(t_0, \cdot)) \equiv \hat{s}_{..e}(0, \cdot)$.
- Orthogonalize the set $\{\Phi_{j.ie}\}_{j=0}^{P_e}$ using the Gram-Schmidt process [61], so that the resulting set $\{\Psi_{j.ie}\}_{j=0}^{P_e}$ is an orthogonal basis in $\mathfrak{Z}_e$. Specifically, for $j \in \{0, 1, \ldots, P_e\}$:

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

- Define $\mathfrak{Z}_{ie}^{[P_e]} = \text{span}\{\Psi_{j.ie}\}_{j=0}^{P_e}$ to be an (h, p)-discretization of $\mathfrak{Z}$ over the region $\mathfrak{R}_{ie}$. Then, because $\mathfrak{Z}_{ie}^{[P_e]}$ is an evolving function space over Ξ_e, expansion (4.6) is now to be read as:

$$u_{.ie}(t, \xi) \approx u_{.ie}^{[P_e]}(t, \xi) = \sum_{j=0}^{P_e} u^j{}_{.ie}(t)\, \Psi_{j.ie}(\xi) \equiv u^j{}_{.ie}(t)\, \Psi_{j.ie}(\xi). \tag{4.6'}$$

Thus, to compute the l-th component of the configuration state, $s^l{}_{.ie}$, we simply need to take the $(l-1)$-th time derivative of (4.6'). Here $l \in \{1, 2, \ldots, n\}$.

(b) Transfer at $t = t_i$ the random modes from the old definition of system's configuration state

$$s_{.(i-1)e}(t_i, \cdot) = (u_{.(i-1)e}(t_i, \cdot), \partial_t u_{.(i-1)e}(t_i, \cdot), \ldots, \partial_t^{n-1} u_{.(i-1)e}(t_i, \cdot))$$

to the new definition of system's configuration state

$$s_{.ie}(t_i, \cdot) = (u_{.ie}(t_i, \cdot), \partial_t u_{.ie}(t_i, \cdot), \ldots, \partial_t^{n-1} u_{.ie}(t_i, \cdot)),$$

given that $i \geq 1$. To do so, we implement the FSC-2 approach presented in [92] to obtain the random modes of each of the components of $s_{.ie}$ at $t = t_i$. The reason why we choose FSC-2 over FSC-1 is that the former has the ability to transfer the probability information exactly at any instant of time. Thus, by resorting to Theorem 1 of [92], one can show that the j-th random mode of the l-th component of $s_{.ie}(t_i, \cdot\,)$ is given by:

$$(s^l)^j{}_{.ie}(t_i) = \begin{cases} \mathbf{E}[\Phi_{l.ie}] & \text{for } j = 0 \\ \dfrac{\det \triangle_j(l)}{\det \square_j} & \text{for } 0 < j < l \\ 1 & \text{for } j = l \\ 0 & \text{otherwise,} \end{cases} \tag{4.8b'}$$

from where we have taken $\{\Phi_{l.ie} := \varphi^l(M)(0, s_{..e}(t_i, \cdot\,)) \equiv s^l{}_{.ie}(t_i, \cdot\,) = s^l{}_{.(i-1)e}(t_i, \cdot\,)\}_{l=1}^n$, and

$$\square_j = \begin{bmatrix} \mathrm{Cov}[\Phi_{1.ie}, \Phi_{1.ie}] & \cdots & \mathrm{Cov}[\Phi_{1.ie}, \Phi_{j.ie}] \\ \vdots & \ddots & \vdots \\ \mathrm{Cov}[\Phi_{j.ie}, \Phi_{1.ie}] & \cdots & \mathrm{Cov}[\Phi_{j.ie}, \Phi_{j.ie}] \end{bmatrix} \in \mathscr{M}(j \times j, \mathbb{R}),$$

$$\triangle_j(l) = \begin{bmatrix} \mathrm{Cov}[\Phi_{1.ie}, \Phi_{1.ie}] & \cdots & \mathrm{Cov}[\Phi_{1.ie}, \Phi_{j.ie}] \\ \vdots & \ddots & \vdots \\ \mathrm{Cov}[\Phi_{(j-1).ie}, \Phi_1] & \cdots & \mathrm{Cov}[\Phi_{(j-1).ie}, \Phi_{j.ie}] \\ \mathrm{Cov}[\Phi_{l.ie}, \Phi_{1.ie}] & \cdots & \mathrm{Cov}[\Phi_{l.ie}, \Phi_{j.ie}] \end{bmatrix} \in \mathscr{M}(j \times j, \mathbb{R})$$

with $\triangle_1(l) = \mathrm{Cov}[\Phi_{l.ie}, \Phi_{1.ie}] \in \mathbb{R}$ and

$$\triangle_2(l) = \begin{bmatrix} \mathrm{Cov}[\Phi_{1.ie}, \Phi_{1.ie}] & \mathrm{Cov}[\Phi_{1.ie}, \Phi_{2.ie}] \\ \mathrm{Cov}[\Phi_{l.ie}, \Phi_{1.ie}] & \mathrm{Cov}[\Phi_{l.ie}, \Phi_{2.ie}] \end{bmatrix} \in \mathscr{M}(2 \times 2, \mathbb{R})$$

as special cases of $\triangle_j$ by definition.

(c) Substitute (4.6') into (4.1) to obtain (4.7).

(d) Project (4.7a) onto $\mathfrak{Z}_{ie}^{[P_e]}$ to obtain (4.8a) subject to (4.8b'). Note that if $i = 0$, (4.8a) is subject to (4.8b).

(e) Integrate (4.8) over time provided that a suitable time integration method has been chosen for the system of equations in hand. This step requires finding the random modes of each of the components of the configuration state $s_{.ie}$ at $t = t_{i+1}$; that is, $\{(s^l)^j{}_{.ie}(t_{i+1})\}_{l=1,j=0}^{n,P_e}$

(f) Compute both the (local) mean and the (local) variance of each of the components of output $y = \boldsymbol{\mathcal{M}}[x][u]$ over $\mathfrak{R}_{ie}$, by resorting to the formulas prescribed by (4.10) and (4.11).

ii. Aggregate results to compute over $\mathfrak{R}_i := \mathfrak{T}_i \times \Xi$ the (global) mean and (global) variance of $y = \boldsymbol{\mathcal{M}}[x][u]$ using the formulas stipulated by (4.12) and (4.14).

2. Post-process results.

4.6. Numerical results

As in [92], we define the local and global errors, $\epsilon : \mathcal{T} \to \mathcal{T}$ and $\epsilon_G : \mathcal{T} \to \mathbb{R}$, using the following expressions:

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

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

In an effort to reduce as much as possible the source of errors coming from the discretization of $\mathcal{T}$, the time-step size used for the simulations is taken as $\Delta t = 0.001$ s in Problems 1 and 2 and $\Delta t = 0.005$ s in Problems 3 and 4. To integrate (4.8) over time, we use the RK4 method over each random element, and in order to obtain accurate results, the random function space is updated at every time step. For simplicity, the partition of the random domain is such that all elements are the same size. The temporal domain employed in Problems 1, 2 and 3 is $\mathfrak{T} = [0, 150]$ s, and in Problem 4 is $\mathfrak{T} = [0, 50]$ s. To ensure that the stochasticity of the system's state is well developed for the analysis with ME-FSC, we sometimes use the

gPC method for the first second of the simulation. This is to prevent that ill-conditioned matrices arise at earlier times of the simulation whenever we have a situation where the initial conditions of the system are deterministic (as is the case with the first three problems). Specifically, in Problems 1 and 2 we employ the gPC method with $P = 7$, and in Problem 3 we employ the gPC method with $P = 9$.

The inner products are computed using a set of Legendre quadrature rules on each random axis defined by

$$\text{Uniform} \sim \text{Gauss-Legendre(10 points/element)}.$$

This means that for distributions other than uniform the probability distribution function must be included in the integrand in order to obtain the numerical value of the inner product.

All problems are run using Apple's Foundation and Accelerate frameworks [93] on a 2020 MacBook Air with Apple M1 chip (*8-Core* CPU at *3.20 GHz*, *8-Core* GPU, *16-Core* Neural Engine, and *16 GB* unified memory) and *1 TB Apple-Fabric SSD* storage (APFS-formatted), running macOS Big Sur (version 11.2). The code is written entirely in the Swift 5.3 language [94].

4.6.1. Problem 1: A linear system governed by a 2nd-order stochastic ODE

We first consider the problem of an undamped single-degree-of-freedom system under free vibration. The law of motion for this system is defined by

$$m\ddot{u} + ku = 0,$$

where the mass of the system is $m = 100$ kg, and the stiffness of the system, $k : \Xi \to \mathbb{R}^+$, is stochastic and given by $k(\xi) = \xi$. The system has an initial displacement of $u(0, \cdot\,) = \boldsymbol{u} \equiv 0.05$ m, and an initial velocity of $\dot{u}(0, \cdot\,) = \boldsymbol{v} \equiv 0.20$ m/s. Formally, one can express this problem in the following way:

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

$$m\ddot{u} + ku = 0 \qquad \text{on } \mathfrak{T} \times \Xi \tag{4.15a}$$

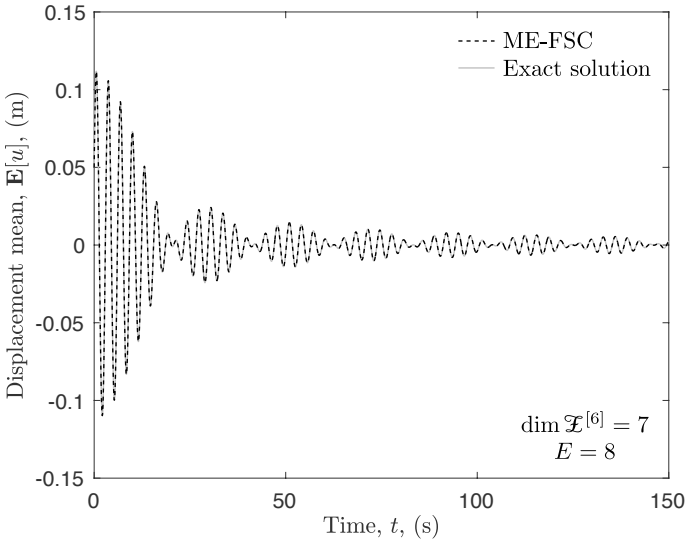
$$\Big\{u(0,\cdot) = \mathfrak{u},\ \dot{u}(0,\cdot) = \mathfrak{v}\Big\} \qquad \text{on } \{0\} \times \Xi. \tag{4.15b}$$

This problem statement is similar to (4.1*), with the only difference being that $p \equiv 0$ and $m, \mathfrak{u}, \mathfrak{v}$ are real numbers.

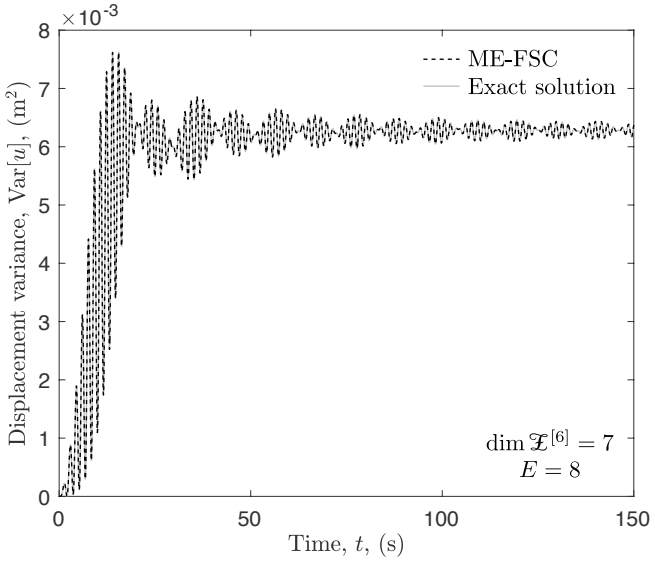
Two different probability distributions are considered for ξ. The first distribution is a *uniform distribution*, Uniform $\sim \xi \in \Xi = [a, b]$, and the second distribution is a *beta distribution*, Beta$(\alpha, \beta) \sim \xi \in \Xi = [a, b]$. The parameters for both these distributions are taken as: $(a, b) = (340, 460)$ N/m and $(\alpha, \beta) = (2, 5)$.

In Fig. 4.3 we show the evolution of the mean and variance of the system's displacement. These results are obtained using 8 elements in the random domain and 7 basis vectors per element. We see that when this particular discretization is used, the ME-FSC method is able to reproduce the exact response with high fidelity, explaining why the two plots look indistinguishable from each other.

Figs. 4.4 to 4.6 present the local errors in mean and variance of the system's displacement using different numbers of elements and basis vectors. For brevity, we only present the case when the probability measure is uniform, although similar convergence trends are also achievable when the probability measure is beta. For sake of comparison, we also include the case when $P = 2$, even though this is not allowed by the FSC method. This is because in FSC the lower bound for P is always $n + 1$, where n is the order of the governing ODE with respect to time. (To keep the uniformity in the presentation, the same is done in Problems 2 and 3.) These plots show that the results get better if the number of elements is increased. However, this is not always the case whenever we increase the number of basis vectors. For instance, when the number of basis vectors is increased from 3 to 5, we observe that the results improve noticeably, but when the number of basis vectors is increased from 5 to 7 they do not. This is chiefly because the quadrature rule employed in this work is not optimal—note that the quadrature points are not those that ensure exponential convergence to the sought integral, as in Gaussian-based quadrature rules, the quadrature points are demanded to be the roots of an orthogonal polynomial that is concordant with the measure defined on the



(a) Mean



(b) Variance

Figure 4.3. *Problem 1* — Evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for the case when the (h, p)-discretization level of RFS is $(P, E) = (6, 8)$ and $\mu \sim$ Uniform

integral's domain (which is clearly not the case here). The above assertion even holds for the case when 1 element is used in the simulations (i.e. when there is no partition in the random domain), since distributing only 10 Legendre quadrature points in the random domain is not sufficiently enough to allow the higher basis vectors to play a role in the global response of the system. However, if more quadrature points were to be employed in the simulations, an improvement between the plots with $P = 4$ and $P = 6$ could have been discerned for the case of using 1 element, but virtually no improvement for the case of using more than 1 element. It is worth pointing out that the accuracy of these results is limited by machine precision, and that as a result of this, better results than those depicted in Figs. 4.4 to 4.6 are difficult to obtain for other values of P and E.

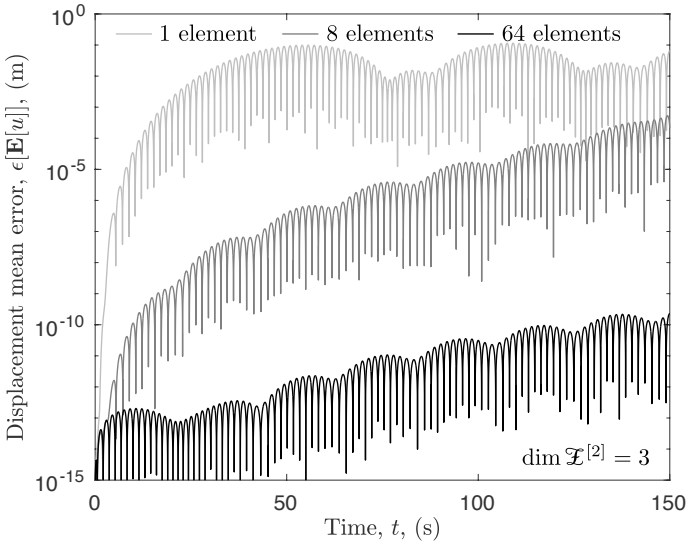
Figs. 4.7 and 4.8 depict the convergence of global errors as a function of the number of elements and the number of basis vectors used. Included in this figure are the cases where the probability measure is uniform or beta. On average, exponential convergence can be attained if the number of elements is exponentially incremented. In particular, when the number of basis vectors is 5, we see that the exponential convergence is much steeper than when it is, say, 3 or 4. However, when the number of basis vectors is greater than 5, no improvement in the accuracy of the results can be achieved. This same outcome occurs consistently in both distributions, which means that when a Legendre quadrature rule with 10 points per element and 5 basis vectors are used in the simulations, the maximum accuracy allowed by the machine and the ME-FSC method is finally reached.

4.6.2. Problem 2: A linear system governed by a 3rd-order stochastic ODE

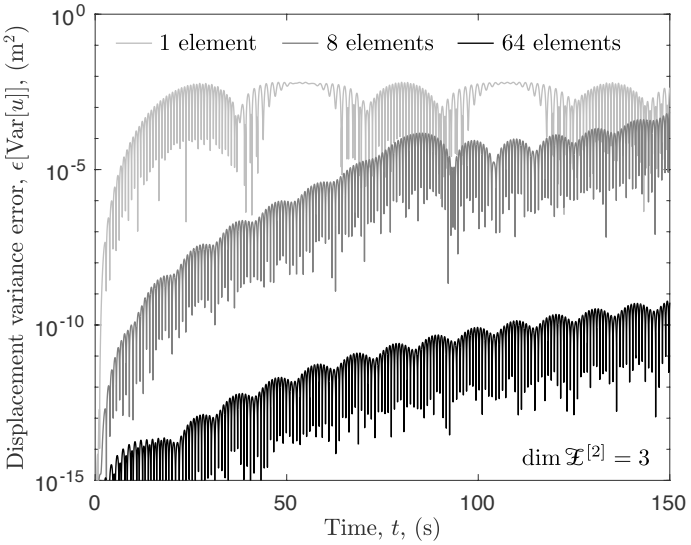
We next consider the problem of a linear mechanical system governed by a third-order stochastic ODE. The governing differential equation for this system is defined by

$$\partial_t^3 u + \tfrac{1}{2}\partial_t^2 u + k\,\partial_t u + u = 0,$$

where $k : \Xi \to \mathbb{R}$ is a stochastic mechanical parameter given by $k(\xi) = \xi$, and $u : \mathfrak{T} \times \Xi \to \mathbb{R}$ denotes the displacement of the system with $\partial_t u, \partial_t^2 u, \partial_t^3 u$ representing the velocity, acceleration

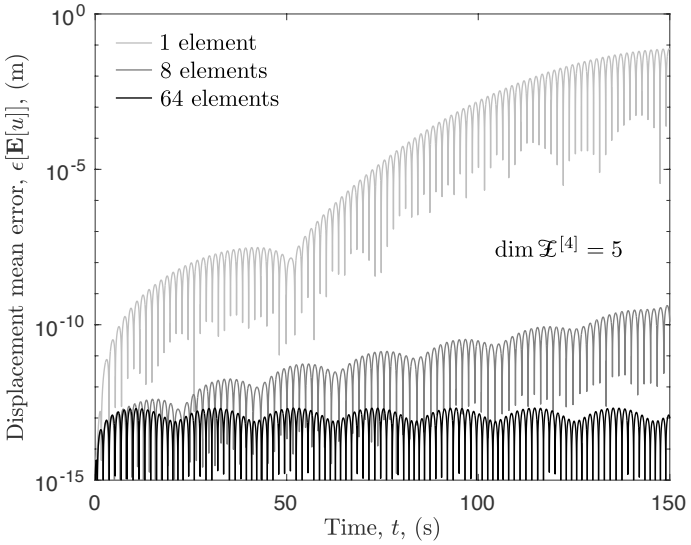


(a) Mean error for $\mathfrak{X}^{[2]}$

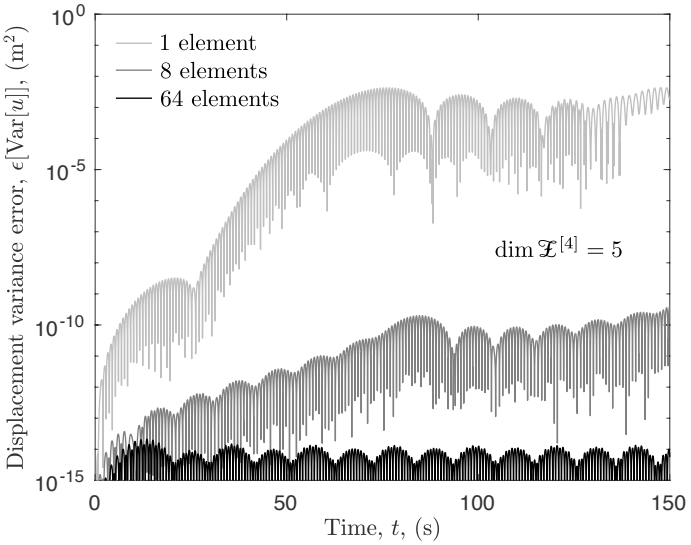


(b) Variance error for $\mathfrak{X}^{[2]}$

Figure 4.4. *Problem 1* — Local error evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different (h, p)-discretization levels of RFS and for $\mu \sim$ Uniform (SET 1/3)

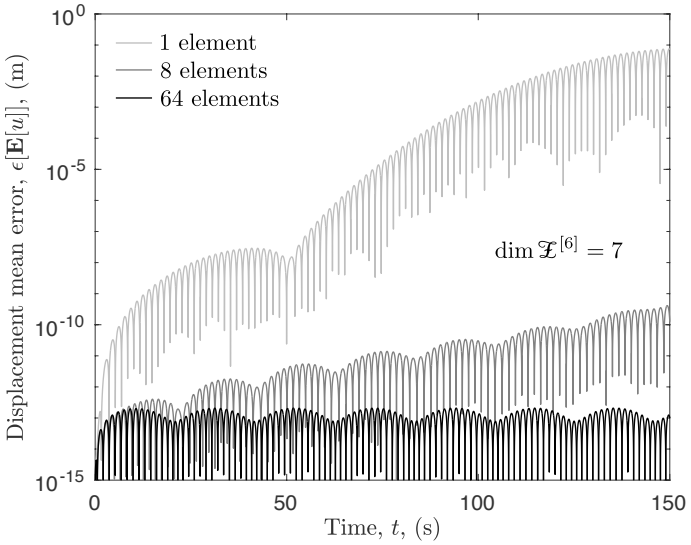


(a) Mean error for $\mathfrak{X}^{[4]}$

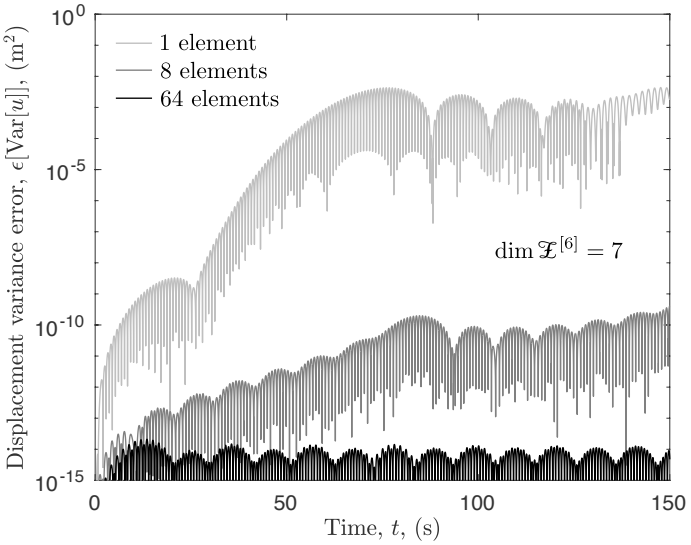


(b) Variance error for $\mathfrak{X}^{[4]}$

Figure 4.5. *Problem 1* — Local error evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different (h, p)-discretization levels of RFS and for $\mu \sim$ Uniform (SET 2/3)

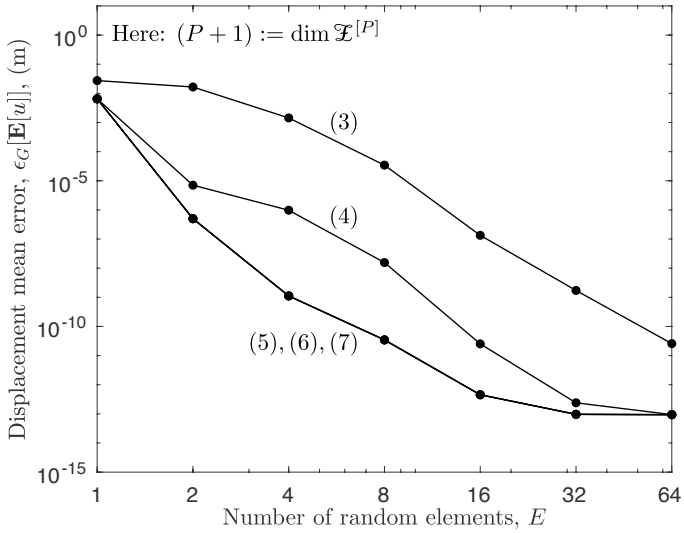


(a) Mean error for $\mathfrak{X}^{[6]}$

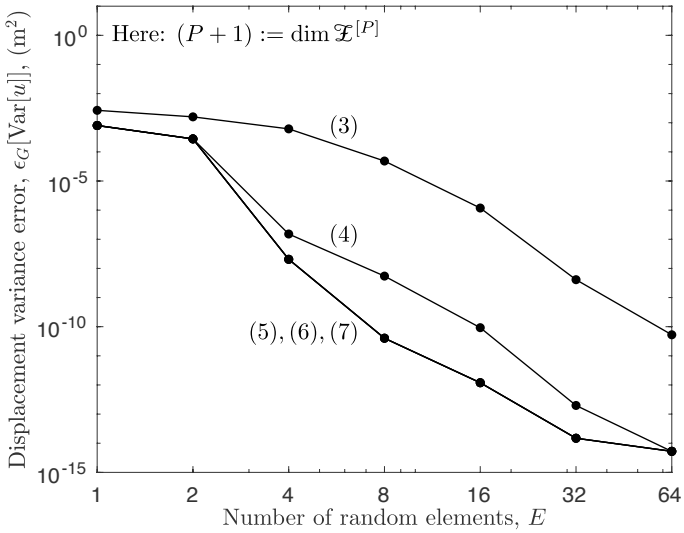


(b) Variance error for $\mathfrak{X}^{[6]}$

Figure 4.6. *Problem 1* — Local error evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different (h, p)-discretization levels of RFS and for $\mu \sim$ Uniform (SET 3/3)

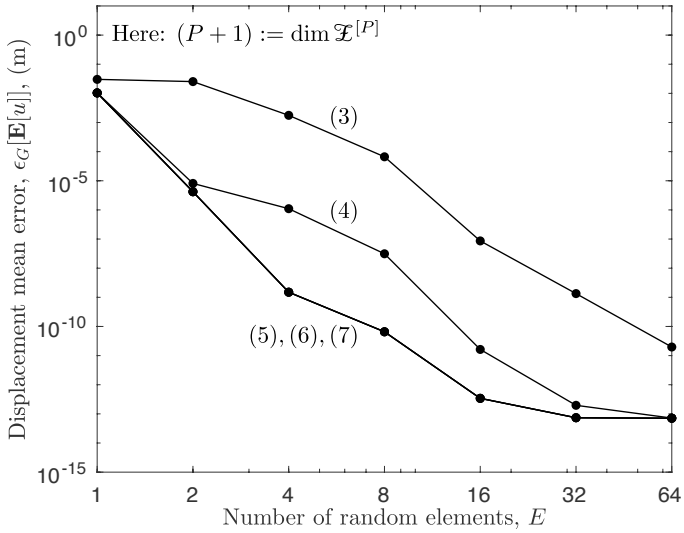


(a) Mean error for $\mu \sim$ Uniform

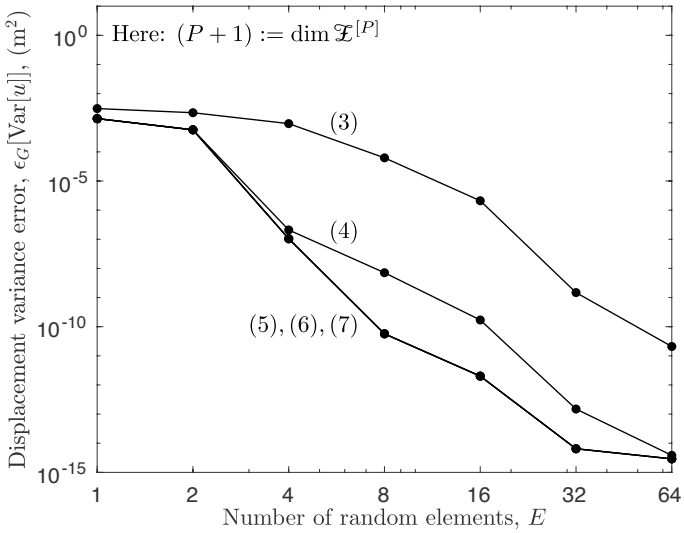


(b) Variance error for $\mu \sim$ Uniform

Figure 4.7. *Problem 1* — Global error of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different (h, p)-discretization levels of RFS ($\mu \sim$ Uniform)



(a) Mean error for $\mu \sim$ Beta



(b) Variance error for $\mu \sim$ Beta

Figure 4.8. *Problem 1* — Global error of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different (h, p)-discretization levels of RFS ($\mu \sim$ Beta)

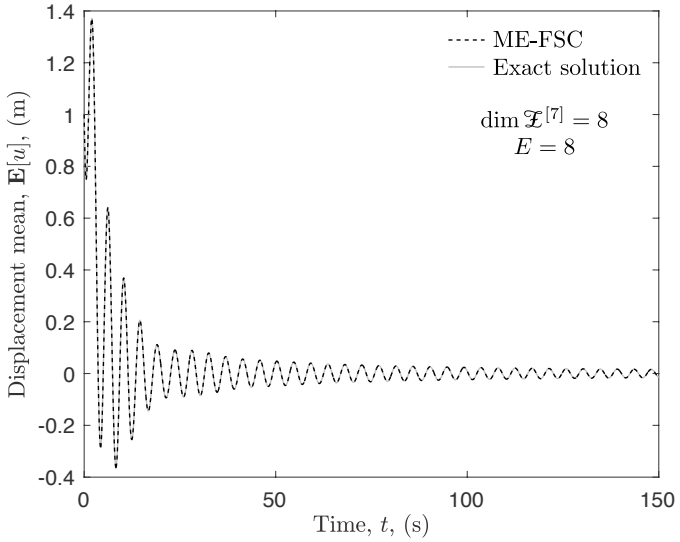
and jerk of the system, respectively. The initial conditions of the system are deterministic and given by: $u(0,\cdot) \equiv 1$ m, $\partial_t u(0,\cdot) \equiv -1$ m/s, and $\partial_t^2 u(0,\cdot) \equiv 2$ m/s^2.

As with Problem 1, two different probability distributions are explored for ξ. Namely, a *uniform distribution* defined by Uniform $\sim \xi \in \Xi = [a,b]$, and a *beta distribution* defined by Beta$(\alpha,\beta) \sim \xi \in \Xi = [a,b]$, whence $(a,b) = (2,3)$ N/m and $(\alpha,\beta) = (2,5)$.

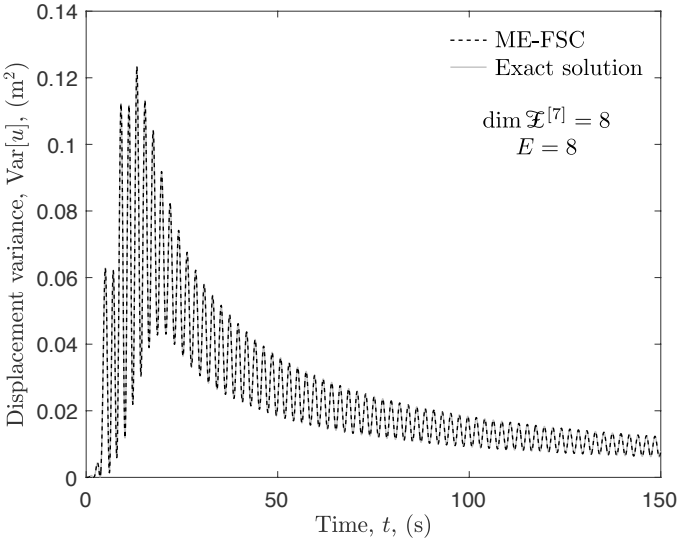
Fig. 4.9 shows the evolution of the mean and variance of the system's displacement for the particular case where the probability measure is uniform and 8 elements and 8 basis vectors are used. As in the previous problem, the results obtained with ME-FSC are again indistinguishable from the exact response.

In Figs. 4.10 to 4.12 we present the local errors in mean and variance of the system's displacement using different numbers of elements and basis vectors. For brevity, the results are only presented for $\mu \sim$ Uniform. Once again, it is observed that the accuracy of the results improves as the number of elements increases, but it necessarily does not as the number of basis vectors increases. This is exemplified in Figs. 4.11 and 4.12 from where it is deduced that the results do not improve if the number of basis vectors is increased from 6 to 8. This once again is due to the quadrature rule used to compute the inner products (i.e. 10 Legendre quadrature points per element) and the limited precision of the machine.

Finally, Figs. 4.13 and 4.14 depict the global errors in mean and variance of the system's displacement. The results are presented as a function of the number of elements and the number of basis vectors used, and for each of the probability distributions explored for ξ. Again, exponential convergence is on average attainable if the number of elements is exponentially incremented. However, as observed, using more than 4 basis vectors in the simulations does not help improve the overall accuracy of the results, except of course when 4 elements and 5 basis vectors are used to obtain the mean response. This figure therefore suggests that in some situations it might be better to refine the partition of the random domain (instead of making the random function space bigger) to obtain more accurate results.

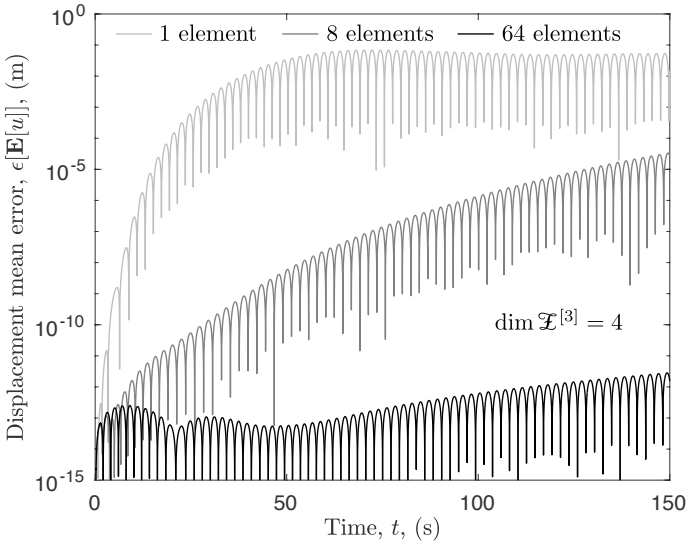


(a) Mean

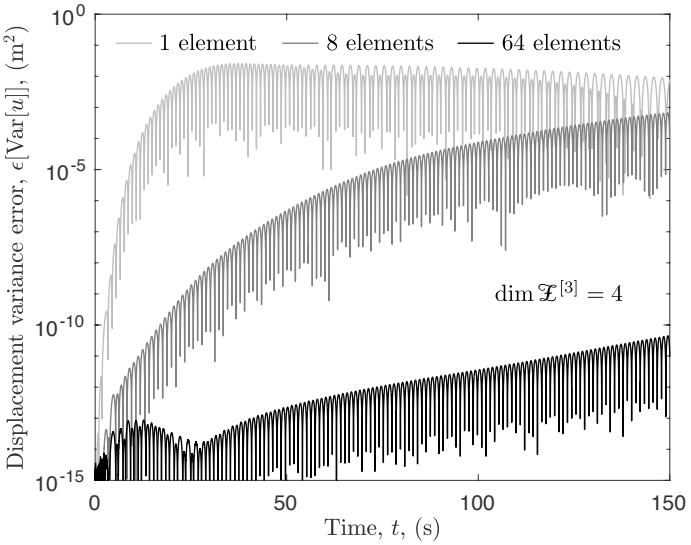


(b) Variance

Figure 4.9. *Problem 2* — Evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for the case when the (h, p)-discretization level of RFS is $(P, E) = (7, 8)$ and $\mu \sim$ Uniform

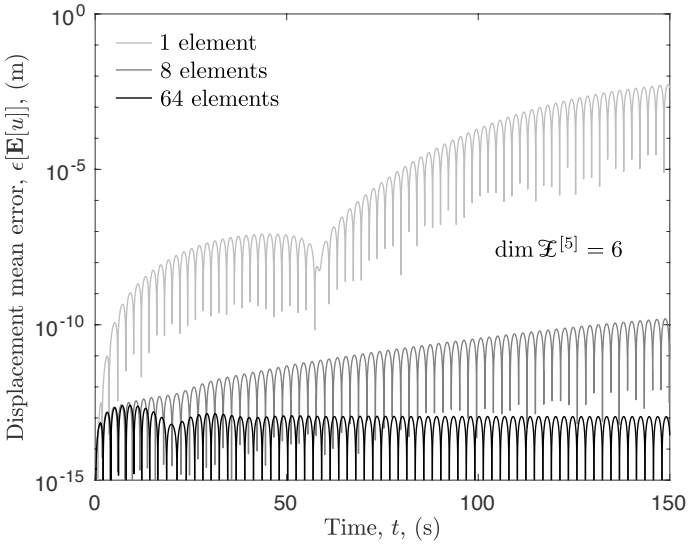


(a) Mean error for $\mathfrak{X}^{[3]}$

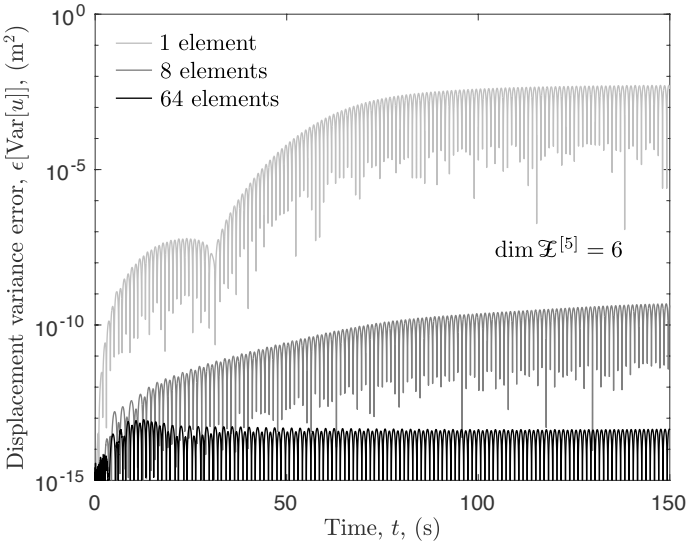


(b) Variance error for $\mathfrak{X}^{[3]}$

Figure 4.10. *Problem 2* — Local error evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different (h, p)-discretization levels of RFS and for $\mu \sim$ Uniform (SET 1/3)

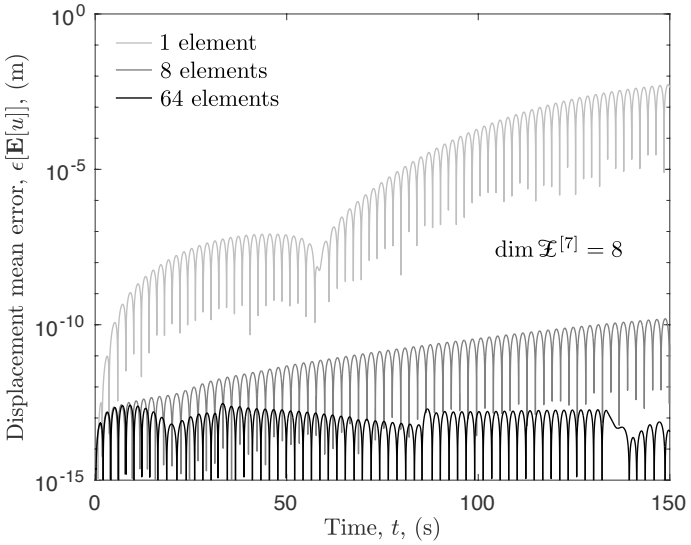


(a) Mean error for $\mathfrak{X}^{[5]}$

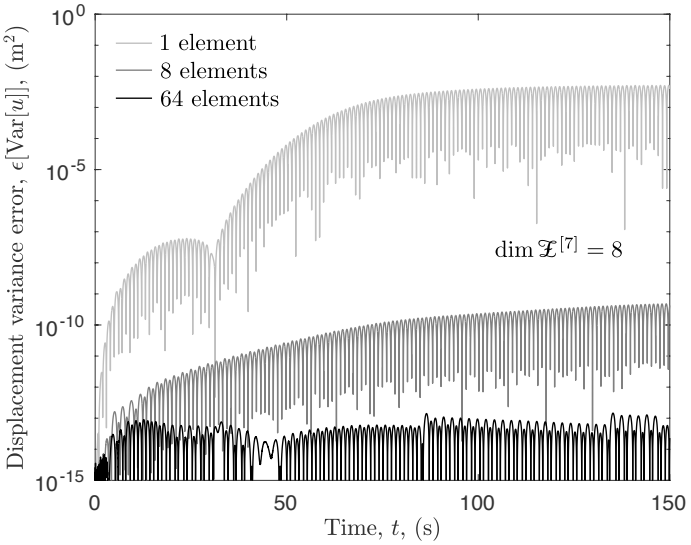


(b) Variance error for $\mathfrak{X}^{[5]}$

Figure 4.11. *Problem 2* — Local error evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different (h, p)-discretization levels of RFS and for $\mu \sim$ Uniform (SET 2/3)

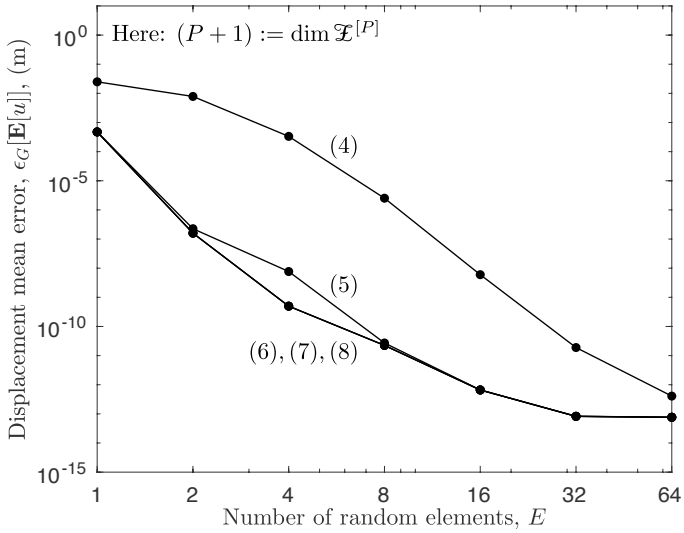


(a) Mean error for $\mathfrak{X}^{[7]}$

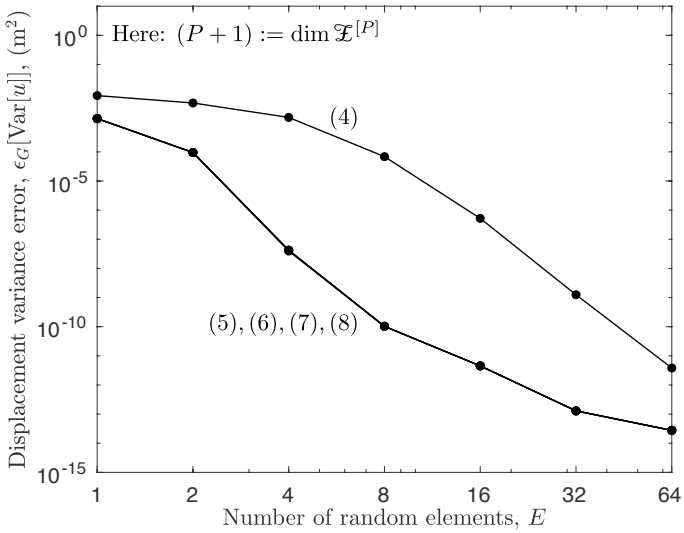


(b) Variance error for $\mathfrak{X}^{[7]}$

Figure 4.12. *Problem 2* — Local error evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different (h, p)-discretization levels of RFS and for $\mu \sim$ Uniform (SET 3/3)

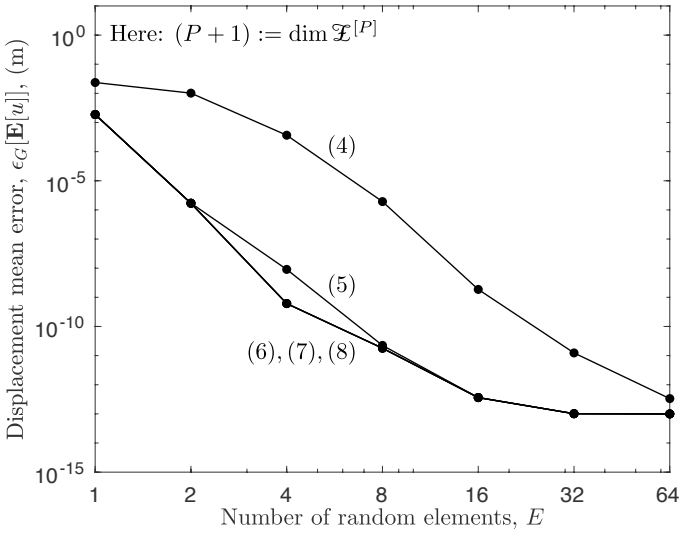


(a) Mean error for $\mu \sim$ Uniform

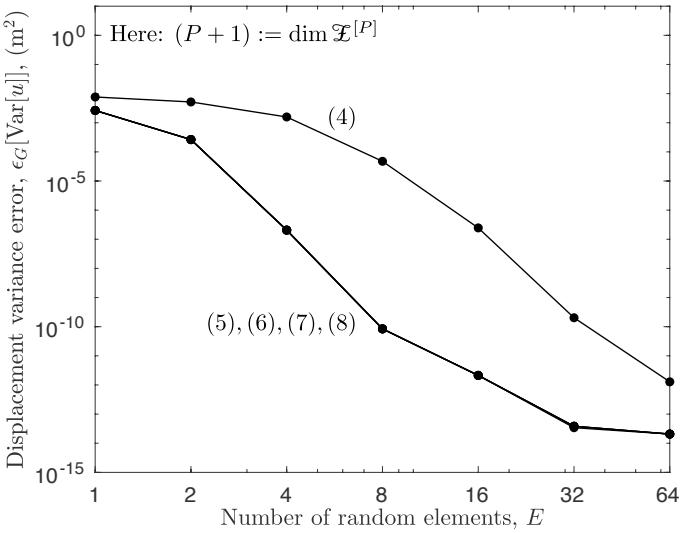


(b) Variance error for $\mu \sim$ Uniform

Figure 4.13. *Problem 2* — Global error of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different (h, p)-discretization levels of RFS ($\mu \sim$ Uniform)



(a) Mean error for $\mu \sim$ Beta



(b) Variance error for $\mu \sim$ Beta

Figure 4.14. *Problem 2* — Global error of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different (h, p)-discretization levels of RFS ($\mu \sim$ Beta)

4.6.3. Problem 3: A nonlinear system governed by a 2nd-order stochastic ODE (the Van-der-Pol oscillator)

In this section, we investigate the nonlinear behavior of a single-degree-of-freedom system with Van-der-Pol damping. The governing differential equation for this system is given by

$$m\ddot{u} - (1 - \rho u^2)c\dot{u} + ku = 0,$$

where $m = 100$ kg is the mass of the system, $\rho = 150$ m^{-2} is the contributing factor to the nonlinearity of the system, $c : \Xi \to \mathbb{R}^+$ is a coefficient representing the strength of the damping in the system, and $k : \Xi \to \mathbb{R}^+$ is the stiffness of the system. The functions c and k are assumed to be given by $c(\xi) = \xi^1$ and $k(\xi) = \xi^2$. The initial conditions of the system are: $u(0, \cdot) \equiv 0.20$ m and $\dot{u}(0, \cdot) \equiv 0.30$ m/s. Notice here that $u : \mathfrak{T} \times \Xi \to \mathbb{R}$ denotes the displacement of the system, and that $\dot{u} := \partial_t u$ and $\ddot{u} := \partial_t^2 u$ represent, respectively, the velocity and acceleration of the system.

For concreteness, we take ξ^1 to be *uniformly distributed* in $\bar{\Xi}_1 = [150, 450]$ kg/s, and ξ^2 to be *beta distributed* with parameters $(\alpha, \beta) = (2, 5)$ in $\bar{\Xi}_2 = [340, 460]$ N/m. Hence, the random space is two-dimensional and defined by $\Xi = \bar{\Xi}_1 \times \bar{\Xi}_2$ with $\mu \sim \text{Uniform} \otimes \text{Beta}$.

Fig. 4.15 depicts the evolution of the mean and variance of the system's displacement using ME-FSC and a Monte Carlo simulation with one million realizations. The reason why this time we use a Monte Carlo simulation as the reference solution is that a closed-form solution for u is not available. However, it is worth pointing out that one drawback of using Monte Carlo as the reference solution is that it will not allow us to compare the degree of accuracy obtained with ME-FSC adequately, since the solution obtained with ME-FSC may be far more accurate than the one given by Monte Carlo. Nonetheless, from this figure we learn that when 64 elements and 5 basis vectors are employed to run the simulation, the ME-FSC method is capable of reproducing the Monte Carlo solution quite well. To compare the level of accuracy of ME-FSC with respect to Monte Carlo, Figs. 4.16 to 4.18 present the local errors in mean and variance of the system's displacement. In general, good convergence can be observed when the number of elements and number of basis vectors are both increased. This observation can be better verified using Fig. 4.19 which plots the global errors in mean

and variance of the system's displacement. As expected, the results only improves if the number of basis vectors is increased up to a certain number, which in this case happens to be 5. However, contrary to what we observed in Problems 1 and 2, increasing the number of elements does not necessarily improve the accuracy of the results, as can be confirmed when $P+1$ is set to 4. This suggests that due to the nonlinear nature of the system's ODE, higher basis vectors can play a major role in the description of the system's state over time.

4.6.4. Problem 4: A nonlinear system governed by a system of 1st-order stochastic ODEs (the Kraichnan-Orszag three-mode problem)

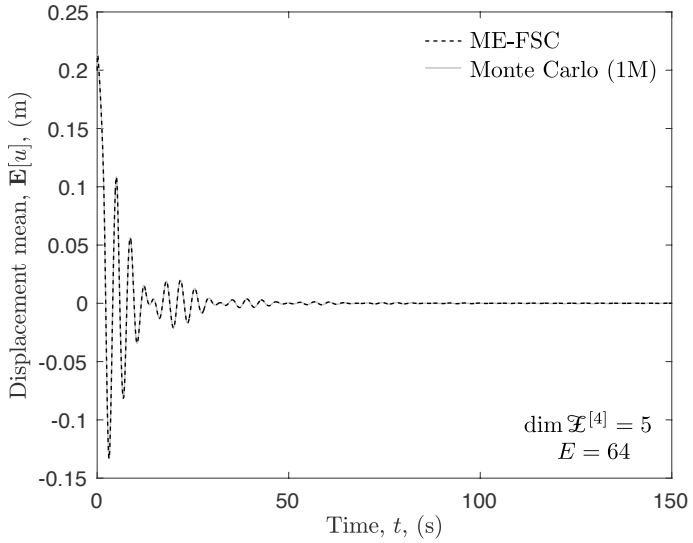
In this last problem, we explore the so-called Kraichnan-Orszag three-mode problem [95] in order to test the performance of the ME-FSC method more throughly. This problem is particularly challenging for methods based on the spectral approach because the solution is known to be discontinuous over the random domain. It has been used as a benchmark problem in various works (e.g. [34, 35, 43]), and so here we also opt to study it using the ME-FSC method. The three-mode problem considered in this work is the same as the one presented in [43] (Pag. 635, Section 4.3.4).

After performing a thoughtful 45° rotation of the random domain (specifically, about the u_3-axis), the solution becomes discontinuous on $u_1 = 0$ and $u_2 = 0$, and the system's governing differential equation reads

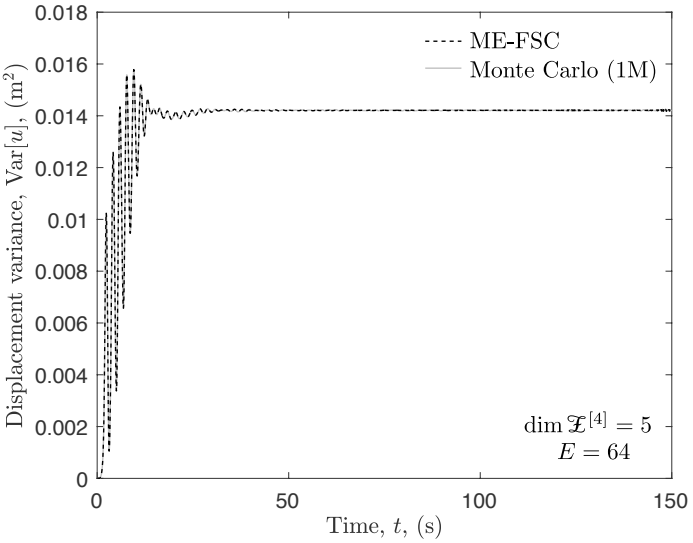
$$\begin{aligned}
\dot{u}_1 &= u_1 u_3, \\
\dot{u}_2 &= -u_2 u_3, \\
\dot{u}_3 &= -u_1^2 + u_2^2,
\end{aligned}$$

where $u_1, u_2, u_3 : \mathfrak{T} \times \Xi \to \mathbb{R}$ represent the three modes of the system, and $\dot{u}_1 := \partial_t u_1, \dot{u}_2 := \partial_t u_2, \dot{u}_3 := \partial_t u_3$ are the corresponding velocities. In this problem, we take the initial conditions of the system to be stochastic and given by: $u_1(0, \xi) = \xi^1$, $u_2(0, \xi) = \xi^2$, and $u_3(0, \xi) = \xi^3$.

Two probability distributions are investigated for $\xi = (\xi^1, \xi^2, \xi^3)$. The first one is a *uniform distribution* defined by $\text{Uniform}^{\otimes 3} \sim \xi \in \Xi = [a, b]^3$, and the second one is a *beta distribution*

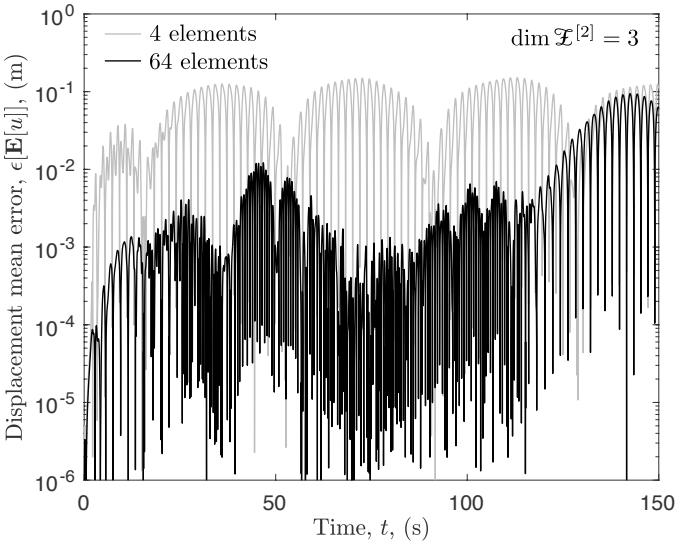


(a) Mean

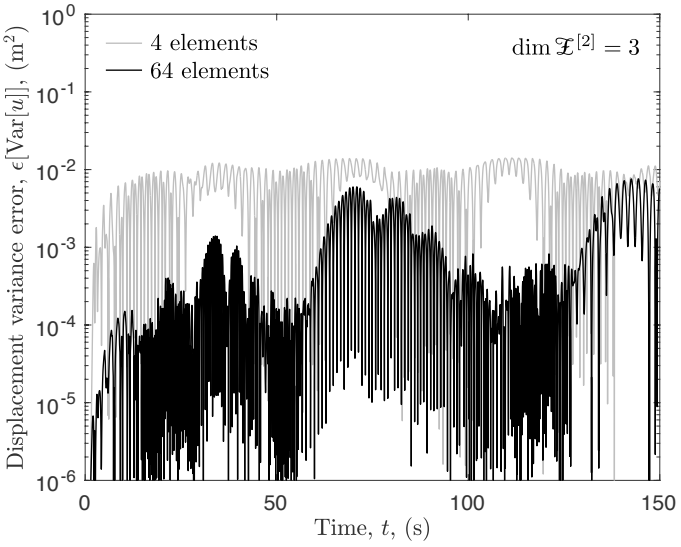


(b) Variance

Figure 4.15. *Problem 3* — Evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for the case when the (h, p)-discretization level of RFS is $(P, E) = (4, 64)$ and $\mu \sim \text{Uniform} \otimes \text{Beta}$

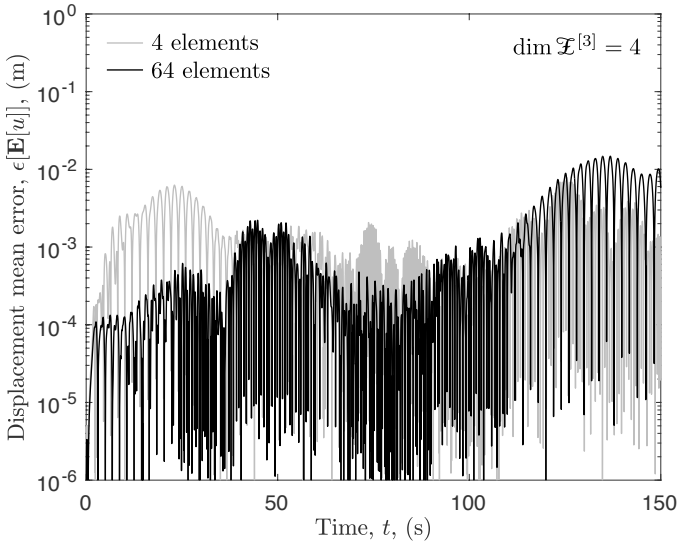


(a) Mean error for $\mathfrak{X}^{[2]}$

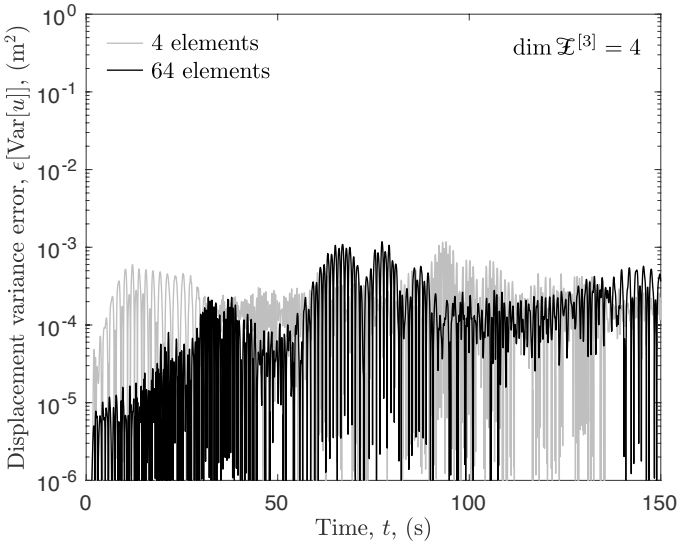


(b) Variance error for $\mathfrak{X}^{[2]}$

Figure 4.16. *Problem 3* — Local error evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different (h, p)-discretization levels of RFS and for $\mu \sim \text{Uniform} \otimes \text{Beta}$ (SET 1/3)

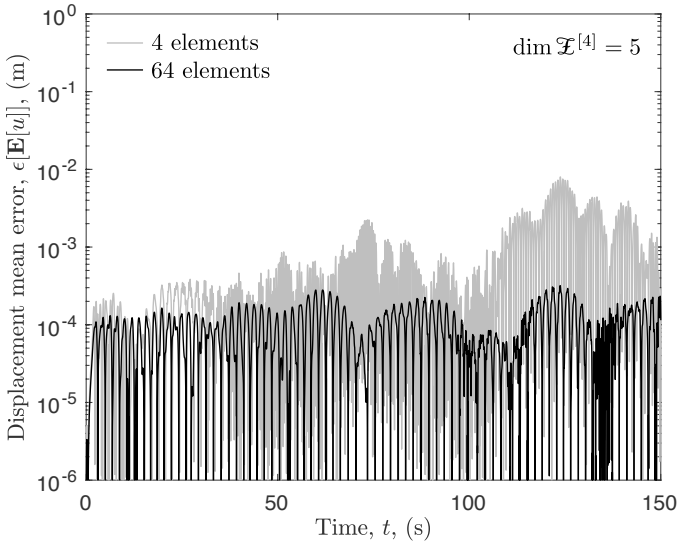


(a) Mean error for $\mathfrak{Z}^{[3]}$

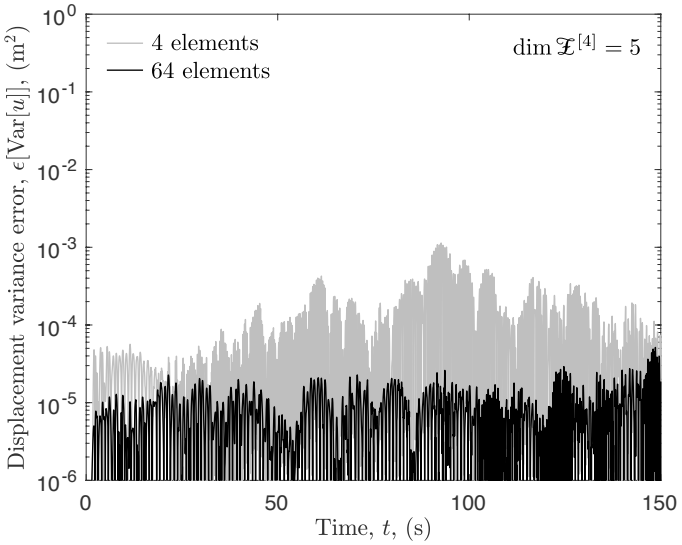


(b) Variance error for $\mathfrak{Z}^{[3]}$

Figure 4.17. *Problem 3* — Local error evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different (h, p)-discretization levels of RFS and for $\mu \sim \text{Uniform} \otimes \text{Beta}$ (SET 2/3)

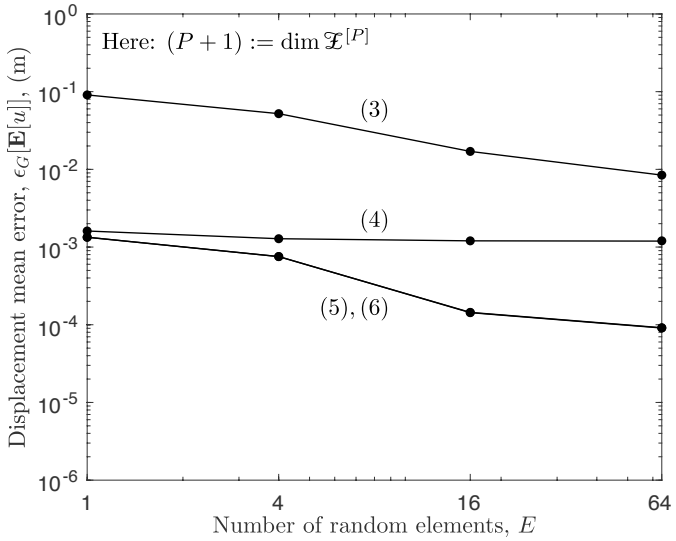


(a) Mean error for $\mathfrak{X}^{[4]}$

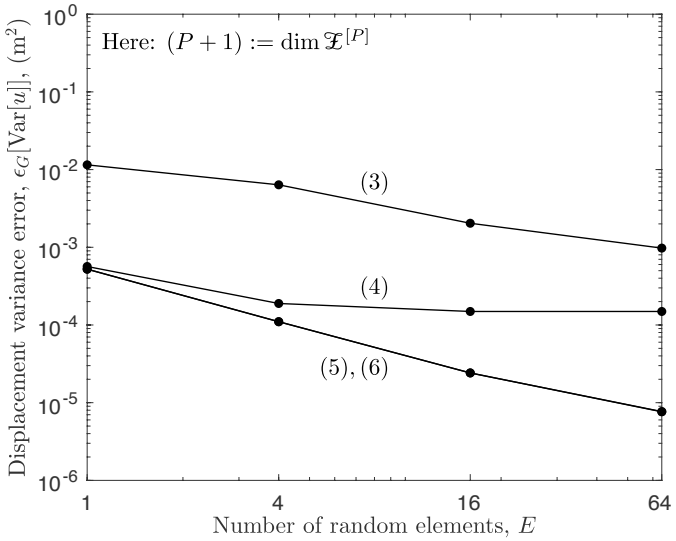


(b) Variance error for $\mathfrak{X}^{[4]}$

Figure 4.18. *Problem 3* — Local error evolution of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different (h, p)-discretization levels of RFS and for $\mu \sim \text{Uniform} \otimes \text{Beta}$ (SET 3/3)



(a) Mean error



(b) Variance error

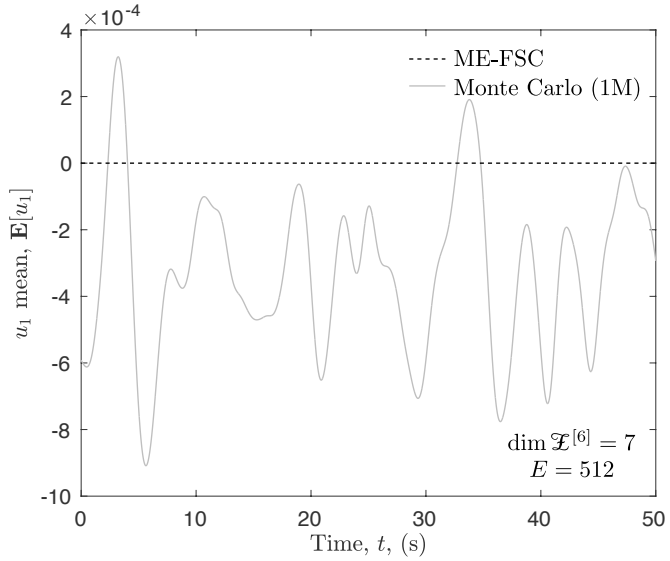
Figure 4.19. *Problem 3* — Global error of $\mathbf{E}[u]$ and $\mathrm{Var}[u]$ for different (h, p)-discretization levels of RFS and for $\mu \sim \text{Uniform} \otimes \text{Beta}$

defined by $\text{Beta}(\alpha,\beta)^{\otimes 3} \sim \xi \in \Xi = [a,b]^3$, from where $(a,b) = (-1,1)$ and $(\alpha,\beta) = (2,5)$. The random space is thus three-dimensional.

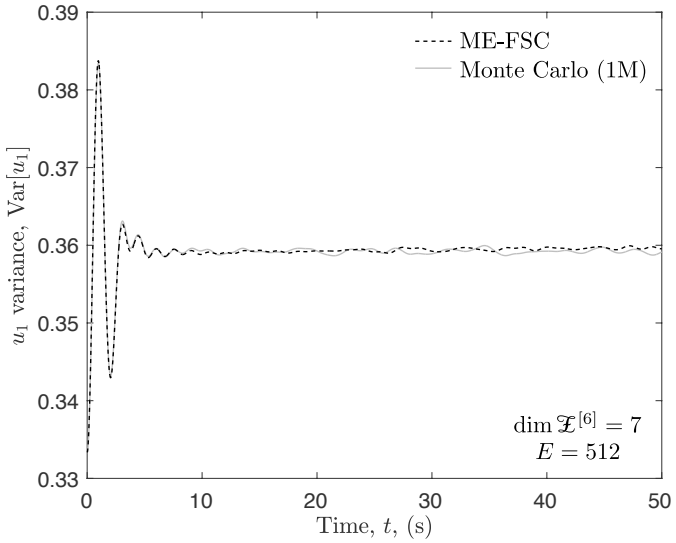
In Figs. 4.20 to 4.23 we depict the evolution of the mean and variance of u_1 and u_3 for the two distributions chosen for ξ. The results are obtained for the case of using 512 elements and 7 basis vectors per element, and then compared against a Monte Carlo simulation with one million realizations. As observed, the ME-FSC solution is in good agreement with the Monte Carlo solution. However, from Figs. 4.20a and 4.21a we notice that the ME-FSC solution is much more accurate than Monte Carlo, since the exact mean of u_1, u_2 and u_3 are known to be identically equal to zero when the probability measure is uniform. As a result, the Monte Carlo solution is demonstrated to be only 3 to 4 orders of magnitude accurate with respect to the exact solution.

Moreover, Figs. 4.24 to 4.29 present the local and global errors in mean and variance of u_1 and u_3 for the two distributions chosen for ξ. From these plots we see that increasing the number of elements help improve the accuracy of the ME-FSC results with respect to Monte Carlo. However, because the Monte Carlo solution is not exact, the errors have the tendency to stagnate around 10^{-3} and 10^{-4} when the ME-FSC results are compared against Monte Carlo. This is the reason why, in the case of the mean, the accuracy of the ME-FSC results does not improve as the number of elements increases from 8 to 512; but, in the case of the variance, they do improve because the errors obtained with ME-FSC are above 10^{-4}. Therefore, to achieve comparable solution accuracy to Monte Carlo, around 512 elements are needed to solve this problem reasonably well with ME-FSC.

For this particular problem, the computational cost associated with a simulation with 512 elements and 7 basis vectors per element was 82 seconds, whereas the computational cost associated with a Monte Carlo simulation with one million realizations was 97 seconds. This outcome indicates that simulations conducted with ME-FSC were at least 15% faster than those conducted with Monte Carlo. For higher-dimensional probability spaces, it would be necessary to implement a different quadrature rule that does not suffer from the curse of dimensionality to help speed up the computation of the inner products (which is where the bottleneck of the ME-FSC method actually is), and thus make the computational cost of ME-FSC an order of magnitude lower than Monte Carlo for comparable solution accuracy.

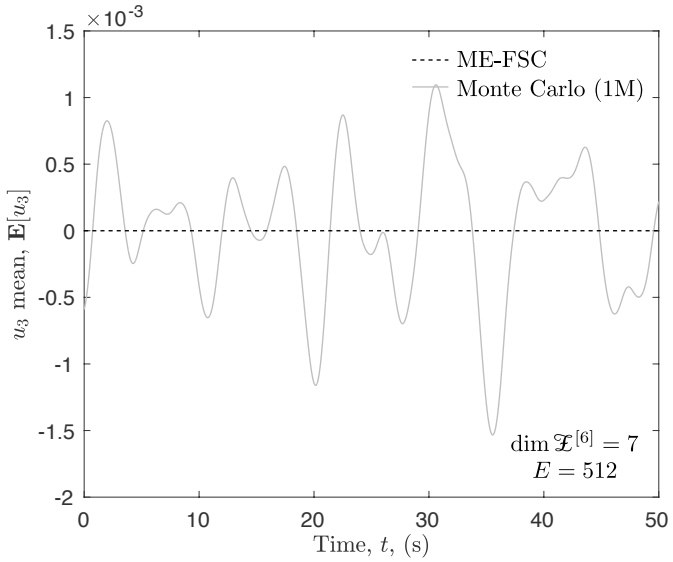


(a) Mean for u_1

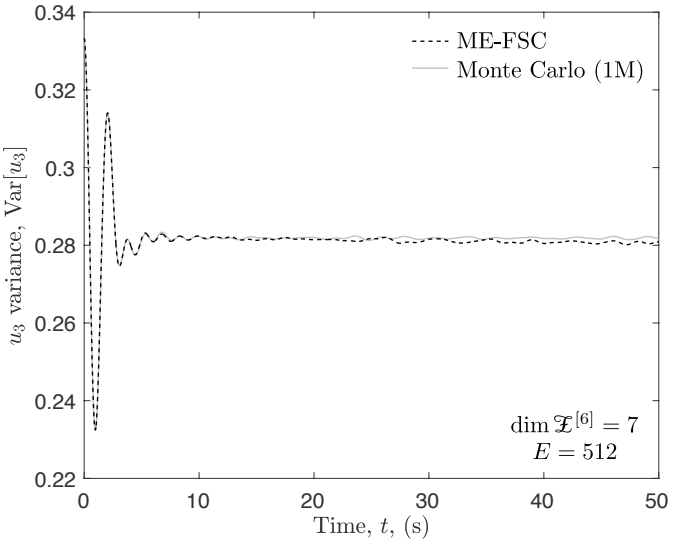


(b) Variance for u_1

Figure 4.20. *Problem 4* — Evolution of $\mathbf{E}[u_1]$ and $\mathrm{Var}[u_1]$ for the case when the (h, p)-discretization level of RFS is $(P, E) = (6, 512)$ and $\mu \sim \mathrm{Uniform}^{\otimes 3}$

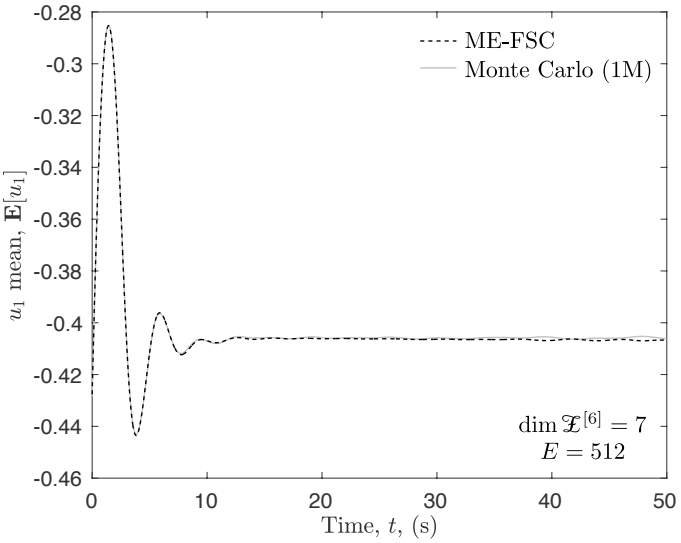


(a) Mean for u_3

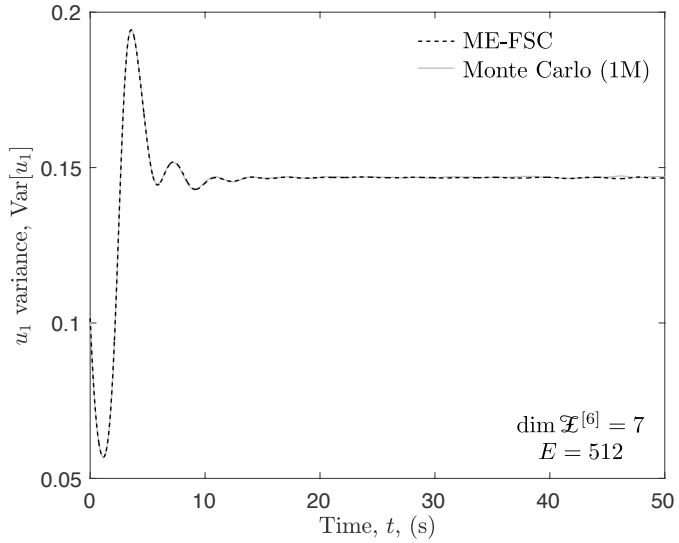


(b) Variance for u_3

Figure 4.21. *Problem 4* — Evolution of $\mathbf{E}[u_3]$ and $\mathrm{Var}[u_3]$ for the case when the (h, p)-discretization level of RFS is $(P, E) = (6, 512)$ and $\mu \sim \mathrm{Uniform}^{\otimes 3}$

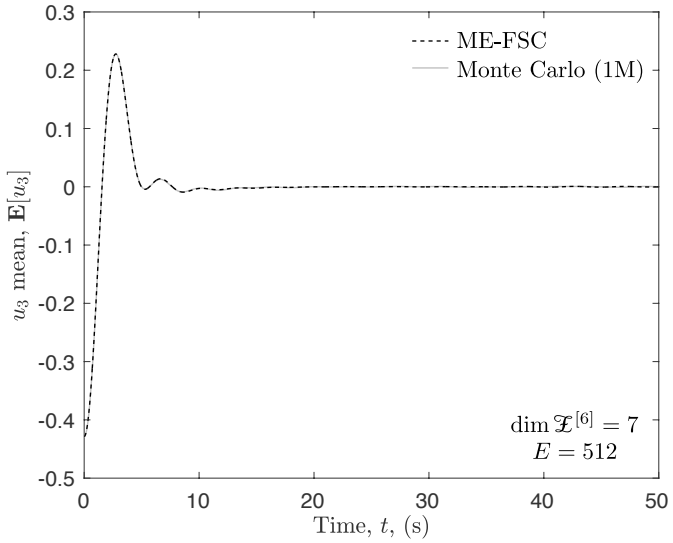


(a) Mean for u_1

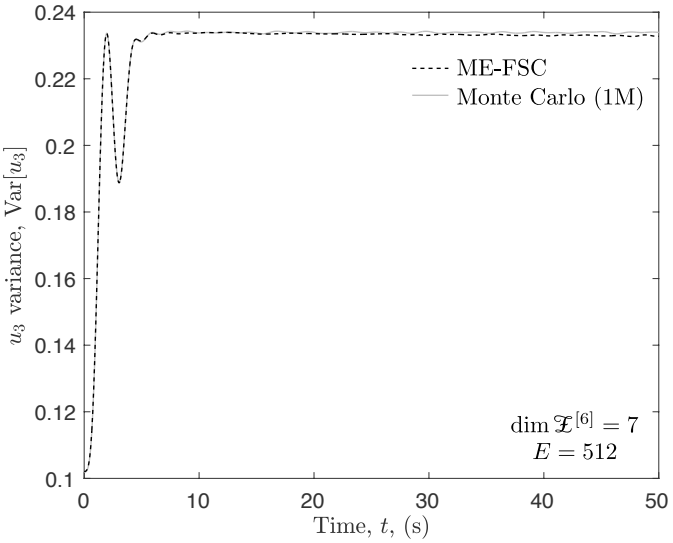


(b) Variance for u_1

Figure 4.22. *Problem 4* — Evolution of $\mathbf{E}[u_1]$ and $\mathrm{Var}[u_1]$ for the case when the (h, p)-discretization level of RFS is $(P, E) = (6, 512)$ and $\mu \sim \mathrm{Beta}^{\otimes 3}$

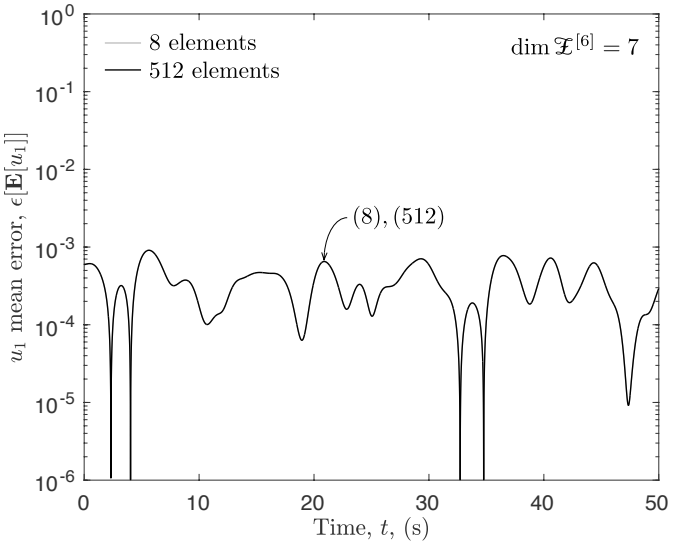


(a) Mean for u_3

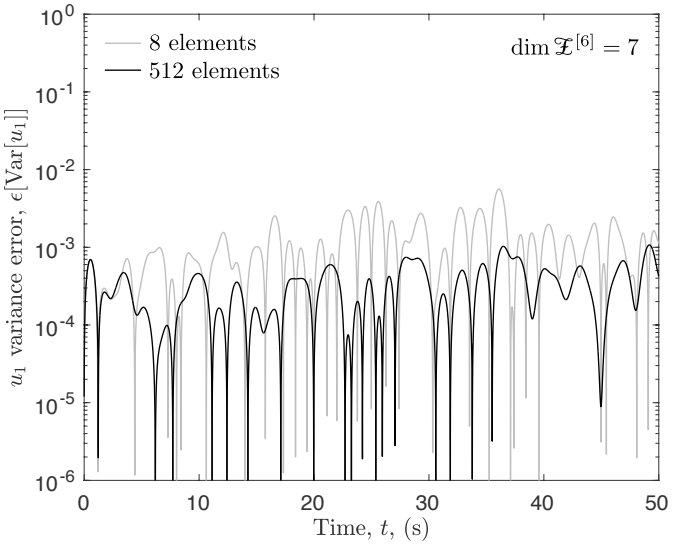


(b) Variance for u_3

Figure 4.23. *Problem 4* — Evolution of $\mathbf{E}[u_3]$ and $\mathrm{Var}[u_3]$ for the case when the (h, p)-discretization level of RFS is $(P, E) = (6, 512)$ and $\mu \sim \mathrm{Beta}^{\otimes 3}$

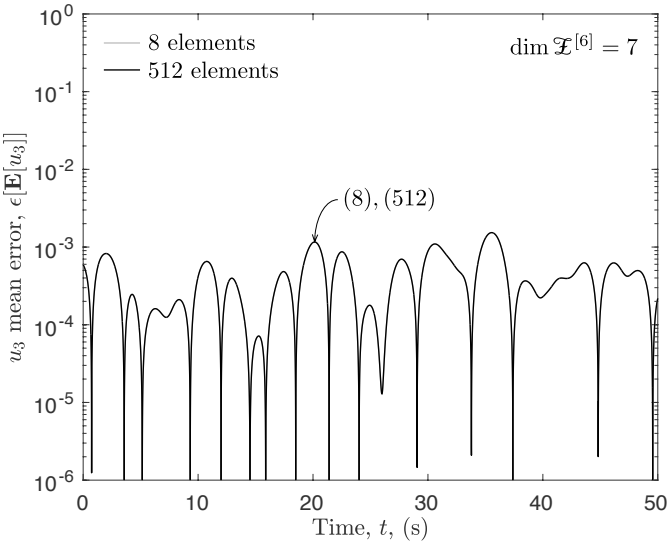


(a) Mean error for u_1

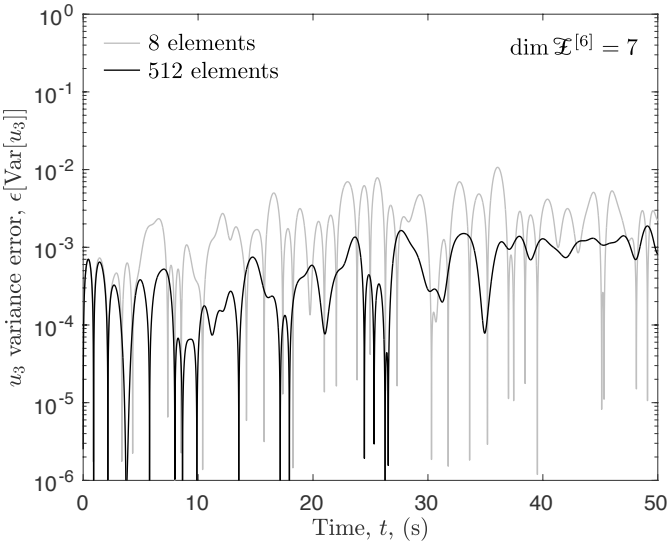


(b) Variance error for u_1

Figure 4.24. *Problem 4* — Local error evolution of $\mathbf{E}[u_1]$ and $\mathrm{Var}[u_1]$ for different (h, p)-discretization levels of RFS and for $\mu \sim \mathrm{Uniform}^{\otimes 3}$

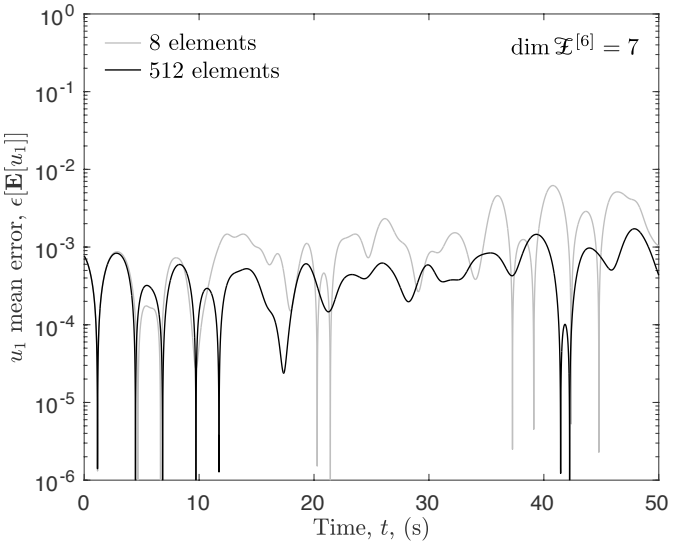


(a) Mean error for u_3

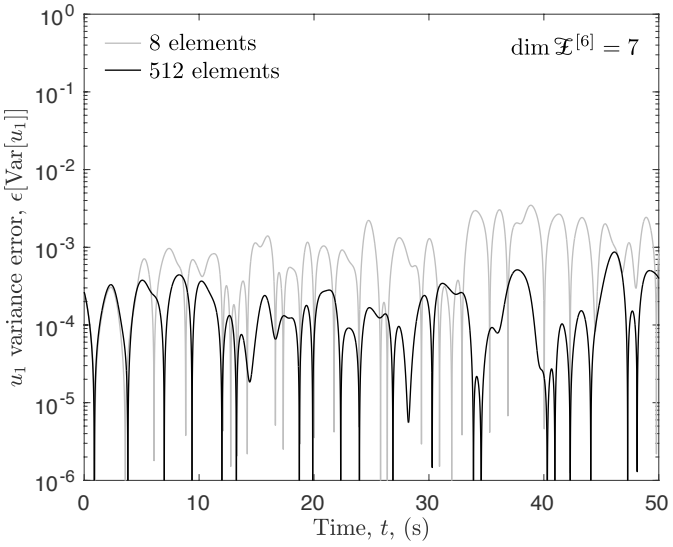


(b) Variance error for u_3

Figure 4.25. *Problem 4* — Local error evolution of $\mathbf{E}[u_3]$ and $\mathrm{Var}[u_3]$ for different (h, p)-discretization levels of RFS and for $\mu \sim \mathrm{Uniform}^{\otimes 3}$

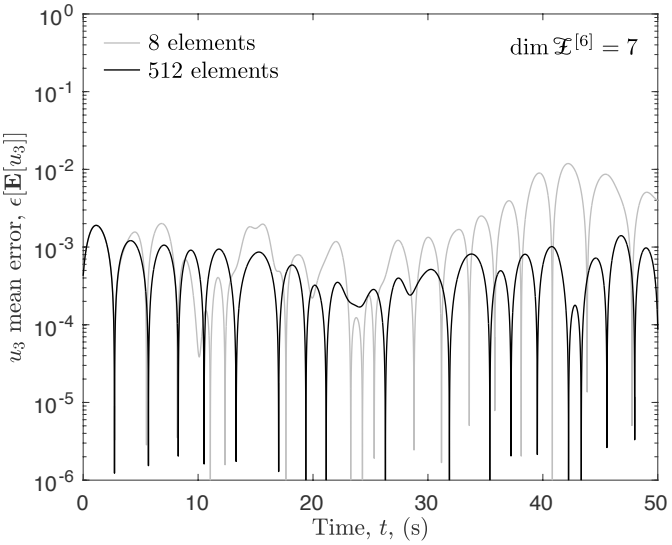


(a) Mean error for u_1

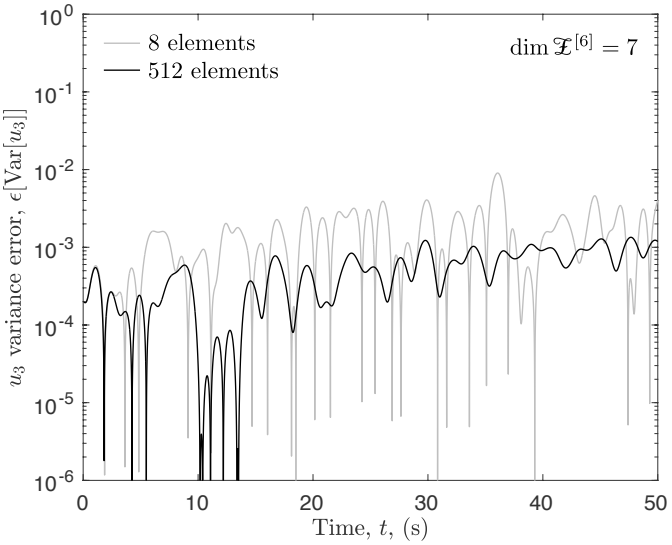


(b) Variance error for u_1

Figure 4.26. *Problem 4* — Local error evolution of $\mathbf{E}[u_1]$ and $\mathrm{Var}[u_1]$ for different (h, p)-discretization levels of RFS and for $\mu \sim \mathrm{Beta}^{\otimes 3}$

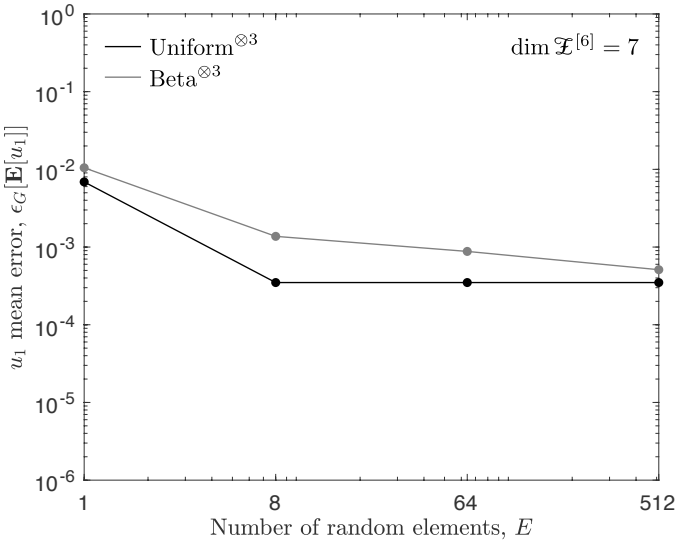


(a) Mean error for u_3

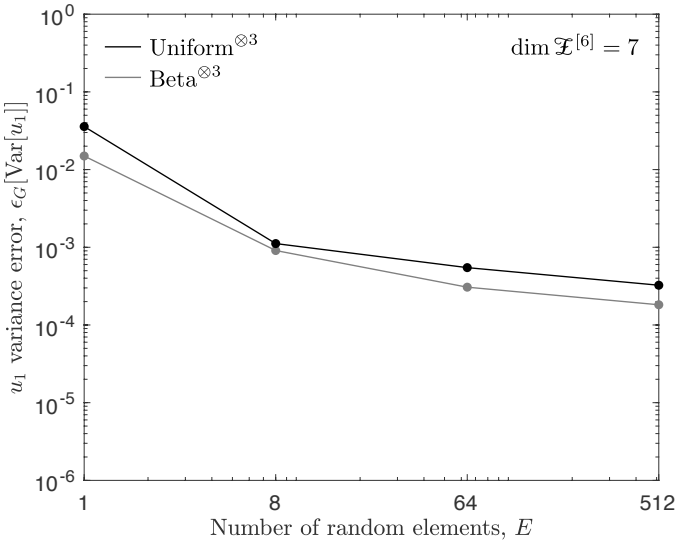


(b) Variance error for u_3

Figure 4.27. *Problem 4* — Local error evolution of $\mathbf{E}[u_3]$ and $\mathrm{Var}[u_3]$ for different (h, p)-discretization levels of RFS and for $\mu \sim \mathrm{Beta}^{\otimes 3}$

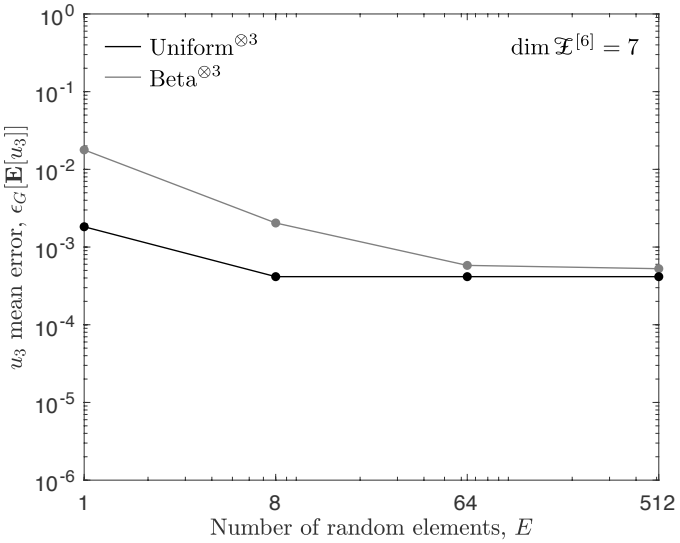


(a) Mean error for u_1

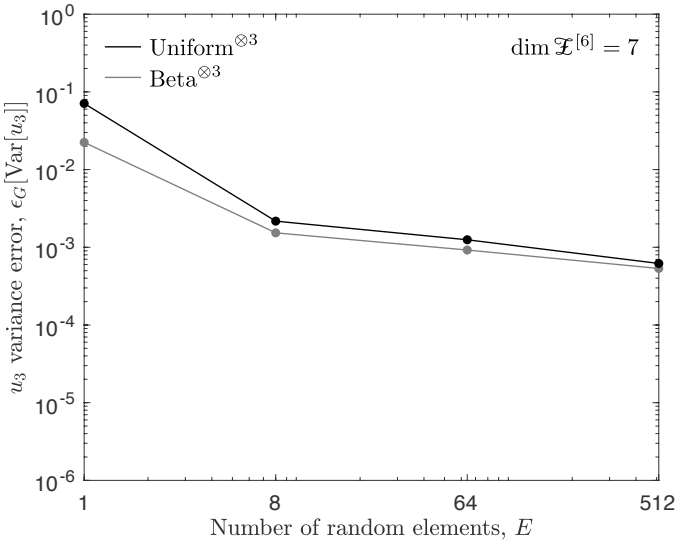


(b) Variance error for u_1

Figure 4.28. *Problem 4* — Global error of $\mathbf{E}[u_1]$ and $\mathrm{Var}[u_1]$ for different (h, p)-discretization levels of RFS



(a) Mean error for u_3



(b) Variance error for u_3

Figure 4.29. *Problem 4* — Global error of $\mathbf{E}[u_3]$ and $\mathrm{Var}[u_3]$ for different (h, p)-discretization levels of RFS

4.7. Conclusion

In this paper we have presented an extension of the FSC method [92], called the *multi-element flow-driven spectral chaos* (ME-FSC) method, to deal with stochastic discontinuities and long-time integration of stochastic dynamical systems more efficiently. The method is also particularly useful when it comes to dealing with large stochastic dynamical systems, since it allows the analyst to decompose the problem into several small subproblems and thus the possibility to solve each of them separately. Furthermore, if each subproblem were to be solved simultaneously on machines with multiple CPU cores or on separate machines, the solution could be obtained in a more reasonable amount of time, making therefore the multi-element approach more appealing than FSC in many other instances.

The key idea behind the ME-FSC method is no different than the ME-gPC method [43]. In ME-FSC, the random domain is partitioned into several elements, and then the FSC method is implemented on each element. In a subsequent step, the results are aggregated and the probability moments of interest are computed using the law of total probability. To ensure that the probability information is transferred exactly from one random function space to another, the FSC-2 approach is implemented on each random element. The significance of implementing the FSC-2 approach into the ME-FSC scheme is that it helps reduce by several orders of magnitude the error propagation over time.

Four representative problems were investigated in this paper. The first two problems dealt with systems governed by a linear stochastic ODE to enable us to obtain an exact solution with which to compare the ME-FSC results against. The third problem dealt with a system governed by a nonlinear stochastic ODE (the Van-der-Pol oscillator) to study the effectiveness of the ME-FSC method in the resolution of nonlinear problems. The last fourth problem dealt with a system governed by a nonlinear system of stochastic ODEs (the Kraichnan-Orszag three-mode problem) so that the performance of ME-FSC could be investigated more throughly. Based on our findings, we can conclude that the ME-FSC method is capable of reproducing the exact solution with high fidelity, and for those problems with no closed-form solutions, the ME-FSC method is capable of resembling reasonably well a

Monte Carlo simulation with one million realizations using a fraction of the computational cost required to do so.

5. MODAL FLOW-DRIVEN SPECTRAL CHAOS (MFSC) METHOD FOR LONG-TIME INTEGRATION OF STOCHASTIC DYNAMICS OF STRUCTURES

5.1. Introduction

The objective of this chapter is to demonstrate that the excessive computational burden of a stochastic simulation can be greatly alleviated if the spatial random function space is discretized using the first few vibration modes of the system. For efficiency reasons, we continue discretizing the random function space with the FSC method. This new technique, called the *modal flow-driven spectral chaos* (MFSC) method, is the author's second attempt to further reduce the dimensionality of the random function space whenever we have systems with several degrees of freedom. The demonstration is conducted herein by studying the dynamic response of a stochastic, linear structural system using the spectral approach. The mathematical spaces employed in this chapter can be readily found in Section 3.2 (Definitions 1 to 5).

5.2. Problem statement

A system of non-autonomous, second-order linear ODEs in the context of stochastic dynamics of structures is considered herein. The problem can be stated formally as follows.

Find the displacement of the system $\mathbf{u} : \mathfrak{T} \times \Xi \to \mathbb{R}^R$, such that each entry of $\mathbf{u}$ is in $\mathcal{U}$ and (μ-a.e.):

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{p} \qquad \text{on } \mathfrak{T} \times \Xi \tag{5.1a}$$

$$\Big\{\mathbf{u}(0,\cdot) = \boldsymbol{u},\, \dot{\mathbf{u}}(0,\cdot) = \boldsymbol{v}\Big\} \qquad \text{on } \{0\} \times \Xi, \tag{5.1b}$$

where $\mathbf{M}, \mathbf{C}, \mathbf{K} : \Xi \to \mathrm{L}(\mathbb{R}^R, \mathbb{R}^R)$ are (respectively) the mass matrix, the damping matrix and the stiffness matrix of the system, $\mathbf{p} : \mathfrak{T} \times \Xi \to \mathbb{R}^R$ is the external force acting on the system, $R \in \mathbb{N}_1$ symbolizes the number of degrees of freedom that the system possesses, and $\boldsymbol{u}, \boldsymbol{v} : \Xi \to \mathbb{R}^R$ are (respectively) the initial displacement and the initial velocity of

the system. Observe that $\dot{\mathbf{u}} := \partial_t \mathbf{u}$ denotes the velocity of the system, and $\ddot{\mathbf{u}} := \partial_t^2 \mathbf{u}$ the acceleration of the system.

In the discussion that follows, $\mathbf{M}(\xi) \neq \mathbf{0}$ represents a symmetric matrix for all $\xi \in \Xi$.

5.3. An overview of the eigenproblem in structural dynamics

In structural dynamics, the generalized eigenproblem discussed below is of special interest since its solution gives the natural frequencies and modes of an undamped structural system. This section is devoted to revise the eigenproblem encountered in structural dynamics using the spectral approach. In Section 5.4, we use the modes of the system to reduce considerably the number of degrees of freedom involved in the mathematical model of the system.

5.3.1. Eigenproblem statement

Find the eigenpair $(\lambda_m, \boldsymbol{\phi}_m) : \Xi \to \mathbb{R}^{R+1}$, such that λ_m and each entry of $\boldsymbol{\phi}_m$ are in $\mathfrak{Z}$ and (μ-a.e.):

$$\mathbf{K}\boldsymbol{\phi}_m = \lambda_m \mathbf{M}\boldsymbol{\phi}_m \qquad \text{on } \Xi \tag{5.2a}$$

$$\boldsymbol{\phi}_m^T \mathbf{M}\boldsymbol{\phi}_m = \rho \qquad \text{on } \Xi, \tag{5.2b}$$

where $\mathbf{M}$ and $\mathbf{K}$ are the mass and stiffness matrices as defined in Section 5.2, $\lambda : \Xi \to \mathbb{R}$ is the m-th eigenvalue of the system, $\boldsymbol{\phi}_m : \Xi \to \mathbb{R}^R$ is the corresponding eigenvector, and $\rho \in \mathbb{R}^+$ is a prescribed positive real number (typically taken as 1 in the literature). Here the superscript ^{T} stands for the transpose operation found in linear algebra.

5.3.2. Spectral solution of eigenproblem

The eigenproblem is solved below using the spectral approach. The solution is similar to that presented by Ghanem and Ghosh in [96] with the only difference being that here there is an explicit mass matrix defined in the problem statement. This is the reason why in this work we orthogonalize the eigenvectors with respect to the mass matrix (via (5.2b)) and not with respect to the identity matrix (as in [96]).

In tensor notation, system (5.2) can be rewritten as:

$$\mathbf{K}[\,\cdot\,,\boldsymbol{\phi}_m] = \lambda_m\,\mathbf{M}[\,\cdot\,,\boldsymbol{\phi}_m] \qquad \Leftrightarrow \qquad K^u{}_v\phi^v{}_m = \lambda_m M^u{}_v\phi^v{}_m \tag{5.3a}$$

$$\boldsymbol{\delta}[\boldsymbol{\phi}_m,\mathbf{M}[\,\cdot\,,\boldsymbol{\phi}_m]] = \rho \qquad \Leftrightarrow \qquad \delta_{uw}\phi^u{}_m M^w{}_v\phi^v{}_m = \rho, \tag{5.3b}$$

where $\mathbf{M} = M^u{}_v\,\mathbf{e}_u\otimes\mathbf{e}^v$, $\mathbf{K} = K^u{}_v\,\mathbf{e}_u\otimes\mathbf{e}^v : \mathbb{R}^{R\prime}\times\mathbb{R}^R\to\mathfrak{Z}$ are (respectively) the spatial mass tensor field and the spatial stiffness tensor field of the system on Ξ, $\mathbf{e}_u\in\mathbb{R}^R$ is the standard unit vector associated with the u-th dimension, $\mathbf{e}^v : \mathbb{R}^R\to\mathbb{R}$ is the standard covector associated with the v-th dimension, $\mathbb{R}^{R\prime}$ is the dual space of $\mathbb{R}^R$ with $\mathbf{e}^v[\mathbf{e}_u] = \delta^v{}_u$, and $\boldsymbol{\delta} = \delta_{uv}\,\mathbf{e}^u\otimes\mathbf{e}^v : \mathbb{R}^R\times\mathbb{R}^R\to\mathbb{R}$ is the standard metric tensor endowed on $\mathbb{R}^R$. In these expressions, $\delta^v{}_u$ and δ_{uv} are two Kronecker deltas, $\boldsymbol{\phi}_m = \phi^v{}_m\mathbf{e}_v$ is a spatial vector field on Ξ, and $M^u{}_v, K^u{}_v, \phi^v{}_m, \lambda_m : \Xi\to\mathbb{R}$ are random functions in $\mathfrak{Z}$ given by

$$M^u{}_v = M^u{}_v(\xi), \quad K^u{}_v = K^u{}_v(\xi), \quad \phi^v{}_m = \phi^v{}_m(\xi) \quad \text{and} \quad \lambda_m = \lambda_m(\xi)$$

for all $u,v\in\{1,2,\ldots,R\}$.

Since it is already presumed that $\lambda_m,\phi^v{}_m\in\mathfrak{Z}$, these functions can be represented in $\mathfrak{Z} = \operatorname{span}\{\tilde{\Psi}_j\}_{j=0}^{\infty}$ by the Fourier series:

$$\lambda_m(\xi) = \sum_{j=0}^{\infty}\lambda_m{}^j\,\tilde{\Psi}_j(\xi) \equiv \lambda_m{}^j\,\tilde{\Psi}_j(\xi) \tag{5.4a}$$

$$\phi^v{}_m(\xi) = \sum_{j=0}^{\infty}\phi^v{}_m{}^j\,\tilde{\Psi}_j(\xi) \equiv \phi^v{}_m{}^j\,\tilde{\Psi}_j(\xi). \tag{5.4b}$$

Substituting (5.4) into (5.3) gives

$$K^u{}_v\phi^v{}_m{}^j\tilde{\Psi}_j = \lambda_m{}^k\tilde{\Psi}_k M^u{}_v\phi^v{}_m{}^j\tilde{\Psi}_j \tag{5.5a}$$

$$\delta_{uw}\phi^u{}_m{}^j\tilde{\Psi}_j M^w{}_v\phi^v{}_m{}^k\tilde{\Psi}_k = \rho. \tag{5.5b}$$

Projecting (5.5) onto $\mathfrak{Z}$ implies

$$\tilde{\Psi}^i[K^u{}_v\phi^v{}_m{}^j\tilde{\Psi}_j] = \tilde{\Psi}^i[\lambda_m{}^k\tilde{\Psi}_k M^u{}_v\phi^v{}_m{}^j\tilde{\Psi}_j] \tag{5.6a}$$

$$\tilde{\Psi}^i[\delta_{uw}\phi^u{}_m{}^j\tilde{\Psi}_j M^w{}_v\phi^v{}_m{}^k\tilde{\Psi}_k] = \tilde{\Psi}^i[\rho], \tag{5.6b}$$

which after simplification yields the following system of (infinitely many) nonlinear equations:

$$\tilde{K}^{u\ i}_{v\ j}\phi^{v}_{\ m}{}^{j} = \tilde{M}^{u\ i}_{v\ jk}\phi^{v}_{\ m}{}^{j}\lambda_{m}{}^{k} \tag{5.7a}$$

$$\delta_{uw}\tilde{M}^{w\ i}_{\ v\ jk}\phi^{u}_{\ m}{}^{j}\phi^{v}_{\ m}{}^{k} = \rho\,\tilde{\delta}^{i}_{0}, \tag{5.7b}$$

where $\tilde{M}^{u\ i}_{v\ jk}, \tilde{K}^{u\ i}_{v\ j}, \tilde{\delta}^{i}_{0} \in \mathbb{R}$ are real numbers given by

$$\tilde{M}^{u\ i}_{v\ jk} = \frac{\langle\tilde{\Psi}_i, M^{u}_{\ v}\tilde{\Psi}_j\tilde{\Psi}_k\rangle}{\langle\tilde{\Psi}_i,\tilde{\Psi}_i\rangle}, \quad \tilde{K}^{u\ i}_{v\ j} = \frac{\langle\tilde{\Psi}_i, K^{u}_{\ v}\tilde{\Psi}_j\rangle}{\langle\tilde{\Psi}_i,\tilde{\Psi}_i\rangle} \quad \text{and} \quad \tilde{\delta}^{i}_{0} = \frac{\langle\tilde{\Psi}_i, 1\rangle}{\langle\tilde{\Psi}_i,\tilde{\Psi}_i\rangle}.$$

Remark 5.1. The real numbers $\tilde{M}^{u\ i}_{v\ jk}$ and $\tilde{K}^{u\ i}_{v\ j}$ represent the components of the random-spatial tensors:

$$\tilde{\boldsymbol{M}} = \tilde{M}^{u\ i}_{v\ jk}\,\mathbf{e}_u \otimes \mathbf{e}^v \otimes \tilde{\Psi}_i \otimes \tilde{\Psi}^j \otimes \tilde{\Psi}^k : \mathbb{R}^{R'} \times \mathbb{R}^R \times \mathfrak{Z}' \times \mathfrak{Z}^2 \to \mathbb{R}$$

$$\tilde{\boldsymbol{K}} = \tilde{K}^{u\ i}_{v\ j}\,\mathbf{e}_u \otimes \mathbf{e}^v \otimes \tilde{\Psi}_i \otimes \tilde{\Psi}^j : \mathbb{R}^{R'} \times \mathbb{R}^R \times \mathfrak{Z}' \times \mathfrak{Z} \to \mathbb{R},$$

where $\tilde{\boldsymbol{M}}$ is symmetric in the indices j and k. Furthermore, the associated tensor:

$$\tilde{\boldsymbol{M}}^{\flat_1} = \tilde{M}_{uv\ jk}^{\ \ \ i}\,\mathbf{e}^u \otimes \mathbf{e}^v \otimes \tilde{\Psi}_i \otimes \tilde{\Psi}^j \otimes \tilde{\Psi}^k : \mathbb{R}^{R} \times \mathbb{R}^R \times \mathfrak{Z}' \times \mathfrak{Z}^2 \to \mathbb{R}$$

is symmetric not only in the indices j and k but also in the indices u and v, since $\mathbf{M}$ is a symmetric matrix. Here $\flat_1$ symbolizes the flat of $\tilde{\boldsymbol{M}}$ with respect to its first slot, giving thereby an expression for the components of $\tilde{\boldsymbol{M}}^{\flat_1}$:

$$\tilde{M}_{uv\ jk}^{\ \ \ i} = \delta_{uw}\tilde{M}^{w\ i}_{\ v\ jk},$$

where δ_{uw} denotes the components of the metric tensor endowed on $\mathbb{R}^R$.

Remark 5.2. If $\mathfrak{Z}$ is represented with $\tilde{\mathfrak{Z}}^{[P]} = \operatorname{span}\{\tilde{\Psi}_j\}_{j=0}^{P}$, then (5.7) is a system of $(R+1)(P+1)$ nonlinear equations with $(R+1)(P+1)$ unknowns. The unknowns being the two points[1]: $\lambda_m = (\lambda_m{}^0, \ldots, \lambda_m{}^P) \in \mathbb{R}^{P+1}$ and $\phi_m := (\phi^1_{\ m}{}^0, \ldots, \phi^R_{\ m}{}^P) \in \mathbb{R}^{R(P+1)}$.

[1]To avoid unnecessary complexity in notation, the symbol λ_m has two meanings in this work, namely: the random function λ_m and the $(P+1)$-dimensional point λ_m. However, the actual meaning of this symbol should be clear from the context it is used.

5.3.3. Iterative solution for system of nonlinear equations

This section is started under the assumption that $\mathfrak{L}$ has been discretized with $\tilde{\mathfrak{L}}^{[P]}$, and that the system of nonlinear equations can be solved numerically using the Newton-Raphson method for all $m \in \{1, 2, \ldots, R\}$ (as in [96]).

For this, we take $g^{ui}, h^i : \mathbb{R}^{(R+1)(P+1)} \to \mathbb{R}$ to be functions identically-equal-to-zero given by (from (5.7))

$$g^{ui}(x_m) = \tilde{K}^{u\ i}_{v\ j}\phi^{v}_{m}{}^{j} - \tilde{M}^{u\ i}_{v\ jk}\phi^{v}_{m}{}^{j}\lambda_{m}{}^{k} \equiv 0 \tag{5.8a}$$

$$h^i(x_m) = \delta_{uw}\tilde{M}^{w\ i}_{v\ jk}\phi^{u}_{m}{}^{j}\phi^{v}_{m}{}^{k} - \rho\,\tilde{\delta}^{i}_{0} \equiv 0, \tag{5.8b}$$

where $x_m = (\lambda_m, \phi_m) = (\lambda_m{}^0, \ldots, \lambda_m{}^P, \phi^1_m{}^0, \ldots, \phi^R_m{}^P) \in \mathbb{R}^{(R+1)(P+1)}$.

In addition, for notational convenience we take

$$f = (g, h) : \mathbb{R}^{(R+1)(P+1)} \to \mathbb{R}^{(R+1)(P+1)} \qquad :\Leftrightarrow \qquad f = f(x_m),$$

where

$$g = (g^{10}, \ldots, g^{ui}, \ldots, g^{RP}) : \mathbb{R}^{(R+1)(P+1)} \to \mathbb{R}^{R(P+1)} \qquad :\Leftrightarrow \qquad g = g(x_m)$$

$$h = (h^0, \ldots, h^i, \ldots, h^P) : \mathbb{R}^{(R+1)(P+1)} \to \mathbb{R}^{P+1} \qquad :\Leftrightarrow \qquad h = h(x_m).$$

Linearizing (5.8) about $x_m = \bar{x}_m$ and evaluating the resulting functions at $x_m = \hat{x}_m$ produces

$$f^\mu(\bar{x}_m) + \sum_{\nu=1}^{(R+1)(P+1)} \frac{\partial f^\mu}{\partial x_m{}^\nu}(\bar{x}_m)\,(\hat{x}_m{}^\nu - \bar{x}_m{}^\nu) = 0, \tag{5.9}$$

where $\mu \in \{1, 2, \ldots, (R+1)(P+1)\}$.

However, in matrix form, system (5.9) can also be expressed as

$$\mathbf{f}(\bar{x}_m) + \mathbf{F}(\bar{x}_m)\,(\hat{\mathbf{x}}_m - \bar{\mathbf{x}}_m) = \mathbf{0}, \tag{5.10}$$

where $\bar{\mathbf{x}}_m, \hat{\mathbf{x}}_m, \mathbf{f}(\bar{x}_m) \in \mathbb{R}^{(R+1)(P+1)}$ and $\mathbf{F}(\bar{x}_m) \in \mathrm{L}(\mathbb{R}^{(R+1)(P+1)}, \mathbb{R}^{(R+1)(P+1)})$ are defined by[2]

$$\bar{\mathbf{x}}_m = \left\{ \bar{x}_m{}^{\nu} \right\}, \quad \hat{\mathbf{x}}_m = \left\{ \hat{x}_m{}^{\nu} \right\},$$

$$\mathbf{f}(\bar{x}_m) = \left\{ f^{\mu}(\bar{x}_m) \right\} \equiv \begin{Bmatrix} g^{ui}(\bar{x}_m) \\ \\ h^{i}(\bar{x}_m) \end{Bmatrix} = \begin{Bmatrix} \tilde{K}^{u\ i}_{v\ j} \bar{\phi}^{v}_{m}{}^{j} - \tilde{M}^{u\ i}_{v\ jk} \bar{\phi}^{v}_{m}{}^{j} \bar{\lambda}_{m}{}^{k} \\ \\ \delta_{uw} \tilde{M}^{w\ i}_{v\ jk} \bar{\phi}^{u}_{m}{}^{j} \bar{\phi}^{v}_{m}{}^{k} - \rho\, \tilde{\delta}^{i}_{0} \end{Bmatrix}$$

and

$$\mathbf{F}(\bar{x}_m) = \left[\frac{\partial f^{\mu}}{\partial x_m{}^{\nu}}(\bar{x}_m) \right] \equiv \begin{bmatrix} \dfrac{\partial g^{ui}}{\partial \lambda_m{}^{\beta}}(\bar{x}_m) & \dfrac{\partial g^{ui}}{\partial \phi^{\alpha}_{m}{}^{\beta}}(\bar{x}_m) \\ \\ \dfrac{\partial h^{i}}{\partial \lambda_m{}^{\beta}}(\bar{x}_m) & \dfrac{\partial h^{i}}{\partial \phi^{\alpha}_{m}{}^{\beta}}(\bar{x}_m) \end{bmatrix}$$

$$= \begin{bmatrix} -\tilde{M}^{u\ i}_{v\ \beta k} \bar{\phi}^{v}_{m}{}^{k} & \tilde{K}^{u\ i}_{\alpha\ \beta} - \tilde{M}^{u\ i}_{\alpha\ \beta k} \bar{\lambda}_{m}{}^{k} \\ \\ 0 & 2\, \delta_{\alpha w} \tilde{M}^{w\ i}_{v\ \beta k} \bar{\phi}^{v}_{m}{}^{k} \end{bmatrix},$$

whence $\alpha \in \{1, 2, \ldots, R\}$ and $\beta \in \{0, 1, \ldots, P\}$.

However, by using matrix notation, the components of $\mathbf{f}(\bar{x}_m)$ and $\mathbf{F}(\bar{x}_m)$ can be written more compactly as:

$$\mathbf{f}(\bar{x}_m) = \begin{Bmatrix} \tilde{\mathbf{K}}^{i}_{\ j} \bar{\boldsymbol{\phi}}_{m}{}^{j} - \bar{\lambda}_{m}{}^{k} \tilde{\mathbf{M}}^{i}_{\ jk} \bar{\boldsymbol{\phi}}_{m}{}^{j} \\ \\ (\bar{\boldsymbol{\phi}}_{m}{}^{j})^{T} \tilde{\mathbf{M}}^{i}_{\ jk} \bar{\boldsymbol{\phi}}_{m}{}^{k} - \rho\, \tilde{\delta}^{i}_{0} \end{Bmatrix} \quad \text{and} \quad \mathbf{F}(\bar{x}_m) = \begin{bmatrix} -\tilde{\mathbf{M}}^{i}_{\ jk} \bar{\boldsymbol{\phi}}_{m}{}^{k} & \tilde{\mathbf{K}}^{i}_{\ j} - \bar{\lambda}_{m}{}^{k} \tilde{\mathbf{M}}^{i}_{\ jk} \\ \\ & 2\, (\bar{\boldsymbol{\phi}}_{m}{}^{k})^{T} \tilde{\mathbf{M}}^{i}_{\ jk} \end{bmatrix},$$

where $\tilde{\mathbf{M}}^{i}_{\ jk} = \left[\tilde{M}^{u\ i}_{v\ jk} \right], \tilde{\mathbf{K}}^{i}_{\ j} = \left[\tilde{K}^{u\ i}_{v\ j} \right] \in \mathrm{L}(\mathbb{R}^{R}, \mathbb{R}^{R})$, and $\bar{\boldsymbol{\phi}}_{m}{}^{j} = \left\{ \bar{\phi}^{u}_{m}{}^{j} \right\} \in \mathbb{R}^{R}$. From a computational standpoint these two matrix representations of $\mathbf{f}$ and $\mathbf{F}$ are very useful, since they allow vectorization of the operations in a numerical computing environment such as BLAS (Basic Linear Algebra Subprograms) [97], and thus, improve the performance of the code considerably.

Thus, after solving for $\hat{\mathbf{x}}_m$ using (5.10) yields the next trial solution for (5.7):

$$\hat{\mathbf{x}}_m = \bar{\mathbf{x}}_m - \mathbf{F}^{-1}(\bar{x}_m)\, \mathbf{f}(\bar{x}_m), \tag{5.11}$$

[2]The formal derivation for the components of $\mathbf{F}$ can be found in Appendix 5.A.

where $\mathbf{F}^{-1}(\bar{x}_m)$ represents the matrix inverse of $\mathbf{F}(\bar{x}_m)$, and $\bar{\mathbf{x}}_m$ (or $\bar{x}_m$) denotes the initial trial solution.

5.4. Modal decomposition and spatial discretization of structural system

In structural dynamics, the displacement of a system can be written as:

$$\mathbf{u} = \boldsymbol{\phi}[\,\cdot\,, \mathbf{q}] \qquad \Leftrightarrow \qquad u^v = \phi^v{}_m \, q^m, \tag{5.12}$$

where $\boldsymbol{\phi} = \phi^v{}_m \, \mathbf{e}_v \otimes \mathbf{e}^m : \mathbb{R}^{R'} \times \mathbb{R}^R \to \mathfrak{Z}$ is the modal spatial tensor field of the system on Ξ, and $\mathbf{q} = q^m \, \mathbf{e}_m$ is a displacement vector field on $\mathfrak{T} \times \Xi$. In these expressions, $\phi^v{}_m : \Xi \to \mathbb{R}$ is a random function in $\mathfrak{Z}$ given by $\phi^v{}_m = \phi^v{}_m(\xi)$ (as in Section 5.3.2), $q^m : \mathfrak{T} \times \Xi \to \mathbb{R}$ is a random-temporal function in $\mathcal{U}$ given by $q^m = q^m(t, \xi)$, and $v, m \in \{1, 2, \ldots, R\}$.

Now, we define $\mathcal{S} = \mathbb{R}^R$ and let $\mathcal{S}^{[S]}$ be a subspace of $\mathcal{S}$ such that $\mathcal{S}^{[S]} = \text{span}\{\mathbf{e}_v\}_{v=1}^S$ and $S < R$. Then, a spatial discretization for the structural system in hand can be carried out by considering the first S eigenvectors of the system, so as to produce an approximate representation of $\mathbf{u}$ in $\mathcal{S}^{[S]}$ as follows:

$$\mathbf{u}(t, \xi) \approx \mathbf{u}^{[S]}(t, \xi) = \sum_{m=1}^{S} \phi^v{}_m(\xi) \, q^m(t, \xi) \, \mathbf{e}_v. \tag{5.13}$$

This approximation is performed here provided that the eigenvectors are ordered such that the corresponding eigenvalues satisfy: $\lambda_1 < \lambda_2 < \cdots < \lambda_R$.

To simplify notation, expansion (5.13) is written hereafter as

$$\mathbf{u}(t, \xi) = \phi^v{}_m(\xi) \, q^m(t, \xi) \, \mathbf{e}_v \qquad \Leftrightarrow \qquad u^v(t, \xi) = \phi^v{}_m(\xi) \, q^m(t, \xi), \tag{5.14}$$

where a summation sign is implied over the repeated index $m \in \{1, 2, \ldots, S\}$ unless indicated otherwise, and $v \in \{1, 2, \ldots, R\}$. We note that the superscript $^{[S]}$ was dropped in (5.14) to avoid notational complexity.

As it will become apparent shortly, system (5.1) is best described using tensor notation, and so we write from now on the system's governing differential equation (and its initial conditions) in the following way:

$$M^u{}_v\,\ddot{u}^v + C^u{}_v\,\dot{u}^v + K^u{}_v\,u^v = p^u \qquad \text{on } \mathfrak{T}\times\Xi \tag{5.15a}$$

$$\left\{u^u(0,\cdot\,) = \boldsymbol{u}^u,\, \dot{u}^u(0,\cdot\,) = \boldsymbol{v}^u\right\} \qquad \text{on } \{0\}\times\Xi, \tag{5.15b}$$

where $M^u{}_v, C^u{}_v, K^u{}_v, \boldsymbol{u}^u, \boldsymbol{v}^u : \Xi \to \mathbb{R}$ are random functions in $\mathfrak{Z}$, $p^u : \mathfrak{T}\times\Xi \to \mathbb{R}$ is a random-temporal function in $\mathscr{V}$, $u^v : \mathfrak{T}\times\Xi \to \mathbb{R}$ is a random-temporal function in $\mathscr{U}$, and $u, v \in \{1, 2, \ldots, R\}$. Furthermore, since $\dot{u}^v := \partial_t u^v$ and $\ddot{u}^v := \partial_t^2 u^v$, then $\dot{u}^v \in \mathscr{T}^1 \otimes \mathfrak{Z}$ and $\ddot{u}^v \in \mathscr{T}^0 \otimes \mathfrak{Z} \equiv \mathscr{V}$.

Therefore, substituting (5.14) into (5.15) gives

$$M^u{}_v\,\phi^v{}_m\,\ddot{q}^m + C^u{}_v\,\phi^v{}_m\,\dot{q}^m + K^u{}_v\,\phi^v{}_m\,q^m = p^u \qquad \text{on } \mathfrak{T}\times\Xi \tag{5.16a}$$

$$\left\{\phi^u{}_m\,q^m(0,\cdot\,) = \boldsymbol{u}^u,\, \phi^u{}_m\,\dot{q}^m(0,\cdot\,) = \boldsymbol{v}^u\right\} \qquad \text{on } \{0\}\times\Xi. \tag{5.16b}$$

To project (5.16) onto $\mathcal{S}^{[S]}$, we apply the spatial tensor field:

$$\boldsymbol{\phi}^T = (\phi^T)^n{}_u\,\mathbf{e}_n \otimes \mathbf{e}^u : \mathbb{R}^{S'} \times \mathbb{R}^R \to \mathfrak{Z}$$

on both sides of each equation to get:

$$\phi_u{}^n M^u{}_v\,\phi^v{}_m\,\ddot{q}^m + \phi_u{}^n C^u{}_v\,\phi^v{}_m\,\dot{q}^m + \phi_u{}^n K^u{}_v\,\phi^v{}_m\,q^m = \phi_u{}^n p^u \qquad \text{on } \mathfrak{T}\times\Xi \tag{5.17a}$$

$$\left\{\phi_u{}^n\phi^u{}_m\,q^m(0,\cdot\,) = \phi_u{}^n\,\boldsymbol{u}^u,\, \phi_u{}^n\phi^u{}_m\,\dot{q}^m(0,\cdot\,) = \phi_u{}^n\,\boldsymbol{v}^u\right\} \qquad \text{on } \{0\}\times\Xi, \tag{5.17b}$$

where $n \in \{1, 2, \ldots, S\}$, $(\phi^T)^n{}_u := \phi_u{}^n = \delta_{uv}\delta^{mn}\phi^v{}_m$, and δ_{uv} and δ^{mn} are two Kronecker deltas representing the components of the metric tensor endowed on $\mathbb{R}^R$ and $\mathbb{R}^{S'}$, respectively.

5.5. Solution based on the spectral approach

5.5.1. Discretization of random function space

To simplify the presentation, in this section we consider a p-discretization of $\mathfrak{Z}$. Let $\mathfrak{Z}^{[P]} = \operatorname{span}\{\Psi_j\}_{j=0}^P$ be a finite subspace of $\mathfrak{Z}$ with $P+1 \in \mathbb{N}_1$ denoting the dimensionality

of the subspace. Then, if we let $q^{m[P]}(t,\cdot\,)$ be an element of $\mathfrak{Z}^{[P]}$, it is apparent that the following approximation can be made, provided that $\{\Psi_j\}_{j=0}^{\infty}$ is well graded:

$$q^m(t,\xi) \approx q^{m[P]}(t,\xi) = \sum_{j=0}^{P} q^{mj}(t)\,\Psi_j(\xi) \equiv q^{mj}(t)\,\Psi_j(\xi). \tag{5.18}$$

For the last equality, we need to keep in mind that $j \in \{0,1,\ldots,P\}$.

Substituting (5.18) and (5.4b) into (5.17) gives

$$\delta_{uw}\delta^{xn}\phi^{w}{}_{x}{}^{k}\tilde{\Psi}_k M^{u}{}_{v}\,\phi^{v}{}_{m}{}^{l}\tilde{\Psi}_l\ddot{q}^{mj}\Psi_j + \cdots = \delta_{uw}\delta^{xn}\phi^{w}{}_{x}{}^{k}\tilde{\Psi}_k p^u \qquad \text{on } \mathfrak{T}\times\Xi \tag{5.19a}$$

$$\Big\{\delta_{uw}\delta^{xn}\phi^{w}{}_{x}{}^{k}\tilde{\Psi}_k\phi^{u}{}_{m}{}^{l}\tilde{\Psi}_l\,q^{mj}(0)\,\Psi_j = \delta_{uw}\delta^{xn}\phi^{w}{}_{x}{}^{k}\tilde{\Psi}_k\boldsymbol{\mathcal{u}}^u,\ \ldots\Big\} \qquad \text{on } \{0\}\times\Xi. \tag{5.19b}$$

Projecting (5.19) onto $\mathfrak{Z}^{[P]}$ yields a system of $(P+1)S$ ordinary differential equations of second order in the variable t, where the unknowns are the random modes $q^{mj} = q^{mj}(t)$ and $\dot{q}^{mj} = \dot{q}^{mj}(t)$:

$$M^{n}{}_{m}{}^{i}{}_{j}\,\ddot{q}^{mj} + C^{n}{}_{m}{}^{i}{}_{j}\,\dot{q}^{mj} + K^{n}{}_{m}{}^{i}{}_{j}\,q^{mj} = p^{ni} \qquad \text{on } \mathfrak{T} \tag{5.20a}$$

$$\Big\{q^{mj}(0) = \boldsymbol{\mathcal{q}}^{mj}, \dot{q}^{mj}(0) = \boldsymbol{\mathcal{r}}^{mj}\Big\} \qquad \text{on } \{0\}, \tag{5.20b}$$

where $M^{n}{}_{m}{}^{i}{}_{j}, C^{n}{}_{m}{}^{i}{}_{j}, K^{n}{}_{m}{}^{i}{}_{j}, \boldsymbol{\mathcal{q}}^{mj}, \boldsymbol{\mathcal{r}}^{mj} \in \mathbb{R}$ and $p^{ni} : \mathfrak{T} \to \mathbb{R}$ are given by

$$\begin{aligned}
M^{n}{}_{m}{}^{i}{}_{j} &= \delta_{uw}\delta^{xn}\,\phi^{w}{}_{x}{}^{k}\phi^{v}{}_{m}{}^{l}\,\langle\Psi_i, M^{u}{}_{v}\,\Psi_j\tilde{\Psi}_k\tilde{\Psi}_l\rangle/\langle\Psi_i,\Psi_i\rangle\\
C^{n}{}_{m}{}^{i}{}_{j} &= \delta_{uw}\delta^{xn}\,\phi^{w}{}_{x}{}^{k}\phi^{v}{}_{m}{}^{l}\,\langle\Psi_i, C^{u}{}_{v}\,\Psi_j\tilde{\Psi}_k\tilde{\Psi}_l\rangle/\langle\Psi_i,\Psi_i\rangle\\
K^{n}{}_{m}{}^{i}{}_{j} &= \delta_{uw}\delta^{xn}\,\phi^{w}{}_{x}{}^{k}\phi^{v}{}_{m}{}^{l}\,\langle\Psi_i, K^{u}{}_{v}\,\Psi_j\tilde{\Psi}_k\tilde{\Psi}_l\rangle/\langle\Psi_i,\Psi_i\rangle\\
\boldsymbol{\mathcal{q}}^{mj} &= (A^{-1})_{n}{}^{m}{}_{i}{}^{j}\delta_{uw}\delta^{xn}\phi^{w}{}_{x}{}^{k}\,\langle\Psi_i, \boldsymbol{\mathcal{u}}^{u}\tilde{\Psi}_k\rangle/\langle\Psi_i,\Psi_i\rangle\\
\boldsymbol{\mathcal{r}}^{mj} &= (A^{-1})_{n}{}^{m}{}_{i}{}^{j}\delta_{uw}\delta^{xn}\phi^{w}{}_{x}{}^{k}\,\langle\Psi_i, \boldsymbol{\mathcal{v}}^{u}\tilde{\Psi}_k\rangle/\langle\Psi_i,\Psi_i\rangle\\
p^{ni}(t) &= \delta_{uw}\delta^{xn}\phi^{w}{}_{x}{}^{k}\,\langle\Psi_i, p^u(t,\cdot\,)\,\tilde{\Psi}_k\rangle/\langle\Psi_i,\Psi_i\rangle,
\end{aligned}$$

where $(A^{-1})_{n}{}^{m}{}_{i}{}^{j}$ represents the entries of the matrix inverse of

$$\mathbf{A} = \Big[A^{\varrho(n,i)}{}_{\varrho(m,j)}\Big] \in \mathrm{L}(\mathbb{R}^{S(P+1)}, \mathbb{R}^{S(P+1)}),$$

$\varrho : \{1,2,\ldots,S\}\times\{0,1,\ldots,P\} \to \mathbb{N}_1$ is given by $\varrho(\alpha,\beta) = S\beta + \alpha$, and

$$A^{\varrho(n,i)}{}_{\varrho(m,j)} := A^{n}{}_{m}{}^{i}{}_{j} = \delta_{uw}\delta^{xn}\phi^{w}{}_{x}{}^{k}\phi^{u}{}_{m}{}^{l}\,\langle\Psi_i, \Psi_j\tilde{\Psi}_k\tilde{\Psi}_l\rangle/\langle\Psi_i,\Psi_i\rangle.$$

Using matrix notation, system (5.20) can be rewritten more compactly as

$$\mathbf{M}^i{}_j \ddot{\mathbf{q}}^j + \mathbf{C}^i{}_j \dot{\mathbf{q}}^j + \mathbf{K}^i{}_j \mathbf{q}^j = \mathbf{p}^i \qquad \text{on } \mathfrak{T} \tag{5.21a}$$

$$\left\{ \mathbf{q}^j(0) = \boldsymbol{q}^j, \dot{\mathbf{q}}^j(0) = \boldsymbol{r}^j \right\} \qquad \text{on } \{0\}, \tag{5.21b}$$

where

$$\mathbf{M}^i{}_j = \left[M^n{}_m{}^i{}_j \right], \ \mathbf{C}^i{}_j = \left[C^n{}_m{}^i{}_j \right], \ \mathbf{K}^i{}_j = \left[K^n{}_m{}^i{}_j \right] \ \in \mathrm{L}(\mathbb{R}^S, \mathbb{R}^S)$$
$$\mathbf{p}^i = \left\{ p^{ni} \right\}, \ \boldsymbol{q}^j = \left\{ q^{mj} \right\}, \ \boldsymbol{r}^j = \left\{ r^{mj} \right\} \ \in \mathbb{R}^S.$$

(Recall that $i, j \in \{0, 1, \ldots, P\}$ and $m, n \in \{1, 2, \ldots, S\}$.)

Remark 5.3. Note that in order to compute $\mathbf{u}$, we can always go back to (5.13) and (5.18) to obtain:

$$\begin{aligned} \mathbf{u}(t, \xi) &= \phi^v{}_m(\xi)\, q^m(t, \xi)\, \mathbf{e}_v \\ &= \phi^v{}_m(\xi)\, q^{mj}(t)\, \Psi_j(\xi)\, \mathbf{e}_v. \end{aligned}$$

5.5.2. Discretization of temporal function space

As usual, the temporal function space $\mathcal{T}$ can be discretized using any suitable time integration method, such as the Newmark-β method [30, 31] or the Runge-Kutta method [29] of fourth order (aka RK4 method). In this work we use the latter to push the state of the system forward in time with high accuracy.

5.5.3. Computation of probability moments

In order to demonstrate how the probability moments can be computed using the spectral approach, suppose that $z(t, \cdot) \in \mathfrak{Z}$ is a square-integrable random variable materialized at some time $t \in \mathfrak{T}$. This random variable may be interpreted as one of the infinitely many

responses that one can obtain from solving (5.1) at a given time. Therefore, an approximate representation of $z(t,\cdot)$ in $\mathfrak{Z}^{[P]} = \text{span}\{\Psi_j\}_{j=0}^{P}$ can be taken as:

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

where the j-th random mode of z is given by

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

Consequently, if we were interested in computing the expectation of z, $\mathbf{E}[z] : \mathfrak{T} \to \mathbb{R}$, or the variance of z, $\text{Var}[z] : \mathfrak{T} \to \mathbb{R}_0^+$, as a function of time, the corresponding formulas for these moments would be given by:

$$\mathbf{E}[z](t) := \int z(t,\cdot)\,\mathrm{d}\mu = z^0(t) \tag{5.22a}$$

$$\text{Var}[z](t) := \int (z(t,\cdot) - \mathbf{E}[z](t))^2\,\mathrm{d}\mu = \sum_{j=1}^{P} \langle \Psi_j, \Psi_j\rangle\, z^j(t)\, z^j(t). \tag{5.22b}$$

The derivation of these two formulas can be found, for instance, in [92].

5.6. Numerical results

The structure considered in this work is a hypothetical, high-rise residential building located in northwestern Colombia. The 25-story building is 77-m tall and has a typical inter-story height of 3 m. Typical plan dimensions of a floor in the building are 25 m by 30 m. Typical spans range from 5 to 7 m. The lateral load resistance of the building is chiefly provided by a reinforced-concrete core wall which is connected to perimeter reinforced-concrete moment frames using post-tensioned concrete flat slabs. Table 5.1 provides further information about the characteristics of this building. The building was designed using performance-based design criteria [40, 41] in order to make the structural design code-compliant. The building's fundamental period of vibration is 1.78 s.

Once the mathematical model of the structural system has been abstracted and constructed, the model is further simplified to get only one degree of freedom per story as shown in Fig. 5.1. This is possible to do since in a one-direction, lateral-load analysis (like the one we are

Table 5.1. Additional characteristics of selected 25-story building

	Slabs (25)	Columns (20)	Core wall (1)
Compressive strength of concrete	$f'_c = 35$ MPa	$f'_c = 45$ MPa	$f'_c = 45$ MPa
1st-7th story	$t = 20$ cm	$s = 55 \times 55$ cm	$t = 50$ cm
8th-18th story	$t = 20$ cm	$s = 50 \times 50$ cm	$t = 45$ cm
19th-25th story	$t = 20$ cm	$s = 45 \times 45$ cm	$t = 40$ cm

about to undertake) the contribution of the rotational mass to the total response is negligible compared to the translational mass.

Assuming free vibration and no damping forces acting in the system, the governing differential equation of motion for the 25-story-building model becomes

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{0}.$$

Although this is an unrealistic governing equation for a building, we use it herein to investigate the performance of the MFSC method more thoroughly. This is because the addition of damping forces in the governing equation contributes at making the system memoryless over time, which in turn weakens any nonlinear dependency that may exist over the probability space as the simulation proceeds. Thus, to make the problem more difficult to solve, the damping forces are removed from the standard governing equation of the building.

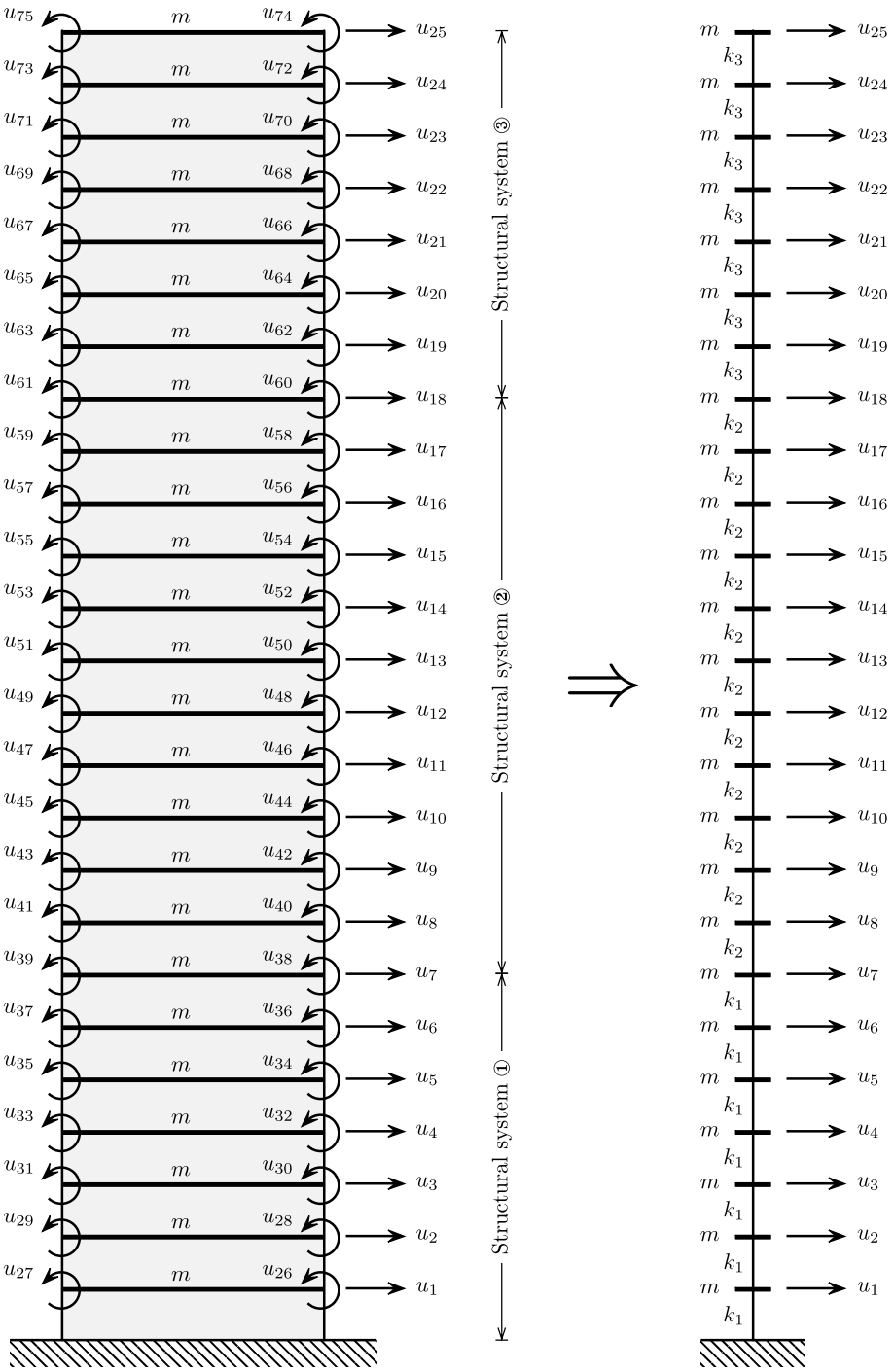
The mass matrix, $\mathbf{M} \in \mathrm{L}(\mathbb{R}^{25}, \mathbb{R}^{25})$, is deterministic and given by:

$$\mathbf{M} = \begin{bmatrix} m & & & & & \\ & m & & & & \\ & & \ddots & & & \\ & & & m & & \\ & & & & m & \end{bmatrix}_{25\times 25},$$

whereas the stiffness matrix, $\mathbf{K} : \Xi \to \mathrm{L}(\mathbb{R}^{25}, \mathbb{R}^{25})$, is stochastic and given by:

$$\mathbf{K}(\xi) = \begin{bmatrix} 2\,k_1(\xi) & -k_1(\xi) & & & & \\ -k_1(\xi) & 2\,k_1(\xi) & & & & \\ & & \ddots & & & \\ & & & 2\,k_3(\xi) & -k_3(\xi) & \\ & & & -k_3(\xi) & k_3(\xi) & \end{bmatrix}_{25\times 25}.$$

As one would expect, $\mathbf{u} : \mathfrak{T} \times \Xi \to \mathbb{R}^{25}$ represents the displacement of the system, and $\dot{\mathbf{u}} := \partial_t \mathbf{u}$ and $\ddot{\mathbf{u}} := \partial_t^2 \mathbf{u}$ are the velocity and acceleration of the system, respectively. Observe that the problem we are trying to solve is that given by (5.1), from where we have taken: $\mathbf{C} \equiv \mathbf{0}$ and



Important note:
This is a schematic representation of the degrees of freedom found in a typical 25-story-building model, and the distribution of the degrees of freedom is not meant to be the actual distribution employed in our 25-story-building model.

Important note:
This is the final distribution of the degrees of freedom employed in our simplified version of the 25-story-building model.

Figure 5.1. Surrogate model of a 25-story building for lateral-load analysis in one direction

$\mathbf{p} \equiv \mathbf{0}$. The initial conditions of the system are $\mathbf{u}(0,\cdot) = \boldsymbol{u}$ and $\dot{\mathbf{u}}(0,\cdot) = \boldsymbol{v} \equiv \mathbf{0}$, where $\boldsymbol{u} \equiv \left[u_1, u_2, \ldots, u_j, \ldots, u_{25}\right]^T$ and $u_j = 0.03j$.

In the above expressions, $\xi = (\xi^1, \xi^2)$, $m = 750$ Mg represents the amount of mass per story, $k_1(\xi) = \xi^1$ is the effective lateral stiffness of the 1st-7th story, $k_2(\xi) = \xi^2$ is the effective lateral stiffness of the 8th-18th story, and $k_3(\xi) = \xi^2$ is the effective lateral stiffness of the 19th-25th story (as depicted in Fig. 5.1).

Two different probability distributions are considered for ξ. The first distribution is a *uniform distribution*, Uniform $\otimes$ Uniform $\sim \xi \in \Xi$, and the second distribution is a *beta distribution*, $\text{Beta}(\alpha,\beta) \otimes \text{Beta}(\alpha,\beta) \sim \xi \in \Xi$. The parameters for both these distributions are taken as follows: $\Xi = [a_1, b_1] \times [a_2, b_2]$ and $(\alpha, \beta) = (2, 5)$, where $(a_1, b_1) = (2.21 \times 10^6, 4.16 \times 10^6)$ kN/m and $(a_2, b_2) = (1.99 \times 10^6, 3.74 \times 10^6)$ kN/m.

To reduce the computational burden of the simulation, we run the simulations with up to $S = 3$ modes of vibration. Using the expected values for k_1, k_2 and k_3, the combined modal mass participations turn out to be about: 82% if $S = 1$, 91% if $S = 2$ and 95% if $S = 3$. The resulting system of equations, as displayed in (5.21), is integrated over time using the RK4 method. The time-step size used is $\Delta t = 0.005$ s. For illustrative purposes, the simulation is set to last $T = 50$ s, which means that the temporal domain of the system is $\mathfrak{T} = [0, 50]$ s. Further, because the initial conditions are deterministic, the gPC method (with $P = 8$) is implemented for the first 2 seconds of the simulation to make sure that the stochasticity of the system's state is well developed for the analysis with MFSC.

To minimize the error propagation over time, the probability information is transferred exactly from one random function space to another using the FSC-2 approach presented in [92]. However, in order to implement the FSC-2 approach correctly, the following modifications are required to be made in the FSC scheme of [92]:

- The boundedness of P is this time given by:

$$(n+1)S \leq P \leq (n+M)S,$$

  which means that the dimensionality of the random function space, $\dim \mathfrak{L}^{[P]}$, is bounded from below by $(n+1)S+1$ and from above by $(n+M)S+1$. We recall that n denotes the order of the system's governing ODE with respect to time, and that M denotes

the order of the stochastic flow map we want to implement. Notice that if $S = 1$, the boundedness of P in [92] is fully recovered. However, because in this work $n = 2$ and $M = 1$ is always assumed for computational efficiency, the above expression reduces to $P = 3S$.

- The system's state involved in the calculations is no longer $(\mathbf{u}, \dot{\mathbf{u}})$ but $(\mathbf{q}, \dot{\mathbf{q}})$ as shown in (5.17). Therefore, the stochastic flow map, $\varphi(1) : \mathbb{R} \times \mathfrak{Z}^{2S} \to \mathfrak{Z}^{2S}$, takes the following form in this work:

$$\begin{aligned}\varphi(1)(h, \mathbf{s}(t_i, \cdot)) =: \mathbf{s}(t_i + h, \cdot) &= \Big(\mathbf{q}(t_i + h, \cdot), \dot{\mathbf{q}}(t_i + h, \cdot)\Big) - O(h^2) \\ &= \Big(\mathbf{q}(t_i, \cdot) + h\, \dot{\mathbf{q}}(t_i, \cdot),\, \dot{\mathbf{q}}(t_i, \cdot) + h\, \ddot{\mathbf{q}}(t_i, \cdot)\Big),\end{aligned}$$

  where $\mathbf{q}(t, \xi) = q^m(t, \xi)\, \mathbf{e}_m$ with $m \in \{1, 2, \ldots, S\}$ (see (5.12)), and

$$\mathbf{s} = (\mathbf{q}, \dot{\mathbf{q}}) \equiv \begin{cases} (q^1, \dot{q}^1) & \text{for } S = 1 \\ (q^1, q^2, \dot{q}^1, \dot{q}^2) & \text{for } S = 2 \\ (q^1, q^2, q^3, \dot{q}^1, \dot{q}^2, \dot{q}^3) & \text{for } S = 3. \end{cases}$$

  Likewise, the enriched stochastic flow map, $\hat{\varphi}(1) : \mathbb{R} \times \mathfrak{Z}^{3S} \to \mathfrak{Z}^{3S}$, takes the form:

$$\hat{\varphi}(1)(h, \hat{\mathbf{s}}(t_i, \cdot)) =: \hat{\mathbf{s}}(t_i + h, \cdot) = (\mathbf{q}(t_i + h, \cdot), \dot{\mathbf{q}}(t_i + h, \cdot), \ddot{\mathbf{q}}(t_i + h, \cdot)),$$

  where

$$\hat{\mathbf{s}} = (\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \equiv \begin{cases} (q^1, \dot{q}^1, \ddot{q}^1) & \text{for } S = 1 \\ (q^1, q^2, \dot{q}^1, \dot{q}^2, \ddot{q}^1, \ddot{q}^2) & \text{for } S = 2 \\ (q^1, q^2, q^3, \dot{q}^1, \dot{q}^2, \dot{q}^3, \ddot{q}^1, \ddot{q}^2, \ddot{q}^3) & \text{for } S = 3. \end{cases}$$

  Note that the $O(h^2)$ term was dropped in the definition of $\hat{\varphi}(1)$ for notational convenience.

- The set of linearly independent functions is therefore taken in this work as: $\{\Phi_{j.i} := \hat{\varphi}^j(1)(0, \hat{\mathbf{s}}(t_i, \cdot))\}_{j=1}^{3S}$.

The inner products are evaluated numerically using the following quadrature rules on each random axis:

$$\text{Uniform} \sim \text{Gauss-Legendre(100 points)} \quad \text{and} \quad \text{Beta} \sim \text{Gauss-Jacobi(80 points)}.$$

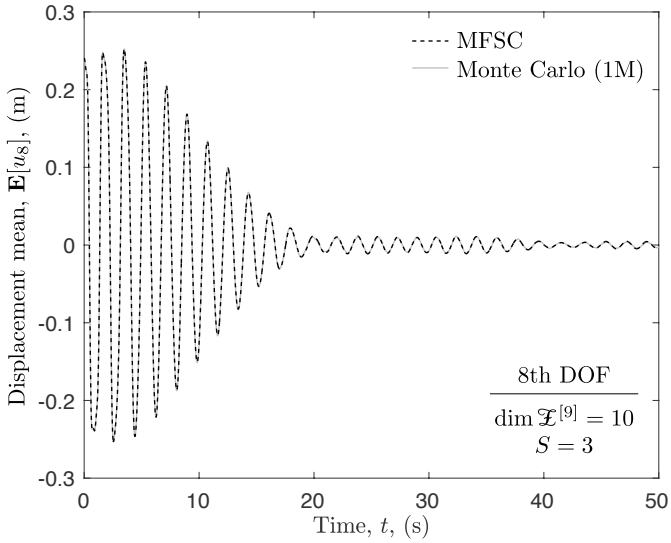
Moreover, as in [92], the local and global errors, $\epsilon : \mathcal{T} \to \mathcal{T}$ and $\epsilon_G : \mathcal{T} \to \mathbb{R}$, are defined with:

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

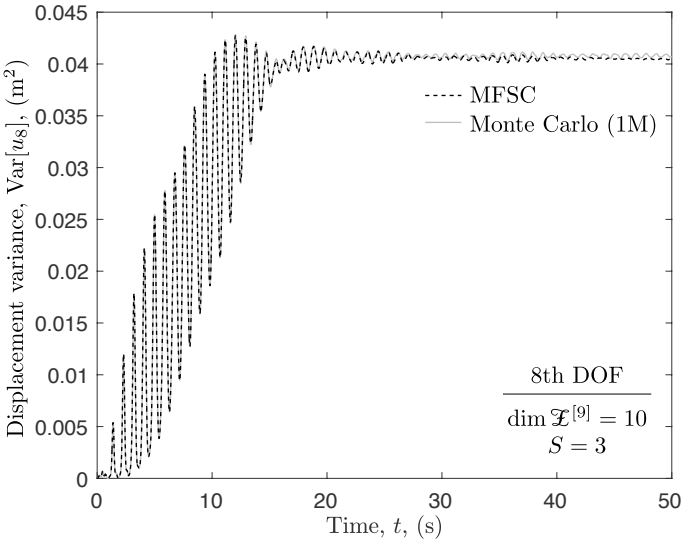
where Δt is the time-step size used for the simulation, $t_i \in \mathfrak{T}$ is the time instant of the simulation, and N is the number of time steps employed in the simulation (with $t_0 = 0$ and $t_N = N\,\Delta t = T$).

All problems are run in MATLAB R2020b [98] on a 2020 MacBook Air with Apple M1 chip (*8-Core* CPU at *3.20 GHz*, *8-Core* GPU, *16-Core* Neural Engine, and *16 GB* unified memory) and *1 TB Apple-Fabric SSD* storage (APFS-formatted), running macOS Big Sur (version 11.2).

In Figs. 5.2 to 5.4 we depict the evolution of the mean and variance for some of the story displacements using MFSC with $S = 3$ and a Monte Carlo simulation with one million realizations. The results depicted correspond to the case when the probability measure is assumed uniform. The dimensionality of the random function space, $\dim \mathfrak{X}^{[P]}$, is in this case 10 because $P = 3S$ in this work. The responses shown are those obtained for u_8, u_{17} and u_{25} (as displayed in Fig. 5.1). As observed, the MFSC solution is capable of reproducing the Monte Carlo solution reasonably well, for the results only happen to be slightly distinguishable from each other after the 30th second of the simulation. This difference in the results is attributable to the fact that higher modes can influence the total response by about 5% at early times of the simulation. (Recall that the combined modal mass participation for $S = 3$ is 95%.) Although this error has the potential to worsen the results significantly over time, here we see that this is not the case for the problem in hand, since the MFSC results are in good agreement with Monte Carlo for the entire simulation.

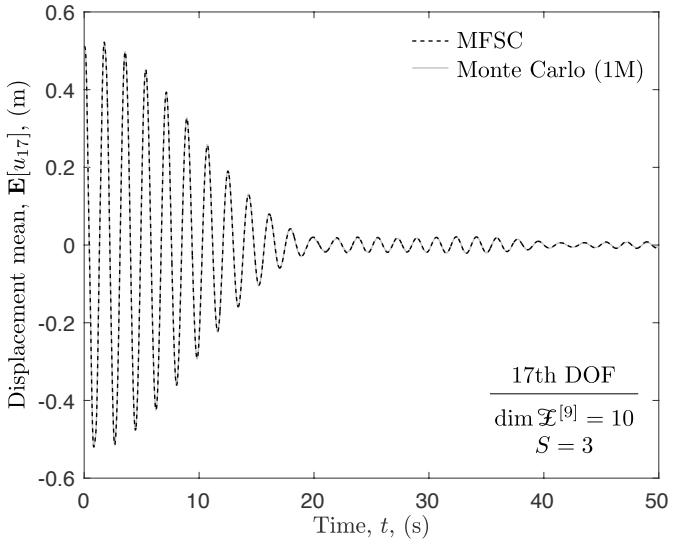


(a) Mean for 8th DOF

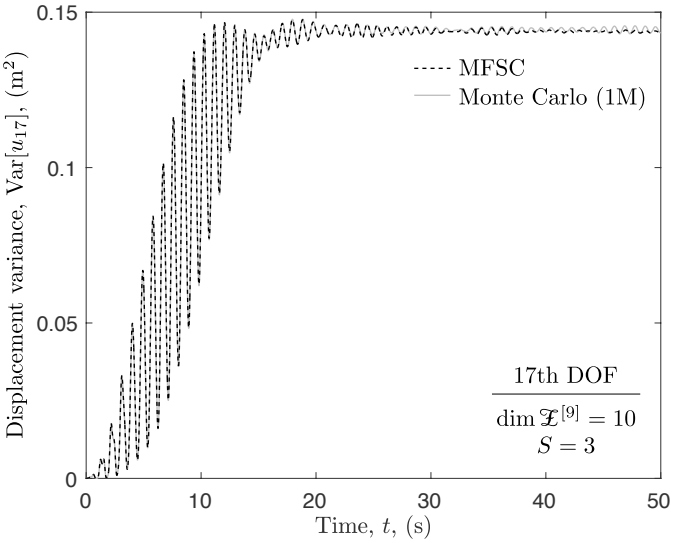


(b) Variance for 8th DOF

Figure 5.2. Evolution of $\mathbf{E}[u_8]$ and $\mathrm{Var}[u_8]$ for the case when the p-discretization level of RFS is $(P, S) = (3, 3)$ and $\mu \sim \text{Uniform} \otimes \text{Uniform}$

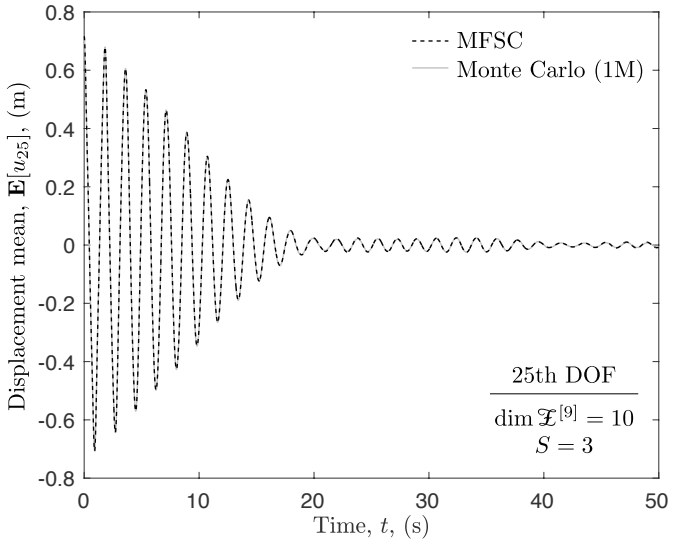


(a) Mean for 17th DOF

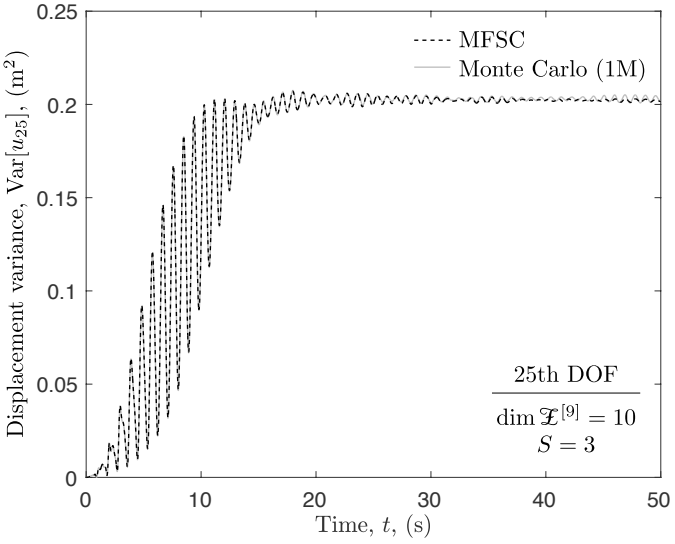


(b) Variance for 17th DOF

Figure 5.3. Evolution of $\mathbf{E}[u_{17}]$ and $\mathrm{Var}[u_{17}]$ for the case when the p-discretization level of RFS is $(P, S) = (3, 3)$ and $\mu \sim \text{Uniform} \otimes \text{Uniform}$



(a) Mean for 25th DOF (roof)



(b) Variance for 25th DOF (roof)

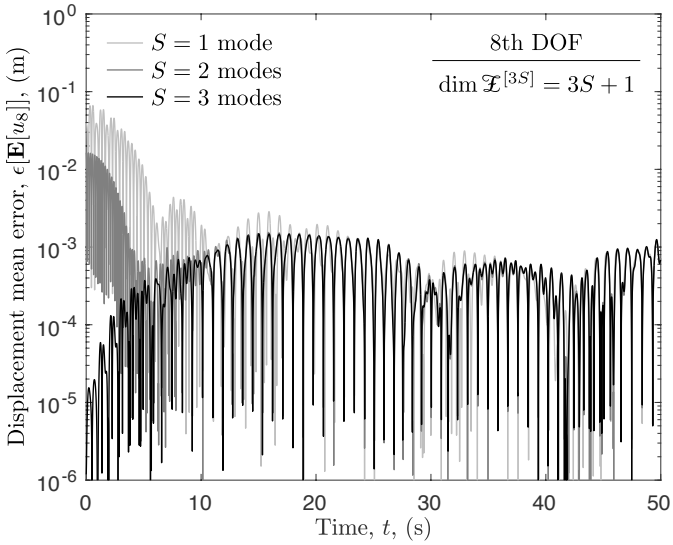
Figure 5.4. Evolution of $\mathbf{E}[u_{25}]$ and $\mathrm{Var}[u_{25}]$ for the case when the p-discretization level of RFS is $(P, S) = (3, 3)$ and $\mu \sim \text{Uniform} \otimes \text{Uniform}$

Figs. 5.5 to 5.7 present the local errors in mean and variance of u_8, u_{17} and u_{25} as a function of the number of vibration modes employed in the simulations with MFSC and for the case when the probability measure is uniform. This figure shows that for the first 5 seconds of the simulation, the total response is largely commanded by the higher vibration modes of the building. This is the reason why choosing $S = 1$ or $S = 2$ in the simulations with MFSC does not produce accurate results at early times of the simulation. However, as time progresses, the total response becomes more and more commanded by the first two vibration modes of the building. This indicates that 2 vibration modes are sufficient to capture the overall response of the building with MFSC, but 3 vibrations modes are still required to capture local phenomena such as those identified in the first 5 seconds of the simulation.

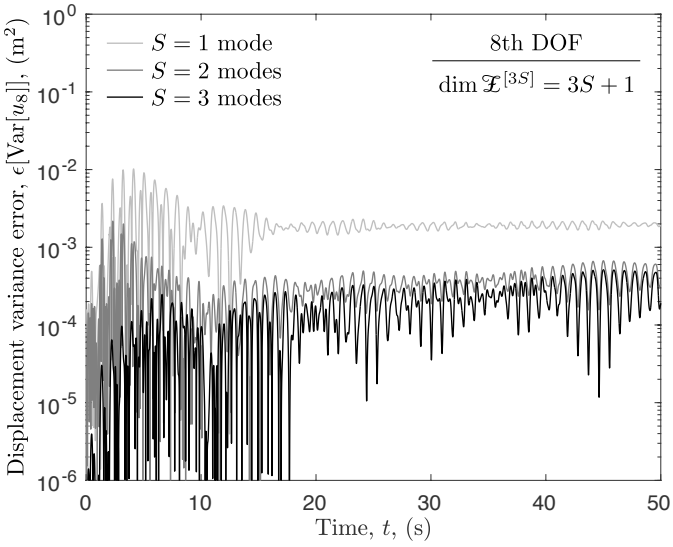
Finally, Figs. 5.8 to 5.10 depict the global errors in mean and variance for u_8, u_{17} and u_{25} as a function of the number of vibration modes and the two distributions chosen for ξ. As shown, the global errors can drop up to 10^{-3} if 2 vibration modes are used in the simulations with MFSC. However, due to the inexactness of the Monte Carlo solution, increasing the number of vibration modes does not have a direct impact at reducing the global errors beyond 10^{-4}. Therefore, comparable results to Monte Carlo are achievable for this problem if only two or three vibration modes are included in the simulations with MFSC.

5.7. Conclusion

This chapter has presented a new technique called the *modal flow-driven spectral chaos* (MFSC) method to deal with large-scale structural dynamical systems subjected to uncertainties more effectively using the spectral approach. Because the aim of this work was to reduce considerably the dimensionality of the random function space, it was necessary to find first the first few vibration modes of the system. To do so, an iterative process based on the Newton-Raphson method was proposed to resolve a nonlinear system of eigen-equations over the spatial-random space (as in [96]). Then, the first few vibration modes of the system were utilized to project the system's governing differential equation of motion, and this way reduce substantially the number of degrees of freedom involved in the initial mathematical model

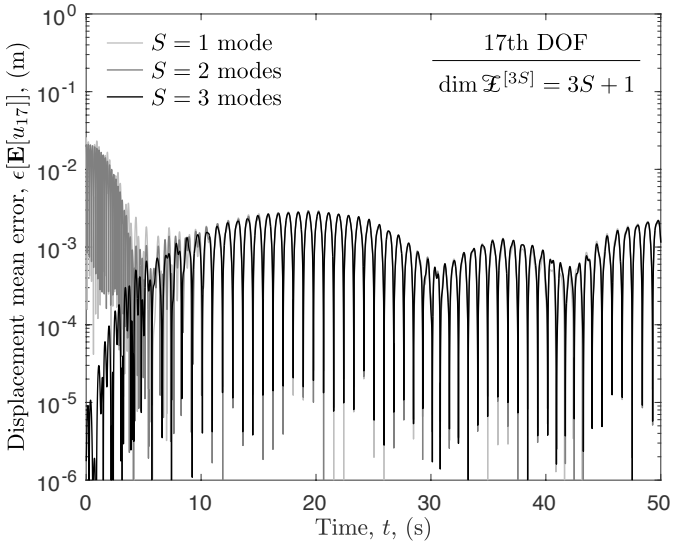


(a) Mean error for 8th DOF

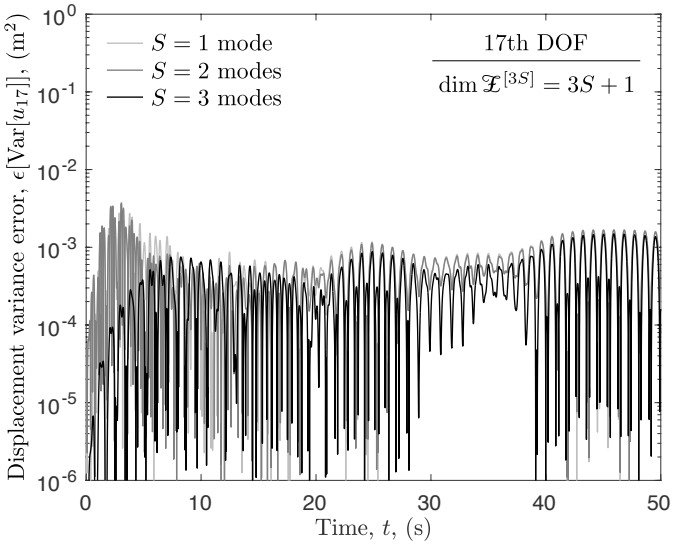


(b) Variance error for 8th DOF

Figure 5.5. Local error evolution of $\mathbf{E}[u_8]$ and $\mathrm{Var}[u_8]$ for different p-discretization levels of RFS and for $\mu \sim \text{Uniform} \otimes \text{Uniform}$

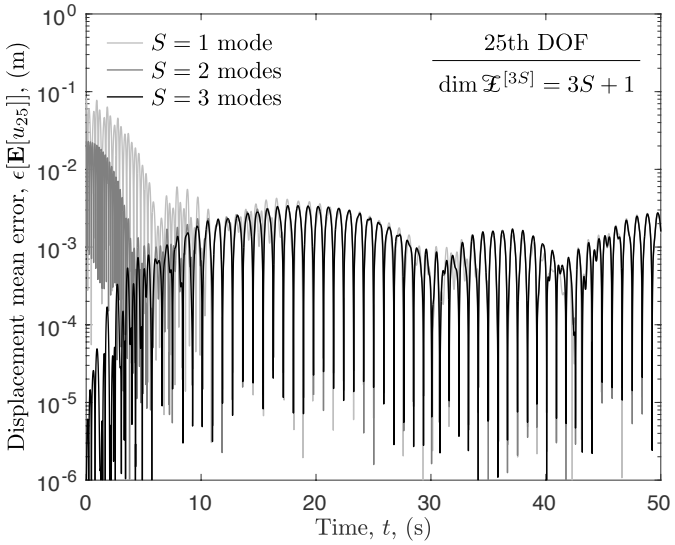


(a) Mean error for 17th DOF

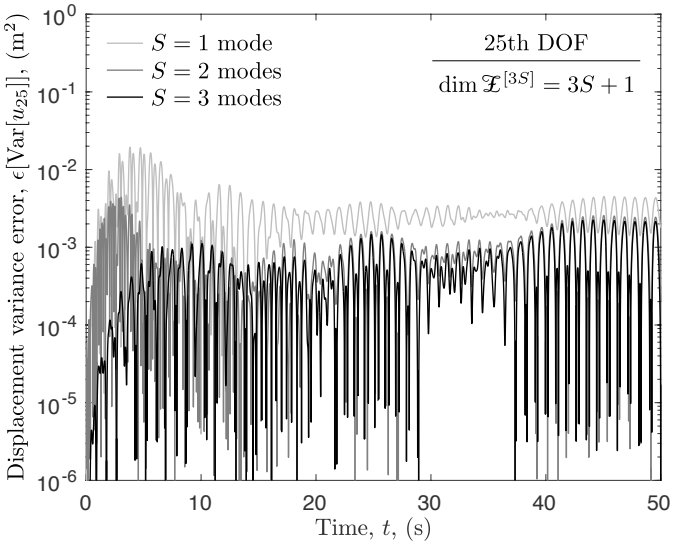


(b) Variance error for 17th DOF

Figure 5.6. Local error evolution of $\mathbf{E}[u_{17}]$ and $\mathrm{Var}[u_{17}]$ for different p-discretization levels of RFS and for $\mu \sim \text{Uniform} \otimes \text{Uniform}$

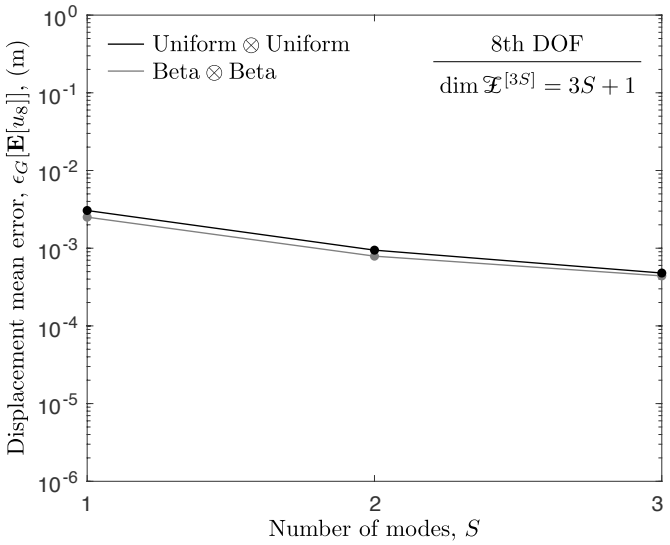


(a) Mean error for 25th DOF (roof)

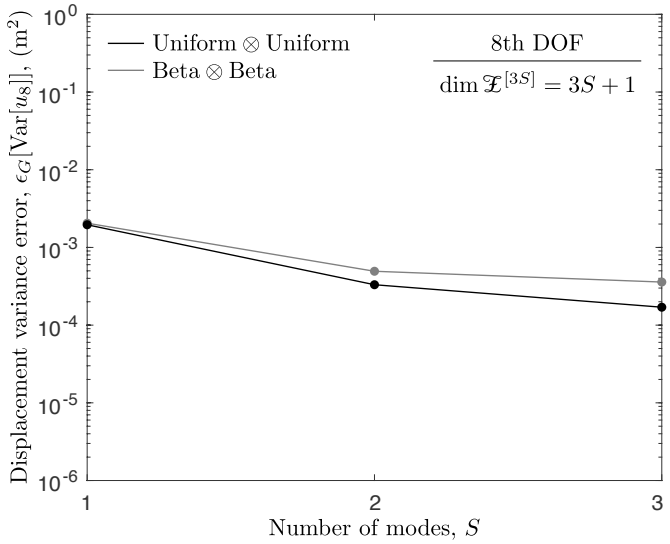


(b) Variance error for 25th DOF (roof)

Figure 5.7. Local error evolution of $\mathbf{E}[u_{25}]$ and $\mathrm{Var}[u_{25}]$ for different p-discretization levels of RFS and for $\mu \sim \text{Uniform} \otimes \text{Uniform}$

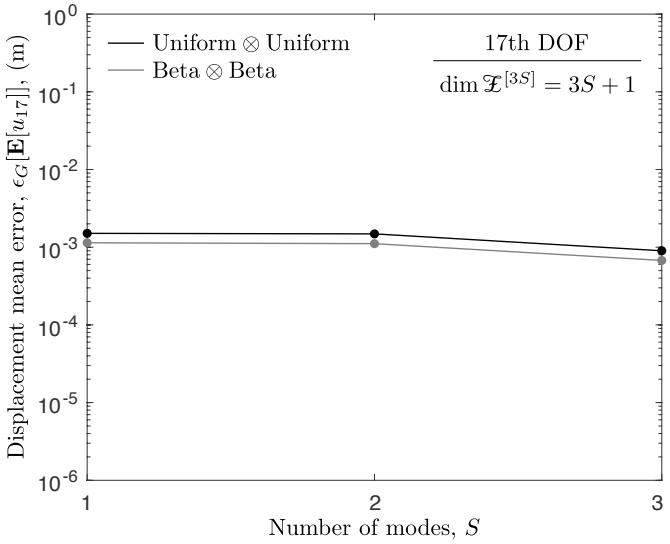


(a) Mean error for 8th DOF

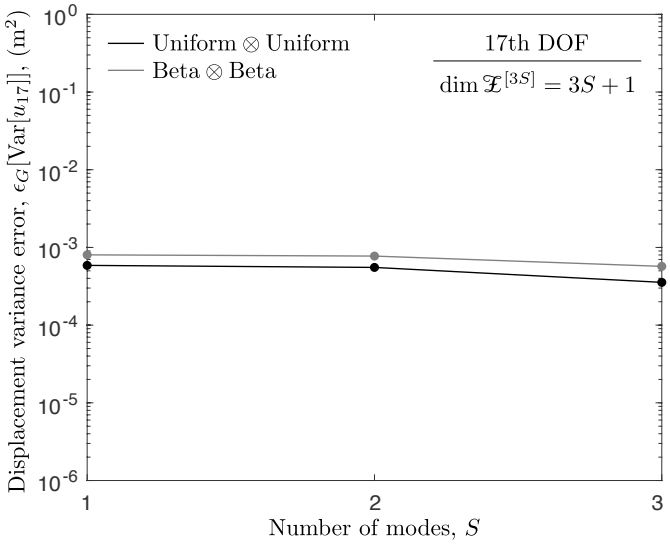


(b) Variance error for 8th DOF

Figure 5.8. Global error of $\mathbf{E}[u_8]$ and $\mathrm{Var}[u_8]$ for different p-discretization levels of RFS

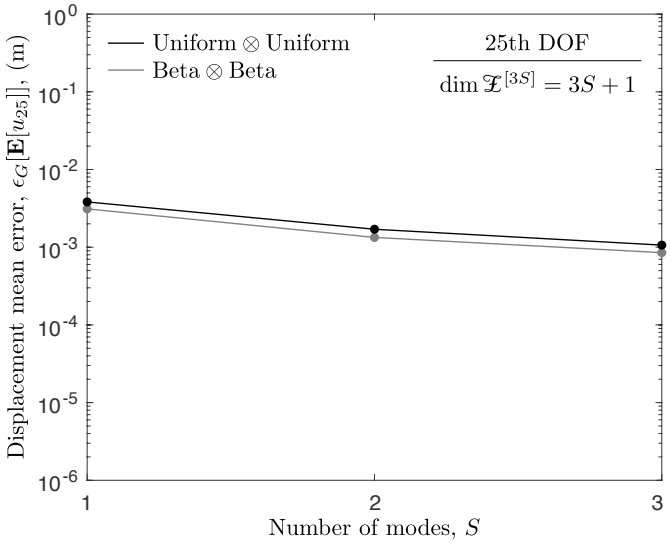


(a) Mean error for 17th DOF

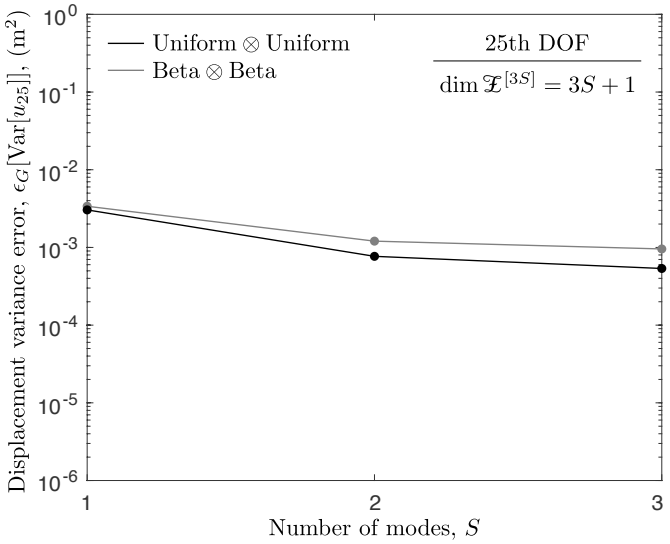


(b) Variance error for 17th DOF

Figure 5.9. Global error of $\mathbf{E}[u_{17}]$ and $\mathrm{Var}[u_{17}]$ for different p-discretization levels of RFS



(a) Mean error for 25th DOF (roof)



(b) Variance error for 25th DOF (roof)

Figure 5.10. Global error of $\mathbf{E}[u_{25}]$ and $\mathrm{Var}[u_{25}]$ for different p-discretization levels of RFS

of the system. The spectral approach was then employed to solve the new resulting set of stochastic equations as a means to find the sought solution afterwards.

To test the effectiveness of the MFSC method, a 25-story building was adapted to comprise at most one degree of freedom per story. Then, a modal analysis was carried out to obtain the first three vibration modes of the building. From this analysis it was learnt that the building had a combined modal mass participation of about 95% for the first three vibration modes, which helped bound the number of vibration modes to use in the MFSC scheme.

Based on our findings, the MFSC method is capable of approximating reasonably well the solution of a 25-story building with stochastic stiffness and subject to free vibration. However, it is important to note here that the success of these results is a consequence of the fact that the total response of the building is chiefly commanded by the first three vibration modes. For systems whose total response is not commanded by the first few vibration modes, the MFSC approach may not be the best technique out there to implement since the number of operations can certainly exceed the number of operations needed in a regular scheme such as FSC. Therefore, a better conceptualization of the modal idea may still be necessitated before considering it a good candidate for handling, at low computational cost, large-scale stochastic dynamical systems with the spectral approach.

5.A. Formal derivation for the components of Jacobian matrix F

From (5.8) it follows that the expressions for the components of $\mathbf{F}$ are as follows (recall that $m \in \{1, 2, \ldots, R\}$ is fixed):

$$\begin{aligned}\frac{\partial g^{ui}}{\partial \lambda_m{}^{\beta}} &= -\tilde{M}^{u\ i}_{v\ jk}\,\phi^{v}{}_{m}{}^{j}\frac{\partial \lambda_m{}^{k}}{\partial \lambda_m{}^{\beta}} = -\tilde{M}^{u\ i}_{v\ jk}\,\phi^{v}{}_{m}{}^{j}\delta^{k}{}_{\beta} \\ &= -\tilde{M}^{u\ i}_{v\ j\beta}\,\phi^{v}{}_{m}{}^{j} = -\tilde{M}^{u\ i}_{v\ k\beta}\,\phi^{v}{}_{m}{}^{k} = -\tilde{M}^{u\ i}_{v\ \beta k}\,\phi^{v}{}_{m}{}^{k}.\end{aligned}$$

$$\begin{aligned}\frac{\partial g^{ui}}{\partial \phi^{\alpha}{}_{m}{}^{\beta}} &= \tilde{K}^{u\ i}_{v\ j}\frac{\partial \phi^{v}{}_{m}{}^{j}}{\partial \phi^{\alpha}{}_{m}{}^{\beta}} - \tilde{M}^{u\ i}_{v\ jk}\frac{\partial \phi^{v}{}_{m}{}^{j}}{\partial \phi^{\alpha}{}_{m}{}^{\beta}}\lambda_m{}^{k} \\ &= \tilde{K}^{u\ i}_{v\ j}\delta^{v}{}_{\alpha}\delta^{j}{}_{\beta} - \tilde{M}^{u\ i}_{v\ jk}\delta^{v}{}_{\alpha}\delta^{j}{}_{\beta}\lambda_m{}^{k} = \tilde{K}^{u\ i}_{\alpha\ \beta} - \tilde{M}^{u\ i}_{\alpha\ \beta k}\lambda_m{}^{k}.\end{aligned}$$

$$\frac{\partial h^{i}}{\partial \lambda_m{}^{\beta}} = 0.$$

$$\begin{aligned}\frac{\partial h^{i}}{\partial \phi^{\alpha}{}_{m}{}^{\beta}} &= \delta_{uw}\tilde{M}^{w\ i}_{v\ jk}\frac{\partial \phi^{u}{}_{m}{}^{j}}{\partial \phi^{\alpha}{}_{m}{}^{\beta}}\phi^{v}{}_{m}{}^{k} + \delta_{uw}\tilde{M}^{w\ i}_{v\ jk}\phi^{u}{}_{m}{}^{j}\frac{\partial \phi^{v}{}_{m}{}^{k}}{\partial \phi^{\alpha}{}_{m}{}^{\beta}} \\ &= \delta_{uw}\tilde{M}^{w\ i}_{v\ jk}\delta^{u}{}_{\alpha}\delta^{j}{}_{\beta}\phi^{v}{}_{m}{}^{k} + \delta_{uw}\tilde{M}^{w\ i}_{v\ jk}\phi^{u}{}_{m}{}^{j}\delta^{v}{}_{\alpha}\delta^{k}{}_{\beta} \\ &= \delta_{\alpha w}\tilde{M}^{w\ i}_{v\ \beta k}\phi^{v}{}_{m}{}^{k} + \delta_{uw}\tilde{M}^{w\ i}_{\alpha\ j\beta}\phi^{u}{}_{m}{}^{j} \\ &= \tilde{M}_{\alpha v}{}^{i}{}_{\beta k}\phi^{v}{}_{m}{}^{k} + \tilde{M}_{u\alpha}{}^{i}{}_{j\beta}\phi^{u}{}_{m}{}^{j} \\ &= \tilde{M}_{\alpha v}{}^{i}{}_{\beta k}\phi^{v}{}_{m}{}^{k} + \tilde{M}_{v\alpha}{}^{i}{}_{k\beta}\phi^{v}{}_{m}{}^{k} \\ &= \tilde{M}_{\alpha v}{}^{i}{}_{\beta k}\phi^{v}{}_{m}{}^{k} + \tilde{M}_{\alpha v}{}^{i}{}_{\beta k}\phi^{v}{}_{m}{}^{k} \\ &= 2\,\tilde{M}_{\alpha v}{}^{i}{}_{\beta k}\phi^{v}{}_{m}{}^{k} = 2\,\delta_{\alpha w}\tilde{M}^{w\ i}_{v\ \beta k}\phi^{v}{}_{m}{}^{k}.\end{aligned}$$

To simplify the last partial derivative, we took into account the symmetry of $\tilde{\boldsymbol{M}}^{\flat_1}$ with respect to its first and second slots and fourth and fifth slots.

6. CONCLUSION

This dissertation has presented a novel numerical method called the *flow-driven spectral chaos* (FSC) for the quantification of uncertainties in long-time response of dynamical systems using the spectral approach. To simplify the FSC presentation, the work was divided into three levels of abstraction.

In Chapter 2, a *specialized version* of the FSC method was presented to deal with structural dynamical systems subjected to uncertainties. In this chapter, the FSC method was introduced for the first time using a specific second-order stochastic ODE to study the computational efficiency of the method in detail. As presented, the FSC method revolves around a newly developed concept called 'enriched stochastic flow maps', which is implemented within the FSC scheme to help track the evolution of a finite-dimensional random function space efficiently in time. It was shown that the computational cost of the FSC method is an order of magnitude lower than TD-gPC for comparable solution accuracy. This gain in computational cost was realized because, unlike most existing methods, the number of basis vectors required to track the stochastic part of the solution space, and consequently the computational cost associated with the solution of the resulting system of equations, does not depend upon the dimensionality of the probability space. Four representative numerical examples were presented to demonstrate the performance of the FSC method for long-time integration of second-order stochastic dynamical systems in the context of stochastic dynamics of structures.

In Chapter 3, a *generalized version* of the FSC method was developed to deal with dynamical systems governed by (nonlinear) stochastic ODEs of arbitrary order. The FSC method was not only found to be computationally more efficient than TD-gPC but also far more accurate. To transfer the probability information from one random function space to another, two approaches were developed and studied in this chapter. 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. It was then concluded that the FSC method is capable of quantifying uncertainties with high fidelity, especially for the long-time response of stochastic dynamical systems governed by (nonlinear) ODEs of arbitrary order. Six representative numerical examples, including a

nonlinear problem (the Van-der-Pol oscillator), were 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 was used to demonstrate that when the FSC method is used in conjunction with Monte Carlo integration, the curse of dimensionality can be overcome altogether.

In Chapter 4, a *multi-element version* of the FSC method was formulated to deal with dynamical systems that exhibit discontinuities over the probability space. The key idea of the multi-element FSC (ME-FSC) method is to partition the random domain into several elements so that the FSC method can be implemented on each random element separately. Then, to compute the probability moments of interest over the entire random domain, the law of total probability is used. The benefit of the ME-gPC method is threefold. First, thanks to this technique, the simulation can be run simultaneously on machines with multiple CPU cores (or if needed on separate machines) to reduce the excessive computational burden associated with the simulation. Second, if the random input is discontinuous over the probability space, the random domain can be partitioned into several elements to assure that at most the discontinuity will only appear on regions of measure zero. Third, if an adaptive criterium is introduced within the ME-FSC scheme (so as to allow the elements to get smaller on-the-fly whenever a threshold value is exceeded), the errors can be kept to a minimum during the simulation. In this chapter, four representative numerical examples were presented to demonstrate the effectiveness of the ME-FSC method in dealing with stochastic discontinuities and long-time integration of stochastic dynamical systems.

In Chapter 5, a modal decomposition of the spatial function space was proposed to deal with the dynamics of large-scale structural systems subjected to uncertainties more effectively. The *modal* FSC (MFSC) method was developed herein as a workaround to reduce significantly the dimensionality of the random function space. It was shown that when the MFSC method is employed to quantify the uncertainties of a 25-story building under free vibration, it is possible to obtain good results out of the simulation. However, a better conceptualization of the modal idea may still be necessitated to address more effectively other types of large-scale stochastic dynamical systems.

In conclusion, through the three levels of abstraction, the FSC method was shown to be effective for capturing uncertainties in a wide variety of (nonlinear) stochastic dynamical systems. This is because the FSC method leverages the finite nature of Taylor-based stochastic flow maps commonly used in practice to construct a relatively small random function space via the system's enriched state. Its strength compared to other spectral methods lies in the fact that the FSC method is both curse-of-dimensionality free at the RFS level and capable of quantifying long-time response of stochastic dynamical systems more reliably and at a lower computational cost. Future research may be aimed at developing an adaptive version of the ME-FSC method to reduce the computational cost associated with standard ME-FSC at early times of the simulation, and at developing a more compelling stochastic flow map to address dynamical systems whose flow maps are not analytic on the temporal domain. All this is in order to continue extending the range of applications of the FSC method in the area of uncertainty quantification.

REFERENCES

[1] Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, Peng Hu, Xiao-Yan Yang, Wei-Jun Zhang, Hao Li, Yuxuan Li, Xiao Jiang, Lin Gan, Guangwen Yang, Lixing You, Zhen Wang, Li Li, Nai-Le Liu, Chao-Yang Lu, and Jian-Wei Pan. Quantum computational advantage using photons. *Science*, 2020.

[2] Dirk P Kroese, Thomas Taimre, and Zdravko I Botev. *Handbook of Monte Carlo methods*, volume 706. John Wiley & Sons, 2013.

[3] George Fishman. *Monte Carlo: Concepts, algorithms, and applications*. Springer Science & Business Media, 2013.

[4] Reuven Y Rubinstein and Dirk P Kroese. *Simulation and the Monte Carlo method*, volume 10. John Wiley & Sons, 2016.

[5] Jon D Collins and William T Thomson. The eigenvalue problem for structural systems with statistical properties. *AIAA journal*, 7(4):642–648, 1969.

[6] Wing Kam Liu, Ted Belytschko, and A Mani. Probabilistic finite elements for nonlinear structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 56(1):61–81, 1986.

[7] Wing Kam Liu, Ted Belytschko, and A Mani. Random field finite elements. *International journal for numerical methods in engineering*, 23(10):1831–1845, 1986.

[8] Michał Kleiber and Tran Duong Hien. *The stochastic finite element method: Basic perturbation technique and computer implementation*. Jhon Wiley & Sons, 1992.

[9] M Shinozuka and G Deodatis. Response variability of stochastic finite element systems. *Journal of Engineering Mechanics*, 114(3):499–519, 1988.

[10] Fumio Yamazaki, Masanobu Shinozuka, and Gautam Dasgupta. Neumann expansion for stochastic finite element analysis. *Journal of Engineering Mechanics*, 114(8):1335–1354, 1988.

[11] George Deodatis. Weighted integral method. I: Stochastic stiffness matrix. *Journal of Engineering Mechanics*, 117(8):1851–1864, 1991.

[12] George Deodatis and Masanobu Shinozuka. Weighted integral method. II: Response variability and reliability. *Journal of Engineering Mechanics*, 117(8):1865–1877, 1991.

[13] Dongbin Xiu. *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton University Press, 2010.

[14] O. P. Le Maître and O. M. Knio. *Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics*. Springer Science & Business Media, 2010.

[15] T. J. Sullivan. *Introduction to Uncertainty Quantification*, volume 63. Springer, 2015.

[16] Michael Stein. Large sample properties of simulations using Latin hypercube sampling. *Technometrics*, 29(2):143–151, 1987.

[17] Wei-Liem Loh et al. On Latin hypercube sampling. *The annals of statistics*, 24(5):2058–2080, 1996.

[18] Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*, volume 63. Siam, 1992.

[19] Harald Niederreiter, Peter Hellekalek, Gerhard Larcher, and Peter Zinterhof. *Monte Carlo and Quasi-Monte Carlo Methods 1996: Proceedings of a Conference at the University of Salzburg, Austria, July 9-12, 1996*. Springer-Verlag New York, 1998.

[20] Bennett L Fox. *Strategies for Quasi-Monte Carlo*, volume 22. Springer Science & Business Media, 1999.

[21] James Yeh. *Real analysis, theory of measure and integration*. World Scientific, 3 edition, 2014.

[22] Walter Rudin. *Real and complex analysis*. McGraw-Hill Education, 3 edition, 1986.

[23] Walter Rudin. *Real and complex analysis*. McGraw-Hill, 1987.

[24] Roger G. Ghanem and Pol D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Springer-Verlag New York, 1991.

[25] Philip J Davis and Philip Rabinowitz. *Methods of numerical integration*. Courier Corporation, 2007.

[26] Erich Novak and Klaus Ritter. High dimensional integration of smooth functions over cubes. *Numerische Mathematik*, 75(1):79–97, 1996.

[27] Hans-Joachim Bungartz and Michael Griebel. Sparse grids. *Acta numerica*, 13:147–269, 2004.

[28] Gunther Leobacher and Friedrich Pillichshammer. *Introduction to quasi-Monte Carlo integration and applications*. Springer, 2014.

[29] John Charles Butcher and JC Butcher. *The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods*, volume 512. Wiley New York, 1987.

[30] Nathan M Newmark. A method of computation for structural dynamics. *Journal of the engineering mechanics division*, 85(3):67–94, 1959.

[31] Anil K Chopra. *Dynamics of Structures: Theory and Applications to Earthquake Engineering*. Prentice Hall New Jersey, 4th edition, 2012.

[32] Norbert Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938.

[33] Dongbin Xiu and George Em Karniadakis. The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM journal on scientific computing*, 24(2):619–644, 2002.

[34] Marc Gerritsma, Jan-Bart Van der Steen, Peter Vos, and George Karniadakis. Time-dependent generalized polynomial chaos. *Journal of Computational Physics*, 229(22):8333–8363, 2010.

[35] Vincent Heuveline and Michael Schick. A hybrid generalized polynomial chaos method for stochastic dynamical systems. *International Journal for Uncertainty Quantification*, 4(1), 2014.

[36] Dirk M Luchtenburg, Steven L Brunton, and Clarence W Rowley. Long-time uncertainty propagation using generalized polynomial chaos and flow map composition. *Journal of Computational Physics*, 274:783–802, 2014.

[37] H Cagan Ozen and Guillaume Bal. Dynamical polynomial chaos expansions and long time evolution of differential equations with random forcing. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):609–635, 2016.

[38] ASCE/SEI 7-16. Minimum design loads and associated criteria for buildings and other structures, 2017.

[39] ATC-40. Seismic evaluation and retrofit of concrete buildings. Technical Report SSC 96-01, Applied Technology Council, November 1996.

[40] PEER TBI-2.0. Guidelines for performance-based seismic design of tall buildings. Technical Report 2017/06, Pacific Earthquake Engineering Center, May 2017.

[41] month=September year=2012-institution=Federal Emergency Management Agency FEMA P-58, title=Seismic performance assessment of buildings. Technical report.

[42] Roger Ghanem and Pol D Spanos. Polynomial chaos in stochastic finite elements. *Journal of Applied Mechanics*, 57(1):197–202, 1990.

[43] Xiaoliang Wan and George Em Karniadakis. An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *Journal of Computational Physics*, 209(2):617–642, 2005.

[44] Xiaoliang Wan and George Em Karniadakis. Multi-element generalized polynomial chaos for arbitrary probability measures. *SIAM Journal on Scientific Computing*, 28(3):901–928, 2006.

[45] Xiaoliang Wan and George Em Karniadakis. Long-term behavior of polynomial chaos in stochastic flow simulations. *Computer methods in applied mechanics and engineering*, 195(41-43):5582–5596, 2006.

[46] Jasmine Foo, Xiaoliang Wan, and George Em Karniadakis. The multi-element probabilistic collocation method (ME-PCM): Error analysis and applications. *Journal of Computational Physics*, 227(22):9572–9595, 2008.

[47] Nitin Agarwal and Narayan R Aluru. A domain adaptive stochastic collocation approach for analysis of MEMS under uncertainties. *Journal of Computational Physics*, 228(20):7662–7688, 2009.

[48] Xiang Ma and Nicholas Zabaras. An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *Journal of Computational Physics*, 228(8):3084–3113, 2009.

[49] Gaurav Kewlani and Karl 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*, pages 1177–1182. IEEE, 2009.

[50] Jasmine Foo and George Em Karniadakis. Multi-element probabilistic collocation method in high dimensions. *Journal of Computational Physics*, 229(5):1536–1557, 2010.

[51] P Prempraneerach, Franz S Hover, Michael S Triantafyllou, and George E Karniadakis. Uncertainty quantification in simulations of power systems: Multi-element polynomial chaos methods. *Reliability Engineering & System Safety*, 95(6):632–646, 2010.

[52] John D Jakeman, Akil Narayan, and Dongbin Xiu. Minimal multi-element stochastic collocation for uncertainty quantification of discontinuous functions. *Journal of Computational Physics*, 242:790–808, 2013.

[53] Themistoklis P Sapsis and Pierre FJ Lermusiaux. Dynamically orthogonal field equations for continuous stochastic dynamical systems. *Physica D: Nonlinear Phenomena*, 238(23):2347–2360, 2009.

[54] Minseok Choi, Themistoklis P Sapsis, and George Em Karniadakis. A convergence study for SPDEs using combined polynomial chaos and dynamically-orthogonal schemes. *Journal of Computational Physics*, 245:281–301, 2013.

[55] MP Ueckermann, Pierre FJ Lermusiaux, and Themistoklis P Sapsis. Numerical schemes for dynamically orthogonal equations of stochastic fluid and ocean flows. *Journal of Computational Physics*, 233:272–294, 2013.

[56] Mulin Cheng, Thomas Y Hou, and Zhiwen Zhang. A dynamically bi-orthogonal method for time-dependent stochastic partial differential equations. I: Derivation and algorithms. *Journal of Computational Physics*, 242:843–868, 2013.

[57] Mulin Cheng, Thomas Y Hou, and Zhiwen Zhang. A dynamically bi-orthogonal method for time-dependent stochastic partial differential equations. II: Adaptivity and generalizations. *Journal of Computational Physics*, 242:753–776, 2013.

[58] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 1965.

[59] Philip Holmes, John L Lumley, Gahl Berkooz, and Clarence W Rowley. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge university press, 2012.

[60] Oliver G Ernst, Antje Mugler, Hans-Jörg Starkloff, and Elisabeth Ullmann. On the convergence of generalized polynomial chaos expansions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46(2):317–339, 2012.

[61] Ward Cheney and David Kincaid. *Linear algebra: Theory and applications*. Jones & Bartlett Learning, 2nd edition, 2010.

[62] The MathWorks Inc. MATLAB, 2016.

[63] Jagmohan L Humar. *Dynamics of Structures*. CRC Press, 2nd edition, 2012.

[64] Timothy D Ancheta, Robert B Darragh, Jonathan P Stewart, Emel Seyhan, Walter J Silva, Brian S-J Chiou, Katie E Wooddell, Robert W Graves, Albert R Kottke, David M Boore, et al. NGA-West2 database. *Earthquake Spectra*, 30(3):989–1005, 2014.

[65] Ivo Babuška, Fabio Nobile, and Raul Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 45(3):1005–1034, 2007.

[66] Fabio Nobile, Raúl Tempone, and Clayton G Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46(5):2309–2345, 2008.

[67] Aretha L Teckentrup, Peter Jantsch, Clayton G Webster, and Max Gunzburger. A multilevel stochastic collocation method for partial differential equations with random input data. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):1046–1074, 2015.

[68] Norbert Wiener and Aurel Wintner. The discrete chaos. *American Journal of Mathematics*, 65(2):279–298, 1943.

[69] Norbert Wiener. *Nonlinear problems in random theory*. MIT press, 1958.

[70] Vito Volterra, M Tomassetti, and FS Zarlatti. *Leçons sur les équations intégrales et les équations intégro-différentielles*. Gauthier-Villars Paris, 1913.

[71] Robert H Cameron and William T Martin. The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals. *Annals of Mathematics*, pages 385–392, 1947.

[72] D Lucor, D Xiu, and 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.

[73] RV Field and M Grigoriu. A new perspective on polynomial chaos. *Computational Stochastic Mechanics, Spanos & Deodatis eds, Millpress*, pages 199–205, 2003.

[74] Kiyosi Itô. Multiple Wiener integral. *Journal of the Mathematical Society of Japan*, 3(1):157–169, 1951.

[75] Hisanao Ogura. Orthogonal functionals of the Poisson process. *IEEE Transactions on Information Theory*, 18(4):473–481, 1972.

[76] Adrian Segall and Thomas Kailath. Orthogonal functionals of independent-increment processes. *IEEE Transactions on Information Theory*, 22(3):287–298, 1976.

[77] Martin Schetzen. Nonlinear system modeling based on the Wiener theory. *Proceedings of the IEEE*, 69(12):1557–1573, 1981.

[78] David Douglas Engel. *The multiple stochastic integral*, volume 265. American Mathematical Soc., 1982.

[79] Roger Ghanem and Pol Dimitrios Spanos. A stochastic Galerkin expansion for nonlinear random vibration analysis. *Probabilistic Engineering Mechanics*, 8(3-4):255–264, 1993.

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

[81] Omar M Knio, Habib N Najm, Roger G Ghanem, et al. A stochastic projection method for fluid flow. I: Basic formulation. *Journal of computational Physics*, 173(2):481–511, 2001.

[82] Olivier P Le Maıtre, Matthew T Reagan, Habib N Najm, Roger G Ghanem, and Omar M Knio. A stochastic projection method for fluid flow. II: Random process. *Journal of computational Physics*, 181(1):9–44, 2002.

[83] Dongbin Xiu and George Em Karniadakis. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of computational physics*, 187(1):137–167, 2003.

[84] Badrinarayanan Velamur Asokan and Nicholas Zabaras. Using stochastic analysis to capture unstable equilibrium in natural convection. *Journal of Computational Physics*, 208(1):134–153, 2005.

[85] OM Knio and OP Le Maitre. Uncertainty propagation in CFD using polynomial chaos decomposition. *Fluid Dynamics Research*, 38(9):616, 2006.

[86] Habib N Najm. Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annual review of fluid mechanics*, 41:35–52, 2009.

[87] Xiaoliang Wan and George Em Karniadakis. Beyond Wiener-Askey expansions: Handling arbitrary PDFs. *Journal of Scientific Computing*, 27(1-3):455–464, 2006.

[88] S Oladyshkin, H Class, R Helmig, and 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):1508–1518, 2011.

[89] Gaurav Kewlani, Justin Crawford, and Karl Iagnemma. A polynomial chaos approach to the analysis of vehicle dynamics under uncertainty. *Vehicle system dynamics*, 50(5):749–774, 2012.

[90] Eleonora Musharbash, Fabio Nobile, and Tao Zhou. Error analysis of the dynamically orthogonal approximation of time dependent random PDEs. *SIAM Journal on Scientific Computing*, 37(2):A776–A810, 2015.

[91] Sergei Abramovich Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. In *Doklady Akademii Nauk*, volume 148, pages 1042–1045. Russian Academy of Sciences, 1963.

[92] Hugo Esquivel, Arun Prakash, and Guang Lin. Flow-driven spectral chaos (FSC) method for simulating long-time dynamics of arbitrary-order non-linear stochastic dynamical systems. *Journal of Computational Physics*, page 110044, 2020.

[93] Apple Inc. Apple's frameworks through the Xcode IDE, 2021.

[94] Apple Inc. et al. Swift programming language, 2021.

[95] Steven A Orszag and LR Bissonnette. Dynamical properties of truncated Wiener-Hermite expansions. *The Physics of Fluids*, 10(12):2603–2613, 1967.

[96] Roger Ghanem and Debraj Ghosh. Efficient characterization of the random eigenvalue problem in a polynomial chaos decomposition. *International Journal for Numerical Methods in Engineering*, 72(4):486–504, 2007.

[97] L Susan Blackford, Antoine Petitet, Roldan Pozo, Karin Remington, R Clint Whaley, James Demmel, Jack Dongarra, Iain Duff, Sven Hammarling, Greg Henry, et al. An updated set of basic linear algebra subprograms (blas). *ACM Transactions on Mathematical Software*, 28(2):135–151, 2002.

[98] The MathWorks Inc. MATLAB, 2020.