Generalized Spectral Decomposition for Stochastic Non Linear Problems

Anthony Nouy, Olivier Le Maitre

Generalized Spectral Decomposition for Stochastic Non Linear Problems [1]

Anthony Nouy [a,*] and Olivier P. Le Maître [b,c,**]

[a] *Research Institute in Civil Engineering and Mechanics (GeM), Nantes Atlantic University, Ecole Centrale Nantes, UMR CNRS 6183, 2 rue de la Houssinière, B.P. 92208, 44322 Nantes Cedex 3, France.*

[b] *LIMSI-CNRS, BP133, F-91143 Orsay, France.*

[c] *DEN/DM2S/SFME, Centre d'Etudes Nucléaires, Saclay, France*

Abstract

We present an extension of the Generalized Spectral Decomposition method for the resolution of non-linear stochastic problems. The method consists in the construction of a reduced basis approximation of the Galerkin solution and is independent of the stochastic discretization selected (polynomial chaos, stochastic multi-element or multiwavelets). Two algorithms are proposed for the sequential construction of the successive generalized spectral modes. They involve decoupled resolutions of a series of deterministic and low dimensional stochastic problems. Compared to the classical Galerkin method, the algorithms allow for significant computational savings and require minor adaptations of the deterministic codes. The methodology is detailed and tested on two model problems, the one-dimensional steady viscous Burgers equation and a two-dimensional non-linear diffusion problem. These examples demonstrate the effectiveness of the proposed algorithms which exhibit convergence rates with the number of modes essentially dependent on the spectrum of the stochastic solution but independent of the dimension of the stochastic approximation space.

Key words: Uncertainty quantification, Stochastic spectral Decompositions, Non-linear problem, Generalized Spectral Decomposition, Eigenproblem

* Corresponding author. Tel: 33-2-51-12-55-20;fax: 33-2-51-12-55-57

**Corresponding author. Tel: 33-1-69-85-80-76; fax: 33-1-69-85-80-88

Email addresses: `anthony.nouy@univ-nantes.fr` (Anthony Nouy), `olm@limsi.fr` (Olivier P. Le Maître).

URLs: `http://www.univ-nantes.fr/nouy-a` (Anthony Nouy), `http://www.limsi.fr/Individu/olm` (Olivier P. Le Maître).

[1] This work is supported by the French National Research Agency (grant ANR-06-JCJC-0064) and by GdR MoMaS with partners ANDRA, BRGM, CEA, CNRS, EDF, IRSN.

1 Introduction

The increasing availability of computational resources and complexity of numerical models has stressed the need for efficient techniques to account for uncertainties in model data and incomplete knowledge of the simulated system. Uncertainty quantification (UQ) methods are designed to address this need by providing a characterization of the uncertainty in the model output. The uncertainty characterization and level of information provided depend on the UQ method selected and range from the construction of simple confidence intervals to the determination of complete probability laws. Among the different UQ methods available, the polynomial chaos (PC) methods [40, 5, 13] are receiving a growing interest as they provide a rich uncertainty characterization thanks to their probabilistic character. In fact, PC methods for UQ have been constantly improved and applied to problems with increasing complexity (*e.g.* non-linear ones) since the early works of Ghanem and Spanos [13].
The fundamental concept of PC methods is to treat the UQ problem in a probabilistic framework, where the uncertain model data are parameterized using a finite set of random variables which are subsequently regarded as the generator of new dimensions along which the model solution is dependent. A convergent expansion along the uncertainty dimensions is then sought in terms of orthogonal basis functions spanning an appropriate stochastic space. The expansion coefficients provide a complete characterization of the uncertain solution in a convenient format allowing for straightforward post-treatment and uncertainty analysis such as the assessment of the impact of specific uncertain data source on specific observables.

There are two distinct classes of techniques for the determination of the expansion coefficients. The non-intrusive techniques, such as quadrature-based projections [34, 20] and regressions [4], offer the advantage of requiring the availability of a deterministic code only, but are limited by the need of computing the solution for a large number of realizations of the uncertain data. Many works are currently focusing on numerical strategies for the minimization of the number of solutions to be computed, essentially through the use of coarse or adaptive quadrature formulas [16, 11]. The second class of techniques relies on the model equations to derive a problem for the expansion coefficients through Galerkin-type procedures. It yields accurate solutions but usually requires the resolution of a large set of equations calling for ad hoc numerical strategies, such as Krylov type iterations [12, 32, 15] and preconditioning techniques [33, 21], as well as an adaptation of the deterministic codes. The method presented in this paper focuses on the minimization of the computational cost in Galerkin methods for non-linear models.

The essential motivation behind PC methods is the promise of obtaining accurate estimates of the uncertain solution with a limited number of terms

in the expansion. However, as applications and uncertainty settings gain in complexity, the dimension of the expansion basis needed to yield accurate estimates quickly increases with significant increase in the computational cost and memory requirements. These limitations have been partially addressed by using better suited stochastic bases both in terms of probability distribution of the random variables [41] and approximation properties of the basis functions using so-called finite element, multi-element or multi-wavelet bases [7, 2, 10, 17, 18, 38]. An interesting feature of finite-element, multi-element and multi-wavelet bases is the possibility to enrich adaptively the stochastic approximation basis to the sought solution (see for instance [18, 38, 39, 19, 22]).

Another way to minimize the size and numerical cost of Galerkin computations is to seek the approximate solution on a reduced space. It is remarked that such reduction approach should not be opposed or understood as an alternative to the adaptive methods mentioned above, but would actually further improve their efficiency since adaptive techniques require the resolution of large Galerkin problems, though local ones. The main idea of reduced approximations is to take advantage of the structure of the full approximation space, which results from the tensor product of the deterministic and stochastic approximation spaces: if one is able to appropriately reduce the deterministic or stochastic approximation space, to a low dimensional sub-space, the size of the Galerkin problem to be solved drastically reduces too. Of course, the determination of a low dimensional sub-space that still accurately captures the essential features of the solution is not immediate since the solution is unknown. In [9], the Galerkin problem is first solved on a coarse deterministic mesh to provide a coarse estimate of the solution which is then decomposed into its principal components through Karhunen-Loeve (KL) expansion. The first random coefficients of the KL expansion are then used as a reduced stochastic basis in the Galerkin problem considered now on a fine deterministic mesh. Alternatively, in [23], a Neumann expansion of the operator is used to obtain an estimate of the covariance operator of the solution. The dominant eigenspace of the approximate covariance operator is then considered as the reduced deterministic (spatial) sub-space to be used subsequently in the Galerkin procedure. In fact, as for the first approach, this can be interpreted as a coarse *a priori* KL expansion of the solution. These two approaches have demonstrated their effectiveness in reducing the size of the Galerkin problem solved *in fine*. However, the second approach, based on Neumann expansion, is dedicated to linear problems, and the extension of the first approach to highly non-linear problems, such as for instance the Navier-Stokes equations, seems critical due to limitations in the possible deterministic coarsening: the reduced basis may simply miss important features of the non-linear solution. Another alternative, called the Stochastic Reduced Basis Method [25, 35], has been proposed for the *a priori* construction of reduced basis. In this method, dedicated to linear problems, the reduced basis is a basis of a low-dimensional Krylov sub-space of the random operator associated with the right hand side.

It captures approximatively the active upper spectrum of the random operator. The main difference with the above techniques is that the reduced basis is random. The method does not take part of the tensor product structure of the function space and then does not circumvent the problem of memory requirements. Moreover, the components of the solution on this basis, obtained through a Galerkin projection, leads to a system of equations which has not a conventional form.

We thus investigate in this paper the extension of the so-called Generalized Spectral Decomposition (GSD) method which does not require one to provide a reduced basis (*a priori* or determined by alternative means) but that instead yields by itself the "optimal" reduced basis.

The Generalized Spectral Decomposition (GSD) method consists in searching an optimal decomposition of the solution u to a stochastic problem under the form $\sum_{i=1}^{M} U_i \lambda_i$, where the U_i are deterministic functions while λ_i are random variables. In this context, the set of λ_i (resp. of U_i) are understood as a reduced basis of random variables (resp. of deterministic functions). Optimal decompositions could be easily defined if the solution u were known. Such a decomposition can for example be obtained by a KL expansion (or classical spectral decomposition) of u, which is the optimal decomposition with respect to a classical inner product. The GSD method consists in defining an optimality criterion for the decomposition which is based on the equation(s) solved by the solution but not on the solution itself. The construction of the decomposition therefore does not require to know the solution *a priori* or to provide a surrogate (approximation on coarser mesh or low order Neumann expansion) as pointed previously. The GSD method was first proposed in [27] in the context of linear stochastic problems. In the case of linear symmetric elliptic coercive problems, by defining an optimal decomposition with respect to the underlying optimization problem, the functions U_i (resp. λ_i) were shown to be solutions of an eigen-like problem. Ad-hoc algorithms, inspired by power method for classical eigenproblems, have been proposed in [27] for the resolution of this eigen-like problem, while improved algorithms and in-depth analysis of the GSD method for a wider class of linear problems (in particular time-dependent problems) can be found in [28]. The main advantage of these algorithms is to separate the resolution of a few deterministic problems and a few reduced stochastic problems (*i.e.* using a reduced basis of deterministic functions). These algorithms lead to significant computational savings when compared to classical resolution techniques of stochastic Galerkin equations. A first attempt for extending the GSD method to non-linear problems has been investigated in [26]: algorithms derived for the linear case were simply applied to subsequent linear stochastic problems arising from a classical non-linear iterative solver. Reduced basis generated at each iteration were stored, sorted and re-used for subsequent iterations. In this paper, we propose a "true" extension of the GSD to non-linear problems, where we directly construct an

optimal decomposition of the solution with regard to the initial non-linear problem.

The outline of the paper is as follows. In section 2, we introduce a general formulation of non-linear stochastic problems and the associated stochastic Galerkin schemes. In section 3, we present the extension of GSD for non-linear problems. In particular, we provide some basic mathematical considerations which motivate this extension. The GSD is interpreted as the solution of an eigen-like problem and two ad-hoc algorithms are proposed for building the decomposition. These algorithms are inspired from the ones proposed in [27] in the context of linear stochastic problems. Then, the GSD method is applied to two non-linear models: the steady viscous Burgers equation (sections 4 and 5) and a stationary diffusion equation (sections 6 and 7). Computational aspects of the GSD are detailed for each of these model problems. Finally, in section 8, we summarize the main findings of this work and we discuss future improvements and extensions of the method.

2 Non-linear stochastic problems

2.1 Variational formulation

We adopt a probabilistic modeling of uncertainties and introduce an abstract probability space $(\Theta, \mathcal{B}, P)$. Θ is the space of elementary events, $\mathcal{B}$ a σ-algebra on Θ and P a probability measure. We consider non linear problems having the following semi-variational form:
Given an elementary event θ, find $u(\theta) \in \mathscr{V}$ such that we have almost surely

$$b(u(\theta), v; \theta) = l(v; \theta) \quad \forall v \in \mathscr{V}, \tag{1}$$

where $\mathscr{V}$ is a given vector space, eventually of finite dimension, b and l are semi-linear and linear forms respectively. The forms b and l may depend on the elementary event θ. In this paper, we consider that $\mathscr{V}$ does not depend on the elementary event. It could be the case when considering partial differential equations defined on random domains [30, 29]. On the stochastic level, we introduce a suitable function space $\mathscr{S}$ for random variables taking values in $\mathbb{R}$. The full variational formulation of the problem writes:
Find $u \in \mathscr{V} \otimes \mathscr{S}$ such that

$$B(u, v) = L(v) \quad \forall v \in \mathscr{V} \otimes \mathscr{S}, \tag{2}$$

where the semi-linear and linear forms B and L have for respective expressions:

$$B(u,v) = \int_\Theta b(u(\theta), v(\theta); \theta)\, dP(\theta) := E(b(u,v;\cdot)), \tag{3}$$

$$L(v) = \int_\Theta l(v(\theta); \theta)\, dP(\theta) := E(l(v;\cdot)). \tag{4}$$

where $E(\cdot)$ denotes the mathematical expectation.

2.2 Stochastic discretization

In this article, we consider a parametric modeling of uncertainties. Semilinear form b and linear form l are parametrized using a finite set of N real continuous random variables $\boldsymbol{\xi}$ with known probability law $P_{\boldsymbol{\xi}}$. Then, by the Doob-Dynkin's lemma [31], we have that the solution of problem (1) can be written in terms of $\boldsymbol{\xi}$, *i.e.* $u(\theta) \equiv u(\boldsymbol{\xi})$. The stochastic problem can then be reformulated in the N-dimensional image probability space $(\Xi, \mathcal{B}_\Xi, P_{\boldsymbol{\xi}})$, where $\Xi \subset \mathbb{R}^N$ denotes the range of $\boldsymbol{\xi}$. The expectation operator has the following expression in the image probability space:

$$E(f(\cdot)) = \int_\Theta f(\boldsymbol{\xi}(\theta))\, dP(\theta) = \int_\Xi f(\mathbf{y}) dP_{\boldsymbol{\xi}}(\mathbf{y}). \tag{5}$$

Since we are interested in finding an approximate stochastic solution of equation (1), function space $\mathscr{S}$ is considered as a finite dimensional subspace of $L^2(\Xi, dP_{\boldsymbol{\xi}})$, the space of real second order random variables defined on Ξ. Different types of approximation are available at the stochastic level: continuous polynomial expansion [13, 41, 36], piecewise polynomial expansion [7], multiwavelets [17, 18]. At this point, it is stressed that the method proposed in this paper is independent of the type of stochastic approximation used.

Remark 1 *The choice of a suitable function space $\mathscr{S}$ is a non trivial question in the infinite dimensional case. Several interpretations of stochastic partial differential equations (SPDE) are generally possible, e.g. by introducing the concept of Wick product between random fields, leading to well posed problems and then to different possible solutions [14, 3, 37]. These mathematical considerations are beyond the scope of this article. For non-linear problems dealt with in this article, where a classical interpretation of products between random fields is used [2, 23], a possible choice could consist in classical Banach spaces $L^p(\Xi, dP_{\boldsymbol{\xi}}) \subset L^2(\Xi, dP_{\boldsymbol{\xi}})$, $2 \leqslant p < \infty$. Usual approximation spaces being contained and dense in these Banach spaces, it ensures the consistency of the approximation.*

In what follows, we will mainly use the initial probability space $(\Theta, \mathcal{B}, P)$. The reader must keep in mind that at each moment, the elementary event $\theta \in \Theta$

can be replaced by $\boldsymbol{\xi} \in \Xi$ in any expression.

3 General Spectral Decomposition for non linear problems

3.1 Principle

The Generalized Spectral Decomposition (GSD) method consists in searching an approximate low-order decomposition of the solution to problem (2):

$$u_M(\theta) = \sum_{i=1}^{M} U_i \lambda_i(\theta), \tag{6}$$

where $U_i \in \mathscr{V}$ are deterministic functions while $\lambda_i \in \mathscr{S}$ are random variables (*i.e.* real-valued functions of the elementary random event). In this context, the set of λ_i (resp. of U_i) can be understood as a reduced basis of random variables (resp. of deterministic functions). In this section, we will see in which sense optimal reduced basis can be thought as solutions of eigen-like problems. Starting from this interpretation, we will propose two simple and efficient algorithms for building the generalized spectral decomposition.

3.2 Definition of an optimal couple (U, λ)

First, let us explain how to define an optimal couple $(U, \lambda) \in \mathscr{V} \times \mathscr{S}$. The proposed definition is a direct extension to the non-linear case of the definition introduced in [28].
It is remarked that if U was known and fixed, the following Galerkin orthogonality criterium would lead to a suitable definition for λ:

$$B(\lambda U, \beta U) = L(\beta U) \quad \forall \beta \in \mathscr{S}. \tag{7}$$

In other words, it consists in defining λU as the Galerkin approximation of problem (2) in the sub-space $U \otimes \mathscr{S} \subset \mathscr{V} \otimes \mathscr{S}$.
Alternatively, if λ was known and fixed, the following Galerkin orthogonality criterium would lead to a suitable definition for U:

$$B(\lambda U, \lambda V) = L(\lambda V) \quad \forall V \in \mathscr{V}. \tag{8}$$

In other words, it consists in defining λU as the Galerkin approximation of problem (2) in the sub-space $\mathscr{V} \otimes \lambda \subset \mathscr{V} \otimes \mathscr{S}$.
As a shorthand notation, we write $\lambda = f(U)$ the solution of equation (7) and $U = F(\lambda)$ the solution of equation (8). It should be clear that a natural

definition of an optimal couple (U, λ) consists in satisfying simultaneously equations (7) and (8). The problem can then write: find $\lambda \in \mathscr{S}$ and $U \in \mathscr{V}$ such that

$$U = F(\lambda) \quad \text{and} \quad \lambda = f(U). \tag{9}$$

The problem can be formulated on U as follows: find $U \in \mathscr{V}$ such that

$$U = F \circ f(U) := T(U), \tag{10}$$

where mapping T is a homogeneous mapping of degree 1:

$$T(\alpha U) = \alpha T(U) \quad \forall \alpha \in \mathbb{R}^*. \tag{11}$$

This property comes from properties of f and F, which are both homogeneous mappings of degree (-1):

$$\forall \alpha \in \mathbb{R}^*, \quad f(\alpha U) = \alpha^{-1} f(U), \quad F(\alpha \lambda) = \alpha^{-1} F(\lambda). \tag{12}$$

The homogeneity property of T allows to interpret equation (10) as an eigen-like problem where the solution U is interpreted as a generalized eigenfunction.

By analogy with classical eigenproblems, each eigenfunction is associated with a unitary eigenvalue. The question is then: how to define the best generalized eigenfunction among all possible generalized eigenfunctions ? A natural answer is: the best U is the one which maximizes the norm $\|Uf(U)\|$ of the approximate solution $Uf(U)$, *i.e.* such that it gives the highest contribution to the generalized spectral decomposition. In order to provide a more classical writing of an eigen-problem, we now rewrite the approximation as $\alpha U f(U)/\|Uf(U)\|$, with $\alpha \in \mathbb{R}^+$. The problem is then to find a couple $(U, \alpha) \in \mathscr{V} \times \mathbb{R}^+$ such that α is maximum and such that the following Galerkin orthogonality criterium is still satisfied:

$$\alpha U = F(f(U)/\|Uf(U)\|) = \|Uf(U)\| T(U) := \widetilde{T}(U). \tag{13}$$

The mapping $\sigma : U \in \mathscr{V} \mapsto \|Uf(U)\| \in \mathbb{R}^+$ is a homogeneous mapping of degree 0. Then, mapping $\widetilde{T}$, which is a simple rescaling of T, is still homogeneous of degree 1, so that equation (13) can be interpreted as an eigen-like problem on $\widetilde{T}$: find $(U, \alpha) \in \mathscr{V} \times \mathbb{R}^+$ such that

$$\widetilde{T}(U) = \alpha U \tag{14}$$

U is a generalized eigenfunction of $\widetilde{T}$ if and only if it is a generalized eigenfunction of T. A generalized eigenfunction is associated with a generalized eigenvalue $\alpha = \sigma(U)$ of mapping $\widetilde{T}$. The best $U \in \mathscr{V}$ then appears to be the generalized eigenfunction associated with the dominant generalized eigenvalue of $\widetilde{T}$.

Remark 2 *In the case where B is a bounded elliptic coercive bilinear form, it is proved in [27] that the dominant generalized eigenfunction U is such that it minimizes the error $(u - Uf(U))$ with respect to the norm induced by B.*

Remark 3 *Let us note that the previous reasoning can be made on a problem formulated on λ, writing: find $(\lambda, \alpha) \in \mathscr{S} \times \mathbb{R}^+$ such that*

$$\widetilde{T}^*(\lambda) = \alpha\lambda, \tag{15}$$

where $\widetilde{T}^*(\lambda) = \sigma^*(\lambda) f \circ F(\lambda)$, with $\sigma^*(\lambda) = \|F(\lambda)\lambda\|$. We can easily show that if U is a generalized eigenfunction of $\widetilde{T}$, then $\lambda = f(U)$ is a generalized eigenfunction of $\widetilde{T}^*$, associated with the generalized eigenvalue $\sigma^*(\lambda) = \sigma(f(U))$. Problems on U and λ are completely equivalent. In this article, we arbitrarily focus on the problem on U.

3.3 A progressive definition of the decomposition

Following the previous observations, we now propose to build progressively the generalized spectral decomposition defined in equation (6). The couples (U_i, λ_i) are defined one after the others. To this end, let us assume that u_M is known. We denote $(U, \lambda) \in \mathscr{V} \otimes \mathscr{S}$ the next couple to be defined. A natural definition of this couple still consists in satisfying the two following Galerkin orthogonality criteria:

$$B(u_M + \lambda U, \beta U) = L(\beta U) \quad \forall \beta \in \mathscr{S}, \tag{16}$$
$$B(u_M + \lambda U, \lambda V) = L(\lambda V) \quad \forall V \in \mathscr{V}. \tag{17}$$

As a shorthand notation, we write $\lambda = f_M(U)$ the solution of equation (16) and $U = F_M(\lambda)$ the solution of equation (17). This problem can still be formulated on U as follows: find $U \in \mathscr{V}$ such that

$$U = F_M \circ f_M(U) := T_M(U). \tag{18}$$

where mapping T_M is an homogeneous mapping of degree 1. Problem (18) can still be interpreted as an eigen-like problem. In fact, by analogy with classical eigenproblems, operator T_M can be interpreted as a "deflation" of the initial operator T (see [28] for details).
Introducing $\sigma_M(U) = \|U f_M(U)\|$ allows to reformulate problem (18) as an eigen-like problem on mapping $\widetilde{T}_M = \sigma_M(U) T_M(U)$: find the dominant generalized eigenpair $(U, \alpha) \in \mathscr{V} \times \mathbb{R}^+$, satisfying:

$$\widetilde{T}_M(U) = \alpha U, \tag{19}$$

where $\alpha = \sigma_M(U)$ appears to be the generalized eigenvalue of $\widetilde{T}_M$ associated with the generalized eigenfunction U.

Finally, denoting by $(U_i, \sigma_{i-1}(U_i))$ the dominant eigenpair of operator $\widetilde{T}_{i-1}$, the generalized decomposition of order M is then defined as

$$u_M = \sum_{i=1}^{M} U_i f_{i-1}(U_i) = \sum_{i=1}^{M} \sigma_{i-1}(U_i) U_i f_{i-1}(U_i) / \|U_i f_{i-1}(U_i)\|, \tag{20}$$

where for consistency, we let $u_0 = 0$.

3.4 Algorithms for building the decomposition

With the previous definition, optimal couples (U_i, λ_i) appears to be dominant eigenfunctions of successive eigen-like problems. The following algorithms, initially proposed in [27] for linear stochastic problems, are here extended to the non-linear framework. In the following, we denote $W_M = (U_1, \ldots, U_M) \in (\mathscr{V})^M$, $\Lambda_M = (\lambda_1, \ldots, \lambda_M) \in (\mathscr{S})^M$ and

$$u_M(\theta) := W_M \cdot \Lambda_M(\theta). \tag{21}$$

3.4.1 Basic power-type method: algorithm 1

In order to find the dominant eigenpair $(U, \sigma_M(U))$ of eigen-like problem (19), we suggest to use a power-type algorithm. It consists in building the series $U^{(k+1)} = \widetilde{T}_M(U^{(k)})$, or equivalenty $U^{(k+1)} = \gamma^{(k)} \widetilde{T}_M(U^{(k)})$, where $\gamma^{(k)} \in \mathbb{R}$ is a rescaling factor. We emphasize that the rescaling factor has no influence on the convergence of this series, due to homogeneity property of mapping $\widetilde{T}_M$ (inherited from those of f_M and F_M). This strategy leads to algorithm 1, which can be interpreted as a power-type algorithm with deflation for building the whole decomposition.

Algorithm 1 *Power-type algorithm*

1: ***for*** $i = 1 \ldots M$ ***do***
2: *Initialize* $\lambda \in \mathscr{S}$
3: ***for*** $k = 1 \ldots k_{max}$ ***do***
4: $U := F_{i-1}(\lambda)$
5: $U := U/\|U\|_{\mathscr{V}}$ *(normalization)*
6: $\lambda = f_{i-1}(U)$
7: *Check convergence on* $\sigma_{i-1}(U)$ *(tolerance* ϵ_s*)*
8: ***end for***
9: $W_i := (W_{i-1}, U)$
10: $\Lambda_i := (\Lambda_{i-1}, \lambda)$
11: *Check convergence*
12: ***end for***

The main advantage of this algorithm is that it only requires the resolution of problems $\lambda = f(U)$ and $U = F(\lambda)$ which are respectively a simple nonlinear equation on λ and a nonlinear deterministic problem.

It is well known for classical eigenproblems that the power method does not necessarily converge or can exhibit a very slow convergence rate. This is the case when the dominant eigenvalue is of multiplicity greater than one or when dominant eigenvalues are very close. However, a convergence criterium based on eigenfunction U is not adapted to our problem. In fact, a pertinent evaluation of convergence should be based on the eigenvalue, which in our case corresponds to the contribution $\sigma_{i-1}(U)$ of a couple $(U, f_{i-1}(U))$ to the generalized spectral decomposition. In the case of multiplicity greater than one, a convergence of the eigenvalue indicates that the current iterate U should be a good candidate for maximizing the contribution to the generalized decomposition. When dominant eigenvalues are very close, a slow convergence rate can be observed on the eigenvalue when approaching the upper spectrum. However, close eigenvalues are associated to eigenfunctions which have similar contributions to the decomposition. Therefore, any of these eigenfunctions seems to be a rather good choice, the rest of the upper spectrum being explored by subsequent "deflations" of the operator. The above remarks indicate that a relatively coarse convergence criterium (tolerance ϵ_s) can be used for the power iterates:

$$|\sigma_{i-1}(U^{(k)}) - \sigma_{i-1}(U^{(k-1)})| \leqslant \epsilon_s \sigma_{i-1}(U^{(k)}) \tag{22}$$

This will be illustrated in numerical examples.

Remark 4 *A natural choice for the norm $\|U\lambda\|$ on $\mathscr{V} \otimes \mathscr{S}$ consists in taking a tensorization of norms defined on $\mathscr{V}$ and $\mathscr{S}$. The contribution of $Uf(U)$ can then be simply written $\|Uf(U)\| = \|U\|_{\mathscr{V}}\|f(U)\|_{\mathscr{S}}$. In algorithm 1, U being normalized, the evaluation of $\sigma_{i-1}(U)$ (step (7)) then only requires the evaluation of $\|\lambda\|_{\mathscr{S}}$.*

Remark 5 *For computational and analysis purposes, one may want to perform an orthonormalization of the decomposition. This orthonormalization can concern the deterministic basis W_M or the stochastic basis Λ_M. In both cases, it involves a non singular $M \times M$ matrix R such that the linear transformation writes $W_M \leftarrow W_M \cdot R$ (resp. $\Lambda_M \leftarrow \Lambda_M \cdot R$) for the orthonormalization of W_M (resp. Λ_M). To maintain the validity of the decomposition, the inverse transformation R^{-1} has also to be applied to the complementary basis, i.e. $\Lambda_M \leftarrow \Lambda_M \cdot R^{-1}$ (resp. $W_M \leftarrow W_M \cdot R^{-1}$).*

3.4.2 Improved power-type method: algorithm 2

A possible improvement of algorithm 1 consists in updating the reduced random basis Λ_M every time a new couple is computed, while keeping unchanged the deterministic basis W_M. We denote $\mathscr{V}_M = span\{U_i, i = 1 \dots M\} \subset \mathscr{V}$ the subspace spanned by W_M; on this subspace, Equation (2) becomes: find $u_M \in \mathscr{V}_M \otimes \mathscr{S}$ such that

$$B(u_M, v_M) = L(v_M) \quad \forall v_M \in \mathscr{V}_M \otimes \mathscr{S}. \tag{23}$$

This problem is equivalent to find $\Lambda_M \in (\mathscr{S})^M$ such that

$$B(W_M \cdot \Lambda_M, W_M \cdot \Lambda_M^*) = L(W_M \cdot \Lambda_M^*) \quad \forall \Lambda_M^* \in (\mathscr{S})^M. \tag{24}$$

We write $\Lambda_M = f_0(W_M)$ the solution to equation (24), which is a set of M coupled non-linear stochastic equations. The improved algorithm including stochastic basis updates is:

Algorithm 2 *Power-type algorithm with updating of the random basis*

1: ***for*** $M = 1 \dots M_{max}$ ***do***
2: *Do steps 2 to 10 of algorithm 1*
3: *Orthonormalize W_M (optional)*
4: *Update $\Lambda_M = f_0(W_M)$*
5: *Check convergence*
6: ***end for***

In the very particular case where $b(\cdot, \cdot)$ is bilinear and deterministic, it can be proved that the updating does not modify the decomposition [28]. This can be explained by the fact that dominant eigenfunctions of successive operators $\widetilde{T}_M$ are optimal regarding the initial problem, *i.e.* are dominant eigenfunctions of the initial operator $\widetilde{T} = \widetilde{T}_0$. In the general case, this property is not verified and makes that this updating can lead to a significant improvement of the accuracy of the decomposition. This will be illustrated in numerical examples.

Remark 6 *The orthonormalization step (3) of algorithm 2 is actually optional, as it does not affect the reduced spaces generated. Still, for numerical and analysis purposes, it is often preferred to work with orthonormal functions.*

3.5 Extension to affine spaces

In many situations, *e.g.* when dealing with non homogeneous boundary conditions, the solution u is to be sought in an affine space, with an associated vector space denoted $\mathscr{V} \otimes \mathscr{S}$. In order to apply the GSD method, the problem is classically reformulated in vector space $\mathscr{V} \otimes \mathscr{S}$ by introducing a particular

function u_0 of the affine space. The variational problem (2) becomes:
Find $u = u_0 + \widetilde{u}$, with $\widetilde{u} \in \mathscr{V} \otimes \mathscr{S}$, such that

$$B(u_0 + \widetilde{u}, v) = L(v) \quad \forall \mathscr{V} \otimes \mathscr{S}. \tag{25}$$

Then, now denoting $\widetilde{u}_M = W_M \cdot \Lambda_M$ and extending the definition of u_M to

$$u_M = u_0 + \widetilde{u}_M = u_0 + W_M \cdot \Lambda_M, \tag{26}$$

it is seen that the algorithms 1 and 2 apply immediately for the construction of the generalized spectral decomposition $\widetilde{u}_M$ of $\widetilde{u}$. This procedure is used in the next section, which details the application of the proposed iterative methods to the Burgers equation.

Remark 7 *The definition of a particular function u_0 is usual in the context of Galerkin approximation methods. For example, when dealing with non-homogeneous Dirichlet boundary conditions and when using finite element approximation at the spatial level, it simply consists in defining a finite element function with ad-hoc nodal values at the boundary nodes. The problem on $\widetilde{u} \in \mathscr{V} \otimes \mathscr{S}$ is then associated with homogeneous Dirichlet boundary conditions.*

4 Application to Burgers equation

4.1 Burgers equation

We consider the stochastic steady Burgers equation on the spatial domain $\Omega = (-1, 1)$, with random (but uniform) viscosity $\mu \in L^2(\Theta, dP)$. The stochastic solution,

$$u : (x, \theta) \in \Omega \times \Theta \mapsto u(x, \theta) \in \mathbb{R}, \tag{27}$$

satisfies almost surely

$$u \frac{\partial u}{\partial x} - \mu \frac{\partial^2 u}{\partial x^2} = 0, \quad \forall x \in \Omega. \tag{28}$$

This equation has to be complemented with boundary conditions. We assume deterministic boundary conditions:

$$u(-1, \theta) = 1, \quad u(1, \theta) = -1 \ (a.s.). \tag{29}$$

We further assume that $\mu(\theta) \geqslant \alpha > 0$ almost surely to ensure a physically meaningful problem. Thanks to the mathematical properties of the Burgers

equation (the solution is bounded by its boundary values), we have almost surely $u(x,\theta) \in [-1,1]$ and $u(x,\cdot) \in L^2(\Theta, dP)$ for all $x \in [-1,1]$.

4.2 Variational formulation

We introduce the following function space:

$$\mathscr{U} = \{v \in H^1(\Omega); v(-1) = 1, v(1) = -1\}. \tag{30}$$

The space $\mathscr{U}$ is affine, and we denote $\mathscr{V}$ the corresponding vector space:

$$\mathscr{V} = \{v \in H^1(\Omega); v(-1) = 0, v(1) = 0\}. \tag{31}$$

The stochastic solution $u(x,\theta)$ is sought in the tensor product function space $\mathscr{U} \otimes \mathscr{S}$. It is solution of the variational problem (25) with

$$b(u, v; \theta) = \int_\Omega \left(\mu(\theta) \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + u \frac{\partial u}{\partial x} v \right) dx, \tag{32}$$

$$l(v; \theta) = 0. \tag{33}$$

Remark 8 *The previous variational formulation implicitly assumes that $\mathscr{S} \subset L^2(\Theta, dP)$ is finite dimensional.*

To detail the methodology, we write

$$b(u, v; \theta) = \mu(\theta) a(u, v) + n(u, u, v), \tag{34}$$

where a and n are bilinear and trilinear forms respectively, defined as:

$$a(u, v) = \int_\Omega \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} \, dx, \tag{35}$$

$$n(u, v, w) = \int_\Omega u \frac{\partial v}{\partial x} w \, dx. \tag{36}$$

Remark 9 *It is seen that the forms a and n have no explicit dependence with regards to the elementary event θ. Generalization of the methodology to situations where forms depend on the event is however immediate.*

The boundary conditions being deterministic, an obvious choice for $u_0 \in \mathscr{U}$ is $u_0(x,\theta) = -x$. Then, to simplify the notations, we define $\lambda_0 = 1$ and $U_0 = u_0$ such that the approximate solution u_M writes:

$$u_M = u_0 + \sum_{i=1}^{M} \lambda_i U_i = \sum_{i=0}^{M} \lambda_i U_i \tag{37}$$

4.3 Application of GSD algorithm to the Burgers equation

Algorithms 1 and 2 can now be applied to perform the generalized spectral decomposition of the solution. We now detail the main ingredients of the algorithms, namely steps (4) and (6) of algorithm 1, and the update step of algorithm 2.

4.3.1 Resolution of $U = F_M(\lambda)$

To compute $U = F_M(\lambda)$, one has to solve for U the equation (17). This is equivalent to solve for U the following deterministic problem (remember that λ is given):

$$B_M(\lambda U, \lambda V) = L_M(\lambda V) \quad \forall V \in \mathscr{V}. \tag{38}$$

where $\forall u, v \in \mathscr{V} \otimes \mathscr{S}$,

$$B_M(u,v) \equiv B(u_M + u, v) - B(u_M, v), \tag{39}$$
$$L_M(v) \equiv L(v) - B(u_M, v). \tag{40}$$

Substracting $B(u_M, v)$ on both sides of (17) to yield (38) ensures that the right-hand side L_M vanishes whenever u_M solves the weak form of the stochastic Burgers equation. This manipulation is however purely formal. With some elementary manipulations, it is easy to show that

$$\begin{aligned} B_M(\lambda U, \lambda V) =& E(\lambda\lambda\mu) a(U,V) + E(\lambda\lambda\lambda) n(U,U,V) \\ &+ \sum_{i=0}^{M} E(\lambda_i \lambda\lambda) \left[n(U_i, U, V) + n(U, U_i, V) \right], \end{aligned} \tag{41}$$

$$L_M(\lambda V) = -\sum_{i=0}^{M} E(\mu\lambda_i\lambda) a(U_i, V) - \sum_{i,j=0}^{M} E(\lambda\lambda_i\lambda_j) n(U_i, U_j, V). \tag{42}$$

Therefore, one can recast the equation on U in the formal way:

$$\begin{aligned} \widetilde{\mu} a(U,V) + n(U,U,V) + n(\widetilde{U}, U, V) + n(U, \widetilde{U}, V) = \\ -a(\breve{U}, V) - n(1, \widehat{Z}, V), \quad \forall V \in \mathscr{V}, \end{aligned} \tag{43}$$

where

$$\widetilde{\mu} = \frac{E(\lambda\lambda\mu)}{E(\lambda^3)}, \quad \widetilde{U} = \sum_{i=0}^{M} \frac{E(\lambda_i\lambda\lambda)}{E(\lambda\lambda\lambda)} U_i, \tag{44}$$

$$\breve{U} = \sum_{i=0}^{M} \frac{E(\mu\lambda_i\lambda)}{E(\lambda\lambda\lambda)} U_i, \quad \widehat{Z} = \frac{1}{2} \sum_{i,j=0}^{M} \frac{E(\lambda_i\lambda_j\lambda)}{E(\lambda\lambda\lambda)} U_i U_j, \tag{45}$$

Equation (43) shows that U is the solution of a non linear deterministic problem, with homogeneous boundary conditions, involving a quadratic non linearity term ($n(U,U,V)$) which reflects the non linearity of the original Burgers equation. In fact, the resulting problem for U has the same structure as the weak form of the deterministic Burgers equations, with some additional (linear) terms expressing the coupling of U with u_M (due to the non linearity) and a right-hand side accounting for the equation residual for $u = u_M$. As a result, a standard non linear solver can be used to solve this equation, *e.g.* one can re-use a deterministic steady Burgers solver with minor adaptations.

Remark 10 *At first thought, equation* (43) *suggests that a robust non linear solver is needed for its resolution, since a priori the effective viscosity* $\tilde{\mu}$ *may become negative and experience changes by orders of magnitudes in the course of the iterative process. However, one can always make use of the homogeneity property*

$$\frac{U}{\alpha} = F_M(\alpha\lambda), \quad \forall \alpha \in \mathbb{R}^*, \tag{46}$$

to rescale the problem and fit solver requirements if any. Note that equation (46) *together with equation* (43) *also indicate that the nature of the non-linear deterministic problems to be solved is preserved along the course of the iterations. For instance, the effective viscosity goes to zero as* $|\lambda| \to \infty$ *but the problem does not degenerate to an hyperbolic one since the right-hand-side also goes to zero and* U *satisfies homogeneous boundary conditions.*

4.3.2 Resolution of $\lambda = f_M(U)$

The random variable $\lambda \in \mathscr{S}$ is solution of the variational problem:

$$B_M(\lambda U, \beta U) = L_M(\beta U) \quad \forall \beta \in \mathscr{S}. \tag{47}$$

After some manipulations, this equation is found to be equivalent to:

$$\begin{aligned} E(\beta\lambda\lambda)n(U,U,U) + E(\beta\mu\lambda)a(U,U) + \sum_{i=0}^{M} E(\beta\lambda_i\lambda)\left[n(U,U_i,U) + n(U_i,U,U)\right] \\ = -\sum_{i=0}^{M} E(\beta\mu\lambda_i)a(U_i,U) - \sum_{i,j=0}^{M} E(\beta\lambda_i\lambda_j)n(U_i,U_j,U). \end{aligned} \tag{48}$$

This is a simple stochastic quadratic equation on λ: a standard non linear solver can be used for its resolution.

4.3.3 Resolution of $\Lambda_M = f_0(W_M)$

To update $\Lambda_M = (\lambda_1, \ldots, \lambda_M) \in (\mathscr{S})^M$, one has to solve:

$$B(u_0 + W_M \cdot \Lambda_M, W_M \cdot \Lambda_M^*) = L(W_M \cdot \Lambda_M^*) \quad \forall \Lambda_M^* \in (\mathscr{S})^M. \tag{49}$$

This equation can be split into a system of M equations:

$$\forall k \in \{1, \ldots, M\}, \quad B(u_0 + W_M \cdot \Lambda_M, U_k \beta_k) = L(U_k \beta_k) \quad \forall \beta_k \in \mathscr{S}. \tag{50}$$

Introducing the previously defined forms, it comes:

$$\sum_{i=0}^{M} \mu(\theta)\lambda_i(\theta) a(U_i, U_k) + \sum_{i,j=0}^{M} \lambda_i \lambda_j n(U_i, U_j, U_k) = 0, \quad \forall k \in \{1, \ldots, M\}. \tag{51}$$

Again, it is seen that the updating step consists in solving a system of quadratic non linear equations for the $\{\lambda_i\}_{i=1}^M$. A standard non linear solver can be used for this purpose.

4.4 Spatial discretization

Let us denote $\mathbb{P}_{\mathrm{N_x}+1}(\Omega)$ the space of polynomials of degree less or equal to $\mathrm{N_x}+1$ on Ω. We define the approximation vector space $\mathscr{V}^h$ as:

$$\mathscr{V}^h = \{v \in \mathbb{P}_{\mathrm{N_x}+1}(\Omega); v(-1) = 0, v(1) = 0\} \subset \mathscr{V}. \tag{52}$$

Let $x_{i=\{0,\ldots,\mathrm{N_x}+1\}}$ be the $\mathrm{N_x}+2$ Gauss-Lobatto points [1] of the interval $[-1, 1]$, such that

$$x_0 = -1 < x_1 < \ldots < x_{\mathrm{N_x}} < x_{\mathrm{N_x}+1} = 1. \tag{53}$$

We denote $L_{i\in\{1,\ldots,\mathrm{N_x}\}}(x) \in \mathbb{P}_{\mathrm{N_x}+1}$, the Lagrange polynomials constructed on the Gauss-Lobatto grid:

$$L_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{\mathrm{N_x}+1} \frac{x - x_j}{x_i - x_j}. \tag{54}$$

These polynomials satisfy

$$L_i(x_j) = \begin{cases} 0 \text{ if } i \neq j \\ 1 \text{ if } i = j \end{cases} \quad \forall j = 0, \ldots, \mathrm{N_x}+1, \tag{55}$$

and form a basis of $\mathscr{V}^h$:

$$\mathscr{V}^h = span\{L_i, i = 1, \ldots, \mathrm{N_x}\}. \tag{56}$$

For any $v \in \mathscr{V}^h$, we have

$$v(x) = \sum_{i=1}^{\mathrm{N_x}} v^i L_i(x), \quad v^i = v(x_i). \tag{57}$$

The derivative of $v \in \mathscr{V}^h$ has for expression:

$$\frac{\partial v}{\partial x} = \sum_{i=1}^{\mathrm{N_x}} v^i L_i'(x), \quad L_i' \equiv \frac{\partial L_i}{\partial x}. \tag{58}$$

The bilinear and trilinear forms a and n are evaluated using the quadrature formula over the Gauss-Lobatto points [6]. Specifically, for $u, v \in \mathscr{V}^h$, we have

$$\begin{aligned} a(u,v) &= \int_\Omega \frac{\partial u}{\partial x}\frac{\partial v}{\partial x} dx = \int_\Omega \left(\sum_{i=1}^{\mathrm{N_x}} u^i L_i'\right)\left(\sum_{i=1}^{\mathrm{N_x}} v^i L_i'\right) dx \\ &= \sum_{i,j=1}^{\mathrm{N_x}} u^i v^j \int_\Omega L_i'(x) L_j'(x) dx = \sum_{i,j=1}^{\mathrm{N_x}} u^i v^j a_{i,j}, \end{aligned} \tag{59}$$

where

$$a_{i,j} \equiv \left(\sum_{k=0}^{\mathrm{N_x}+1} L_i'(x_k) L_j'(x_k)\omega_k\right), \tag{60}$$

with $\omega_{k\in\{0,\ldots,\mathrm{N_x}+1\}}$ the Gauss-Lobatto quadrature weights [1]. Similarly, for $u, v, w \in \mathscr{V}^h$, we have

$$\begin{aligned} n(u,v,w) &= \int_\Omega u\frac{\partial v}{\partial x} w\, dx \approx \sum_{k=0}^{\mathrm{N_x}+1}\left(u(x_k)\sum_{i=0}^{\mathrm{N_x}+1} v^i L_i'(x_k) w(x_k)\right)\omega_k \\ &\approx \sum_{k=1}^{\mathrm{N_x}}\sum_{i=0}^{\mathrm{N_x}+1} n_{i,k} u^k v^i w^k, \end{aligned} \tag{61}$$

where $n_{i,k} \equiv L_i'(x_k)\omega_k$. The same expression holds for $u_0 \notin \mathscr{V}^h$.

4.5 Stochastic discretization

In the results presented hereafter, the random viscosity μ is parametrized using a set of N independent real continuous second order random variables, $\boldsymbol{\xi} = \{\xi_1, \ldots, \xi_\mathrm{N}\}$,

$$\mu(\theta) = \mu(\boldsymbol{\xi}(\theta)). \tag{62}$$

We denote Ξ the range of $\boldsymbol{\xi}$ and $P_{\boldsymbol{\xi}}$ the known probability law of $\boldsymbol{\xi}$. Since random variables ξ_i are independent, we have for $\boldsymbol{y} = (y_1, \ldots, y_{\mathrm{N}}) \in \mathbb{R}^{\mathrm{N}}$

$$dP_{\boldsymbol{\xi}}(\boldsymbol{y}) = \prod_{i=1}^{\mathrm{N}} p_{\xi_i}(y_i) dy_i, \tag{63}$$

Let $(\Xi, \mathcal{B}_\Xi, P_{\boldsymbol{\xi}})$ be the associated probability space. The stochastic solution is then sought in the image probability space $(\Xi, \mathcal{B}_\Xi, P_\xi)$ instead of $(\Theta, \mathcal{B}, P)$, *i.e.* we compute $u(\boldsymbol{\xi})$. Furthermore, the expectation operator has the following expression in the image probability space:

$$E(f(\cdot)) = \int_\Theta f(\boldsymbol{\xi}(\theta))\, dP(\theta) = \int_\Xi f(\boldsymbol{y}) dP_{\boldsymbol{\xi}}(\boldsymbol{y}). \tag{64}$$

It is clear from this relation that if $f \in L^2(\Theta, dP)$ then $f \in L^2(\Xi, dP_{\boldsymbol{\xi}})$, the space of second order random variables spanned by $\boldsymbol{\xi}$. To proceed with the determination of the numerical solution, one has to construct a finite dimensional approximation space $\mathscr{S} \subset L^2(\Xi, dP_\Xi)$. Different discretizations are available at the stochastic level (continuous polynomial expansion, piecewise polynomial expansions, multiwavelets, ...). At this point, it is stressed that the proposed GSD algorithms are independent of the type of stochastic discretization used. In the following, we rely on classical Generalized Polynomial Chaos expansions, which consist in defining the stochastic space as

$$\mathscr{S} = span\{\Psi_0, \ldots, \Psi_{\mathrm{P}}\}, \tag{65}$$

where the Ψ_i are mutually orthogonal random polynomials in $\boldsymbol{\xi}$, with total degree less or equal to $\mathrm{N_o}$. The orthogonality of the random polynomials writes

$$E(\Psi_i \Psi_j) = E(\Psi_i^2)\delta_{ij}. \tag{66}$$

The dimension of the stochastic subspace is therefore given by

$$\dim(\mathscr{S}) = \mathrm{P} + 1 = \frac{(\mathrm{N} + \mathrm{N_o})!}{\mathrm{N}!\mathrm{N_o}!}, \tag{67}$$

and a random variable $\beta \in \mathscr{S}$ has for expansion

$$\beta(\boldsymbol{\xi}) = \sum_{i=0}^{\mathrm{P}} \beta^i \Psi_i(\boldsymbol{\xi}). \tag{68}$$

Specifically, the $\lambda_i \in \mathscr{S}$ of the GSD of the solution will have expansions of the form:

$$\lambda_i = \sum_{k=0}^{\mathrm{P}} \lambda_i^k \Psi_k(\boldsymbol{\xi}).$$

4.6 Solvers

4.6.1 $U = F_M(\lambda)$

With the spatial discretization introduced previously, one has to solve for $U \in \mathscr{V}^h$ the following set of $\mathrm{N_x}$ non linear equations (corresponding to (43)):

$$G_k(U^1, \ldots, U^{\mathrm{N_x}}; \lambda) = 0, \quad k = 1, \ldots, \mathrm{N_x}, \tag{69}$$

where

$$\begin{aligned} G_k(U^1, \ldots, U^{\mathrm{N_x}}; \lambda) = \tilde{\mu} \sum_{i=1}^{\mathrm{N_x}} a_{i,k} U^i + \sum_{i=1}^{\mathrm{N_x}} n_{i,k} \left(U^k U^i + \widetilde{U}^k U^i + U^k \widetilde{U}^i \right) \\ + \sum_{i=1}^{\mathrm{N_x}} a_{i,k} \breve{U}^i + \sum_{i=1}^{\mathrm{N_x}} n_{i,k} \widehat{Z}^i, \end{aligned} \tag{70}$$

with

$$\widetilde{\mu} = \frac{E(\lambda\lambda\mu)}{E(\lambda\lambda\lambda)}, \quad \widetilde{U}^k = \sum_{i=0}^{M} \frac{E(\lambda_i\lambda\lambda)}{E(\lambda\lambda\lambda)} U_i^k, \tag{71}$$

$$\breve{U}^k = \sum_{i=0}^{M} \frac{E(\mu\lambda_i\lambda)}{E(\lambda\lambda\lambda)} U_i^k, \quad \widehat{Z}^k = \frac{1}{2} \sum_{i,j=0}^{M} \frac{E(\lambda_i\lambda_j\lambda)}{E(\lambda\lambda\lambda)} U_i^k U_j^k, \tag{72}$$

and the coefficients $a_{i,k}$ and $n_{i,k}$ defined in paragraph 4.4. Also, since the stochastic expansion coefficients of λ and the λ_i are given, the expectations are classically evaluated analytically. For instance,

$$E\left(\lambda_i\lambda_j\lambda\right) = \sum_{l=0}^{\mathrm{P}} \sum_{m=0}^{\mathrm{P}} \sum_{n=0}^{\mathrm{P}} T_{lmn} \lambda_i^l \lambda_j^m \lambda^n, \quad T_{lmn} = E\left(\Psi_l \Psi_m \Psi_n\right).$$

In this work, we have used a classical Newton method to solve (69).

4.6.2 $\lambda = f_M(U)$

Introducing the stochastic expansions of μ and of the λ_i, the expansion coefficients of λ satisfy the following set of $\mathrm{P}+1$ non linear equations:

$$g_k(\lambda^0, \ldots, \lambda^{\mathrm{P}}; U) = \sum_{i,j=0}^{\mathrm{P}} c_{ijk} \lambda^i \lambda^j + \sum_{i=0}^{\mathrm{P}} d_{ik} \lambda^i + e_k = 0, \quad k = 0, \ldots, \mathrm{P}, \tag{73}$$

where

$$
\begin{aligned}
c_{ijk} &= E(\Psi_i\Psi_j\Psi_k)n(U,U,U),\\
d_{ik} &= \sum_{j=0}^{\mathrm{P}} E(\Psi_i\Psi_j\Psi_k)\left[\mu^j a(U,U) + \sum_{l=0}^{M}\lambda_l^j(n(U,U_l,U)+n(U_l,U,U))\right],\\
e_k &= \sum_{i,j=0}^{\mathrm{P}} E(\Psi_i\Psi_j\Psi_k)\left[\mu^i\sum_{l=0}^{M}\lambda_l^j a(U_l,U) + \sum_{l,m=0}^{M}\lambda_l^i\lambda_m^j n(U_l,U_m,U)\right].
\end{aligned}
$$

This set of equations can be solved using efficient standard techniques involving exact Jacobian computation. In this work, we have used the minpack subroutines [24] to solve (73).

4.6.3 $\Lambda_M = f_0(W_M)$

The stochastic expansion of Λ_M is

$$\Lambda_M = \sum_{i=0}^{\mathrm{P}} \Lambda_M^i \Psi_i. \tag{74}$$

Introducing this expansion in (51), one obtains a set of $M\times(\mathrm{P}+1)$ non-linear equations, which are:

$$
\begin{aligned}
g_{k,q}(\Lambda_M^0,\ldots,\Lambda_M^{\mathrm{P}};W_M) = \sum_{l,m=0}^{\mathrm{P}} E(\Psi_l\Psi_m\Psi_q)\Bigg[&\sum_{i=0}^{M}\mu^l\lambda_i^m a(U_i,U_k)\\
&+\sum_{i,j=0}^{M}\lambda_i^l\lambda_j^m n(U_i,U_j,U_k)\Bigg] = 0,\\
&k=1,\ldots,\mathrm{M},\quad q=0,\ldots,\mathrm{P}.
\end{aligned}
\tag{75}
$$

Again, we rely on the minpack library to solve this set of non linear equations.

Remark 11 *It is seen that on the contrary of the determination of U and λ, the size of the non linear system of equations for the updating of Λ_M increases with M.*

5 Results

5.1 Error estimation

For the purpose of convergence analysis, we define the stochastic residual of the equation as

$$\mathcal{R}_M(x,\theta) = u_M \frac{\partial u_M}{\partial x} - \mu \frac{\partial^2 u_M}{\partial x^2} \tag{76}$$

and the corresponding L^2-norm

$$\|\mathcal{R}_M\|^2 = \int_\Omega \|\mathcal{R}_M(x,\cdot)\|^2_{L^2(\Xi,dP_\xi)}\, dx = \int_\Omega E(\mathcal{R}_M(x,\cdot)^2)\, dx. \tag{77}$$

It is observed that this norm measures the errors due to both stochastic and spatial discretizations. As a results, when $(M, \dim(\mathscr{S})) \to \infty$, this error is not expected to go to zero but to level off to a finite value corresponding to the spatial discretization error. However, thanks to the spectral finite element approximation in space, the errors in the following numerical tests are dominated by the stochastic error due to $\dim(\mathscr{S}) < \infty$. In fact, in this work, we are more interested by the analysis of the convergence with M of u_M toward the discrete exact solution on $\mathscr{V}^h \otimes \mathscr{S}$, and the comparison of the convergence rates of the two algorithms, than in the absolute error. For this purpose, we define the stochastic residual $R_M(x,\theta)$ as the orthogonal projection of $\mathcal{R}_M(x,\theta)$ on $\mathscr{S}$:

$$\mathcal{R}_M(x,\theta) = R_M(x,\theta) + R_M^\perp(x,\theta), \tag{78}$$

such that

$$R_M(x,\cdot) \in \mathscr{S}, \quad E\left(R_M^\perp(x,\cdot)\beta\right) = 0,\ \forall \beta \in \mathscr{S}. \tag{79}$$

In other words, $R_M(x,\cdot)$ is the classical Galerkin residual on $\mathscr{S}$,

$$R_M(x,\theta) = \sum_{k=0}^{\mathrm{P}} R_M^k(x)\Psi_k(\theta),$$

where

$$\begin{aligned} E\left(\Psi_k\Psi_k\right) R_M^k(x) &= E\left(\mathcal{R}_M(x,\cdot)\Psi_k(\cdot)\right) \\ &= \sum_{i,j=0}^{M} E(\lambda_i\lambda_j\Psi_k) U_i \frac{\partial U_j}{\partial x} - \sum_{i=0}^{M} E\left(\mu\lambda_i\Psi_k\right) \frac{\partial^2 U_i}{\partial x^2}. \end{aligned}$$

Its L^2-norm is

$$\|R_M\|^2 = \int_\Omega \left[\sum_{k=0}^{\mathrm{P}} \left(R_M^k(x) \right)^2 E(\Psi_k \Psi_k) \right] dx. \tag{80}$$

It is seen that $\|R_M\|$, though containing a contribution of the spatial discretization error deemed negligible, essentially measures the reduced basis approximation error (*i.e.* by substituting the "exact" discrete solution $u^h \in \mathscr{V}^h \otimes \mathscr{S}$ by $u_M = W_M \cdot \Lambda_M$ in the equations). Consequently, we shall refer to $\mathcal{R}_M$ as the equation residual and to R_M as the reduction residual.

5.2 Convergence analysis

To analyze the convergence of the GSD algorithms, we consider the following random viscosity setting:

$$\mu(\boldsymbol{\xi}) = \mu^0 + \sum_{i=1}^{\mathrm{N}} \mu' \xi_i, \tag{81}$$

with all ξ_i being uniformly distributed on $(-1, 1)$, leading to $\Xi = (-1, 1)^{\mathrm{N}}$. To ensure the positivity of the viscosity, we must have $\mu^0 > \mathrm{N}|\mu'|$. We set $\mu' = c\mu^0/\mathrm{N}$, with $|c| < 1$. For this parametrization, the variance of the viscosity is

$$E((\mu - \mu^0)^2) = \frac{\mathrm{N}}{3}(\mu')^2 = \frac{c^2}{3\mathrm{N}}(\mu^0)^2. \tag{82}$$

It is remarked that for this parametrization, the density of μ depends on N and experience less and less variability as N increases. For the discretization of the stochastic space $\mathscr{S}$, we use multidimensional Legendre polynomials. The mean viscosity is set to $\mu^0 = 0.2$ and $c = 0.85$.

In a first series of tests, we set $\mathrm{N} = 4$ and $\mathrm{N_o} = 6$, so $\dim(\mathscr{S}) = 210$, while for the spatial discretization $dim(\mathscr{V}^h) = \mathrm{N_x} = 200$ is used. This spatial discretization allows for accurate deterministic solutions for any realization $\mu(\boldsymbol{\xi})$, $\boldsymbol{\xi} \in \Xi$. If the stochastic solution was to be found in the full approximation space $\mathscr{V}^h \otimes \mathscr{S}$, the size of the non-linear problem to be solved would be $dim(\mathscr{V}^h) \times \dim(\mathscr{S}) = 42,000$. In contrast, the reduced basis solution $W_M \cdot \Lambda_M$ has for dimension $M \times (dim(\mathscr{V}^h) + \dim(\mathscr{S})) = 410M$.

In Figure 1, we compare the convergence of algorithms 1 and 2, as measured by the two residual norms $\|\mathcal{R}_M\|$ and $\|R_M\|$, with the size M of the reduced basis (left plot) and with the total number of iterations performed on $U = F_M(\lambda)$ and $\lambda = f_M(U)$ (right plot). The stopping criteria is here $\epsilon_s = 10^{-3}$.
Focusing first on the reduction residual R_M in the left plot, we can conclude

that both algorithms converge to the discrete solution on $\mathscr{V}^h \otimes \mathscr{S}$ with exponential rate as the dimension M of the reduced basis increases. However, the algorithm 2 is more effective in reducing R_M, compared to algorithm 1. Specifically, the exponential convergence rates for $\|R_M\|$ are ~ 1.2 and ~ 0.3 for algorithms 2 and 1 respectively. Also, the norms $\|\mathcal{R}_M\|$ of the equation residual is seen to decrease with the same rate as $\|R_M\|$, though thanks to the higher convergence rate of algorithm 2 it quickly saturate to a finite value (the discretization error) within just 5 iterations. For algorithm 1, the norm of $\mathcal{R}_M$ has not yet reach it asymptotic value for $M = 10$, reflecting the slowest convergence of the solution in $\mathscr{V}^h \otimes \mathscr{S}$.

Moreover, inspection of the right plot of Figure 1 shows that algorithm 2 requires less iterations on problems $U = F_M(\lambda)$ and $\lambda = f_M(U)$ to yield the next term of the decomposition. Specifically, algorithm 2 needs 3 to 4 iterations to meet the stopping criteria, while algorithm 1 needs a variable number of iterations between 3 to 8. This difference is essentially explained by the updating of Λ_M. Indeed, when the orthonormalization of W_M in algorithm 2 is disregarded, the convergence of the resulting decomposition and number of iterations to yield the couples (U, λ) is unchanged (not shown). This confirm the claim made previously that the orthonormalization of W_M is optional. The lower number of iterations needed to yield the couples and faster convergence of the residuals for algorithm 2 does not imply a lower computational cost, since the resolution of $U = F_M(\lambda)$ is inexpensive for the 1-D Burgers equation. In fact, algorithm 2 requires a significantly larger computational time for this problem, as most of the CPU-time is spent solving the stochastic update problem $\Lambda_M = f_M(W_M)$. This conclusion will not hold in general for larger problems (*e.g.* for Navier-Stokes flows) when the resolution of the deterministic problems will dominate the overall CPU-time. Also, computational times are not the only concern and one may prefer to spent more time computing the reduced modes, to achieve a better reduced basis approximation in order to lower memory requirements, especially for problems involving large spatial approximation spaces.

To understand the higher efficiency of algorithm 2, we compare in Figure 2 the 8 first reduced modes $U_i(x)$ computed using the two algorithms. Only half of the domain is shown as the reduced modes are odd functions of x, because of the symmetry of the problem. The comparison clearly shows that algorithm 2 yields a deterministic reduced basis $W_{M=8}$ with a higher frequency content than for this of algorithm 1. This is explained by the improvement of the approximation brought by the updating of Λ_M. In fact, because the updating procedure cancels the equation residual in the subspace $span\{W_M\} \otimes \mathscr{S}$, the following deterministic mode U constructed will be essentially orthogonal to W_M. On the contrary, algorithm 1 only approximatively solve the equations in the subspace $span\{W_M\} \otimes \mathscr{S}$ (*i.e.* $\Lambda_M \neq f_0(W_M)$), with a delayed exploration of the deterministic space $\mathscr{V}^h$ as a result. This point is further

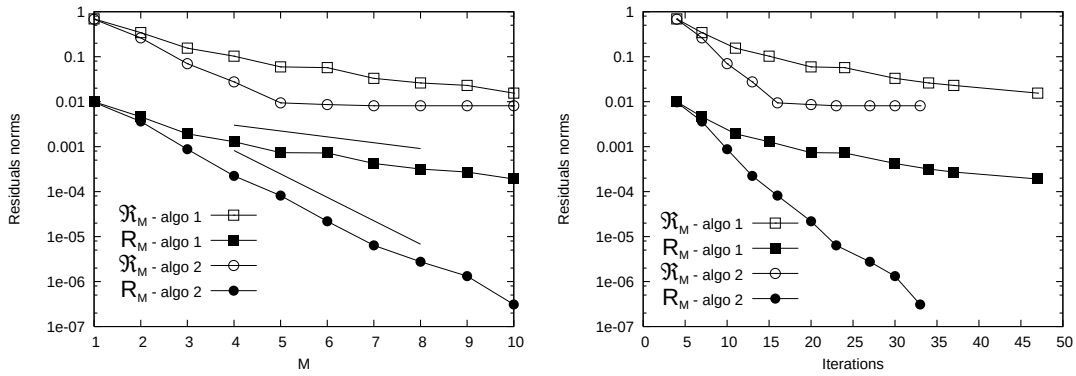

Figure 1. Convergence of the reduction residual R_M (close symbols) and equation residuals $\mathcal{R}_M$ (open symbols) for algorithms 1 (squares) and 2 (circles). The left plot displays the residual norms as a function of the reduced basis dimension M, while the right plot displays the residual norms as a function of the total (cumulated) number of power-type iterations for the computation of successive couples (U, λ). In the left plot, also reported using solid lines are fits of $\|R_M\|$ with $\sim \exp(-1.2M)$ and $\sim \exp(-0.3M)$.

illustrated in Figure 3, where plotted are the second moment of the equation residual, $E(\mathcal{R}_M(x,\cdot)^2)$, for different M and the two algorithms. The plot of $E(\mathcal{R}_M(x,\cdot)^2)$ for algorithm 2 highlights the efficiency of the GSD in capturing the full discrete solution on $\mathscr{V}^h \otimes \mathscr{S}$ in just few modes and indicates that the stochastic discretization mostly affect the equation residual in the area where the solution exhibits the steepest gradients, *i.e.* where the uncertainty has the most impact on the solution.

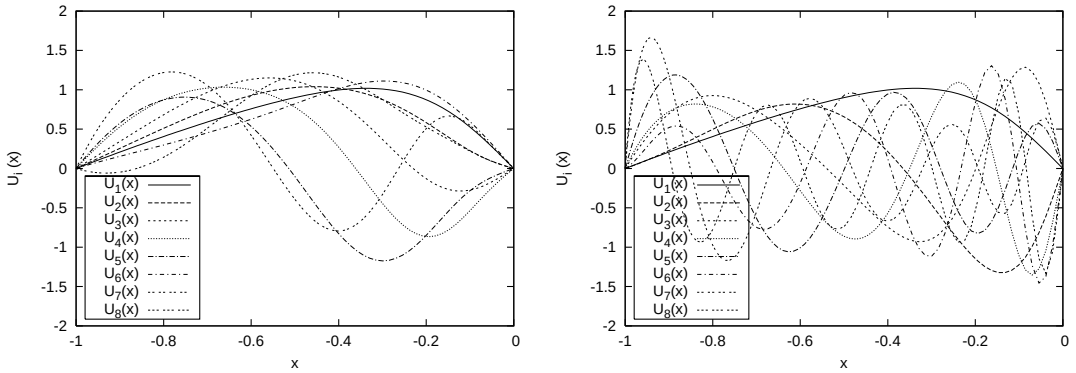

Figure 2. Comparison of the 8 first reduced modes U_i obtained with algorithms 1 (left plot) and 2 (without orthonormalization of W_M).

It is also remarked that even though the equation residual norm provides a measure of how well the reduced basis approximation satisfies the Burgers equation, it is not a direct measure of the error on the solution. Specifically, the somehow large magnitude of $\|\mathcal{R}_M\|$ does not imply that the error ϵ_M on the solution is as high. The L^2-error of the stochastic solution can in turn be

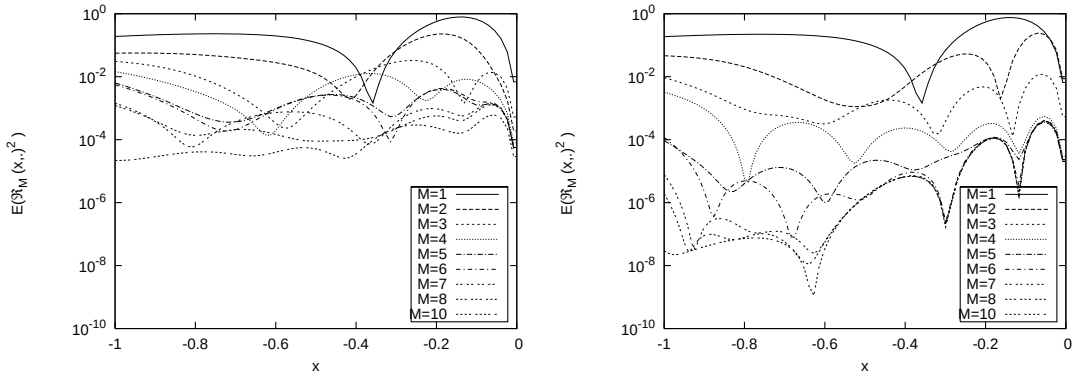

Figure 3. Evolution of the second moment of the equation residual, $E(\mathcal{R}_M(x,\cdot)^2)$, for different M and algorithms 1 (left plot) and 2 (right plot).

measured using the following norm:

$$\|\epsilon_{\mathrm{M}}\|^2 = \int_\Omega \|u_M(x,\cdot) - u(x,\cdot)\|^2_{L^2(\Xi,dP_{\boldsymbol{\xi}})} dx, \tag{83}$$

where u_M is the GSD solution and u the exact stochastic solution. The exact solution being unknown, one has to rely on approximate expression for $\|\epsilon_M\|$. Here, using the fact that the stochastic error dominates the spatial error, we use a vanilla Monte-Carlo (MC) method to estimate the solution error. We denote $u^d(x;\boldsymbol{\xi}) \in \mathscr{V}^h$ the deterministic solution of the Burgers equation for the viscosity realization $\mu(\boldsymbol{\xi})$. We then rely on a uniform random sampling of Ξ, with m sampling points, to construct the stochastic estimator of the local mean square error:

$$\|u_M(x,\cdot) - u(x,\cdot)\|^2_{L^2(\Xi,dP_{\boldsymbol{\xi}})} \approx \frac{1}{m}\sum_{i=1}^{m}\Big(u_M(x,\boldsymbol{\xi}(\theta_i)) - u^d(x;\boldsymbol{\xi}(\theta_i))\Big)^2. \tag{84}$$

Using a sample set with dimension $m = 10,000$ we obtained for the solution computed with algorithm 2 the estimate $\|\epsilon_{M=10}\| = (1.55 \pm 0.1)\,10^{-4}$, showing that the reduced solution u_M is indeed much more accurate than suggested by the norm of the equation residual. As for the equation residual, we provide in Figure 4 the spatial distribution for the mean square error on the solution, for the MC estimate given in equation (84) using $m = 10,000$ MC samples.

For a better appreciation of the convergence of the solution on the reduced basis, we have plotted in Figures 5 and 6 the evolutions of the computed solution mean and standard deviation ($E(u_M)$ and $Std-dev(u_M)$) for different M and for the two algorithms. Again, only half of the domain is shown, the mean (resp. standard deviation) being an odd (resp. even) function of x. Figures 5 shows a fast convergence of the mean for the two algorithms: curves are essentially indistinguishable for $M \geq 3$. Analysis of the standard deviation plots in Figure 6 also reveal a fast convergence, although the faster convergence of algorithm 2 compared to algorithm 1 appears more clearly than for the mean.

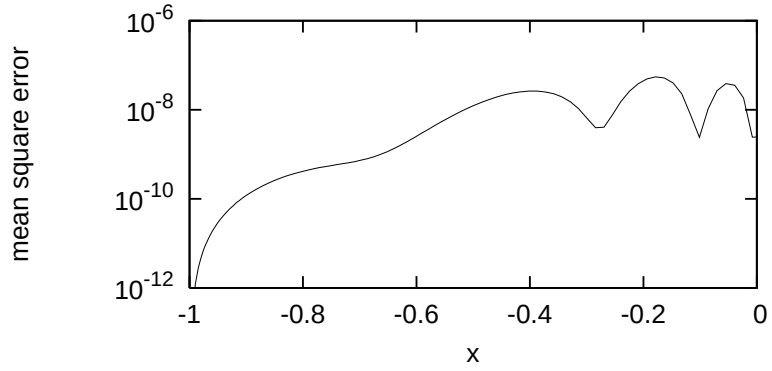

Figure 4. MC estimate of the local mean square error on the solution, $E((u_M - u)^2)$ for $M = 10$ and algorithm 2.

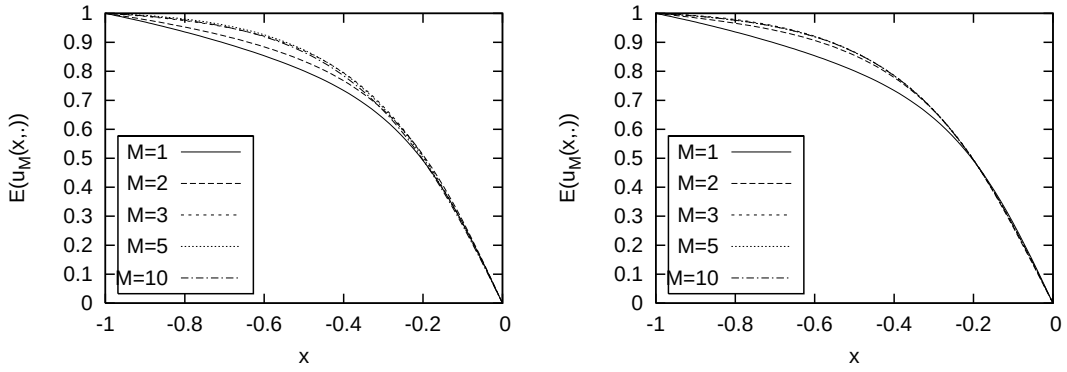

Figure 5. Convergence of the solution mean with the size M of the reduced basis, as indicated, and algorithms 1 (left plot) and 2 (right plot).

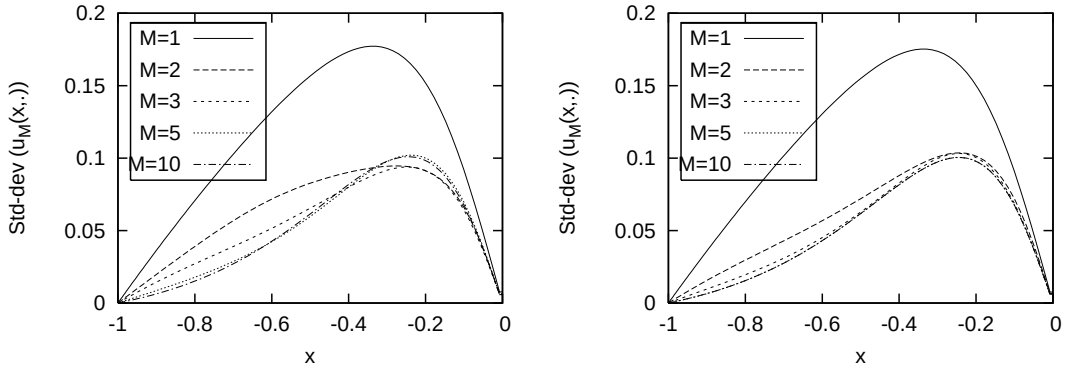

Figure 6. Convergence of the solution standard deviation with the size M of the reduced basis, as indicated, and algorithms 1 (left plot) and 2 (right plot).

5.3 Robustness of the algorithms

We now investigate the robustness of the method with regards to stochastic discretization and numerical parameters.

5.3.1 Impact of ϵ_s

The two algorithms require a criteria ϵ_s to stop the iterations associated with the construction of a new couple (U, λ) (see Section 3.4.1). Non convergence has not been encountered in our computation. Still, in order to avoid performing unnecessary iterations, the selection of an appropriate value for ϵ_s in an important issue as slow convergence was reported in some computations. It also raises questions regarding the accuracy on the computed couples (U, λ) needed to construct an appropriate reduced basis (see discussion in section 3.4.1). This aspect is numerically investigated by considering less and less stringent stopping criteria ϵ_s and monitoring the convergence of $\|R_M\|$. These experiments are reported in Figure 7, for the previous viscosity settings, discretization parameters and for $\epsilon_s = 10^{-2,-3,-4,-6}$. It is seen that for both algorithms, the selection of ϵ_s on the range tested has virtually no effect on the convergence of the decomposition, but to be computationally more demanding as ϵ_s decreases. Similar experiences for other viscosity settings (see below) have demonstrated that one usually has no interest in performing more than 3 to 4 iterations on the computation of couple (U, λ).

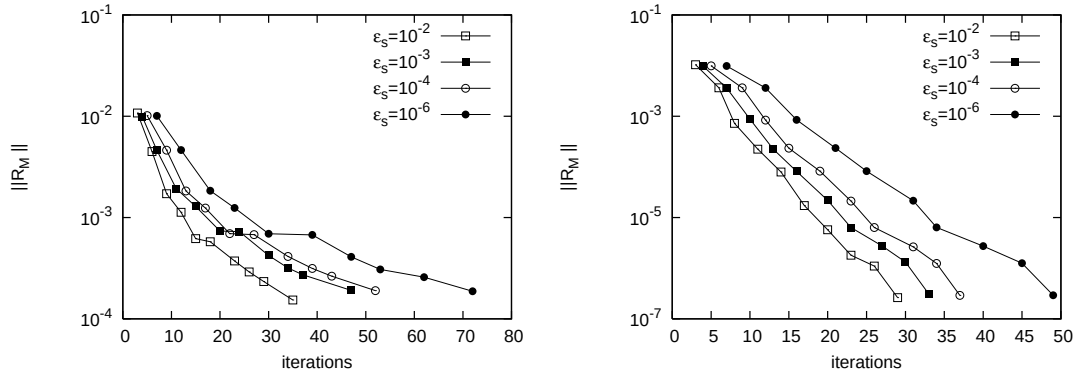

Figure 7. Convergence with the number of iterations of the reduction residual for different stopping criteria ϵ_s as indicated, and algorithms 1 (left plot) and 2 (right plot).

5.3.2 Impact of stochastic polynomial order

In a next series of computations, we vary the polynomial order $\mathrm{N_o} = 3, \ldots, 7$ of the stochastic approximation space $\mathscr{S}$, while holding $\mathrm{N} = 4$ fixed. Other parameters are the same as previously. These experiments can be understood as a refinement of the stochastic discretization, since $\dim(\mathscr{S})$ is directly related to $\mathrm{N_o}$ (see equation (67)). We then monitor the convergence of the two GSD algorithms with M for the different orders $\mathrm{N_o}$. Results are reported in Figure 8. The plots show that the convergence of the algorithms get slower as $\mathrm{N_o}$ increases. This is not surprising since increasing $\mathrm{N_o}$ allows to capture more variability in the solution so that more modes are needed to achieve the same level of accuracy in reduction. Still, one can observe that the convergence

rates tend to level off, denoting the convergence of the stochastic approximation as N_o increases. In fact, these results essentially highlight the need of a high polynomial order to obtain an accurate solution for the viscosity settings used. This is consistent with the decrease in the asymptotic value of the equation residual norm as N_o increases, as shown in Figure 9. Conversely, these computations demonstrate the robustness and stability of the power-type algorithms in constructing approximations on under-resolved stochastic space $\mathscr{S}$.

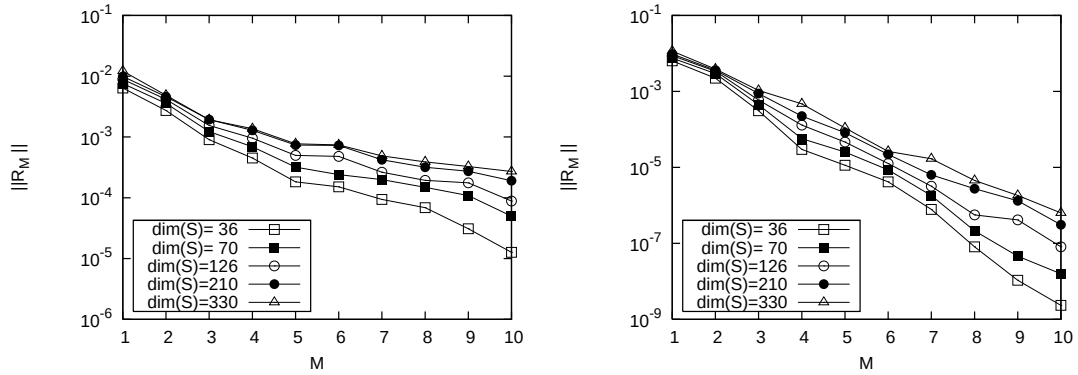

Figure 8. Convergence of the reduction residual R_M with M for different dimensions of the stochastic space $\mathscr{S}$ (corresponding to $N_o = 3, \ldots, 7$ and fixed $N = 4$): algorithms 1 (left plot) and 2 (right plot).

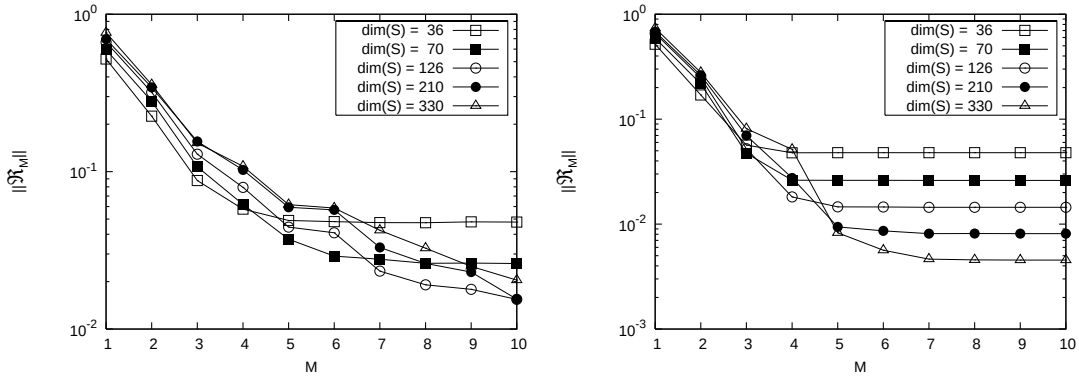

Figure 9. Convergence of the equation residual $\mathcal{R}_M$ with M for different dimensions of the stochastic space $\mathscr{S}$ (corresponding to $N_o = 3, \ldots, 7$ and fixed $N = 4$): algorithms 1 (left plot) and 2 (right plot).

5.3.3 Impact of the stochastic dimensionality

As in the previous tests, we want to compare the efficiencies of the algorithms when the dimension of $\mathscr{S}$ varies, but now due to different stochastic dimensionality N of the problem. Since the random viscosity, as previously parameterized, has decreasing variability when N increases, we need a different parameterization for a fair comparison. The viscosity distribution is now assumed Log-Normal, with median value $\overline{\mu}$ and coefficient of variation $C_{LN} > 1$. It means that the probability of having $\mu(\theta) \in]\overline{\mu}/C_{LN}, \overline{\mu}C_{LN}[$ is equal to 0.99.

Consequently, μ can be parameterized using a normalized normal random variable ζ as:

$$\mu = \exp\left[\overline{\zeta} + \sigma_\zeta \zeta\right], \quad \overline{\zeta} = \ln \overline{\mu}, \quad \sigma_\zeta = \frac{\ln C_{LN}}{2.95}. \tag{85}$$

The random variable ζ can in turns be decomposed as the sum of N independent normalized random variables ξ_i as follows:

$$\zeta = \frac{1}{\sqrt{\mathrm{N}}} \sum_{i=1}^{\mathrm{N}} \xi_i, \quad \xi_i \sim N(0,1). \tag{86}$$

Therefore, the parameterization $\mu(\boldsymbol{\xi})$ with $\boldsymbol{\xi} = \{\xi_1, \ldots, \xi_{\mathrm{N}}\} \in \Xi = (-\infty, \infty)^{\mathrm{N}}$ is

$$\mu(\boldsymbol{\xi}) = \overline{\mu} \exp\left[\frac{\ln C_{LN}}{\sqrt{\mathrm{N}}} \sum_{i=1}^{\mathrm{N}} \xi_i\right], \quad \xi_i \sim N(0,1). \tag{87}$$

It is stressed that for this parameterization the distribution of μ is the same for any $\mathrm{N} \geq 1$. Indeed, μ keeps a log-normal distribution with constant median and coefficient of variation for any N. However, changing N implies that the stochastic solution is sought in function space $L^2(\Xi, dP_{\boldsymbol{\xi}})$ with variable dimensionality for Ξ, such that even if the initial stochastic problem remains unchanged, the resulting problem to be solved on $\mathscr{S} \subset L^2(\Xi, dP_{\boldsymbol{\xi}})$ depends on N. In fact, this parametrization of μ is designed to investigate the efficiency of the GSD for the same problem but considered on probability spaces with increasing dimensionalities. Specifically, we use the Hermite Polynomial Chaos system as a basis of $\mathscr{S}$, so for fixed PC order $\mathrm{N_o}$ the dimension of $\mathscr{S}$ increases with N as given by (67). However, the PC solution for $\mathrm{N} > 1$ involves many hidden symmetries, and we expect the GSD algorithms to "detect" these structures and to construct effective reduced basis.

We set $\overline{\mu} = 0.3$, $C_{LN} = 3$ and $\mathrm{N_o} = 6$. The projection of μ on $\mathscr{S}$ can be determined analytically or numerically computed by solving a stochastic ODE [8]. We compute the GSD of the solutions for $\mathrm{N} = 2, \ldots, 5$ using the two algorithms with $\epsilon_s = 10^{-2}$. Results are reported in Figure 10 where plotted are the norms of residuals $\mathcal{R}_M$ and R_M as a function of the reduced basis dimension M. The plots show that the convergence of the two algorithms is indeed essentially unaffected by the dimension of Ξ.

5.4 Robustness with regards to input variability

In this paragraph, we investigate the robustness of the power-type algorithms with regards to the variability in μ. We rely on the previous parameterization of the Log-Normal viscosity, with $\mathrm{N} = 3$ and $\mathrm{N_o} = 6$ ($\dim(\mathscr{S}) = 84$). In a first

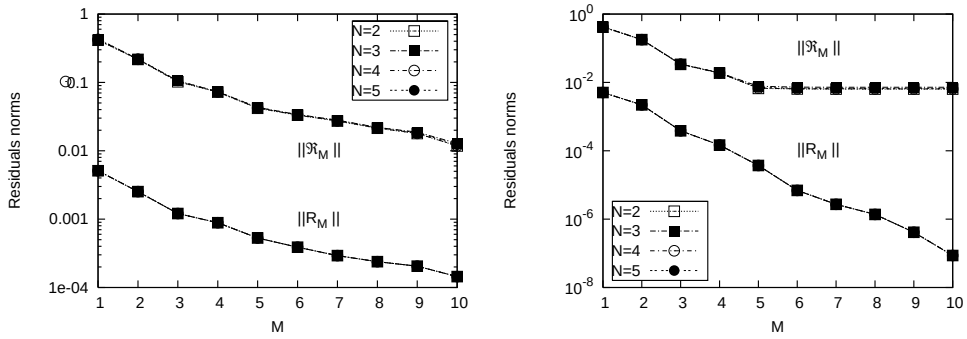

Figure 10. Convergence of the equation residual $\mathcal{R}_M$ and reduction residual R_M norms with M, for different dimensionality N of the stochastic space $\mathscr{S}$, as indicated, using algorithms 1 (left plot) and 2 (right plot).

series of computations we fix $\overline{\mu} = 0.3$ and we vary the coefficient of variability C_{LN} in the range $[1.5, 4]$. In a second series of computation, we fix $C_{LN} = 2.5$ and we vary the median value $\overline{\mu}$ in the range $[0.1, 0.4]$. Results are presented for algorithm 2 only, similar trends being found for algorithm 1.

In Figure 11 we have plotted the reduced basis approximation $u_{M=10}(x)$ for all the computations, using the classical mean value $\pm$ 3 standard deviation bars representation (even so this representation is not well suited here as the solution is clearly non-Gaussian). The plots of the left column correspond to $\overline{\mu} = 0.3$ and increasing coefficient of variability C_{LN} (from top to bottom). They show the increasing variability of the solution with C_{LN} while the mean of the solution is roughly unaffected. On the contrary, the plots of the right column corresponding to $C_{LN} = 2.5$ and increasing $\overline{\mu}$ (from top to bottom), show a large impact of the median value of the viscosity on the mean of the solution, together with a non trivial evolution of the solution variability. Specifically, although the variance of the log-normal viscosity is fixed, the maximum of variance in the solution increases as $\overline{\mu}$ decreases. This complex dependence of the solution with regards to the viscosity distribution underlines the strong non linear character of the Burgers equation.

Having shortly described the evolutions of the solution with the Log-Normal viscosity distribution, we can now proceed with the analysis of the convergence of the residuals $\|R_M\|$ shown in Figure 12. Focusing first on the convergence curves when $\overline{\mu}$ is fixed (left plot of Figure 12), it is first observed that the residual magnitude increases with C_{LN}, as one may have expected. Then, for the two lowest values of C_{LN} the convergence rates are found roughly equal, while slower convergences are reported for $C_{LN} = 3$ and 4. This trend can be explained by the increasing level of variability in the solutions for large COV, that demands more spectral modes to approximate the solution. Note that we have checked that $\dim(\mathscr{S})$ (*i.e.* $\mathrm{N_o}$) was sufficiently large to account for all the variability in the solution, when $C_{LN} = 4$, by performing a computation with $\mathrm{N_o} = 8$, without significant change in the solution.

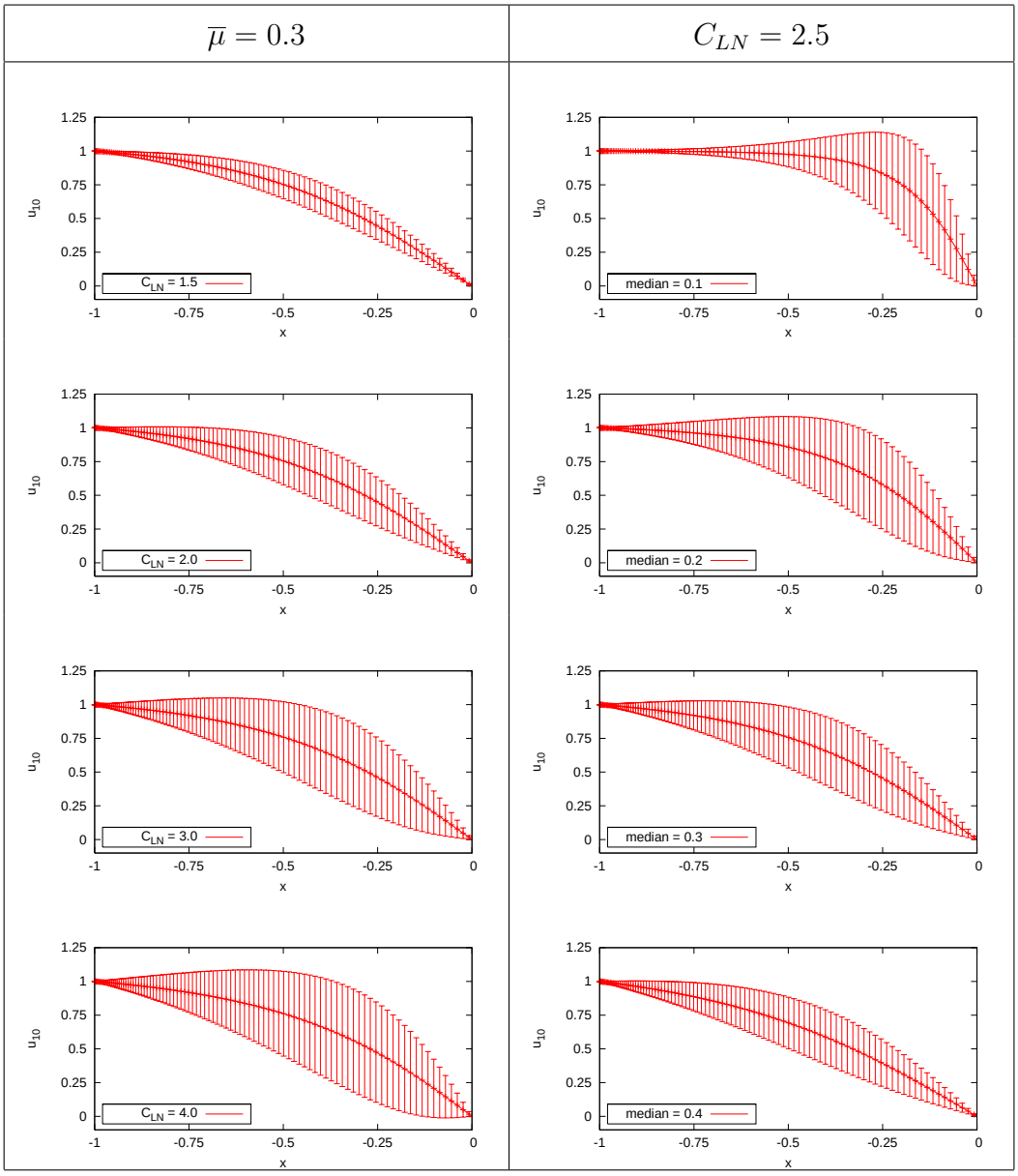

Figure 11. Mean and ±3 standard deviation bars representation of the reduced solutions $u_{M=10}$ for $\overline{\mu} = 0.3$ and $C_{LN} = 1.5$ to 4 (left plots from top to bottom) and $C_{LN} = 2.5$ and $\overline{\mu} = 0.1$ to 0.4 (right plots from top to bottom). Computations with algorithm 2, $\mathrm{N_o} = 6$ and $\mathrm{N} = 3$ ($\dim(\mathscr{S}) = 84$).

Next, the convergence of the GSD is analyzed for fixed $C_{LN} = 2.5$ of the viscosity distribution but increasing median value from 0.1 to 0.4 (right plots of Figure 12, from top to bottom). A degradation of the convergence rate, and an increasing residual magnitude, is observed as $\overline{\mu}$ decreases. This can be jointly explained by the increasing variability in the solution as seen from Figure 11, and by the more complex dependence with μ of the spatial structure of the solution as $\overline{\mu}$ decreases.

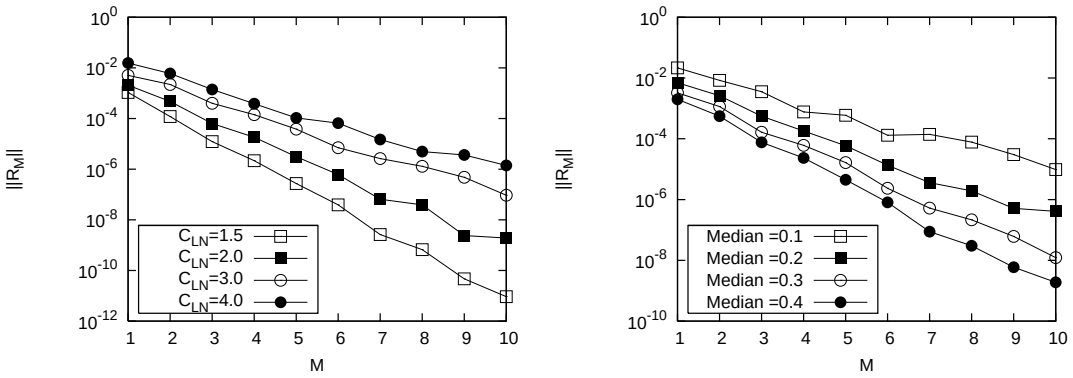

Figure 12. Convergence with M of the reduction residual R_M for $\overline{\mu} = 3$ and different C_{LN} (left plot) and $C_{LN} = 2.5$ and different $\overline{\mu}$ (right plot). Computations with algorithm 2, $\mathrm{N_o} = 6$ and $\mathrm{N} = 3$ ($\dim(\mathscr{S}) = 84$).

5.5 Convergence of probability density functions

To complete this section, we provide in this paragraph an appreciation of the GSD efficiency in terms of convergence of the resulting probability density function of the solution u_M as M increases. To this end, we set $\overline{\mu} = 0.3$ and $C_{LN} = 3$. The parameterization of the random viscosity uses $\mathrm{N} = 5$ with an expansion order $\mathrm{N_o} = 5$, such that the dimension of the stochastic approximation space is $\dim(\mathscr{S}) = 252$. The reduced solution u_M is computed using algorithm 2 with stopping criteria $\epsilon_s = 0.01$. We estimate the probability density function of $u_M(x, \boldsymbol{\xi})$, from a Monte-Carlo sampling of Ξ. For each sample $\boldsymbol{\xi}^{(i)}$ we reconstruct the corresponding solution $u_M(x, \boldsymbol{\xi}^{(i)})$ from:

$$u_M(x, \boldsymbol{\xi}^{(i)}) = \sum_{l=0}^{M} U_l(x) \lambda_l \left(\boldsymbol{\xi}^{(i)}\right) = \sum_{l=0}^{M} U_l(x) \sum_{k=0}^{\mathrm{P}} \lambda_l^k \Psi_k \left(\boldsymbol{\xi}^{(i)}\right). \tag{88}$$

These samples are then used to classically estimate the probability density functions (pdfs) of the solution at some prescribed points. For the analysis, we choose four mesh points which are the closest to $x = -1/8$, $-1/4$, $-1/2$ and $-3/4$. Since the reconstruction of the samples has a low computational cost, we use 10^6 samples to estimate the pdfs. Note that the samples may also be used to estimate other statistics of the solution (*e.g.* its moments).

In Figure 13, we show the computed pdfs at the four mesh points for different dimensions M of the reduced basis. It is seen that for $M = 1$, the reduced approximation provides poor estimates of the pdfs, especially for the points $x \approx -3/4$ and $x \approx -1/2$ where the probabilities of having $u > 1$ are significant. For $M = 2$, we already obtain better estimates of the pdfs, except for the closest point to the boundary, where $M = 3$ is necessary to achieve a smooth pdf. Increasing further M leads to no significant changes in the pdfs. These results are consistent with the previous observations on the convergence of the mean and standard deviation.

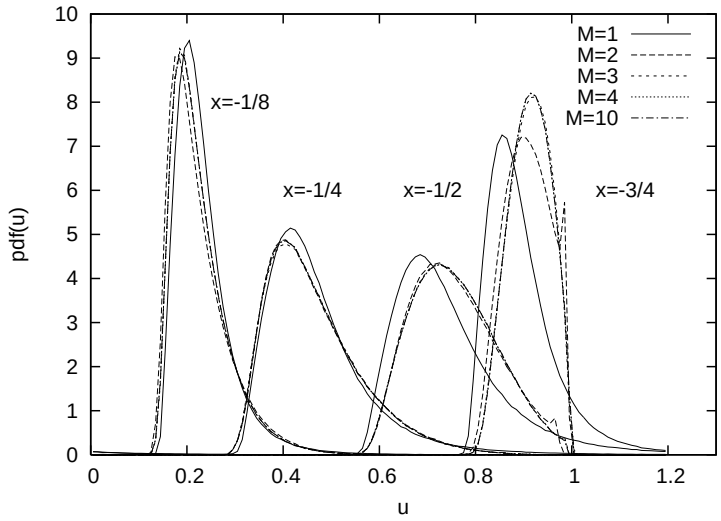

Figure 13. Convergence with M of the probability density function of u at some selected points as indicated. The problem uses $\overline{\mu} = 0.3$ and $C_{LN} = 3$, with a stochastic approximation space N = 5, $\mathrm{N_o} = 5$ ($\dim(\mathscr{S}) = 252$). Computations with algorithm 2 with $\epsilon_s = 0.01$.

To gain further confidence in the accuracy of the reduced basis approximation, we provide in Figure 14 a comparison of the pdfs for $u_{M=10}$ with the pdfs constructed from the classical Galerkin polynomial chaos solution on $\mathscr{S}$ and a Monte-Carlo simulation. The Galerkin solution is computed using an exact Newton solver, yielding a quadratic convergence rate: it can be considered as the exact Galerkin solution on $\mathscr{S}$. The Monte-Carlo simulation is based on a direct sampling of the log-normal viscosity distribution (and not of Ξ). Only 10^4 Monte-Carlo samples are used to estimate the pdfs, due to its computational cost, while the pdfs for the Galerkin solution uses the same 10^6 samples as the reduced approximation. It is seen in Figure 14 that the reduced approximation with only $M = 10$ modes leads to essentially the same pdfs as the full Galerkin solution which involves 252 modes. Also, they are in close agreement with the Monte-Carlo solution, with only small differences caused by the lower sampling used.

6 Application to a nonlinear stationary diffusion equation

In this section, we apply the GSD method to a nonlinear stationary diffusion equation with a cubic nonlinearity for which the mathematical framework can be found in [23]. Associated numerical experiments will be presented in the following section 7.

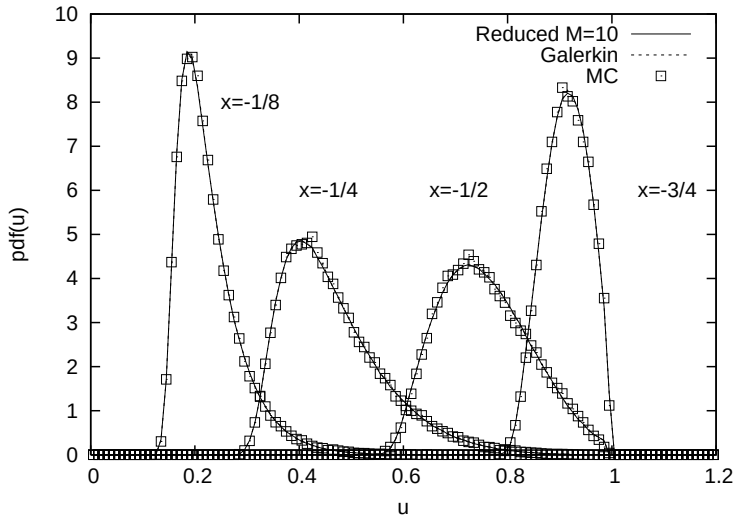

Figure 14. Comparison of the probability density functions of u at some selected points as indicated, for the reduced approximation $u_{M=10}$, the Galerkin solution and Monte-Carlo simulation. The problem corresponds to $\overline{\mu} = 0.3$ and $C_{LN} = 3$, with a stochastic approximation space $\mathrm{N} = 5$, $\mathrm{N_o} = 5$ ($\dim(\mathscr{S}) = 252$) for the Galerkin and reduced solutions, and direct sampling of the log-normal distribution of μ in the Monte-Carlo simulation.

6.1 Stationary diffusion equation

We consider a stationary diffusion problem defined on a L-shape domain $\Omega \subset \mathbb{R}^2$ represented on figure 15: $\Omega = ((0,1) \times (0,2)) \cup ((1,2) \times (1,2))$.

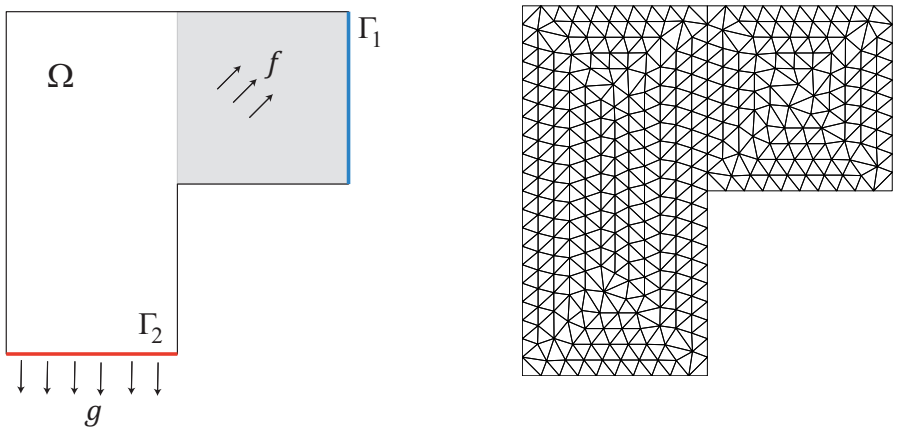

Figure 15. Diffusion problem: geometry, boundary conditions and sources (left) and finite element mesh (right).

Homogeneous Dirichlet boundary conditions are applied on a part Γ_1 of the boundary. A normal flux g is imposed on another part Γ_2 of the boundary. The complementary part of the boundary, denoted by Γ_0, is subjected to a zero flux condition. A volumic source f is imposed on a part $\Omega_1 = (1,2) \times (1,2)$ of

the domain.

The stochastic solution,

$$u : (x, \theta) \in \Omega \times \Theta \mapsto u(x, \theta) \in \mathbb{R}, \tag{89}$$

must satisfy almost surely

$$-\nabla \cdot ((\kappa_0 + \kappa_1 u^2)\nabla u) = \begin{cases} 0 \ on \ \Omega \backslash \Omega_1 \\ f \ on \ \Omega_1 \end{cases}, \tag{90}$$

$$-(\kappa_0 + \kappa_1 u^2)\frac{\partial u}{\partial n} = \begin{cases} 0 \ on \ \Gamma_0 \\ g \ on \ \Gamma_2 \end{cases}, \tag{91}$$

$$u = 0 \quad \text{on} \quad \Gamma_1, \tag{92}$$

where κ_0 and κ_1 are conductivity parameters. We consider that conductivity parameters and source terms are uniform in space. Then, they are modeled with real-valued random variables. The variational formulation writes (2) with:

$$b(u, v; \theta) = \int_\Omega (\kappa_0(\theta) + \kappa_1(\theta)u^2)\nabla u \cdot \nabla v \, dx, \tag{93}$$

$$l(v; \theta) = \int_{\Omega_1} f(\theta)\, v \, dx + \int_{\Gamma_2} g(\theta)\, v \, ds. \tag{94}$$

Remark 12 *Generalization of the methodology to situations where conductivity parameters or source terms are discretized stochastic fields is immediate.*

6.2 Application of GSD algorithms

We now detail the main ingredients of the GSD algorithms, namely steps (4) and (6) of algorithm 1, and the update step of algorithm 2. To detail the methodology, we write

$$b(u, v; \theta) = \kappa_0(\theta)a(u, v) + \kappa_1(\theta)n(u^2, u, v), \tag{95}$$

$$l(v; \theta) = f(\theta)l_1(v) + g(\theta)l_2(v), \tag{96}$$

where a and n are bilinear and trilinear forms respectively, defined as:

$$a(u, v) = \int_\Omega \nabla u \cdot \nabla v \, dx, \tag{97}$$

$$n(w, u, v) = \int_\Omega w\nabla u \cdot \nabla v \, dx. \tag{98}$$

6.2.1 Resolution of $U = F_M(\lambda)$

To compute $U = F_M(\lambda)$, one has to solve for U the following deterministic problem:

$$B_M(\lambda U, \lambda V) = L_M(\lambda V) \quad \forall V \in \mathscr{V}. \tag{99}$$

where $\forall u, v \in \mathscr{V} \otimes \mathscr{S}$,

$$B_M(u, v) = B(u_M + u, v) - B(u_M, v), \tag{100}$$
$$L_M(v) = L(v) - B(u_M, v). \tag{101}$$

After some manipulations, one obtains for the left-hand side:

$$\begin{aligned} B_M(\lambda U, \lambda V) =& \widetilde{\kappa}_0 a(U, V) + \widetilde{\kappa}_1 n(U^2, U, V) \\ &+ n(\widetilde{U}, U^2, V) + n(U^2, \widetilde{U}, V) + n(\overline{Z}, U, V) + n(U, \overline{Z}, V), \end{aligned} \tag{102}$$

where

$$\widetilde{\kappa}_0 = E(\kappa_0 \lambda\lambda), \quad \widetilde{\kappa}_1 = E(\kappa_1 \lambda\lambda\lambda\lambda), \tag{103}$$
$$\widetilde{U} = \sum_{i=1}^{M} E(\kappa_1 \lambda\lambda\lambda\lambda_i) U_i, \tag{104}$$
$$\overline{Z} = \sum_{i,j=1}^{M} E(\kappa_1 \lambda\lambda\lambda_i\lambda_j) U_i U_j. \tag{105}$$

We observe that the left hand side contains the classical linear and cubic terms with deterministic parameters $\widetilde{\kappa}_0$ and $\widetilde{\kappa}_1$ but also linear and quadratic additional terms.
For the right-hand side, one obtains the following expression:

$$L_M(\lambda V) = \widetilde{f} l_1(V) + \widetilde{g} l_2(V) - a(\breve{U}, V) - n(1, \widehat{Z}, V), \tag{106}$$

where

$$\widetilde{f} = E(f\lambda), \quad \widetilde{g} = E(g\lambda), \tag{107}$$
$$\breve{U} = \sum_{i=1}^{M} E(\kappa_0 \lambda\lambda_i) U_i, \tag{108}$$
$$\widehat{Z} = \frac{1}{3} \sum_{i,j,k=1}^{M} E(\kappa_1 \lambda\lambda_i\lambda_j\lambda_k) U_i U_j U_k. \tag{109}$$

In the numerical application, this deterministic problem is solved with a classical Newton-Raphson algorithm.

Remark 13 *Of course, various equivalent notations could have been introduced for writing left and right-hand sides of the deterministic problem. The*

above choice, introducing functions $\overline{Z}$ and $\widehat{Z}$, allows obtaining a compact writing, without summation on spectral modes. When introducing an approximation at the spatial level (e.g. finite element approximation), pre-computing an approximation of functions $\overline{Z}$ and $\widehat{Z}$ allows reducing the number of operations to be performed. This leads to an approximation in the evaluation of left and right-hand sides, and then in the obtained approximate solution, but it can also lead to significant computational savings.

6.2.2 Resolution of $\lambda = f_M(U)$

The random variable $\lambda \in \mathscr{S}$ is solution of the variational problem:

$$B_M(\lambda U, \beta U) = L_M(\beta U) \quad \forall \beta \in \mathscr{S}. \tag{110}$$

After some manipulations, this equation is found to be equivalent to:

$$E(\beta(\alpha^{(1)}\lambda + \alpha^{(2)}\lambda\lambda + \alpha^{(3)}\lambda\lambda\lambda)) = E(\beta\delta), \tag{111}$$

where

$$\alpha^{(1)} = \kappa_0 a(U,U) + \sum_{i,j=1}^{M} \kappa_1 \lambda_i \lambda_j \left[n(U_i U_j, U, U) + 2n(U_i U, U_j, U) \right], \tag{112}$$

$$\alpha^{(2)} = \sum_{i=1}^{M} \kappa_1 \lambda_i \left[2n(U_i U, U, U) + n(U^2, U_i, U) \right], \tag{113}$$

$$\alpha^{(3)} = \sum_{i,j=1}^{M} \kappa_1 \lambda_i \lambda_j \left[n(U_i U_j, U, U) + 2n(U_i U, U_j, U) \right], \tag{114}$$

$$\delta = f l_1(U) + g l_2(U) - \sum_{i=1}^{M} \kappa_0 \lambda_i a(U_i, U) - \sum_{i,j,k=1}^{M} \frac{1}{3} \kappa_1 \lambda_i \lambda_j \lambda_k n(1, U_i U_j U_k, U). \tag{115}$$

In the numerical application, this non-linear equation is solved with a classical Newton algorithm.

6.2.3 Resolution of $\Lambda_M = f_0(W_M)$

To update the random variables $\Lambda_M = (\lambda_1, \ldots, \lambda_M) \in (\mathscr{S})^M$, one has to solve:

$$B(W_M \cdot \Lambda_M, W_M \cdot \Lambda_M^*) = L(W_M \cdot \Lambda_M^*) \quad \forall \Lambda_M^* \in (\mathscr{S})^M. \tag{116}$$

This equation can be split into a system of M equations:

$$\forall k \in \{1, \ldots, M\}, \quad B(W_M \cdot \Lambda_M, U_k \beta_k) = L(U_k \beta_k) \quad \forall \beta_k \in \mathscr{S}. \tag{117}$$

Introducing the previously defined forms, it comes: $\forall k \in \{1, \ldots, M\}$,

$$\sum_{i=1}^{M} \kappa_0 a(U_i, U_k)\lambda_i + \sum_{i,j,l=1}^{M} \kappa_1 n(U_i, U_j, U_l, U_k)\lambda_i\lambda_j\lambda_l = f l_1(U_k) + g l_2(U_k). \quad (118)$$

This is a set of M coupled stochastic equations with a polynomial non-linearity. In the numerical application, this set of equations is solved with a classical Newton algorithm.

7 Results for the stationary diffusion equation

7.1 Discretization

At the stochastic level, we consider that random variables κ_0, κ_1, f and g are parametrized as follows:

$$\begin{aligned} \kappa_0 &= \mu_{\kappa_0}(1 + c_{\kappa_0}\sqrt{3}\xi_1) \\ \kappa_1 &= \mu_{\kappa_1}(1 + c_{\kappa_1}\sqrt{3}\xi_2) \\ f &= \mu_f(1 + c_f\sqrt{3}\xi_3) \\ g &= \mu_g(1 + c_g\sqrt{3}\xi_4) \end{aligned}$$

where the ξ_i are 4 independent random variables, uniformly distributed on $(-1, 1)$. Parameters $\mu_{(.)}$ and $c_{(.)}$ respectively correspond to the means and coefficients of variations of the random variables. We then work in the associated 4-dimensional image probability space $(\Xi, \mathcal{B}_\Xi, P_{\boldsymbol{\xi}})$, where $\Xi = (-1, 1)^4$, and use the same methodology as in section 4.5 for defining an approximation space $\mathscr{S} \subset L^2(\Xi, dP_{\boldsymbol{\xi}})$ based on a generalized polynomial chaos basis (multi-dimensional Legendre polynomials). We denote by $\mathrm{N_o}$ the polynomial chaos order.

At the space level, we introduce a classical finite element approximation space $\mathscr{V}^h \subset \mathscr{V}$ associated with a mesh of Ω composed by 3-nodes triangles (see Figure 15).

7.2 Reference solution and error indicator

The reference Galerkin approximate solution $u^h \in \mathscr{V}^h \otimes \mathscr{S}$ solves:

$$B(u^h, v^h) = L(v^h), \quad \forall v^h \in \mathscr{V}^h \otimes \mathscr{S}. \quad (119)$$

To obtain this reference solution, the non-linear set of equations associated with (119) is solved using a classical modified Newton method with a very high precision (see section 7.5 for details on the reference solver).

In order to analyze the convergence of the GSD method, we introduce an error indicator based on the residual of the discretized problem (119). This error indicator evaluates an error between the truncated GSD and the reference approximate solution u^h but not the error due to spatial and stochastic approximations. A given function $v \in \mathscr{V}^h \otimes \mathscr{S}$ is associated with a vector $\mathbf{v} \in \mathbb{R}^{\mathrm{N_x}} \otimes \mathscr{S}$. We denote by $R_M \in \mathscr{V}^h \otimes \mathscr{S}$ the reduction residual associated with $u_M \in \mathscr{V}^h \otimes \mathscr{S}$ and by $\mathbf{R}_M \in \mathbb{R}^{\mathrm{N_x}} \otimes \mathscr{S}$ the associated discrete residual, defined as follows: $\forall v \in \mathscr{V}^h \otimes \mathscr{S}$, associated with $\mathbf{v} \in \mathbb{R}^{\mathrm{N_x}} \otimes \mathscr{S}$,

$$E(\mathbf{v}^T \mathbf{R}_M) = L(v) - B(u_M, v). \tag{120}$$

An error indicator is then simply defined by the natural L^2-norm of the discrete residual, defined by

$$\|\mathbf{R}_M\|^2 = E(\mathbf{R}_M^T \mathbf{R}_M) \equiv \|R_M\|^2. \tag{121}$$

In the following, we will implicitly use a normalized error criteria $\|R_M\| \leftarrow \|R_M\|/\|R_0\|$, where R_0 stands for the right-hand side of the initial non-linear problem.

7.3 Convergence analysis

To analyze the convergence of the GSD algorithms, we choose the following parameters for defining the basic random variables:

$$\begin{aligned} &\mu_{\kappa_0} = 3, \quad \mu_{\kappa_1} = 1.5, \quad \mu_f = 6, \quad \mu_g = 2.25 \\ &c_{\kappa_0} = .2, \quad c_{\kappa_1} = .2, \quad c_f = .2, \quad c_g = .2 \end{aligned}$$

The basis of function space $\mathscr{S}$ is composed by multidimensional Legendre polynomials up to degree 5 ($\mathrm{N_o} = 5$), so that $\dim(\mathscr{S}) = \frac{(4+\mathrm{N_o})!}{4!No!} = 126$. For the spatial finite element discretization, we have $dim(\mathscr{V}^h) = 368$. If the stochastic solution was to be found in the full approximation space $\mathscr{V}^h \otimes \mathscr{S}$, the size of the non-linear problem to be solved would be $dim(\mathscr{V}^h) \times \dim(\mathscr{S}) = 46,368$. In contrast, the reduced basis solution $W_M \cdot \Lambda_M$ has for dimension $M \times (dim(\mathscr{V}^h) + \dim(\mathscr{S})) = 494M$.

In Figure 16, we compare the convergence of algorithms 1 and 2 with the size M of the reduced basis (left plot) and with the total number of power-type iterations performed for the computation of successive couples (U, λ) (right plot). The stopping criteria for power iterations is here $\epsilon_s = 10^{-2}$. Both algorithms rapidly converge to the discrete solution on $\mathscr{V}^h \otimes \mathscr{S}$ as the dimension

M of the reduced basis increases. Algorithm 2 is more effective in reducing R_M, compared to algorithm 1. Although Figure 16 shows that algorithm 2 requires less power iterations, both algorithms yields relatively similar computational costs on this particular example. Indeed, the faster convergence of algorithm 2 is balanced with computational efforts needed for the updating of random variables. This conclusion will not hold in general for large spatial approximation spaces.

Remark 14 *On this example, we observe a quasi-exponential convergence rate for small M and a decreasing of this rate for larger M. In fact, this is not due to a lack of robustness of the GSD method. It is related to the spectral content of the solution of this 2-dimensional problem. A classical spectral decomposition of the reference solution would reveal the same convergence behavior.*

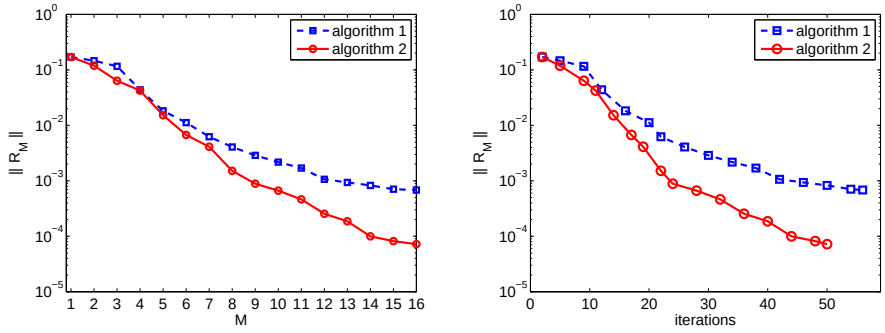

Figure 16. Convergence of the reduction residual R_M for algorithms 1 (squares) and 2 (circles). The left plot displays the residual norm as a function of the reduced basis dimension M, while the right plot displays the residual norm as a function of the total (cumulated) number of power-type iterations for the computation of successive couples (U, λ).

We compare in Figure 17 the 12 first deterministic functions U_i computed using the two algorithms. It is seen that algorithm 2 yields a deterministic reduced basis with a higher frequency content than for this of algorithm 1. In particular, we observe that the last modes obtained by algorithm 2 are essentially orthogonal to the first ones. This is further illustrated in Figure 18, where plotted are the second moment of the equation residual, $E(R_M^2)$, for different M and for the two algorithms. This plot also highlights the efficiency of the GSD in capturing the full discrete solution on $\mathscr{V}^h \otimes \mathscr{S}$ in just few modes and indicates that the stochastic discretization mostly affects the equation residual in the area where the solution exhibits the steepest gradients, *i.e.* where the uncertainty has the most impact on the solution.

Even though the equation residual norm provides a measure of the quality of the approximate solution, it is not a direct measure of the error on the solution. On figure 19, we plot the convergence curves of both algorithms with respect to the residual norm and also with respect to the L^2-norm on the

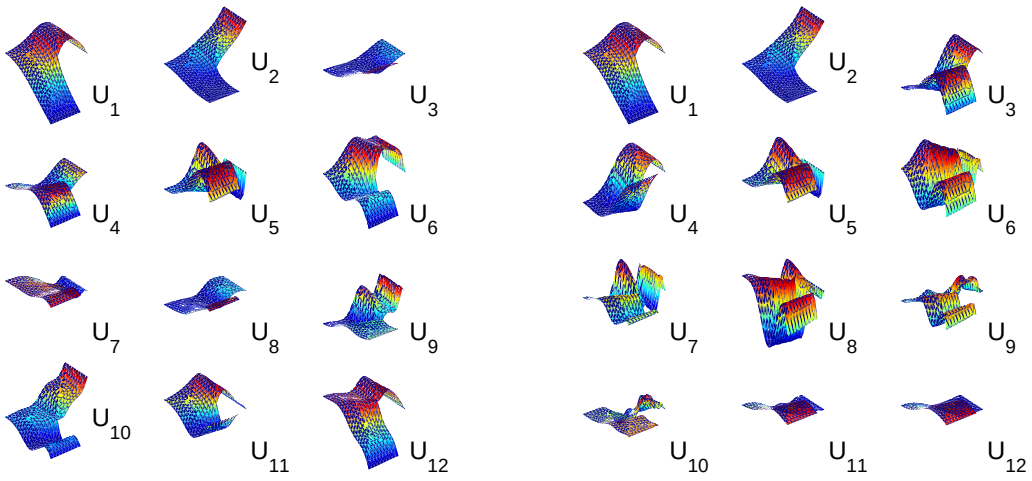

Figure 17. Comparison of the 12 first reduced modes with algorithms 1 (left plot) and 2 (right plot).

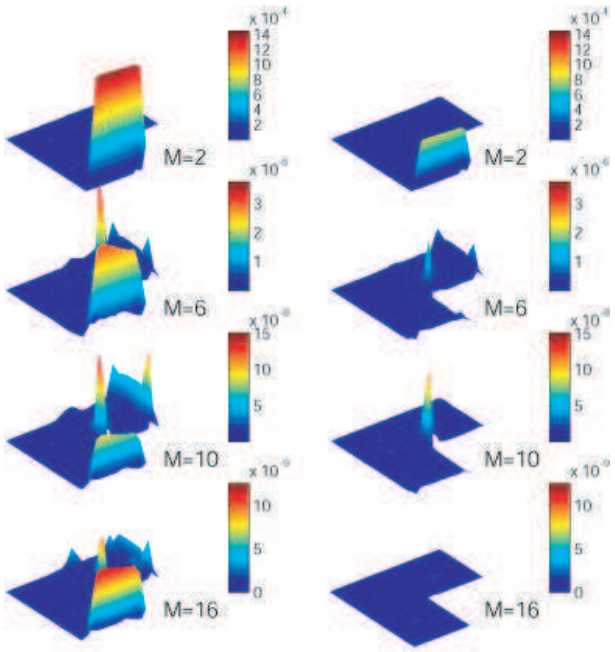

Figure 18. Evolution of the distribution of the second moment of the residual, $E(R_M^2)$, for different M and for algorithms 1 (left column) and 2 (right column).

solution. We observe that the error on the solution is significantly lower than the error based on the residual.

For a better appreciation of the convergence of the GSD, we have plotted in Figure 20 the distributions of the relative errors in mean ε_{mean} and standard

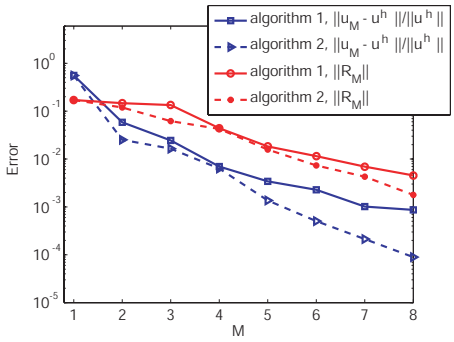

Figure 19. Convergence of $\|R_M\|$ and $\|u_M - u^h\|/\|u^h\|$ for algorithm 1 (solid line) and algorithm 2 (dashed line).

deviation ε_{Std} for different M and for the two algorithms:

$$\varepsilon_{mean} = \frac{|E(u_M) - E(u^h)|}{sup(|E(u^h)|)}$$
$$\varepsilon_{Std} = \frac{|Std(u_M) - Std(u^h)|}{sup(Std(u^h))}$$

We observe a very fast convergence of the GSD decomposition with both algorithms, with a faster convergence of algorithm 2. With only $M = 4$ modes, the relative error on these first two moments is inferior to 10^{-3}. On Figure 21, we have also plotted the convergence of probability density functions (pdfs) of the solution at two different points. We observe that approximate pdfs and reference pdf are essentially indistinguishable for $M \geq 5$. We also observe the superiority of algorithm 2, which yields more accurate pdfs with a lower order M of decomposition.

7.4 Robustness of the algorithms

We now investigate the robustness of the method with regards to stochastic discretization and numerical parameters.

7.4.1 Impact of ϵ_s

We first evaluate the impact of the criterium ϵ_s to stop the power iterations associated with the construction of a new couple (U, λ) (see Section 3.4.1). For that, we here consider less and less stringent stopping criteria ϵ_s and monitor the convergence of R_M. These experiments are reported in Figures 22 and 23, for the previous probabilistic setting and discretization parameters, and for $\epsilon_s = \{5.10^{-1}, 10^{-1}, 10^{-2}, 10^{-3}\}$. It is seen that for both algorithms, the selection of ϵ_s on the range tested has virtually no effect on the convergence of the decomposition, but is computationally more demanding as ϵ_s decreases.

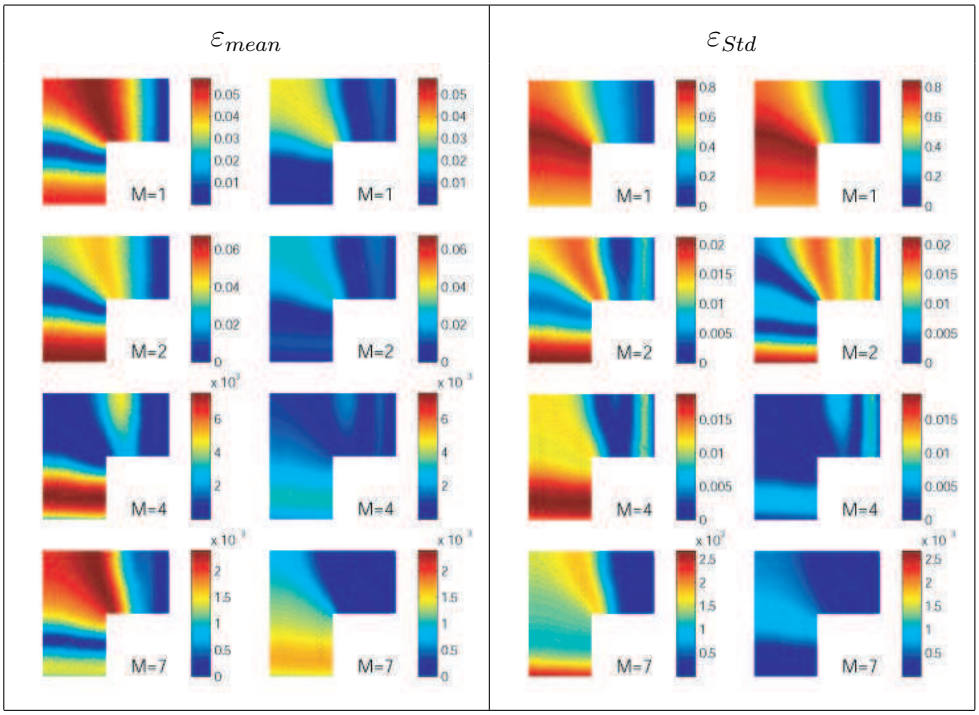

Figure 20. Distribution of the relative error in mean (ε_{mean}) and standard deviation (ε_{Std}) for algorithms 1 (first and third columns) and 2 (second and fourth columns) and different M.

In practise, it is not necessary to perform more than 3 or 4 power iterations to build a new couple (U, λ) (same observation as for the Burgers problem).

7.4.2 Impact of stochastic polynomial order

In a next series of computations, we vary the polynomial order $\mathrm{N_o} = 4, 5, 6$ of the stochastic approximation space $\mathscr{S}$, respectively corresponding to $dim(\mathscr{S}) = 70, 126, 210$. Figure 24, where plotted are the convergence curves for algorithm 1 (left plot) and algorithm 2 (right plot), shows that the polynomial order have a very low influence on the convergence. On this example, this can be explained by the fact that the error induced by the approximation at the stochastic level is lower that the error induced by the truncation of the GSD.

7.4.3 Impact of the input variability

We now investigate the robustness of GSD algorithms with respect to the input variability. We first vary the coefficients of variations $c_{(.)}$ of all random variables at the same time. Figure 25 shows the convergence with M for algorithm 1 (left plot) and algorithm 2 (right plot) for different coefficients of variation: $c_{(.)} = 0.1, 0.2, 0.3$. It is observed that the convergence rate decreases with the coefficient of variation, which is a usual property of spectral decom-

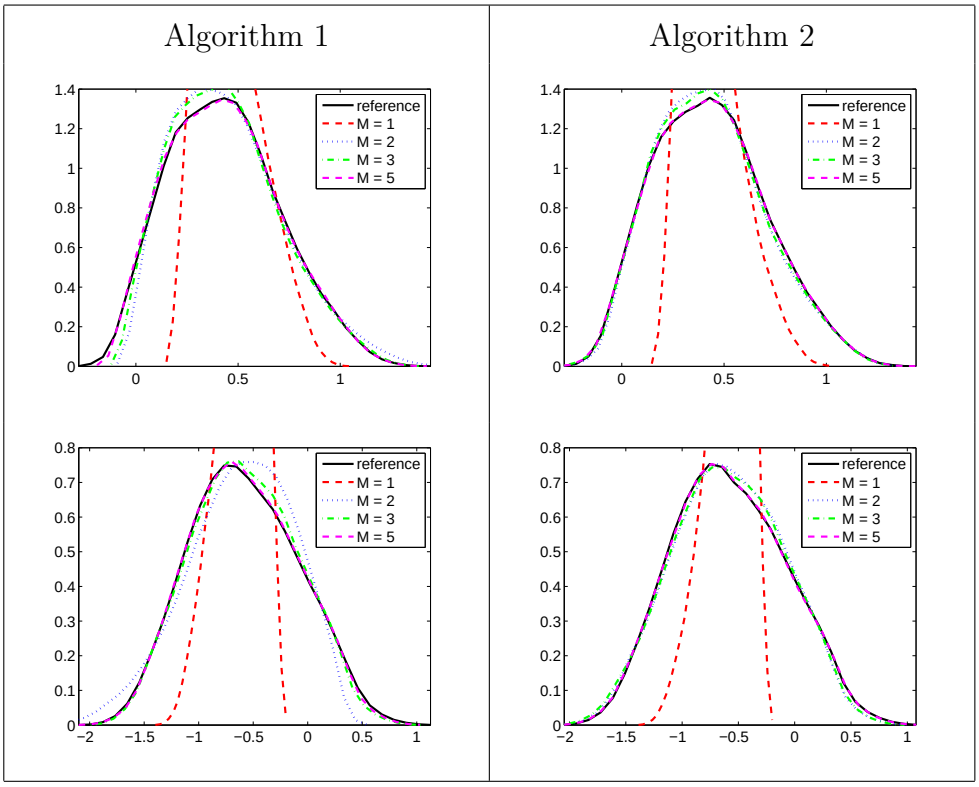

Figure 21. Convergence with M of the probability density function of u_M at points $P_1 = (1.5, 1.5)$ (top row) and $P_2 = (0.5, 0.1)$ (bottom row) and for algorithms 1 and 2.

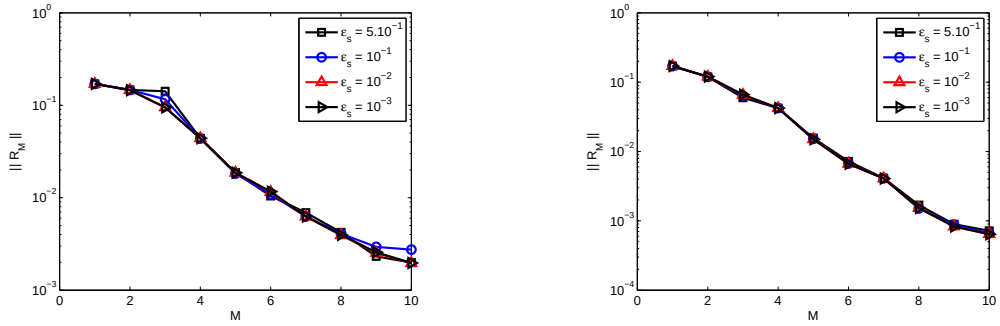

Figure 22. Impact of ϵ_s. Convergence with M for algorithms 1 (left plot) and 2 (right plot).

positions. However, the monotonic convergence illustrates the robustness of GSD algorithms in a wide range of input variability.

We now investigate the impact of the non-linearity by varying the mean μ_{κ_1} of parameter κ_1, letting all the coefficients of variations equal to $c_{(\cdot)} = 0.2$. Figure 25 shows the convergence with M for algorithm 1 (left plot) and algorithm 2 (right plot) for different $\mu_{\kappa_1} = 1.5, 0.5, 0.1, 0.01, 0$. We first observe that the convergence rate decreases as the non-linear term magnitude increases. This can be explained by the fact that the nonlinearity induces a more complex

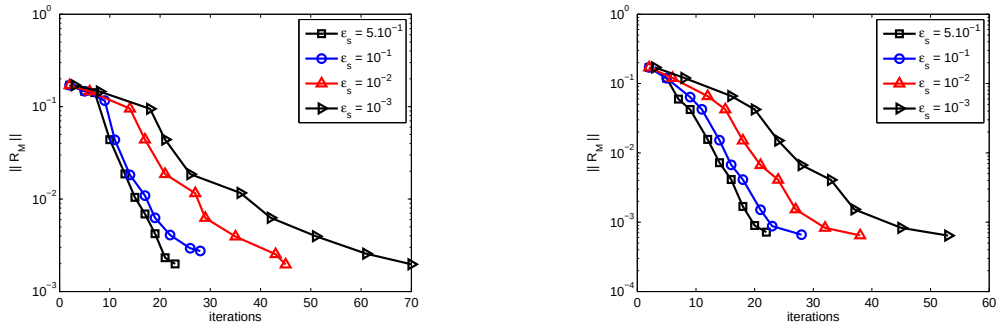

Figure 23. Impact of ϵ_s. Convergence with the number of power-type iterations for algorithms 1 (left plot) and 2 (right plot).

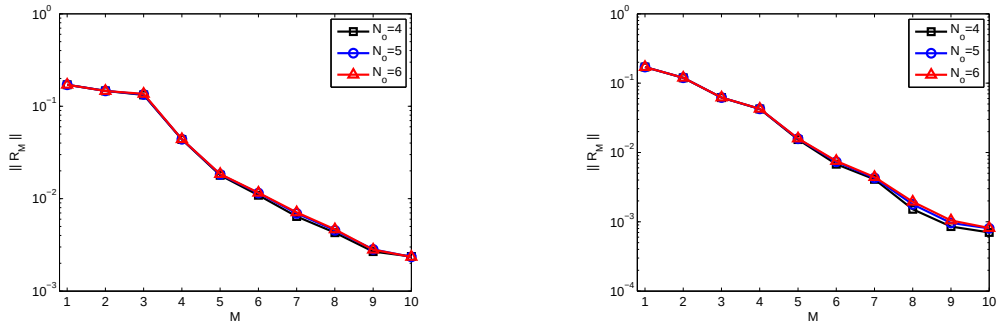

Figure 24. Impact of the polynomial chaos order. Convergence of algorithms 1 (left plot) and 2 (right plot).

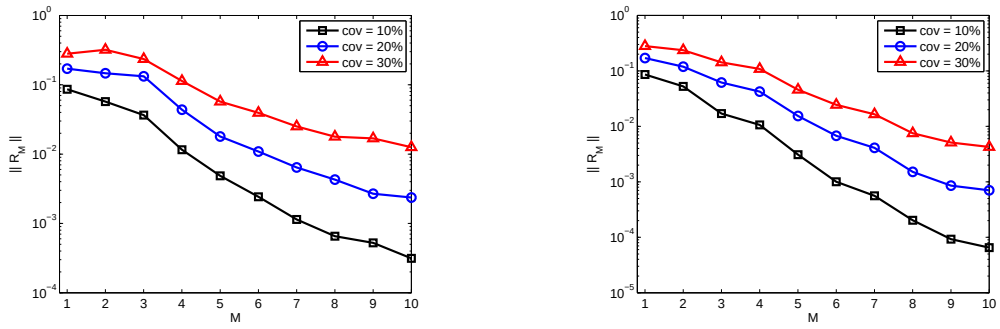

Figure 25. Impact of the input variability: convergence of algorithms 1 (left plot) and 2 (right plot), for different coefficients of variation (cov) of the four random variables, as indicated.

solution, which requires more spectral modes to be correctly captured.

For the case $\mu_{\kappa_1} = 0$, corresponding to the limit linear case, we observe that both algorithms capture the exact discrete solution in only 2 modes (at the computer numerical precision). We could have expected this property since it is clear on this example that only two modes are required to exactly represent the solution of the linear problem. Indeed, the two deterministic functions U_1

and U_2 which solves

$$a(U_1, V) = l_1(V) \; and \;\; a(U_2, V) = l_2(V), \forall V \in \mathscr{V}^h,$$

yield an exact decomposition when associated to the ad-hoc random variables. In fact, every couple of deterministic functions in the span of these functions yields an exact decomposition. This example shows that in this particular case, GSD algorithms allows capturing these ideal decompositions automatically.

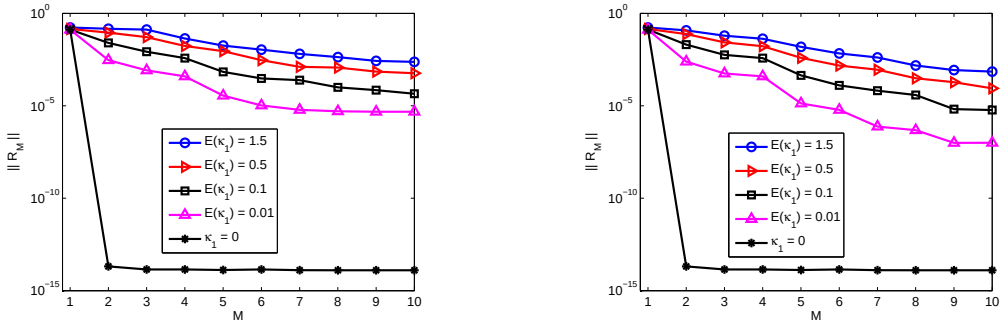

Figure 26. Impact of the non-linearity term (variable $E(\kappa_1)$). Convergence of algorithms 1 (left plot) and 2 (right plot).

7.5 Computation times

In this section, we illustrate the efficiency of the GSD method in terms of computation times. GSD algorithms are compared with a classical modified Newton algorithm for solving the reference Galerkin system of equations (119). A classical Newton method consists in the following iterations: starting from $u^{h,(0)} = 0$, iterates $u^{h,(i)} = 0$ are defined by

$$B'(u^{h,(i+1)}, v^h; u^{h,(i)}) = L(v^h) - B(u^{h,(i)}, v^h) \quad \forall v^h \in \mathscr{V}^h \otimes \mathscr{S} \tag{122}$$

where $B'(\cdot, \cdot; u)$ is the Gateaux derivative of semilinear form B evaluated at u:

$$\begin{aligned} B'(w, v; u) &= \lim_{\epsilon \to 0} \frac{1}{\epsilon} \left(B(u + \epsilon w, v) - B(u, v) \right) \\ &= E\left(\kappa_0 a(w, v) + \kappa_1 (2n(wu, u, v) + n(u^2, w, v)) \right) \end{aligned} \tag{123}$$

In order to reduce computation times of this reference solver, we use the following modification of iteration (122):

$$\begin{aligned} &\widetilde{B}'(u^{h,(i+1)}, v^h; E(u^{h,(i)})) = L(v^h) - B(u^{h,(i)}, v^h) \quad \forall v^h \in \mathscr{V}^h \otimes \mathscr{S} \\ &\widetilde{B}'(w, v; u) := E\left(\mu_{\kappa_0} a(w, v) + \mu_{\kappa_1} (2n(wu, u, v) + n(u^2, w, v)) \right) \end{aligned} \tag{124}$$

where $\widetilde{B}'$ is a simple approximation of B obtained by replacing random parameters κ_0 and κ_1 by their respective mean values. Moreover, $\widetilde{B}'$ is evaluated at $E(u^{h,(i)})$ instead of $u^{h,(i)}$. With these approximations, iteration (124) corresponds to a stochastic problem with a random right-hand side and a deterministic operator. The computation cost of this reference solver is then essentially due to the computation of the residual (right-hand side).

For the present example and moderate input variability, the proposed modified Newton algorithm have good convergence properties.

Remark 15 *For large variability of the input data, the efficiency of the proposed modified Newton method deteriorates. A better approximation of* $B'(\cdot,\cdot;u^{h,(i)})$ *should be provided in order to keep good convergence properties of the Newton algorithm. The robustness and efficiency of GSD algorithms are less affected by this increase in the input variability, as seen in section 7.4.3.*

For both GSD algorithms 1 and 2, we take $\epsilon_s = 10^{-1}$ for the stopping criteria for power iterations. Figure 27 shows the evolution of the residual norm with respect to computational time for the reference solver and for GSD algorithms. We clearly observe a computational gain with GSD algorithms (factor ≈ 6). We also observe that GSD algorithms 1 and 2 lead to similar computational times. In fact, the computational time required by the updating step in algorithm 2 is balanced by the fact that algorithm 2 needs for a lower order of decomposition than algorithm 1 for the same accuracy.

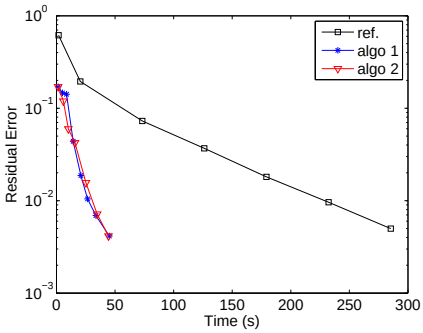

Figure 27. Residual error versus computation time for reference solver and GSD algorithms (reference discretization)

To go further in the comparison of computational costs, we analyze the influence on convergence properties of the dimensions P and $\mathrm{N_x}$ of stochastic and deterministic approximation spaces. We consider four finite element meshes corresponding respectively to $\mathrm{N_x} = 178$, 368, 726 and 1431. We also consider different polynomial chaos degrees $\mathrm{N_o} = 3$, 4, 5 and 6, respectively corresponding to $P = 34$, 69, 125 and 209.

Figures 28 and 29 show the convergence curves (residual norm versus computation time) for different $\mathrm{N_x}$ and $\mathrm{N_o}$. We observe that when increasing the di-

mension of approximation spaces, the efficiency of the reference solver rapidly deteriorates. GSD algorithms are far less affected by this increase of the dimension of approximation spaces.

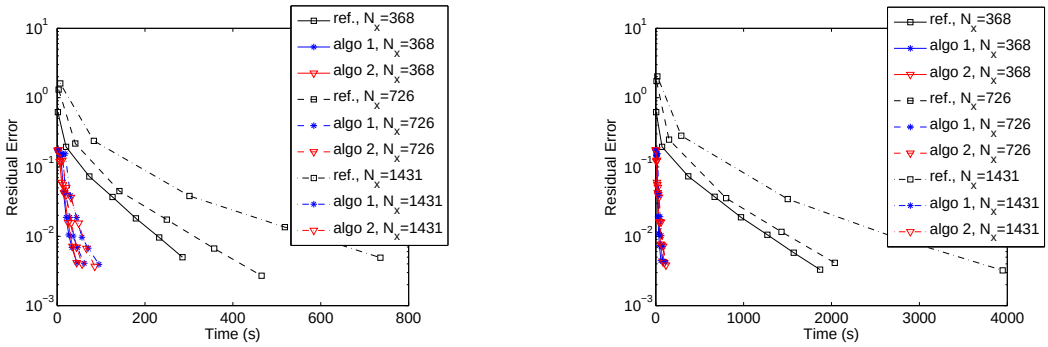

Figure 28. Influence of the dimension of approximation spaces. Residual error versus computation time for the reference solver and GSD algorithms 1 and 2 for different N_x and for $N_o = 4$ (left plot) or $N_o = 5$ (right plot)

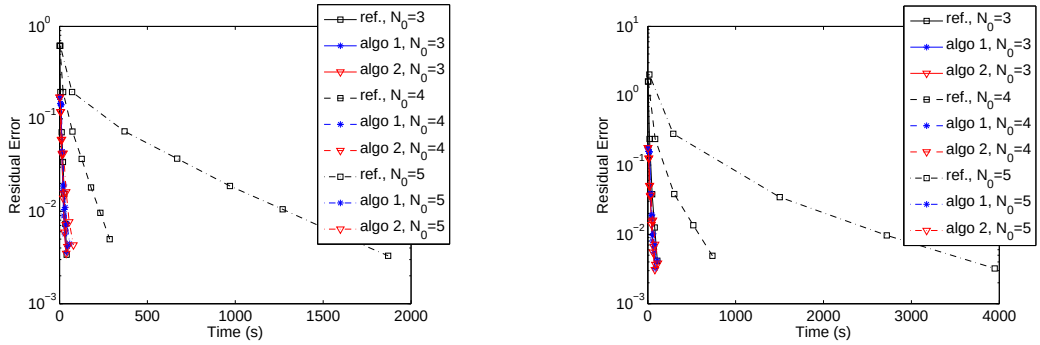

Figure 29. Influence of the dimension of approximation spaces. Residual error versus computation time for the reference solver and GSD algorithms 1 and 2 for different N_o and for $N_x = 368$ (left plot) or $N_x = 1431$ (right plot)

Figure 30 shows the gains in terms of computational times with respect to $N_x \times P$ (for different discretizations at stochastic level and deterministic level). The gain is computed by comparing computational times for the different algorithms to reach a given relative residual error of 5.10^{-2}. This accuracy is sufficient to obtain very accurate approximations in terms of moments, pdfs... This accuracy corresponds to the computation of 4 or 5 GSD modes. We clearly observe that GSD algorithms lead to computational savings which increase with the dimension of approximation spaces. GSD algorithms 1 and 2 lead to similar computational savings. For the finest discretizations, computational times are here divided by a factor up to 100.

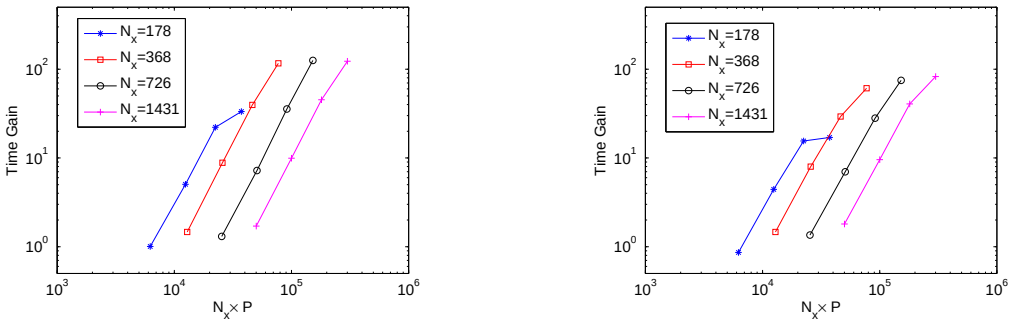

Figure 30. Time gain factor $T_g = \frac{time(reference\ solver)}{time(GSD\ algorithm)}$ with respect to $\mathrm{N_x} \times P$ for a given accuracy (relative residual error of 5.10^{-2}). GSD algorithm 1 (left) and GSD algorithm 2 (right plot)

8 Conclusion

In this paper, we have proposed an extension of the Generalized Spectral Decomposition method and related numerical procedures, initially proposed in [27, 28] for linear problems, to the resolution of non-linear stochastic problems in the context of Galerkin methods. The main features of the method is the approximation of the solution on reduced bases, automatically generated by the algorithms, with significant reduction of the computational requirements compared to the classical Galerkin projection schemes, and the independence of the methodology with regard to the type of stochastic discretization used.

Two non-linear test problems have served as examples to detail the methodology and to show the effectiveness of the proposed algorithms. Specifically, it has been shown that the algorithms lead to solution methods consisting in the resolution of a series of decoupled deterministic and low dimensional stochastic problems. An interesting point to be underlined is the structure of the deterministic problems to be solved which inherit the properties and dimension of the initial deterministic problem, with the introduction of few additional terms: only slight adaptations of available deterministic codes are required compared to the classical Galerkin method. Although being closely related to the polynomial character of the non-linearities in the test problems, this property already makes the GSD very attractive as a generic solution method for a large class of models (*e.g.* the incompressible Navier-Stokes equations).

For the two test problems, the numerical experiments have shown the effectiveness of the proposed algorithms to yield reduced decompositions that approximate the stochastic solution with a small number of modes compared to the dimension of the complete approximation space. For the second algorithm, the convergence of the reduced approximation is essentially governed by the actual spectrum of the stochastic solution, and not by the dimension of

the approximation space, as one may have anticipated from theoretical considerations. Also, algorithm 1 is less efficient than algorithm 2 in terms of accuracy for an equal number of modes in decomposition, but is computationally less expensive and simpler. This is however not enough to establish the general superiority of an algorithm over the other, as different aspects such as relative computational times for the deterministic and stochastic (update) problems, memory requirement and computational complexity intervene depending on the considered model and available resources. However, a common character of the two algorithms is their ability to yield the successive modes of the decomposition in only a few resolutions of the deterministic problem, thus implying large computational savings compared to the classical stochastic Galerkin method.

A potential improvement of the method, currently under investigation, concerns the implementation of alternative algorithms for the construction of the decomposition modes using advanced sub-space techniques (*e.g.* Arnoldi, see [28]) in order to drastically decrease the number of deterministic and reduced stochastic problems to be solved. Ongoing works are also focusing on applications of GSD to large scale problems (*e.g.* the Navier-Stokes equations) and extension to non-linear unsteady problems.

References

[1] M. Abramowitz and I.A. Stegun. *Handbook of Mathematical Functions.* Dover, 1970.

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

[3] F. E. Benth and J. Gjerde. Convergence rates for finite element approximations of stochastic partial differential equations. *Stochastics and Stochastics Rep.*, 63(3-4):313–326, 1998.

[4] M. Berveiller, B. Sudret, and M. Lemaire. Stochastic finite element : a non intrusive approach by regression. *Eur. J. Comput. Mech.*, 15:81–92, 2006.

[5] R.H. Cameron and W.T. Martin. The orthogonal development of nonlinear functionals in series of Fourier-Hermite functionals. *Ann. Math.*, 48:385–392, 1947.

[6] C. Canuto, M.Y. Hussaini, A. Quateroni, and T.A. Zang. *Spectral methods in fluid dynamics.* Springer-Verlag, 1988.

[7] M. Deb, I. Babuška, and J. T. Oden. Solution of stochastic partial differential equations using galerkin finite element techniques. *Comput. Methods Appl. Mech. Engrg.*, 190:6359–6372, 2001.

[8] B.J. Debusschere, H.N. Najm, P.P. Pébray, O.M. Knio, R.G. Ghanem, and O.P. Le Maître. Numerical challenges in the use of Polynomial Chaos representations for stochasic processes. *J. Sci. Comp.*, 26(2):698–719, 2004.

[9] A. Doostan, R.G. Ghanem, and J. Red-Horse. Stochastic model reduction for chaos representations. *Comput. Methods Appl. Mech. Engrg.*, 196:3951–3966, 2007.

[10] P. Frauenfelder, C. Schwab, and R. A. Todor. Finite elements for elliptic problems with stochastic coefficients. *Comput. Methods Appl. Mech. Engrg.*, 194(2-5):205–228, 2005.

[11] B. Ganapathysubramanian and N. Zabaras. Sparse grid collacation schemes for stochastic natural convection problems. *J. Comput. Phys.*, 225:652–685, 2007.

[12] R.G. Ghanem and R.M. Kruger. Numerical solution of spectral stochastic finite element systems. *Comput. Methods Appl. Mech. Engrg.*, 129:289–303, 1996.

[13] R.G. Ghanem and P.D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Dover, 2002. 2nd edition.

[14] H. Holden, B. Oksendal, J. Ubøe, and T. Zhang. *Stochastic Partial Differential Equations*. Birkhäuser, 1996.

[15] A. Keese and H. G. Mathhies. Hierarchical parallelisation for the solution of stochastic finite element equations. *Comput. Methods Appl. Mech. Engrg.*, 83:1033–1047, 2005.

[16] A. Keese and H.G. Matthies. Numerical methods and Smolyak quadrature for nonlinear stochastic partial differential equations. Technical report, Institute of Scientific Computing TU Braunschweig Brunswick, 2003.

[17] O.P. Le Maître, O.M. Knio, H.N. Najm, and R.G. Ghanem. Uncertainty propagation using Wiener-Haar expansions. *J. Comput. Phyics*, 197(1):28–57, 2004.

[18] O.P. Le Maître, H.N. Najm, R.G. Ghanem, and O.M. Knio. Multi-resolution analysis of Wiener-type uncertainty propagation schemes. *J. Comput. Physics*, 197(2):502–531, 2004.

[19] O.P. Le Maître, H.N. Najm, P.P. Pébay, R.G. Ghanem, and O.M. Knio. Multi-resolution-analysis scheme for uncertainty quantification in chemical systems. *J. Sci. Comp.*, 29(2):864–889, 2007.

[20] O.P. Le Maître, M.T. Reagan, H.N. Najm, R.G. Ghanem, and O.M. Knio. A stochastic projection method for fluid flow. ii. random process. *J. Comput. Phyics*, 181:9–44, 2002.

[21] O. Le Maître. A Newton method for the resolution of steady stochastic Navier-Stokes equations. *Computers and Fluids*, 2007. submitted.

[22] L. Mathelin and O. Le Maître. Dual based a posteriori error estimation for stochastic finite element methods. *Comm. Appl. Math. Comp. Sci.*, 2(1):83–115, 2007.

[23] H. G. Matthies and A. Keese. Galerkin methods for linear and nonlinear

elliptic stochastic partial differential equations. *Comput. Methods Appl. Mech. Engrg.*, 194(12-16):1295–1331, 2005.

[24] Jorge J. Moré, Burton S. Garbow, and Kenneth E. Hillstrom. User guide for MINPACK-1. Technical Report ANL-80-74, Argone National Lab, 1980.

[25] P. B. Nair and A. J. Keane. Stochastic reduced basis methods. *AIAA Journal*, 40(8):1653–1664, 2002.

[26] A. Nouy. Construction of generalized spectral bases for the approximate resolution of stochastic problems. *Mécanique & Industries*, 8(3):283–288, 2007.

[27] A. Nouy. A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations. *Comput. Methods Appl. Mech. Engrg.*, 196(45-48):4521–4537, 2007.

[28] A. Nouy. Generalized spectral decomposition method for solving stochastic finite element equations: invariant subspace problem and dedicated algorithms. *Comput. Methods Appl. Mech. Engrg.*, 2008. doi:10.1016/j.cma.2008.06.012.

[29] A. Nouy, A. Clément, F. Schoefs, and N. Moës. An extended stochastic finite element method for solving stochastic partial differential equations on random domains. *Comput. Methods Appl. Mech. Engrg.*, 2008. doi:10.1016/j.cma.2008.06.010.

[30] A. Nouy, F. Schoefs, and N. Moës. X-SFEM, a computational technique based on X-FEM to deal with random shapes. *Eur. J. Comput. Mech.*, 16(2):277–293, 2007.

[31] B. Oksendal. *Stochastic Differential Equations. An Introduction with Applications, fifth ed.* Springer-Verlag, 1998.

[32] M.F. Pellissetti and R.G. Ghanem. Iterative solution of systems of linear equations arising in the context of stochastic finite elements. *Adv. Engrg. Softw.*, 31:607–616, 2000.

[33] C.E. Powell and H.C. Elman. Block-diagonal precondioning for the spectral stochastic finite elements systems. Technical Report TR-4879, University of Maryland, Dept. of Computer Science, 2007.

[34] M.T. Reagan, H.N. Najm, R.G. Ghanem, and O.M. Knio. Uncertainty quantification in reacting flow simulations through non-intrusive spectral projection. *Combustion and Flames*, 132:545–555, 2003.

[35] S. K. Sachdeva, P. B. Nair, and A. J. Keane. Hybridization of stochastic reduced basis methods with polynomial chaos expansions. *Probabilistic Engineering Mechanics*, 21(2):182–192, 2006.

[36] C. Soize and R. Ghanem. Physical systems with random uncertainties: chaos representations with arbitrary probability measure. *SIAM J. Sci. Comput.*, 26(2):395–410, 2004.

[37] T.G. Theting. Solving wick-stochastic boundary value problems using a finite element method. *Stochastics and Stochastics Rep.*, 70:241–270, 2000.

[38] X. Wan and G.E. Karniadakis. An adaptative multi-element generalized

polynomial chaos method for stochastic differential equations. *J. Comput. Physics*, 209:617–642, 2005.

[39] X. Wan and G.E. Karniadakis. Multi-element generalized polynomial chaos for arbitrary propability measures. *SIAM J. Sci. Comp.*, 28(3):901–928, 2006.

[40] S. Wiener. The Homogeneous Chaos. *Amer. J. Math.*, 60:897–936, 1938.

[41] D.B. Xiu and G.E. Karniadakis. The Wiener-Askey Polynomial Chaos for stochastic differential equations. *SIAM J. Sci. Comput.*, 24:619–644, 2002.