Multiscale stochastic preconditioners in non-intrusive spectral projection[☆]

Alen Alenxanderian[a], Olivier P. Le Maître[b], Habib N. Najm[c], Mohamed Iskandarani[d], Omar M. Knio[a,*]

[a] *Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD 21218, USA*
[b] *LIMSI-CNRS, UPR-3251, Orsay, France*
[c] *Sandia National Laboratories, Livermore, CA 9455*
[d] *Rosenstiel School of Marine and Atmospheric Science, University of Miami, Miami, FL 33149-1098*

Abstract

A preconditioning approach is developed that enables efficient polynomial chaos (PC) representations of uncertain dynamical systems. The approach is based on the definition of an appropriate multiscale stretching of the individual components of the dynamical system which, in particular, enables robust recovery of the unscaled transient dynamics. Efficient PC representations of the stochastic dynamics are then obtained through non-intrusive spectral projections of the stretched measures. Implementation of the present approach is illustrated through application to a chemical system with large uncertainties in the reaction rate constants. Computational experiments show that, despite the large stochastic variability of the stochastic solution, the resulting dynamics can be efficiently represented using sparse low-order PC expansions of the stochastic multiscale preconditioner and of stretched variables. The present experiences are finally used to motivate several strategies that promise to yield further advantages in spectral representations of stochastic dynamics.

1. Introduction

Physical systems involving multiple variables changing dynamically over time are often modeled by systems of differential equations. In practice, the values of model parameters are not known exactly, and are best modeled by random variables. This gives rise to dynamical systems with random parameters. When studying such systems, one often aims to compute the evolution of the stochastic model variables, particularly to construct a statistical characterization of these variables over time. Straightforward Monte Carlo approaches, which involve simulating the random dynamical system with a large number of realizations of the model parameters, are too costly when working with dynamical systems which are computationally expensive to solve; this in turn, motivates development of more efficient computational approaches.

An increasingly popular class of methods for quantifying uncertainty in dynamical systems utilizes Polynomial Chaos (PC) expansions of the model variables [12, 21]. Such methods provide an approximation of model variables in terms of a (finite) spectral expansion involving random polynomials. Once available, the spectral expansion can be readily used to efficiently approximate the distribution of the model components over time, with substantially smaller computational effort compared to Monte Carlo sampling.

There are two major approaches for computing the spectral coefficients in a PC expansion, the *Galerkin* method and the so-called *non-intrusive* methods. The Galerkin method involves reformulating the original

[☆] Research supported by the U.S. Department of Energy, Office of Advanced Scientific Computing Research under Award DE-SC0001980 (AA, OMK), by Office of Naval Research under Award N00014-10-1-0498 (AA, OMK, MI), and by the French National Research Agency Grant ANR-08-JCJC-0022 (OLM). HNN acknowledges the support of the US Department of Energy (DOE), Office of Basic Energy Sciences, Division of Chemical Sciences, Geosciences, and Biosciences. Sandia National Laboratories is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the US DOE under contract DE-AC04-94-AL85000.

[*] Corresponding author. Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD 21218. Phone: (410) 516-7736; Fax: (410) 516-7254

Email addresses: (Alen Alenxanderian), olm@limsi.fr (Olivier P. Le Maître), hnnajm@sandia.gov (Habib N. Najm), miskandarani@rsmas.miami.edu (Mohamed Iskandarani), knio@jhu.edu (Omar M. Knio)

URL: http://www.limsi.fr/Individu/olm (Olivier P. Le Maître)

random dynamical system, through its Galerkin projection onto the approximation space spanned by the PC basis [12, 21]. One then has to solve a large system of deterministic differential equations for the time evolution of the PC coefficients. In contrast, the non-intrusive methods share the common characteristic of not requiring the reformulation of the original dynamical system, but aim to approximate the PC coefficients by reusing deterministic solvers. Within this paradigm, different types of non-intrusive methods have been proposed which can be distinguished according to means used to obtain or approximate the PC coefficients. Existing alternatives include the non-intrusive spectral projection (NISP), the collocation method (CM), and regression-like approaches. In the NISP method, the PC coefficients are obtained based on an L^2-projection of the model variables onto the space spanned by the PC basis [21, 37]. The CM considers the PC basis as a set of interpolants and defines the PC coefficients as interpolation coefficients [35, 28, 44, 3]. Finally, in regression-like approaches, the PC coefficients are obtained from the solution of a minimization problem, namely of the distance between the spectral expansion and a set of observations for the model variables [4].

In general, PC methods face the so-called curse of dimensionality. In the context of non-intrusive methods, this phenomenon manifests itself by the rapid increase in the number of deterministic realizations needed for adequate determination of the PC coefficients. This limitation has motivated many research efforts in the past years towards complexity reduction, and great progress has been achieved through the introduction of sparse tensorization and sparse grid techniques [17, 16, 10, 9, 26].

In addition to the issue of dimensionality, a well-known difficulty that occurs with time-dependent dynamical systems concerns the broadening of the spectrum of the PC representation over time. This in turns makes it difficult to approximate model variables via fixed-order PC expansions and to select a truncation order that remains suitable over the entire integration horizon. The underlying difficulty is that different realizations of the random solution trajectory fail to stay in phase as time progresses. In the context of chemical systems, for instance, some realizations can reach equilibrium very quickly, while others can exhibit a transient behavior over substantially larger timescales. This decorrelation of trajectories entails the excitation of higher-order modes in a straightforward PC representation of the model variables over time. Therefore, the corresponding PC expansions require a large number of terms for a reasonably accurate approximation of the stochastic behavior. The effect of such issues has been observed for example in [36]. In Section 4.3, we will see further examples illustrating this phenomenon. Within the Galerkin framework, the approximation of stochastic dynamics having dependence on random model parameter has given birth to alternative techniques based on expansion over piecewise (local) polynomial bases [30, 22, 24, 40, 41], as well as multi-resolution schemes [25, 31]. Although they enable propagation of uncertainty into systems with complex dynamics, these advanced methods mitigate but do not address the present issue in a satisfactory fashion, as they merely reduce the computational effort by dynamically adapting the approximation basis, which nevertheless continues to grow and eventually becomes very large [29]. In contrast, the objective of the present paper is to introduce suitable transformations of the model variables in such a way that a low-order spectral PC basis remains suitable for arbitrarily large times.

In [23], the authors proposed the method of asynchronous integration in the context of the Galerkin method. Targeting random dynamical systems having almost surely stable periodic trajectories, the key idea of the approach is that, assuming a smooth dependence of the limit-cycle with regard to the uncertain system parameters, the random trajectory can be expanded on a low order PC basis provided that one can maintain the process nearly in phase. This is achieved by introducing a time rescaling that depends on the random parameters.

Inspired by the work in [23], we presently explore an alternative stochastic preconditioning approach, in the context of non-intrusive methods. The method addresses the issue of widening spectra by introducing a global multiscale linear or affine transformation of the solution variables so as to synchronize realizations, and consequently control their variance and PC spectrum. Compared to the developments in [23], we relax the assumption of periodic dynamics, and exploit the non-intrusive character of the method to introduce random scaling parameters for each of the system variables. The method, which is illustrated for stiff chemical systems, is outlined in Section 3. In doing so, we exclusively rely on a non-intrusive formalism based on NISP. However, we emphasize the fact that extension to other type of non-intrusive methods is immediate. The basic idea behind the method is to work with transformed variables in scaled time which are in phase and have controlled spectra, *i.e.* the transformed variables have tight low order PC representations which can be computed with a limited number of realizations. Subsequently, the distribution of the original variables can be easily recovered, from that of transformed ones, through a sampling strategy.

The structure of the paper is as follows. In Section 2, we introduce the basic notation and the background ideas needed throughout the paper. In particular, we provide a brief overview of the PC expansion of square integrable random variables, dynamical systems with random inputs, and the classical NISP method. Section 3 focuses on our proposed preconditioning method for NISP. We start by introducing the random scaling parameters and define the associated scaled variables. Next, we highlight the projection of the scaled variables onto the PC basis, and outline the recovery of the original random model variables from their stretched counterparts. Implementation of preconditioned NISP is illustrated for the case of a stiff chemical system that is introduced in Section 4. The system consists of a simplified mechanism for supercritical water oxidation. Its selection is motivated by prior analysis of similar systems [36, 25], which had revealed complex dependence of the dynamics on random rate parameters. Computational experiments are then presented in Section 5 where we demonstrate projection of the stretched variables and the recovery of the original variables. The computations are further analyzed in Section 6 in light of error and convergence estimates, and in Section 7 where we discuss an extension of the preconditioning framework introduced earlier. A brief summary is provided in Section 8 together with a discussion of possible generalizations.

2. Background

In this section we fix the notation and collect background results used in the rest of the paper. In what follows $(\Omega, \mathcal{F}, \mu)$ denotes a probability space, where Ω is the sample space, $\mathcal{F}$ is an appropriate σ-algebra on Ω, and μ is a probability measure. A real-valued random variable ξ on $(\Omega, \mathcal{F}, \mu)$ is an $\mathcal{F}/\mathcal{B}(\mathbb{R})$-measurable mapping $\xi : (\Omega, \mathcal{F}, \mu) \to (\mathbb{R}, \mathcal{B}(\mathbb{R}))$, where $\mathcal{B}(\mathbb{R})$ denotes the Borel σ-algebra on $\mathbb{R}$. The space $L^2(\Omega, \mathcal{F}, \mu)$ denotes the Hilbert space of real-valued square integrable random variables on Ω.

For a random variable ξ on Ω, we write $\xi \sim \mathcal{N}(0, 1)$ to mean that ξ is a standard normal random variable and we write $\xi \sim \mathcal{U}(a, b)$ to mean that ξ is uniformly distributed on the interval $[a, b]$. We use the term iid for a collection of random variables to mean that they are independent and identically distributed. The distribution function [15, 43] of a random variable ξ on $(\Omega, \mathcal{F}, \mu)$ is given by $F_\xi(x) = \mu(\xi \le x)$ for $x \in \mathbb{R}$.

2.1. Polynomial Chaos

In this paper, we consider dynamical systems with finitely many random parameters, parameterized by a finite collection of real-valued iid random variables $\xi_1, \ldots, \xi_M$ on Ω. We let $\mathcal{V} = \sigma(\{\xi_i\}_1^M)$ be the σ-algebra generated by $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_M$ and, disregarding any other sources of uncertainty, work in the space $(\Omega, \mathcal{V}, \mu)$. By $F_{\boldsymbol{\xi}}$ denote the joint distribution function of the random vector $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_M)^T$. Note that since the ξ_j are iid $F_{\boldsymbol{\xi}}(\boldsymbol{x}) = \prod_{j=1}^M F(x_j)$ for $\boldsymbol{x} \in \mathbb{R}^M$. Let us denote by $\Omega^* \subseteq \mathbb{R}^M$ the image of Ω under $\boldsymbol{\xi}$, $\Omega^* = \boldsymbol{\xi}(\Omega)$, and by $\mathcal{B}(\Omega^*)$ the Borel σ-algebra on Ω^*. It is often convenient to work in the *image probability space* $(\Omega^*, \mathcal{B}(\Omega^*), F_{\boldsymbol{\xi}})$ instead of the abstract probability space $(\Omega, \mathcal{V}, \mu)$. We denote the expectation of a random variable $X : \Omega^* \to \mathbb{R}$ by $\langle X \rangle = \int_{\Omega^*} X(\boldsymbol{s})\, dF_{\boldsymbol{\xi}}(\boldsymbol{s})$. The space $L^2(\Omega^*, \mathcal{B}(\Omega^*), F_{\boldsymbol{\xi}})$ is endowed with the inner product $(\cdot, \cdot) : L^2(\Omega^*) \times L^2(\Omega^*) \to \mathbb{R}$ given by $(X, Y) = \int_{\Omega^*} X(\boldsymbol{s})Y(\boldsymbol{s})\, dF_{\boldsymbol{\xi}}(\boldsymbol{s}) = \langle XY \rangle$.

In the case $\xi_i \overset{iid}{\sim} \mathcal{N}(0, 1)$, any $X \in L^2(\Omega^*, \mathcal{B}(\Omega^*), F_{\boldsymbol{\xi}})$ admits an expansion of form,

$$X = \sum_{k=0}^{\infty} c_k \Psi_k, \tag{1}$$

where $\{\Psi_k\}_0^\infty$ is a complete orthogonal set consisting of M-variate Hermite polynomials [1], and the series converges in $L^2(\Omega^*, \mathcal{B}(\Omega^*), F_{\boldsymbol{\xi}})$. The expansion (1) is known as the (Wiener-Hermite) polynomial chaos expansion [42, 6, 14, 21] of U. In practice, the Wiener-Hermite expansion is most useful when the model parameters are parameterized by normally distributed random variables. In cases where the sources of uncertainty follow other distributions, it is convenient to adopt alternative parameterizations and polynomial bases. For example, in the case $\xi_i \overset{iid}{\sim} \mathcal{U}(-1, 1)$, we will use M-variate Legendre polynomials for the basis $\{\Psi_k\}_0^\infty$, and the expansion in (1) is a Generalized Polynomial Chaos [45] expansion of U.

Finally, in practical computations, we will be approximating $X(\boldsymbol{\xi})$ with a truncated series,

$$X(\boldsymbol{\xi}) \doteq \sum_{k=0}^{P} c_k \Psi_k(\boldsymbol{\xi}), \tag{2}$$

where P is finite and depends on the truncation strategy adopted. In the following, we consider truncations based on the total degree of the polynomials in the series, such that P depends of the stochastic dimension M and expansion "order" p according to: $1+P = \dfrac{(M+p)!}{M!p!}$, where p refers to the largest polynomial degree in the expansion.

2.2. Uncertain dynamical systems

Consider the autonomous ODE system,

$$\begin{cases} \dot{\boldsymbol{X}} = F(\boldsymbol{X}), \\ \boldsymbol{X}(0) = \boldsymbol{X}_0, \end{cases} \tag{3}$$

where the solution $\boldsymbol{X}$ of the system is a function $\boldsymbol{X} : [0, T_{fin}] \to \mathbb{R}^n$, with

$$\boldsymbol{X}(t) = [X^1(t), \ldots, X^n(t)]^T.$$

We consider the case of parametric uncertainty in the source term, i.e. $F = F(\boldsymbol{X}, \boldsymbol{\xi})$. Thus, the solution of (3) is a stochastic process,

$$\boldsymbol{X} : [0, T_{fin}] \times \Omega^* \to \mathbb{R}^n.$$

We can rewrite (3) more precisely as

$$\begin{cases} \dot{\boldsymbol{X}}(t, \boldsymbol{\xi}) = F\big(\boldsymbol{X}(t, \boldsymbol{\xi}), \boldsymbol{\xi}\big) \\ \boldsymbol{X}(0, \boldsymbol{\xi}) = \boldsymbol{X}_0, \quad a.s. \end{cases} \tag{4}$$

To capture the distribution of $X^i(t, \boldsymbol{\xi})$ at a given time, we will rely on the truncated PC expansion,

$$X^i(t, \boldsymbol{\xi}) = \sum_{k=0}^{P} c_k^i(t) \Psi_k(\boldsymbol{\xi}),$$

where the time-dependent coefficients $c_k^i(t)$ are to be computed non-intrusively, as outlined in the following section.

2.3. Non Intrusive Spectral Projection

Let us focus on a generic component of the uncertain dynamical system at a fixed time $t \in [0, T_{fin}]$ and simply denote this by X. As mentioned in the introduction, non-intrusive methods aim at computing the PC coefficients in the finite expansion (2) via a set of deterministic evaluations of $X(\boldsymbol{\xi})$ for specific realizations of $\boldsymbol{\xi}$. The NISP method, that will be used in throughout this paper, defines the expansion coefficients c_k in the expansion of $X(\boldsymbol{\xi})$ as the coordinates of its orthogonal projection on the space spanned by the $\{\Psi_k\}_0^P$. Observe that since $\{\Psi_k\}_0^P$ form an orthogonal system we have (cf. for example [32]):

$$\left(X - \sum_{l=0}^{P} c_l \Psi_l, \Psi_k \right) = 0, \quad k = 0, \ldots, P.$$

Therefore,

$$(X, \Psi_k) = \left(\sum_{l=0}^{P} c_l \Psi_l, \Psi_k \right) = \sum_{l=0}^{P} c_l \left(\Psi_l, \Psi_k \right) = c_k \left(\Psi_k, \Psi_k \right), \tag{5}$$

so that the coefficient c_k is given by

$$c_k = \frac{\langle X \Psi_k \rangle}{\langle \Psi_k^2 \rangle}. \tag{6}$$

In the case of Hermite or Legendre polynomials, the moments $\langle \Psi_k^2 \rangle$ in (6) can be computed analytically and hence, the determination of coefficients c_k amounts to the evaluation of the moments $\langle X\Psi_k \rangle$. We note that

$$\langle X\Psi_k \rangle = \int_{\Omega^*} X(\boldsymbol{s})\Psi_k(\boldsymbol{s})\, dF_{\boldsymbol{\xi}}(\boldsymbol{s}), \quad k = 0, \ldots, P.$$

leading to the evaluation of a set of $P+1$ integrals over $\Omega^* \subseteq \mathbb{R}^M$ to obtain the set of projection coefficients. These integrals are classically discretized as finite sums of the form

$$\int_{\Omega^*} X(\boldsymbol{s})\Psi_k(\boldsymbol{s})\, dF_{\boldsymbol{\xi}}(\boldsymbol{s}) = \sum_{j=1}^{N_q} w_j X(\boldsymbol{\xi}_j)\Psi_k(\boldsymbol{\xi}_j) + R_{k,N_q}(X), \tag{7}$$

where $\boldsymbol{\xi}_j \in \Omega^*$ and w_j are the nodes and weights of an appropriate quadrature formula, and $R_{k,N_q}(X)$ is the integration error. Note that the same set of nodes is used to compute all the coefficients c_k, so the complexity of NISP scales with N_q, the number of nodes where one has to compute X. Therefore, the challenge is to design quadrature formulae yielding the lowest integration error for the minimal number of nodes. In general, this is a difficult problem, and one often proceeds by tensorization of one-dimensional quadrature rules. Considering a 1-D quadrature rule with n_1 nodes, its full tensorization gives a M-variate formula having $N_q = n_1^M$ nodes, showing that this approach is limited to low M. This exponential scaling with M is often referred to as *the curse of dimensionality*. The latter can be significantly tempered by relying on sparse tensorizations of sequences of 1-D formulae using Smolyak's formula [38], which leads to sparse-grid integration techniques [11, 33, 34].

Irrespective of the formula considered for the computation of the c_k, the set of integration nodes comprise what we call the NISP sample and denote by

$$\mathcal{S} = \{\boldsymbol{\xi}_j\}_{j=1}^{N_q} \subset \Omega^*.$$

Thus, to evaluate (7) we need to compute $X(\boldsymbol{\xi}_q)$ for all $\boldsymbol{\xi}_q \in \mathcal{S}$. Let the matrix $\boldsymbol{\Pi} = (\Pi_{kj})$, $k = 0, \ldots, P$ and $j = 1, \ldots, N_q$ be given by

$$\Pi_{k,j} = \frac{w_j \Psi_k(\boldsymbol{\xi}_j)}{\langle \Psi_k^2 \rangle}.$$

We call the matrix $\boldsymbol{\Pi}$ the NISP projection matrix. If we denote by $\boldsymbol{x}$ the vector with coordinates $x_j = X(\boldsymbol{\xi}_j)$, then the vector $\boldsymbol{c} = [c_0, \ldots, c_k]^T$ of the spectral coefficients in (6) is given by the matrix vector product

$$\boldsymbol{c} = \boldsymbol{\Pi}\boldsymbol{x},$$

or in components,

$$c_k = \sum_{j=1}^{N_q} \Pi_{kj} x_j = \sum_{j=1}^{N_q} \Pi_{kj} X(\boldsymbol{\xi}_j), \quad k = 0, \ldots, P.$$

Going back to the original problem of projecting the n components of the solution $\boldsymbol{X}(t, \boldsymbol{\xi})$ to the uncertain dynamical system (4), we observe that the projection matrix $\boldsymbol{\Pi}$ is time independent and the same for all the components. As a result, the NISP projection of $\boldsymbol{X}(t, \boldsymbol{\xi})$ requires the resolution of the deterministic dynamics corresponding to $\boldsymbol{X}(t, \boldsymbol{\xi}_q)$, for each $\boldsymbol{\xi}_q \in \mathcal{S}$. We outline the calculation of PC coefficients of $X(t, \boldsymbol{\xi}) \doteq \sum_{k=0}^{P} c_k(t)\Psi_k(\boldsymbol{\xi})$ over time in Algorithm 1.

3. Preconditioned Non Intrusive Spectral Projection

When applying NISP (or any non-intrusive method), the computational complexity mainly depends on the number of sampling points needed to construct the approximation. This in turn is function of the complexity of the distribution of the random quantity one seeks to approximate, which as far as NISP is concerned, relates to the polynomial order needed for sufficient accuracy. As a result, the higher the polynomial order needed to obtain a suitable representation, the larger will be the NISP sample required. Therefore, preconditioning the random quantity so as to reduce the order of the PC expansion is of practical

Algorithm 1 The classical NISP algorithm

```
N_t = T_fin/Δt                                   {number of time steps}
N_q = |S|                                        {number of elements of S}
for j = 1 to N_q do
  Compute {X(iΔt, ξ_j), i = 0, ..., N_t}         {deterministic solve}
  for i = 0 to N_t do
    t_i = iΔt
    for k = 0 to P do
      c_k(t_i) = c_k(t_i) + Π_kj X(t_i, ξ_j)     {NISP sum}
    end for
  end for
end for
```

interest. This is the essential idea proposed in the present paper, where we pursue this goal through the appropriate transformation of an original time-dependent random vector $\boldsymbol{X}$ into a new one $\boldsymbol{Y}$ having a tight sparse PC expansion requiring less efforts to be projected. Such a procedure generally involves three distinct steps, namely the definition of the transformation (Section 3.1), the projection of the transformed variables (Section 3.2), and finally the recovery of the original variable (Section 3.3) for the purpose of sampling and analysis.

Taking advantage of the non-intrusive character of the proposed method, the preconditioning is performed separately for each component, X^i, of the system. Therefore, for convenience, we drop in this section the superscript i and consider a generic component X and its transform Y. Since in the present applications we will restrict ourselves to scaling-type transformations, we shall often refer to the transformed variable Y as the scaled variable.

3.1. Variable transformations

Recall that the state variable X is a stochastic process, $X : [0, T_{fin}] \times \Omega^* \to \mathbb{R}$. We introduce the transformed variable $Y = Y(\tau, \boldsymbol{\xi})$ that depends on a scaled time τ. The scaled time $\tau = \tau(t, \boldsymbol{\xi})$ is defined through,

$$\tau(t, \boldsymbol{\xi}) = \frac{t}{\hat{t}(\boldsymbol{\xi})}, \quad t \in [0, +\infty), \tag{8}$$

where $\hat{t} : \Omega^* \to (0, \infty)$ is a random variable which we call the *time scaling factor*. We assume that there exist positive constants $\kappa_1 < \kappa_2$ such that

$$\kappa_1 \leq \hat{t}(\boldsymbol{\xi}) \leq \kappa_2, \quad \text{for almost all } \boldsymbol{\xi} \in \Omega^*.$$

In general, we define the scaled variable Y by

$$Y(\tau(t, \boldsymbol{\xi}), \boldsymbol{\xi}) = \Phi[X(t, \boldsymbol{\xi})].$$

where Φ is an invertible mapping on $L^2(\Omega^*)$. In the present work, we will mainly focus on the case of *linear scaling* defined through

$$\Phi[X(t, \boldsymbol{\xi}); \hat{c}] = \frac{1}{\hat{c}(\boldsymbol{\xi})} X(t, \boldsymbol{\xi}), \tag{9}$$

where $\hat{c} : \Omega^* \to (0, \infty)$ is an *amplitude scaling factor*; we assume that there exist positive constants ν_1 and ν_2 that are independent of $\boldsymbol{\xi}$ and

$$\nu_1 \leq \hat{c}(\boldsymbol{\xi}) \leq \nu_2, \quad \text{for almost all } \boldsymbol{\xi} \in \Omega^*. \tag{10}$$

Note that (10) ensures that $\hat{c} \in L^\infty(\Omega^*) \subset L^2(\Omega^*)$; moreover, we have $1/\hat{c} \in L^\infty(\Omega^*)$ and since $X(t, \cdot) \in L^2(\Omega^*)$ it follows that $\Phi[X(t, \cdot); \hat{c}] \in L^2(\Omega^*)$, for every $t \in [0, T_{fin}]$. In the case where $X(t, \boldsymbol{\xi})$ is positive valued, as is the case for chemical species concentrations, we may also consider *log-linear scaling*,

$$\Phi[X(t, \boldsymbol{\xi}); \hat{c}] = \log\left(\frac{1}{\hat{c}(\boldsymbol{\xi})} X(t, \boldsymbol{\xi})\right). \tag{11}$$

An advantage of the log-linear scaling is in preserving positivity of the recovered state variables, as further discussed below. Moreover, since we will ultimately approximate the scaled variables with their PC expansions, sometimes the log-linear scaling may result in a better behaved PC representations for the scaled variables.

Remark 3.1 One may consider more general types of transformations $\Phi[X]$ and the scaling method introduced above can be seen as a particular instance of a broader class of preconditioners. An affine transformation is later discussed, though more general mappings are yet to be explored.

Remark 3.2 A fundamental question that we will be concerned with is: *How to choose the scaling factors $\hat{t}$ and $\hat{c}$ so that the scaled variable Y has a tight sparse PC representation that can be efficiently obtained on the basis of a coarse non-intrusive sampling?* In addition, we will eventually have to approximate $\hat{t}$ and $\hat{c}$ via PC expansions, and hence it is of practical concern that these scaling factors themselves have controlled PC representations. In the current work, we base this selection on the observed behavior of the state variables of the chemical model in our study, see Section 4. A general strategy for obtaining suitable scaling factors (and ideally selecting more general transformations) remains an open issue, as further discussed in Section 8.

3.2. Projection of the scaled variables

Here we describe in detail the projection of the scaling factors and the scaled variables into a PC basis.

3.2.1. Expansion of the scaling factors

We will need PC representations of the scaling factors, $\hat{c}$ and $\hat{t}$. These expansions will be computed using the NISP sample set $\mathcal{S}$. We will assume that for each $\boldsymbol{\xi}_j \in \mathcal{S}$ one is able to define a corresponding pair of factors $(\hat{t}(\boldsymbol{\xi}_j); \hat{c}(\boldsymbol{\xi}_j))$; as discussed in Section 4, the corresponding realization $X(\boldsymbol{\xi}_j, t)$ is used for this purpose. The expansions of the scaling factors are

$$\hat{c} \doteq \sum_{k=0}^{P} c_k \Psi_k, \qquad \hat{t} \doteq \sum_{k=0}^{P} t_k \Psi_k,$$

where the expansion coefficients are computed through a classical NISP procedure:

$$c_k = \sum_{j=1}^{N_q} \Pi_{k,j} \hat{c}(\boldsymbol{\xi}_j), \quad t_k = \sum_{j=1}^{N_q} \Pi_{k,j} \hat{t}(\boldsymbol{\xi}_j).$$

Since $\hat{c}$ and $\hat{t}$ are positive we may also consider the expansion of their logarithms. This again has the advantage of preserving their positivity and in some cases yielding nicer (tighter and/or sparser) PC spectra. In particular, we will project the logarithm of the normalized variables, $\hat{c}(\boldsymbol{\xi})/\hat{c}(\boldsymbol{\xi} = \mathbf{0})$ and $\hat{t}(\boldsymbol{\xi})/\hat{t}(\boldsymbol{\xi} = \mathbf{0})$:

$$\log\left(\frac{\hat{c}(\boldsymbol{\xi})}{\hat{c}(\mathbf{0})}\right) \doteq \sum_{k=0}^{P} \sigma_k \Psi_k, \qquad \log\left(\frac{\hat{t}(\boldsymbol{\xi})}{\hat{t}(\mathbf{0})}\right) \doteq \sum_{k=0}^{P} \theta_k \Psi_k,$$

where

$$\sigma_k = \sum_{j=1}^{N_q} \Pi_{k,j} \log\left(\frac{\hat{c}(\boldsymbol{\xi}_j)}{\hat{c}(\mathbf{0})}\right), \quad \theta_k = \sum_{j=1}^{N_q} \Pi_{k,j} \log\left(\frac{\hat{t}(\boldsymbol{\xi}_j)}{\hat{t}(\mathbf{0})}\right). \tag{12}$$

Accordingly $\hat{c}$ and $\hat{t}$ are approximated through,

$$\hat{c} \doteq \hat{c}(\mathbf{0}) \exp\left(\sum_{k=0}^{P} \sigma_k \Psi_k\right), \qquad \hat{t} \doteq \hat{t}(\mathbf{0}) \exp\left(\sum_{k=0}^{P} \theta_k \Psi_k\right). \tag{13}$$

In the computations below, we will use (13) to approximate $\hat{c}$ and $\hat{t}$.

3.2.2. Scaled variable discretization

The major computational work in the method involves the non-intrusive projection of the transformed variable $Y(\tau(t,\boldsymbol{\xi}),\boldsymbol{\xi}) = \Phi\big[X(t,\boldsymbol{\xi});\hat{c}\big]$ to get its PC representation. This calls for time discretization. Owing to the assumed properties of the time scaling in (8), for every $t \in [0,+\infty)$, $\tau(t,\cdot) \in [0,+\infty)$ almost surely. Moreover, for a given $(t',\boldsymbol{\xi}) \in [0,+\infty)\times\Omega^*$, there corresponds an (unscaled) time t given by $t = t' \times \hat{t}(\boldsymbol{\xi})$ and we have that

$$Y(t',\boldsymbol{\xi}) = \Phi\Big[X\big(t' \times \hat{t}(\boldsymbol{\xi}),\boldsymbol{\xi}\big);\hat{c}\Big].$$

This allows us to discretize the transformed variable Y on a fixed *deterministic* grid of times t' in the scaled space. With the slight abuse of notation where there is no confusion between τ and t', the PC expansion sought is then expressed as

$$Y(\tau,\boldsymbol{\xi}) \doteq \sum_{k=0}^{P} Y_k(\tau)\Psi_k(\boldsymbol{\xi}) \approx \Phi\Big[X(\tau \times \hat{t}(\boldsymbol{\xi}),\boldsymbol{\xi});\hat{c}\Big], \tag{14}$$

where the PC coefficients $Y_k(\tau)$ are to be computed on a fixed grid of scaled times τ. Specifically, given $\Delta\tau > 0$ and defining $\tau_l \doteq l\Delta\tau$, $l = 0,1,\ldots$ the NISP projection of the k-PC mode of Y would be

$$Y_k(\tau_l) = \sum_{j=1}^{N_q} \Pi_{k,j}\Phi\Big[X(\tau_l \times \hat{t}(\boldsymbol{\xi}_j),\boldsymbol{\xi}_j);\hat{c}\Big]. \tag{15}$$

3.2.3. Preconditioned projection algorithm

The expression in (15) suggests an implementation where one first selects the time step $\Delta\tau$ and then performs N_q successive deterministic solves for $X(t,\boldsymbol{\xi}_j)$, $\boldsymbol{\xi}_j \in \mathcal{S}$, recording and projecting the current state when t reaches $t = (l\Delta\tau)\times\hat{t}(\boldsymbol{\xi}_j)$. However, since each component X^i of the state vector $\boldsymbol{X}$ has its own time scale factor $\hat{t}^i(\boldsymbol{\xi})$, we have preferred a different approach resulting in a deterministic time integration of $\boldsymbol{X}(t,\boldsymbol{\xi}_j)$ with a fixed time step Δt for all the elements in $\mathcal{S}$, and to rely on an interpolation procedure to retrieve the values $X^i(\tau_l^i \times \hat{t}^i(\boldsymbol{\xi}_j),\boldsymbol{\xi}_j)$ needed for the projection in (15). The projection procedure is reported in Algorithm 2.

Algorithm 2 Algorithm for projection of $Y(\tau,\boldsymbol{\xi})$

```
Set N                                                    {number of deterministic time-steps}
Δt = T_end/N                                             {deterministic time-step}
Pick an appropriate Δτ                                   {time-step in scaled space}
Y = 0                                                    {Initialization}
for j = 1 to Nq do
  Compute χ = {X(iΔt, ξ_j), i = 0,...,N}                 {deterministic solve}
  [c̄, t̄] = GetScalings(ξ_j, χ)                           {compute scaling factors}
  for m = 0, 1, ... do
    τ = m × Δτ                                           {scaled time}
    t = τ_m × t̄                                          {unscaled time}
    if t > T_end break                                   {reach ending time}
    Interpolate X(t, ξ_j) from χ                         {preconditioned value for projection}
    for k = 0 to P do
      Y_k(τ) ← Y_k(τ) + Π_kj × Φ[X(t, ξ_j); c̄]           {NISP sum}
    end for                                              {loop on PC modes}
  end for                                                {loop on time in scaled space}
end for                                                  {loop on NISP sample}
```

Let us examine in detail a few aspects of Algorithm 2.

- Observe that only the inner loop on m (scaled time steps) differs from a component of $\boldsymbol{X}$ to another, as the corresponding scaling factors may differ. In fact, the most computationally demanding part of the

projection algorithm, namely the resolution of the deterministic problem for each of element $\boldsymbol{\xi}_j \in \mathcal{S}$, need not be repeated n times. Instead, one generates n data sets χ^i for the components $i = 1, \ldots, n$ of the state vector $\boldsymbol{X}$, which are subsequently projected independently using their corresponding scaling factors.

- An additional time $T_{end} > T_{fin}$ has been introduced. This is necessary for recovering the original variables from the transformed one up to T_{fin}, the desired final time. Letting m be the lowest integer such that $T_{fin} \leq m\Delta\tau \min_{\boldsymbol{\xi}\in\mathcal{S}} \hat{t}(\boldsymbol{\xi})$, then T_{end} should be taken such that $m\Delta\tau \max_{\boldsymbol{\xi}\in\mathcal{S}} \hat{t}(\boldsymbol{\xi}) \leq T_{end}$ to ensure that Algorithm 2 can project all the $Y(\tau_l)$, $l = 0, \ldots, m$. Obviously, T_{end} is taken as its maximum over the n components of $\boldsymbol{X}$ in the system case.

- The procedure **GetScalings**, which provides the values of the scaling factors to be used to transform the realization before projection, can be implemented in different ways. Here we list two possibilities which will compared in the applications below.

  Option 1: The scaling factors have already been projected and their PC expansions are known at this stage. Accordingly, **GetScalings** simply returns the values $\bar{c}$ and $\bar{t}$ given by the respective PC expansions in (13) evaluated at $\boldsymbol{\xi} = \boldsymbol{\xi}_j$.

  Option 2: The PC expansions of the scaling factors are not known at this stage. In this case, **GetScalings** returns values $\bar{c}$ and $\bar{t}$ which are the scaling factors determined from the analysis of the current realization dynamics, that is using the deterministic time series in χ (see Section 4).

  The first option supposes that the scaling factors have been previously projected, possibly using a different set of NISP nodes. This approach thus seems to waste computational resources, because, as discussed above, the projection of the scaling factors requires the simulation of the whole system for each point in $\mathcal{S}$ to determine $\hat{c}(\boldsymbol{\xi}_j)$ and $\hat{t}(\boldsymbol{\xi}_j)$ used in the projection through (12). This was not the case for the example proposed in this paper, as we could afford to store simulation results for every $\boldsymbol{\xi}_j \in \mathcal{S}$, and readily reuse them in Algorithm 2 with Option 1. However, this luxury may not be an option for larger $\mathcal{S}$ and/or more complex systems. This limitation leads to considering Option 2 where the transformation of the realization uses scaling factors determined on the *fly*, *i.e.* based uniquely on the current realization. For this second option, we also insert in Algorithm 2 an additional step (inside the loop on NISP points, but external to the loop on the scaled time-step) where $\hat{c}$ and $\hat{t}$ are projected, simultaneously with Y, using $\hat{t}(\boldsymbol{\xi}_j) = \bar{t}$ and $\hat{c}(\boldsymbol{\xi}_j) = \bar{c}$ in (12). It is important to remark that, due to projection errors for the scaling factors, the two options result in different PC expansions for the scaled variables Y. Specifically, one does not recover exactly the values $\hat{c}(\boldsymbol{\xi}_j)$ and $\hat{t}(\boldsymbol{\xi}_j)$ used in (12) when evaluating (13) at $\boldsymbol{\xi} = \boldsymbol{\xi}_j$, because of NISP integration and truncation errors. When using Option 2, these differences have to be controlled, as the recovery procedure (to be discussed shortly) uses the PC expansions of the scaling factors to invert the transformation. Therefore, inconsistencies in the definition of the scaling factors between Φ (the forward transform) and its inverse Φ^{-1} (the backward transform) may constitute an additional source of error.

- For the interpolation step we use a simple linear interpolation scheme to approximate $X(\cdot, \boldsymbol{\xi}_j)$ between elements in χ. Specifically, the interpolated value $X(t, \boldsymbol{\xi}_j)$ is evaluated from

$$X(t, \boldsymbol{\xi}_j) = X(t_l, \boldsymbol{\xi}_j) + \left[\frac{X(t_l + \Delta t, \boldsymbol{\xi}_j) - X(t_l, \boldsymbol{\xi}_j)}{\Delta t}\right](t - t_l),$$

  where $t_l = l\Delta t$ is such that $t_l \leq t < t_l + \Delta t$. Note that higher order interpolation schemes could be used, but this was found unnecessary in the application presented in this paper as we could afford to store computed solutions using sufficiently small time increments Δt.

- We have kept the algorithm general by not specifying the transformation Φ used in computation of the scaled variables. Throughout the paper, unless otherwise mentioned, we will rely on the *linear scaling* as defined in (9).

3.3. Recovering the original variables

We now turn to the problem of reconstructing $\boldsymbol{X}(t,\boldsymbol{\xi})$ from the transformed variable $\boldsymbol{Y}(\tau,\boldsymbol{\xi})$. The objective is to derive an efficient procedure to resample $\boldsymbol{X}(t,\boldsymbol{\xi})$, particularly to perform various statistical analyses. Given $t \in [0, T_{fin}]$ and a realization $\boldsymbol{\xi}_l \in \Omega^*$, we would like to to reconstruct the corresponding value of $\boldsymbol{X}(t,\boldsymbol{\xi}_l)$. Again, we focus on the treatment of a single generic component X of $\boldsymbol{X}$, as the generalization is immediate.

3.3.1. Backward transformation

The recovery amounts to the inversion of the transformation:

$$X(t,\boldsymbol{\xi}) = \Phi^{-1}\left[Y(\tau(t,\boldsymbol{\xi}),\boldsymbol{\xi});\hat{c}(\boldsymbol{\xi})\right]. \tag{16}$$

For the case of linear scaling, we have

$$X(t,\boldsymbol{\xi}) = \Phi^{-1}\left[Y(\tau(t,\boldsymbol{\xi}),\boldsymbol{\xi});\hat{c}(\boldsymbol{\xi})\right] = \hat{c}(\boldsymbol{\xi})Y(\tau(t,\boldsymbol{\xi}),\boldsymbol{\xi}), \tag{17}$$

while for the case of log-linear scaling, we have

$$X(t,\boldsymbol{\xi}) = \Phi^{-1}\left[Y(\tau(t,\boldsymbol{\xi}),\boldsymbol{\xi})\right] = \hat{c}(\boldsymbol{\xi})\exp\big(Y(\tau(t,\boldsymbol{\xi}),\boldsymbol{\xi})\big). \tag{18}$$

Of course, in practice we will approximate X by inserting the PC expansions of the scaling factors and scaled variables in (16). Let us introduce the notation $\tilde{X}$ to denote the recovered state variable X from the PC expansions of the scaled variable through the above relations. We have,

$$\begin{cases} \tilde{X}(t,\boldsymbol{\xi}) = \hat{c}(\boldsymbol{\xi})\Big\{\sum_{k=0}^{P} Y_k(\tau(t,\boldsymbol{\xi}))\Psi_k(\boldsymbol{\xi})\Big\} & \text{linear scaling,} \\ \tilde{X}(t,\boldsymbol{\xi}) = \hat{c}(\boldsymbol{\xi})\Big\{\exp\big(\sum_{k=0}^{P} Y_k(\tau(\boldsymbol{\xi},t))\Psi_k(\boldsymbol{\xi})\big)\Big\} & \text{log-linear scaling.} \end{cases} \tag{19}$$

3.3.2. Recovery and sampling algorithms

At a given $t^* \in [0, T_{fin}]$, we can sample the expansion of $\tilde{X}(t^*,\cdot)$ in (19) to generate an approximate distribution of $X(t^*,\cdot)$ fairly efficiently. The only issue remaining is that the PC coefficients of the scaled variable Y are known on the fixed grid of scaled times τ_l which generally will not coincide with the time $\tau(t^*,\cdot)$ in (19). Again, we rely on linear interpolation to compute the PC coefficients Y_k at the needed scaled time τ. The resulting procedure for computing $\tilde{X}$ is provided in Algorithm 3. Note that in Algorithm 3 the procedure **GetScalings** returns the values of the scaling factors evaluated from their PC expansions (as for Option 1).

Algorithm 3 Procedure for computing $\tilde{X}(t^*,\boldsymbol{\xi}^*)$ at $t^* \in [0, T_{fin}]$ and $\boldsymbol{\xi}^* \in \Omega^*$

$[\bar{c},\bar{t}] =$ **GetScalings**$(\boldsymbol{\xi}^*)$ {compute scaling factors from PC expansions}
$\tau^* = t^*/\bar{t}$ {scaled time}
Find τ_l such that $\tau^* \in [\tau_l, \tau_{l+1}]$
for $k = 0$ to P **do**
  $Y_k(\tau^*) = Y_k(\tau_l) + \left[\dfrac{Y_k(\tau_{l+1}) - Y_k(\tau_l)}{\Delta\tau}\right](\tau^* - \tau_l)$ {Interpolate $Y_k(\tau^*)$}
end for {loop on PC modes}
Compute $\tilde{X}(t^*,\boldsymbol{\xi}^*)$ from (19) with $\tau(t^*,\boldsymbol{\xi}^*) = \tau^*$ and $\hat{c}(\boldsymbol{\xi}^*) = \bar{c}$. {apply inverse transformation}

Finally, the complete sampling method is immediately constructed through a Monte Carlo procedure wrapped around the recovery procedure in Algorithm 3. For fixed t^*, one only has to sample Ω^* from the distribution $F_{\boldsymbol{\xi}}$ to obtain a sampling of the distribution of $\tilde{X}(t^*,\cdot)$. This is summarized in Algorithm 4 which will be used in Section 6 to assess the efficiency of the Preconditioned NISP.

Algorithm 4 Algorithm for sampling $\tilde{X}(t^*, \cdot)$ at a given $t^* \in [0, T_{fin}]$

Set Ns {number of sample points to be generated}
for $\ell = 1$ to Ns **do**
  Draw at random $\boldsymbol{\xi}_\ell$ from the distribution $F_{\boldsymbol{\xi}}$ {random sampling of Ω^*}
  Compute $\tilde{X}(t^*, \boldsymbol{\xi}_\ell)$ using procedure in Algorithm 3 {recover realization value}
end for {loop on sample}

4. Model problem

In this section, we describe the dynamical system that will be used to illustrate our proposed preconditioning strategy. The system corresponds to a hydrogen oxidation problem, based on the reduced mechanism that was proposed in [35]. This problem was chosen because it is known to yield solutions with broad spectra when decomposed on PC bases [37]. Previous studies based on stochastic Galerkin projection methods showed that the solution has complex dynamics, is highly sensitive to model parameters (rate constants), and therefore requires a fine stochastic discretization and adaptive strategies to be properly approximated [25, 29].

4.1. Stochastic model

The reduced mechanism from [35] involves the concentrations of seven species, namely $[OH]$, $[H]$, $[H_2O]$, $[H_2]$, $[O_2]$, $[HO_2]$, and $[H_2O_2]$, whose evolution is governed by eight elementary reversible reactions. We let $k_{f,j}$ and $k_{r,j}$ respectively denote the forward and reverse rate constant of the j-th reaction, $j = 1, \ldots, 8$. The randomness enters the model through uncertainty in the forward reaction rates $k_{f,j}$, $j = 1, \ldots 8$. Isothermal conditions are assumed, and the standard heats and entropies of the formation of the species are assumed deterministic. Accordingly, the equilibrium constant constants $K_{c,j}$ are deterministic, and the reverse rates are given by

$$k_{r,j} = K_{c,j}^{-1} \times k_{f,j} \ (a.s.), \quad j = 1, \ldots, 8.$$

We further assume that the model forward reaction rates are independent uniformly-distributed random variables. As a result, they can be parametrized using a set of 8 IID random variables ξ_j, according to

$$k_{f,j} = a_j + b_j \xi_j, \quad \xi_j \sim \mathcal{U}(-1, 1), \quad j = 1, \ldots 8,$$

where the constants a_j and b_j define the range of variability in $k_{f,j}$.

Table 1 lists the eight elementary reactions, the distributions of the forward reaction rates, as well as the value of the equilibrium constant.

Index	Reaction	Distribution of $k_{f,j}$	Equil. const. $K_{c,j}^{-1}$
$j = 1$	$OH + H \rightleftharpoons H_2O$	$\mathcal{U}(4.68 \times 10^{13}, 4.67 \times 10^{14})$	0.3491×10^{-30}
$j = 2$	$H_2 + OH \rightleftharpoons H_2O + H$	$\mathcal{U}(5.00 \times 10^{11}, 7.93 \times 10^{11})$	0.4380×10^{-03}
$j = 3$	$H + O_2 \rightleftharpoons HO_2$	$\mathcal{U}(5.26 \times 10^{13}, 1.31 \times 10^{14})$	0.1045×10^{-08}
$j = 4$	$HO_2 + HO_2 \rightleftharpoons H_2O_2 + O_2$	$\mathcal{U}(5.16 \times 10^{11}, 1.03 \times 10^{12})$	0.9879×10^{-13}
$j = 5$	$H_2O_2 + OH \rightleftharpoons H_2O + HO_2$	$\mathcal{U}(2.20 \times 10^{12}, 5.48 \times 10^{12})$	0.3382×10^{-08}
$j = 6$	$H_2O_2 + H \rightleftharpoons HO_2 + H_2$	$\mathcal{U}(8.48 \times 10^{10}, 3.39 \times 10^{11})$	0.7723×10^{-05}
$j = 7$	$H_2O_2 \rightleftharpoons OH + OH$	$\mathcal{U}(1.26 \times 10^{1}, 1.26 \times 10^{2})$	$0.1589 \times 10^{+12}$
$j = 8$	$OH + HO_2 \rightleftharpoons H_2O + O_2$	$\mathcal{U}(1.24 \times 10^{13}, 1.24 \times 10^{14})$	0.3534×10^{-17}

Table 1: Specification of the random reaction model. Listed are the elementary reactions, the forward rate distributions, and the equilibrium constants. Units are in the cm-mol-s-K system.

In the following, we denote by $\boldsymbol{X}$ the vector of random concentrations, with $X^1 = [OH]$, $X^2 = [H]$, $X^3 = [H_2O]$, $X^4 = [H_2]$, $X^5 = [O_2]$, $X^6 = [HO_2]$ and $X^7 = [H_2O_2]$. The dynamics of $\boldsymbol{X}(t, \boldsymbol{\xi})$ is governed by a non-linear system of first-order ODEs similar to (4) with $n = 7$. The initial condition $\boldsymbol{X}(t = 0)$ is assumed deterministic with $[H_2](t = 0) = 2.06 \times 10^{-6}\text{mol/cm}^3$, $[O_2](t = 0) = 1.04 \times 10^{-6}\text{mol/cm}^3$, and $[H_2O](t = 0) = 4.281 \times 10^{-3}\text{mol/cm}^3$; the rest of the concentrations are zero initially. The system's temperature and pressure are fixed at $T = 823$ K and $P = 246$ bar.

4.2. The sample set $\mathcal{S}$

Here we briefly describe the NISP sample $\mathcal{S}$ we used for the purposes of computations which follow in the subsequent sections. Recall that $\mathcal{S}$ is the set of integration nodes for evaluation of the PC coefficients through (7). In our case, we used full tensorization of a (one-dimensional) 5-point Gauss-Legendre quadrature formula [1, 2]. If we denote by $\mathcal{S}_0$ the set of five Gauss-Legendre nodes, the set $\mathcal{S}$ is then given by

$$\mathcal{S} = \underbrace{\mathcal{S}_0 \times \mathcal{S}_0 \times \cdots \times \mathcal{S}_0}_{8 \text{ times}}.$$

Thus, $\mathcal{S}$ has $5^8 = 390,625$ elements. While this may seem like a fairly large sample, in fact it is a fairly coarse sampling of the space Ω^*. Specifically, consistent with prior experiences in [25], we will later illustrate that direct NISP using the present sample set does not lead to a suitable PC representation.

4.3. Analysis of the stochastic dynamics

Figure 1 shows the evolution of species concentrations corresponding to deterministic realizations of the system taken in $\mathcal{S}$. Shown are the species O_2, H_2O, H, and HO_2, which are typical; the case of OH requires special treatment as later discussed in Section 8.

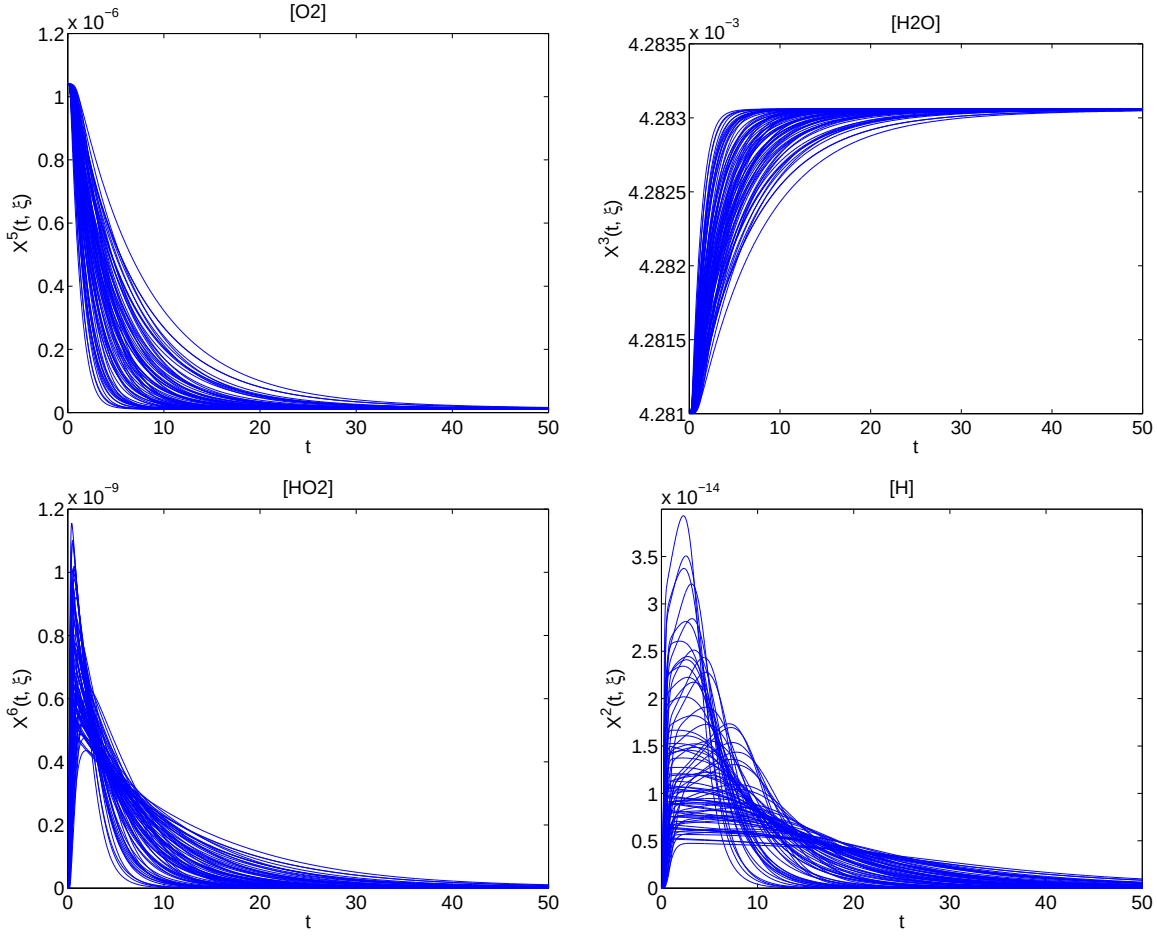

Figure 1: Concentration versus time selected realizations in $\mathcal{S}$. Shown are curves for O_2, H_2O, HO_2, and H, as indicated.

Figure 1 depicts the different types of dynamics observed for the species considered. The curves for O_2 (top-left) indicate that for all the realizations shown the concentration decreases monotonically with time. Uncertainty in the model parameters only affects the decay rate of $[O_2](t)$. A similar behavior is also reported for H_2 (not shown). In contrast, the case of H_2O (top-right) corresponds to a monotonic increase

of its concentration, which asymptotes to a deterministic equilibrium value (as must be the case since the equilibrium constants are assumed deterministic). However, the random model parameters are again seen to affect the characteristic response time of $[H_2O]$. HO_2 exhibits a more complex behavior, with a very fast initial increase in concentration, followed by a substantially slower monotonic decay. Unlike the previous cases where the random inputs were seen to modulate the response time scales, in the present case the peak amplitudes are substantially impacted as well. The case of H is essentially similar to HO_2, although with slower response time during the production stage. Also noticeable is the presence of some plateaus of different widths, during which $[H]$ remains nearly constant before it starts to decay.

These typical dynamics underscore potential difficulties faced in direct projection of the species concentrations. First, for the simplest case of monotonic dynamics, it is seen that time-scale variability leads to substantial variability in the stochastic concentration at a given time. This may not result in complex PC spectra, provided that the time-scale variability remains moderate. In fact, the occurrence of high variability in fast time-scales is frequently observed in ignition problems, which are known to be extremely challenging [31]. At any rate, it appears that an appropriate time transformation reducing the time-scale variability would certainly help controlling the corresponding PC spectrum. For non-monotonic dynamics, the difficulties above are further compounded by substantial uncertainty in the position and amplitude of the maxima. This variability is anticipated to lead to broad PC spectra, and consequently challenges in the representation of the stochastic solution. For the case of H, the occurrence of plateaus of variable durations is anticipated to make these challenges even more severe.

4.4. Scaling of the stochastic dynamics

We now propose appropriate transformations of the individual realizations in view of the PC projection of the transformed signals. The objective here is simply to support our choice of transformations; the actual projection and the effectiveness of the resulting preconditioning are extensively studied in the following section. As discussed previously in section 3, we take advantage of the non-intrusive character of NISP, which allows us to freely define a different transformation suited to each of the model variables (species concentrations). This is particularly important as we have just seen that different species experience different types of dynamics, some having simple monotonic behavior with variability in the characteristic time scale only, others presenting more complex evolution that prevent trivial scaling laws. However, we retain transformations of the type discussed in section 3.1, which incorporate only an amplitude scaling factor $\hat{c}^i$ and a time scaling factor $\hat{t}^i$.

We base the selection of scaling factors for the model variables on their observed behavior. In view of the plots in Figure 1, we have set the following types of scaling laws. Regarding amplitude scaling, we distinguish the case of the species having monotonic evolutions from the case of species with non-monotonic behavior. In the former case (monotonic), we define the amplitude scaling factor either as the initial concentration (monotonic decay) or as the equilibrium value (monotonic increase). Note that the asymptotic equilibrium value is known a priori as it is function of the initial concentrations and equilibrium constants which are all deterministic. For the case of non-monotonic evolutions, the amplitude scaling factor is defined as the maximal concentration achieved over time. Concerning the time scaling factors $\hat{t}^i$, they are defined as the time at which the corresponding species concentrations reaches a specific concentration value which depends on the amplitude scaling factor selected and eventually the initial concentration. We list our choice of (amplitude) $\hat{c}^i$ and (time) $\hat{t}^i$ scale factor definitions for $i = 2, \ldots, 7$ in Table 2. Note that for $X^{i=3,4,5}$ the scaling factors $\hat{c}^i$ are deterministic because they depend on initial or steady-state values which are deterministic in the present experiments.

Using these definitions of the scaling factors, we can compute for each realization (Figure 1) the corresponding value of the amplitude and time scale factors, and thus apply the transformations

$$X^i(t, \boldsymbol{\xi}) \mapsto Y^i(t/\hat{t}^i, \boldsymbol{\xi}) = \frac{1}{\hat{c}^i} X^i(t, \boldsymbol{\xi}),$$

to obtain the corresponding realization of the (linearly) scaled model variables $Y^i(\tau(\boldsymbol{\xi}, t), \boldsymbol{\xi})$. These realizations of the scaled variables are reported in Figure 2. Inspection of Figures 1 and 2 shows that the realizations of the scaled variables present a much lower level of (relative) variability at any time, which is even zero (by construction) at $\tau = \hat{t}$. Thus, the scaled variables appear to be better conditioned for a low-order PC projection than the original variables. However, this claim is not obvious for the two following reasons. First,

Amplitude scaling	Time scaling	Type
$\hat{c}^2(\boldsymbol{\xi}) = \max_t X^2(t,\boldsymbol{\xi})$	$\hat{t}^2(\boldsymbol{\xi}) =$ time for $X^2(t,\boldsymbol{\xi})$ to return to $\frac{1}{2}\hat{c}^2(\boldsymbol{\xi})$	non-monotonic
$\hat{c}^3(\boldsymbol{\xi}) = \lim_{t\to\infty} X^3(t,\boldsymbol{\xi})$	$\hat{t}^3(\boldsymbol{\xi}) =$ time for $X^3(t,\boldsymbol{\xi})$ to reach $\frac{1}{2}(X^3(0) + \hat{c}^2(\boldsymbol{\xi}))$	monotonic $\uparrow$
$\hat{c}^4(\boldsymbol{\xi}) = X^4(0)$	$\hat{t}^4(\boldsymbol{\xi}) =$ time for $X^4(t,\boldsymbol{\xi})$ to reach $\frac{1}{2}\hat{c}^4(\boldsymbol{\xi})$	monotonic $\downarrow$
$\hat{c}^5(\boldsymbol{\xi}) = X^5(0)$	$\hat{t}^5(\boldsymbol{\xi}) =$ time for $X^5(t,\boldsymbol{\xi})$ to reach $\frac{1}{2}\hat{c}^5(\boldsymbol{\xi})$	monotonic $\downarrow$
$\hat{c}^6(\boldsymbol{\xi}) = \max_t X^6(t,\boldsymbol{\xi})$	$\hat{t}^6(\boldsymbol{\xi}) =$ time for $X^6(t,\boldsymbol{\xi})$ to return to $\frac{1}{2}\hat{c}^6(\boldsymbol{\xi})$	non-monotonic
$\hat{c}^7(\boldsymbol{\xi}) = \max_t X^7(t,\boldsymbol{\xi})$	$\hat{t}^7(\boldsymbol{\xi}) =$ time for $X^7(t,\boldsymbol{\xi})$ to return to $\frac{1}{2}\hat{c}^7(\boldsymbol{\xi})$	non-monotonic

Table 2: Definition of the scaling factors $\hat{c}^i(\boldsymbol{\xi})$ and $\hat{t}^i(\boldsymbol{\xi})$ for the transformation of the model variables X^i, $i = 2, \ldots, 7$.

the scaling factors may not have simple PC expansions thus preventing low-order expansions. Second, even if the factors admit low-order expansion, the associated transformations may result in scaled variables Y^i with reduced variance but still having complex (broad) spectra. These two aspects are carefully investigated in the next two sections, where we show that, unlike the original variables, both the scale factors and scaled variables have indeed tight sparse spectra, and can be suitably represented by low-order PC expansions.

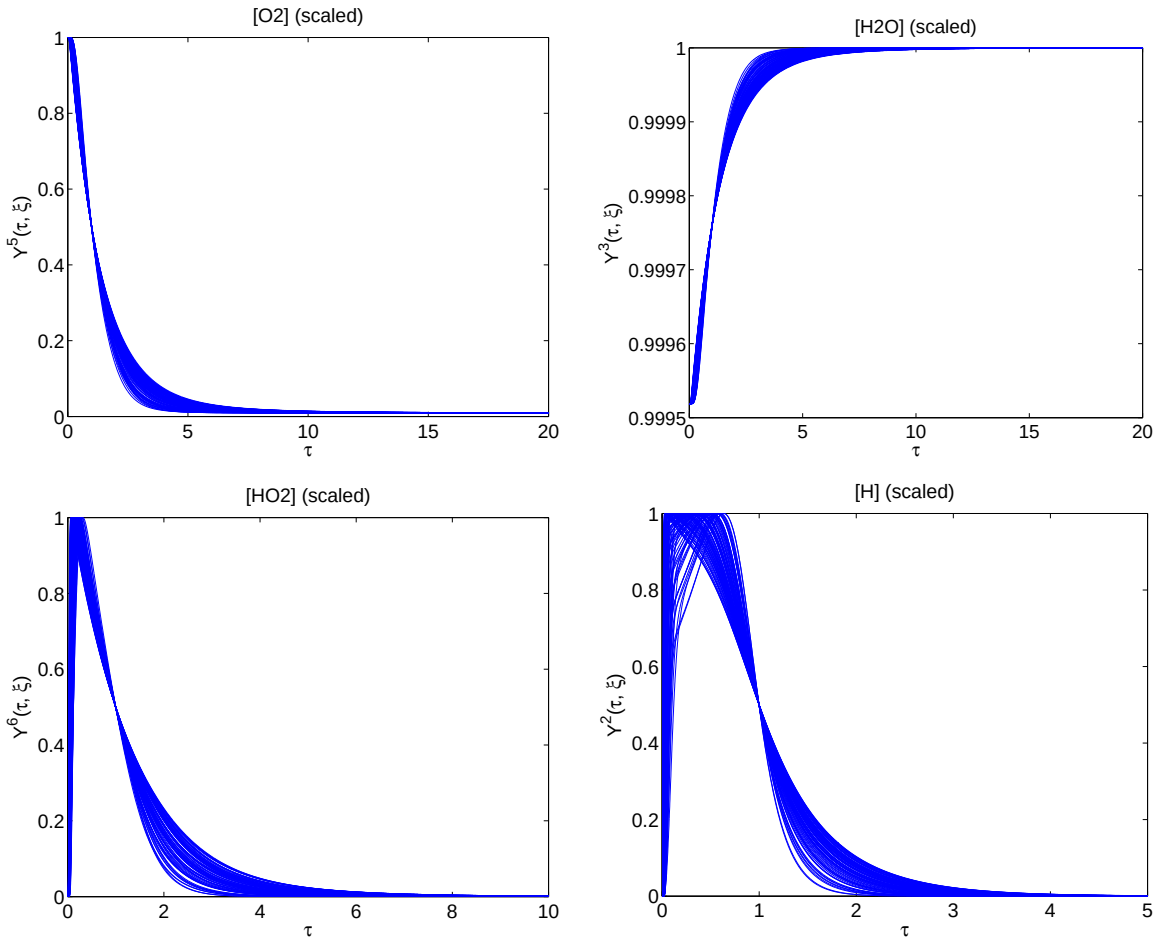

Figure 2: Scaled variables versus stretched time. Individual curves are obtained by transforming the original realizations shown in Figure 1.

5. Application of the Preconditioned NISP

In this section, the preconditioning strategy is applied to the hydrogen oxidation problem. We focus our discussion on the species whose dynamics were discussed in the previous section. First, we carefully examine the scaling factors $\hat{c}$ and $\hat{t}$ and demonstrate that they are well behaved and admit low order PC representations. Next, we show that the preconditioning effectively allows for low order PC representations in the transformed space, while still capturing complex features of the original variables through the recovery procedure. In particular, we show that the preconditioned NISP allows for recovery of complicated (possibly bimodal) distributions via low order PC expansions. We delay quantitative assessments of the method's accuracy to Section 6.

5.1. Expansion of the scaling factors

We start by projecting the amplitude and time scaling factors following the methodology introduced in section 3.2.1. Recall that we actually compute the expansions of the logarithms of the normalized scaling factors through (13) and (12). As mentioned in the previous section, the amplitude scaling factors $\hat{c}^3$, $\hat{c}^4$, and $\hat{c}^5$ are deterministic and hence we apply the above expansion only to the remaining (random) ones.

In Figure 3, we report the results for the projection of $\hat{c}^i$, $i = 2, 6, 7$ corresponding to the amplitude scaling factors for H, HO_2 and H_2O_2 respectively. The top row shows, for the three species, the expansion coefficients σ_k^i of the logarithms (as in (12)), for polynomial expansions up to third order. The coefficients are sorted in increasing (total) polynomial degree and the vertical dashed lines in the plots separate coefficients at successive orders $p = 1$, 2 and 3. Note that for $p = 3$ the expansion involves 165 terms. We first observe that the fixed order spectral bands are sparser as the corresponding p increases; the fraction of PC coefficients σ_k^i with significant magnitude decreases with p, indicating that the different random parameters in the model do not contribute equally to the variability in the species concentrations. This is consistent with our prior experiences in [25]. This observation also applies to the amplitude scaling factors, as expected.

Also note that the magnitudes of the PC coefficients decay with p, which reflects the convergence of the expansions. This is further examined by plotting in the second row of Figure 3 the normalized cumulative "energy" $E_k^2(\sigma^i)$ contained in the first k-th modes:

$$E_k^2(\sigma^i) \doteq \frac{\sum_{l=1}^{k} \left(\sigma_l^i\right)^2 \left\langle \Psi_l^2 \right\rangle}{\sum_{l=1}^{\infty} \left(\sigma_l^i\right)^2 \left\langle \Psi_l^2 \right\rangle}. \tag{20}$$

Because the denominator is unknown, it is replaced in practice with the energy of the highest order expansion, namely here $p = 3$. Plots of $E_k^2(\sigma^i)$ indicate that, for all the species shown, the first order expansion ($p = 1$) accounts for greater than 95% of the total energy, the second order expansion ($p = 2$) over 99%, indicating that the higher order terms have indeed negligible contributions.

Plots in the last row of Figure 3 present the probability density functions (pdfs) of the $\hat{c}^i$ obtained by Kernel-Density-Estimation through the sampling of (13) with third-order expansions of the logarithms ($p = 3$). The pdfs show that the approximate amplitude scaling factors have smooth unimodal distributions, similar to log-normal distributions with longer tails toward high values and sharper tails toward the lower amplitudes. The large coefficients of variation of these distributions have to be underlined since they directly reflect the sensitivity of the chemical system dynamics to the random parameters. The unimodal character of the pdfs also confirms the validity of the amplitude scaling factors definitions. Finally, the results for H and H_2O_2 highlight the advantage of projecting the logarithms, namely in preserving the positivity of $\hat{c}^i$.

Figure 4 shows the results of projecting the time scaling factors $\hat{t}^i$ for the same species as in Figure 3. Briefly, we observe again the decay and sparse character in the coefficients θ_k^i (first row). This is also reflected in the normalized cumulative energy plots for $E_k^2(\theta^i)$ (second row), which again show that a second-order expansion accounts more than 99% of the energy. Regarding the Kernel Density Estimations of the third order approximate of the random time scaling factors (last row), smooth log-normal like distributions, with large coefficients of variations, are again obtained. This lends confidence that the time scaling factors are suitably defined.

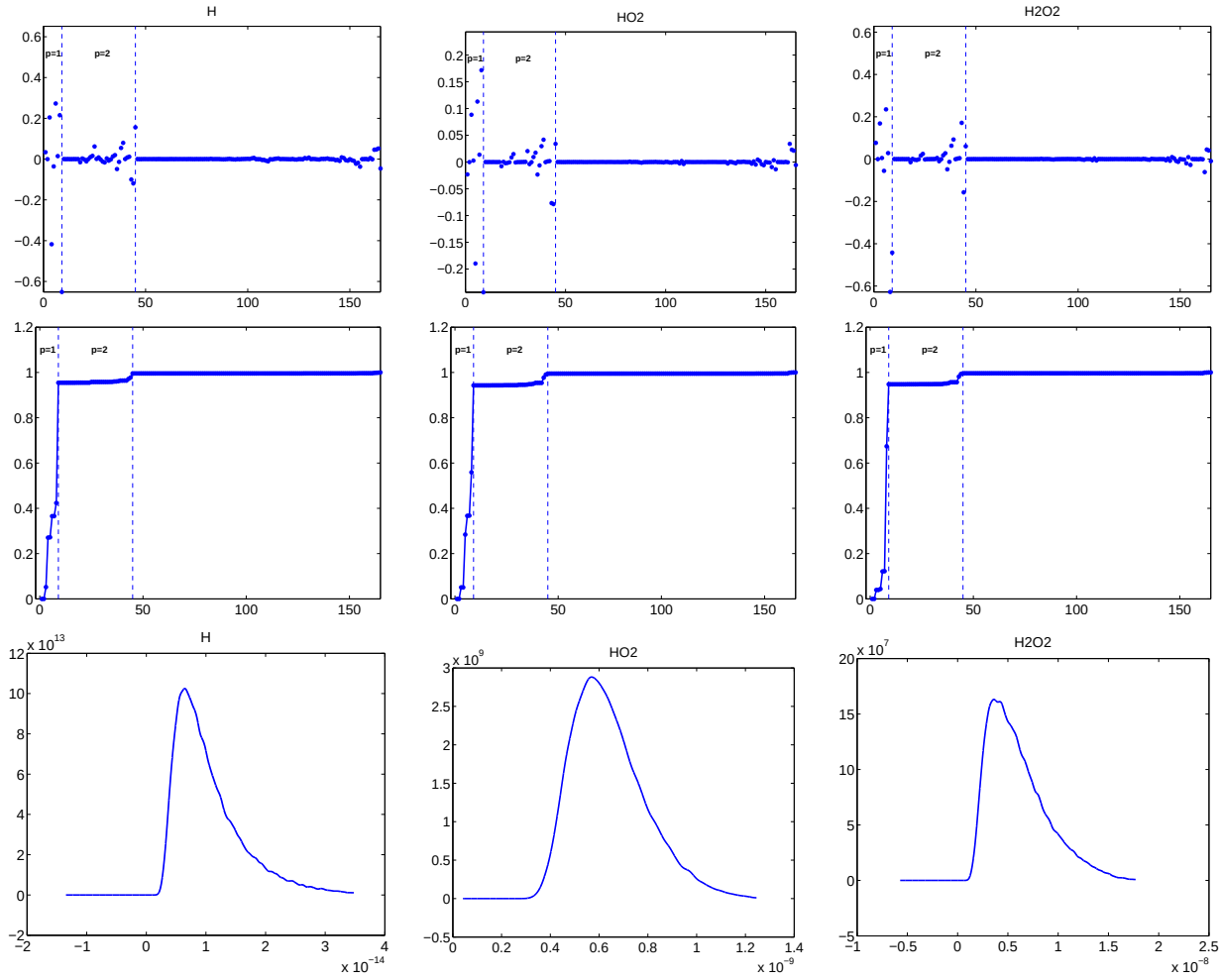

Figure 3: Amplitude scaling factor for H (left), HO_2 (center) and H_2O_2 (right). Top row: expansion coefficients σ_k^i of the logarithms (see (12)) sorted by increasing total degree. Second row: normalized cumulative energies $E_k^2(\sigma^i)$ as defined in (20). Last row: Kernel Density Estimation of the third order ($p = 3$) projection of the amplitude scaling factors.

5.2. Expansion of the scaled variables

We now analyze, in a qualitative fashion, the PC expansion of the scaled variables:

$$Y^i(\tau^i, \boldsymbol{\xi}) = \sum_{k=0}^{P} Y_k^i(\tau^i)\Psi_k(\boldsymbol{\xi}). \tag{21}$$

Specifically, we wish to examine whether the scaling transformation enables representation of the scaled variables using an expansion having fixed, moderate truncation order, and on the basis of a sample set $\mathcal{S}$ that is expected to be too coarse for a direct non-intrusive spectral projection.

Note that the values of $X^i(t, \boldsymbol{\xi})$ are known for all $\boldsymbol{\xi}$ in the NISP sample $\mathcal{S}$ and a fixed time grid. Specifically, assuming N time steps, and letting $\Delta t = T_{fin}/N$, available at our disposal are $X^i(t, \boldsymbol{\xi})$ for all $\boldsymbol{\xi}$ in $\mathcal{S}$ and all discrete times $t_n = n\Delta t$, $n = 1, \ldots, N$. Thus, when projecting the transformed variables Y^i in the scaled time we rely on linear interpolation to get values of X^i not in the time grid, as reported in Algorithm 2.

For the results presented in this section, we rely on third-order PC expansions, which in our case (since we have eight stochastic dimensions) involves 165 terms in (21). For brevity, attention is focused on the concentration of H which is representative of cases involving non-monotonic behavior, and the concentration of O_2 which is representative of all cases involving monotonic behavior.

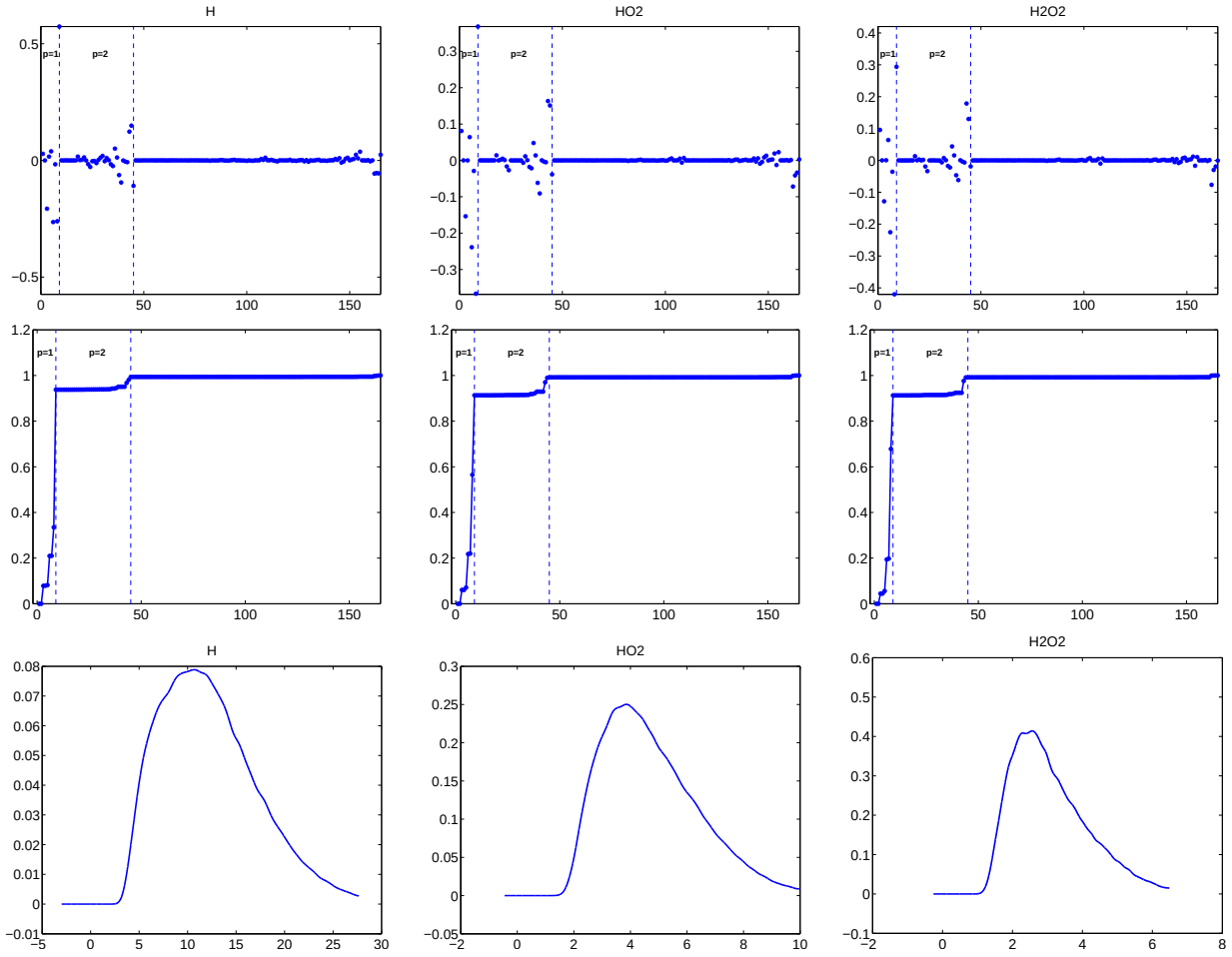

Figure 4: Time scaling factor for H (left), HO_2 (center) and H_2O_2 (right). Top row: expansion coefficients θ_k^i of the logarithms (see (12)) sorted by increasing total degree. Second row: normalized cumulative energies $E_k^2(\theta^i)$ as defined in (20). Last row: Kernel Density Estimation of the third order ($p = 3$) projection of the time scaling factors.

Figure 5 shows the evolution of the mean value of the scaled H and O_2 concentrations. Also plotted are dashed lines indicating the range of $\pm 2\sigma$ bounds, where σ is the local standard deviation. The latter is evaluated from the coefficients of the truncated PC expansion (21), according to:

$$\sigma^i(\tau^i) = \left(\sum_{k=1}^{P} \left[Y_k^i(\tau_i) \right]^2 \langle \Psi_k \rangle^2 \right)^{1/2}$$

Comparing the results in Figure 5 to the corresponding realization plotted in Figure 2, suggests that, though of moderate order, the PC expansion of the scaled variables is able to capture broad features of their evolution. In particular, the variance of the scaled solution, estimated from the PC expansion and visualized using dashed lines, appears to collapse at the stretched time $\tau = 1$.

Figures 6–7 provide PC spectra of the preconditioned H and O_2 concentrations at selected values of the stretched time. Plotted are mode amplitudes Y_k^i for $k = 0, \ldots, P$ for a third-order PC expansion. For O_2, the spectrum has rather simple structure, showing clear dominance of the mean mode over all others. Furthermore, as τ becomes large, the spectrum essentially degenerates, with modes Y_k^5, $k \geq 1$, becoming negligible compared to Y_0^5. This indicates that in the present case, the variability of the solution is essentially entirely absorbed by the stochastic transformation.

For the scaled H concentration, the trends are generally similar. Note that near $\tau = 1$, the spectrum essentially degenerates, as is expected based on the definition of the stochastic time scale. Unlike O_2, however,

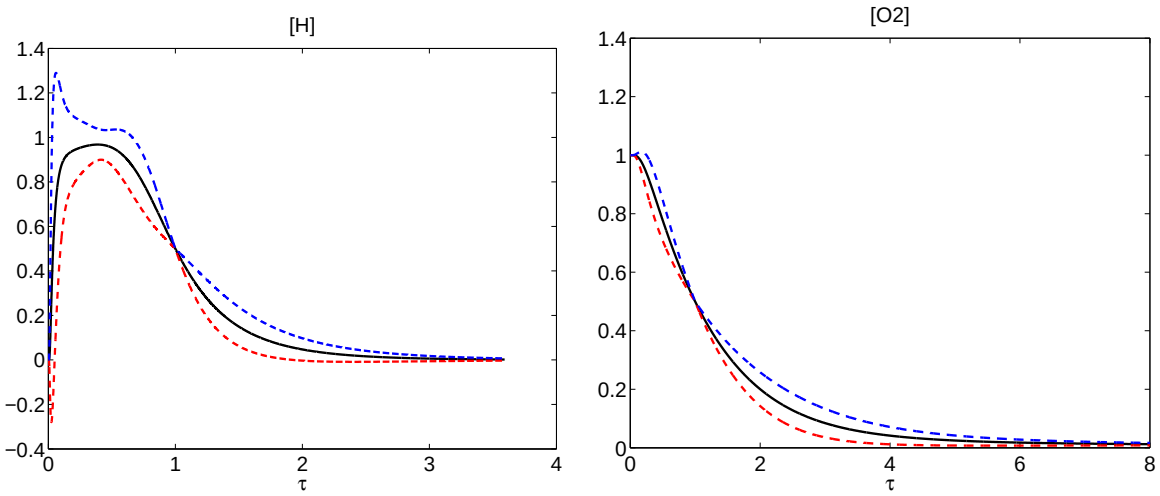

Figure 5: Expected value (solid line) of the scaled concentration $\langle Y^i \rangle$ versus the stretched time variable, τ. The dashed lines show curves of $\langle Y^i \rangle \pm 2\sigma$, where σ is the local standard deviation. Plotted are results for H and O_2, as indicated.

at larger τ values the spectra are richer, though the mode amplitudes become smaller as τ increases. Thus, while the global transformation simplifies the representation of the corresponding signal, the scaled solution retains a non-trivial structure due to the complexity of the corresponding response.

An additional qualitative check that the scaled H concentration is adequately captured by the truncated spectral expansion is provided in Figure 8, which shows PDFs of the scaled concentration at selected values of τ. The latter have been specifically chosen at times where the variance is large, see Figure 5. The results show that the PDFs have smooth behavior that lacks any peculiar feature. This also points to the possibility that a low order truncation may lead to a suitable representation.

Note finally that for both H and O_2, the spectra of the corresponding scaled components are sparse, exhibiting substantial bands where the PC coefficients have vanishing amplitudes. This illustrates the potential of preconditioning in leading to efficient or "compact" representations of possibly complex stochastic dynamics.

5.3. Recovering the original variables

As discussed earlier, once the PC representations of the scaled variables are available, the methodology of section 3 can be readily applied to recover the distributions of the concentrations in the unstretched time variable. Examples reconstructions are given in Figures 9–10, which respectively illustrate the distributions of $[H]$ and $[O_2]$, as recovered from the corresponding scaled variables at selected time instants.

The figures demonstrate that the PDFs of the original variables in the original time are much richer than those of the scaled variables in the stretched timescale. In particular, intricate features are observed, including extended tails, shoulders, and multiple peaks. This provides clear illustration that complex stochastic behavior may be recovered from relatively much simpler variables, provided suitable preconditioners are used.

Qualitatively, the reconstructed PDFs also appear to be consistent with the behavior of individual realizations (Figure 1), at least when one compares the ranges of the variables at corresponding times. A quantitative analysis of the adequacy of PDFs will be conducted in the following section, namely through comparison with Monte Carlo sampling.

The complexity of the reconstructed PDFs in the original time variable also suggests that direct NISP of the original variables onto a low-order basis would not lead to a suitable spectral representation. This would already be anticipated based on earlier experiences using a an intrusive, multiresolution Galerkin formalism [25]. In the following section, experiments are conducted that demonstrate direct NISP of the same realizations onto low-order bases actually results in large errors and generally unsuitable representations.

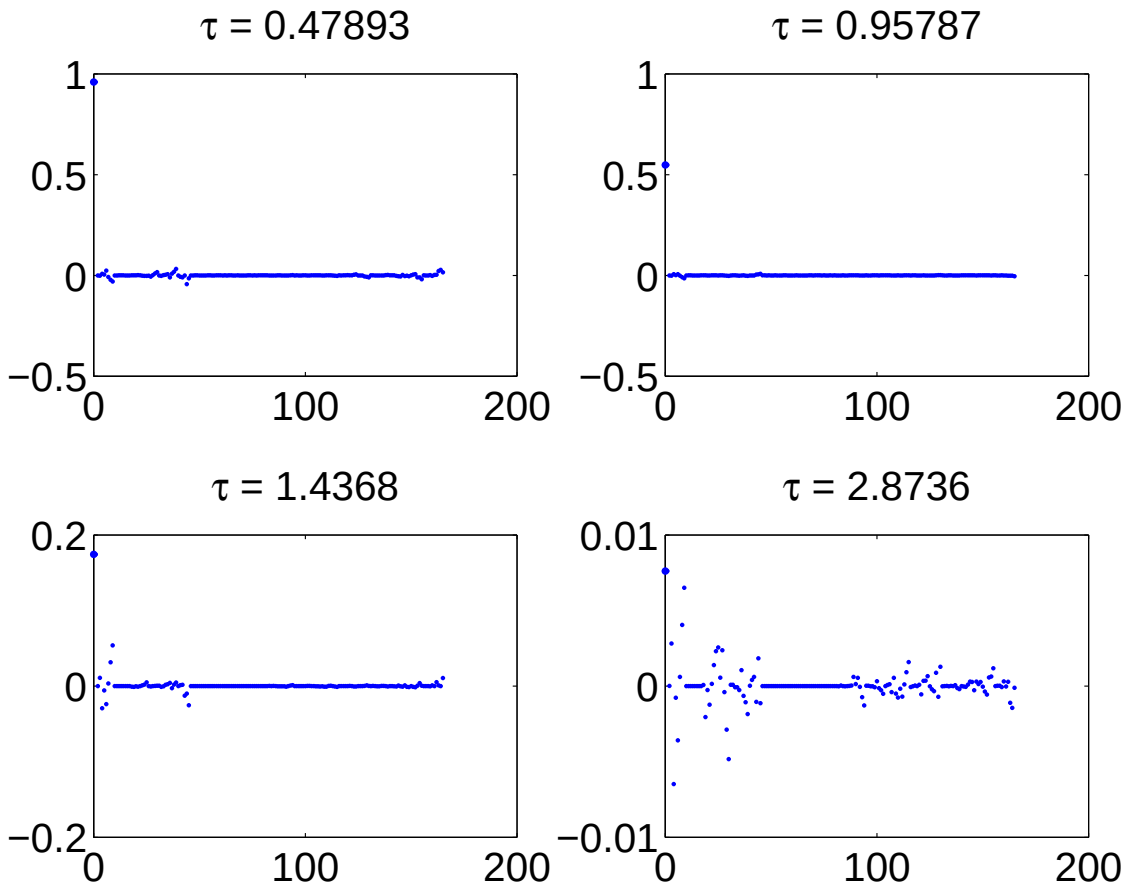

Figure 6: Spectra of the preconditioned H concentration, Y^2, generated at selected values of the stretched time, as indicated.

6. Convergence study

In this section, we assess the advantages achieved by preconditioning, particularly by contrasting the performance of preconditioned and conventional NISP.

6.1. Error estimates

We denote $\|\cdot\|_{L^2(\Omega^*)}$ the L^2-norm in the image space $(\Omega^*, \mathcal{B}(\Omega^*), F_{\boldsymbol{\xi}})$ (see section 2):

$$\|X(t,\cdot)\|_{L^2(\Omega^*)} = \left(X(t,\cdot), X(t,\cdot)\right)^{1/2}.$$

For $t \geq 0$, we define the normalized approximation error on $X(t,\cdot)$, denoted by $\varepsilon(t)$, according to:

$$\varepsilon(t) = \frac{\left\|X(t,\cdot) - \tilde{X}(t,\cdot)\right\|_{L^2(\Omega^*)}}{\|X(t,\cdot)\|_{L^2(\Omega^*)}}, \tag{22}$$

where, following the notations of section 3.3, $\tilde{X}$ is the recovered variable that approximates X. Obviously, since the exact solution $X(t,\cdot)$ is not known we shall instead rely on Monte Carlo samples to estimate ε. Specifically, denoting S a sample set of $\boldsymbol{\xi} \in \Omega^*$ drawn from the distribution $F_{\boldsymbol{\xi}}$, and letting $|S|$ denote the size of the sample set S, we shall make use of

$$\varepsilon(t) = \lim_{|S|\to\infty} \left[\frac{\sum_{\boldsymbol{\xi}\in S} \left(X(t,\boldsymbol{\xi}) - \tilde{X}(t,\boldsymbol{\xi})\right)^2}{\sum_{\boldsymbol{\xi}\in S} \left(X(t,\boldsymbol{\xi})\right)^2} \right]^{1/2}, \tag{23}$$

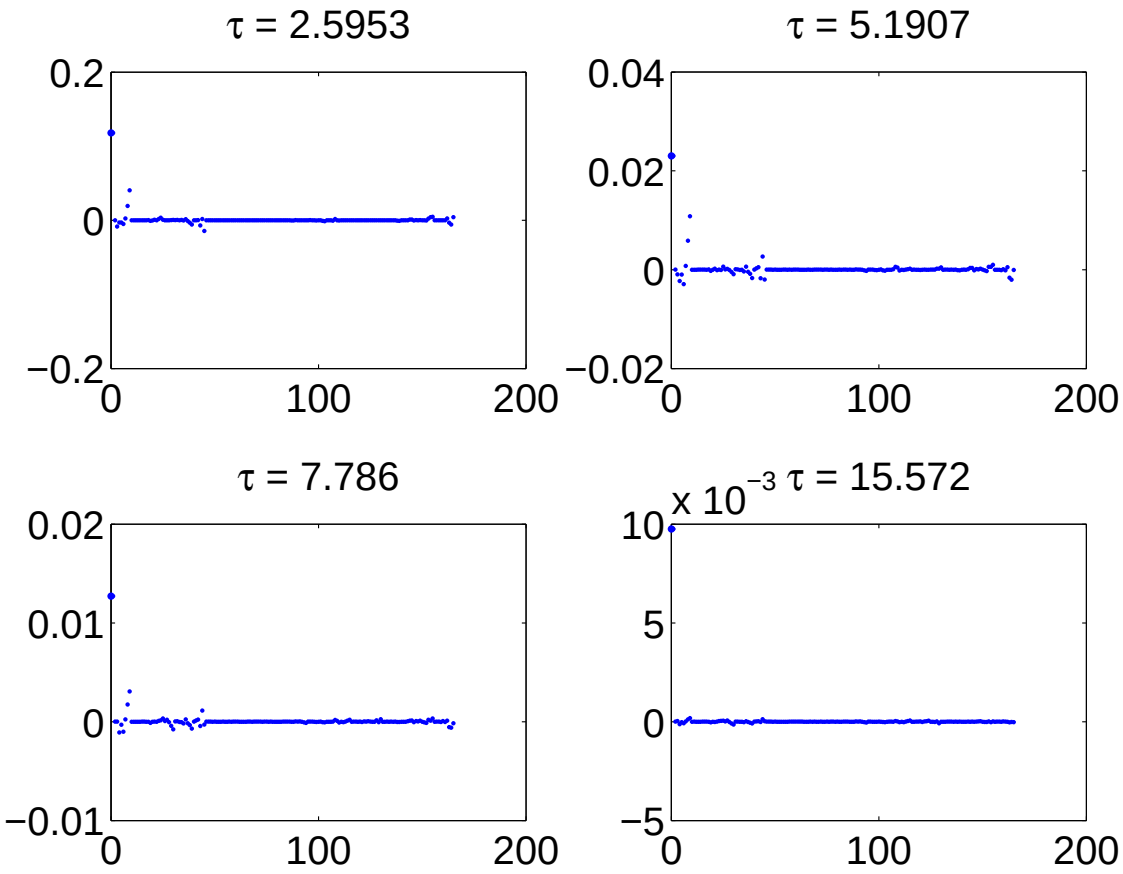

Figure 7: Spectra of the preconditioned O_2 concentration, Y^5, generated at selected values of the stretched time, as indicated.

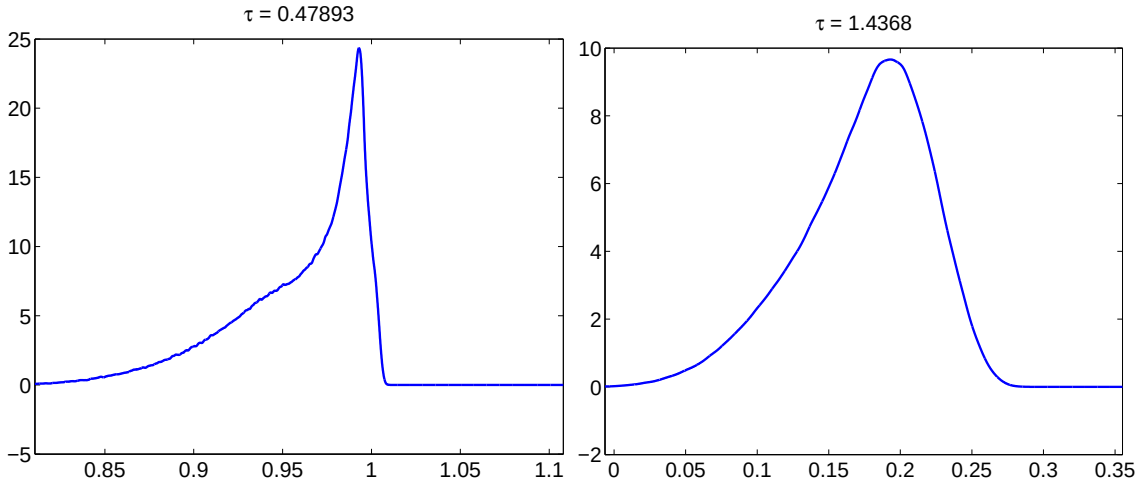

Figure 8: PDFs of the preconditioned solution generated at selected values of the stretched time, as indicated. Shown are results corresponding to H concentration (i.e. Y^2).

to estimate $\varepsilon(t)$ from finite sample sets S. This error measurement will serve to quantify the approximation errors associated to the different types of preconditioning strategies (linear / log-linear projection, Option 1 / Option 2 in Algorithm 2). Also of interest will be the convergence of $\varepsilon(t)$ as the expansion order p increases, and comparison of the errors with the case of NISP without preconditioning, that is when $\hat{c} = \hat{t} = 1$. In

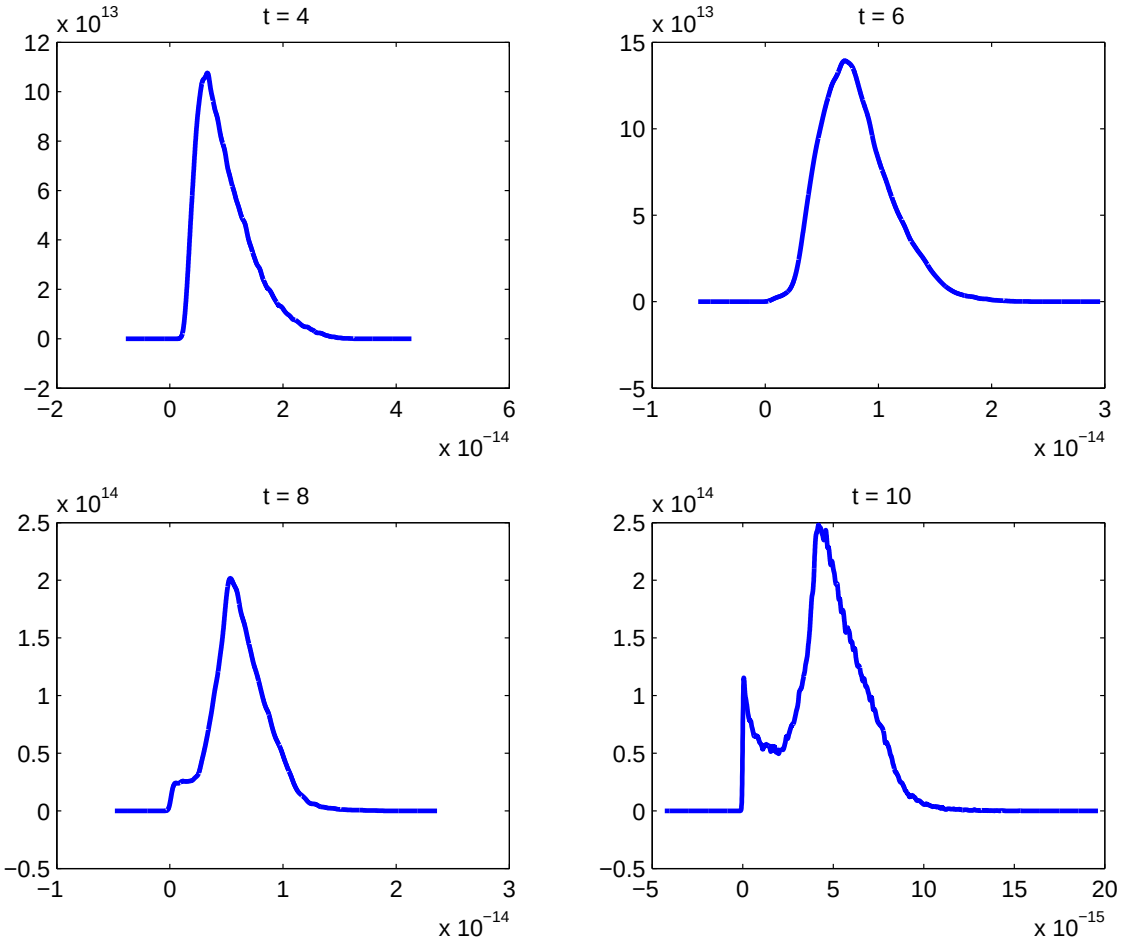

Figure 9: PDF of $[H](t, \boldsymbol{\xi})$ at selected values of time, t, as indicated.

addition, because all species with index $i = 2, \ldots, 7$ exhibit similar error dependencies with regard to the expansion order, we only report measurements for the case of $X^{i=2} = [H]$, which has the most complex behavior. Also, for convenience, we drop the superscript 2 from X^2 and its transform Y^2 and simply refer to them as X and Y throughout this section.

6.2. Error analysis

In a first series of tests, we monitor the convergence of the error estimate (23) for increasingly large Monte-Carlo sample set. A typical illustration of the convergence of $\varepsilon(t)$ as $|S|$ increases is reported in Figure 11. The plot shows that ε is fairly well estimated using a sample set with $|S| = 50,000$, a value that is used in all subsequent computations.

Next, we compare in Figure 12 the errors in the recovered concentration of H at different unscaled times. The left plot corresponds to the case of the recovery from the preconditioned projections Algorithm 2 using Option-1 for the definition of the factors in the transformation, while the right plot corresponds to Option 2. In both cases, results are shown for truncated PC expansions with order $p = 1$, 2, and 3. We first remark that the two options used in the projection algorithm yield comparable error levels. This demonstrates the robustness of the proposed projection method, which is credited to the well-conditioned scale factors, $\hat{c}$ and $\hat{t}$, or more precisely their logarithms. In particular, the latter are accurately approximated using low order expansions such that using either their actual (point-value) or PC approximates for the projection yields insignificant differences at the recovery stage (which we recall uses the PC approximation in the inversion of the transformation). This is an interesting finding as it indicates the possibility of deriving efficient "on the fly" strategies for the preconditioned projection, without having to rely on predetermination of the factors

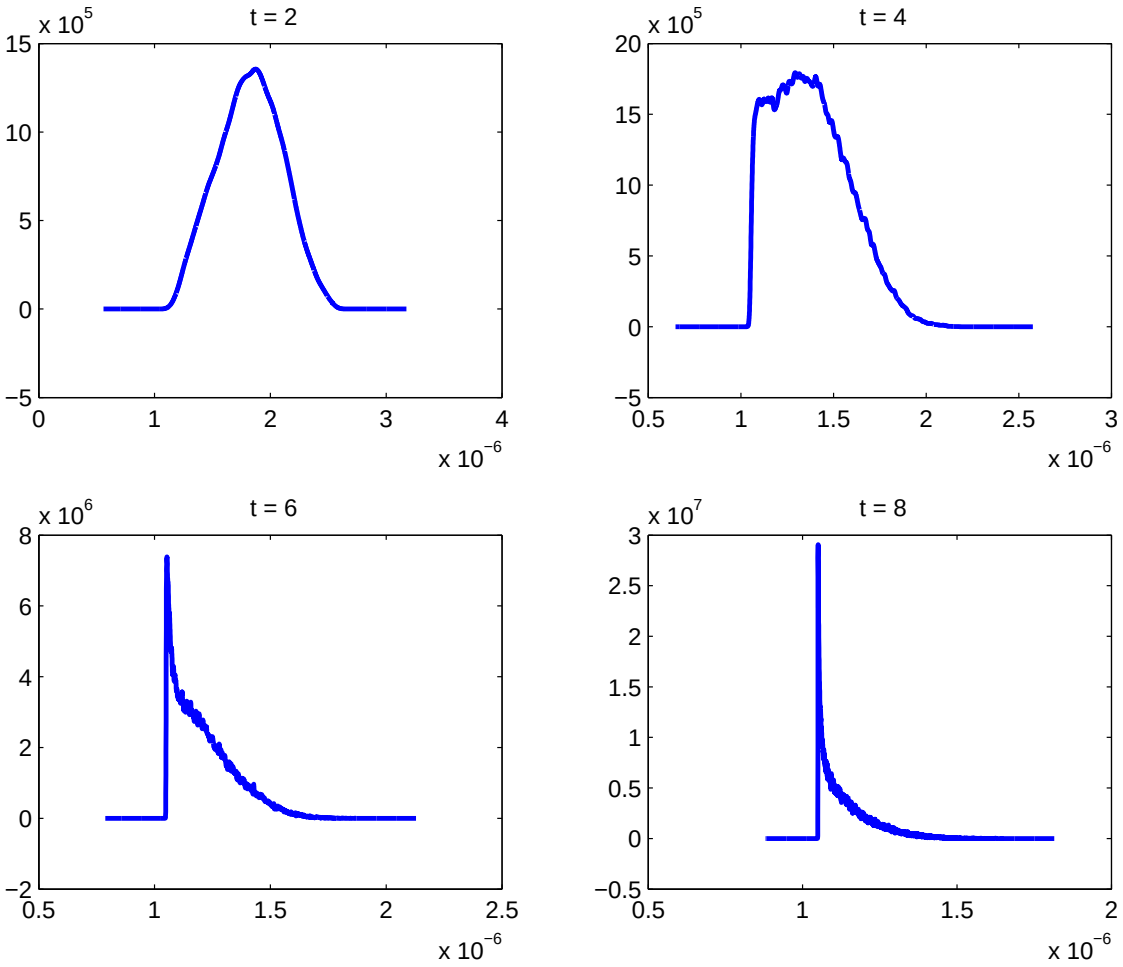

Figure 10: PDF of $[O_2](t, \boldsymbol{\xi})$ at selected values of time, t, as indicated.

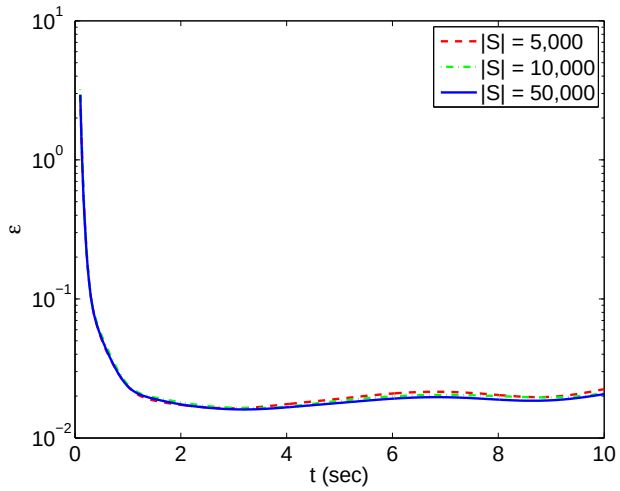

Figure 11: Dependence of $\varepsilon(t)$ (23) on the size of the size of Monte Carlo sample set, $|S|$. The recovered solution is obtained from a the third-order projection of the scaled variable Y^2 through Algorithm 2 with Option 2.

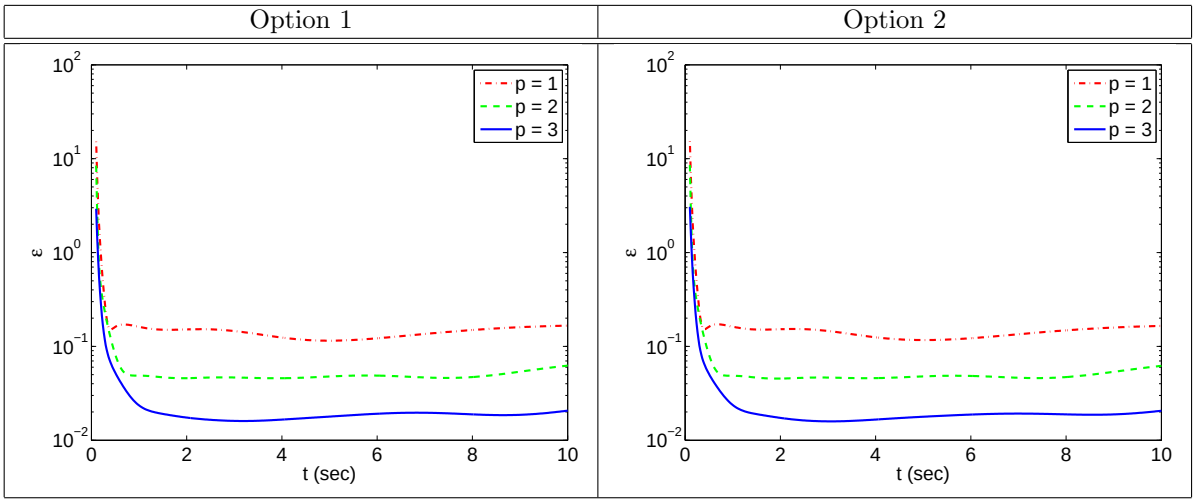

Figure 12: Normalized errors $\varepsilon(t)$ for the recovered concentration $[H]$ using the preconditioned projection with Option 1 (left) and Option 2 (right). Plotted are curves for different PC orders, $p = 1$, 2 and 3, as indicated. The Monte-Carlo estimate of ε in (23) uses $|S| = 50,000$.

before realizing the actual projection. This characteristic is of particular importance in view of adaptive strategies, yet to be deployed, where the NISP sample set would be progressively completed to effectively reduce the projection error.

The second important observation arising from inspection of Figure 12 is the reduction of the error when p increases. For $t > 1$ the error is roughly divided by two each time the expansion order increases by one unit. However, one certainly notices the large error level for all orders tested and at early times, $t < 1/2$. In fact, for $t < 1/4$ the normalized error $\varepsilon(t)$ is even larger than 1. This is mainly caused by the normalization in (22), which magnifies errors at early time because $\|X(t = 0, \cdot)\|_{L^2(\Omega^*)} = 0$, and is also due to the definition of the transformation. In fact, we have defined the scale factors from considerations of the global dynamics of the random system; consequently one can only expect the transformation to be effective *globally*, but not necessarily at early times when the system experiences dynamics with characteristic time-scales much different than $\hat{t}$. (See for instance the plot for the scaled $[H]$ in Figure 2.)

The large error level at early time can be reduced to some extent by projecting the logarithm of the scaled variable Y, thereby ensuring the positivity of the of the recovered variable $\tilde{X}$. This can be seen from Figure 13, which depicts the resulting errors in the recovered variable for the different PC orders (Option 2 is used in the projection algorithm). In the early stages, the error levels resulting from the projecting $\log Y$ are smaller than those observed when Y is directly projected; $\varepsilon(t)$ remains large, however. This observation confirms the limitation of the global transformation and motivates further developments, including implementation of local transformations, or transformations over finite horizons.

Figure 13 also provides the estimated errors for the direct NISP projection of $X(t, \cdot)$, *i.e.* without preconditioning or, equivalently, using $\hat{c} = \hat{t} = 1$ in the definition of the transformation which then reduces trivially to the identity. We observe that for all the orders tested, the preconditioned NISP always provides recovered variables with higher accuracy than the direct NISP projection. Actually, for $p = 3$ the achieved error reduction can be roughly one order of magnitude. Note the improvement brought by the preconditioning varies with time, being lower at early times than at later stages, reflecting again the potential for local time transformations. In addition, the reported results show an apparent larger convergence rate with p of the preconditioned projection compared to the direct one. Combined, the observations of substantially smaller error levels and faster error decay rate provide strong motivation for the potential of preconditioned PC.

6.3. Convergence in distribution

To better appreciate the improvement brought by the preconditioning, we provide in Figure 14 a comparison of the probability density functions of $[H]$ at time $t = 8$, obtained with preconditioned and direct (unpreconditioned) NISP with $p = 1$, 2 and 3. The left plot shows the PDFs of the recovered variable for the preconditioned NISP, based on Algorithm 2 with PC expansion of the logarithm of the scaled variable

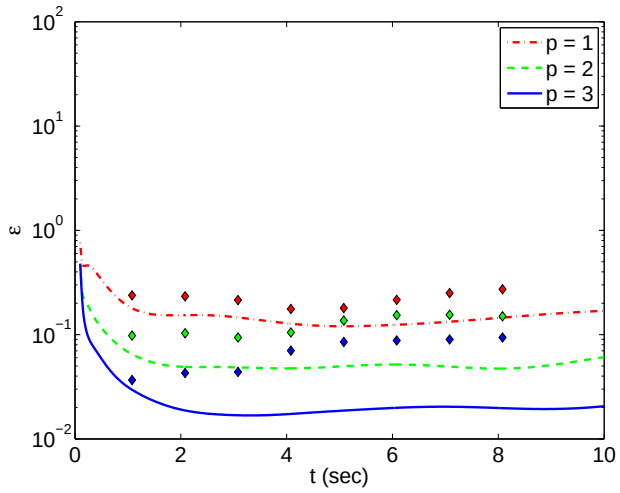

Figure 13: Normalized error $\varepsilon(t)$ of the recovered concentration of $[H]$, based on projecting $\log Y$ with Option 2 (lines) and direct spectral projection of X (symbols). Results are shown for PC expansion orders $p = 1$, 2 and 3, as indicated.

(log-linear scaling) using Option 2. The right plot shows the PDFs for the case of the direct NISP projection along with a PDF obtained with direct Monte Carlo sampling used as surrogate for the "exact" PDF. All curves are generated using the same number of samples and Kernel Density Estimation (KDE) procedure [5].

Focusing first on the case of the preconditioned NISP, we remark that the PDFs for $p = 2$ and $p = 3$ are essentially the same, while for $p = 1$ significant differences are observed. In particular, with $p = 1$ the maximum of the density seems under-estimated and the plateau for $[H] \approx 0$ is not evident. Comparison with the exact PDF shown in the right plot (curve labeled Monte Carlo) highlights the quality of the preconditioned projections at $p = 2$ and 3. In fact, even a second order expansion provides a reasonably accurate approximation.

The fast convergence of preconditioned NISP is to be contrasted with the results of direct projection, which for the three PC orders tested remain far off the exact (Monte Carlo) result. Comparing preconditioned and direct NISP, with $p = 1$ we observe that while both PDFs significantly depart from the exact PDF, preconditioned NISP is already providing a fairly better approximation. Increasing the order to $p = 2$ and 3, the convergence of the direct projection appears much slower in the case of the direct projection. In particular, at $p = 3$ the direct projection is still unable to capture the plateau at the lowest concentration values, while it also suffers from long tails that extend into a range of negative concentrations. Thus, the present observations clearly highlight some of the advantages of preconditioning.

Remark 6.1 As noted earlier, in the present early application of stochastic preconditioners, we have relied on a fully tensorized Gauss quadrature to generate the NISP sample. No attempt has been made to optimize the number of realizations in the NISP sample, though, as shown earlier, the selected quadrature is not suitable for a straightforward spectral projection.

It is interesting to note that with a Monte-Carlo approach, a sample size of $50,000$ was found sufficient to approximate the PDF of the solution. This is significantly smaller than the NISP sample that includes $5^8 = 390,625$ realizations. This may be misleading since it may suggest that the stochastic preconditioning approach is less efficient than MC. We should point out, however, that in practice sparse quadratures are frequently used, which can dramatically reduce the number of NISP samples. This would be feasible in the present case, because the stochastic preconditioning approach enables the use of low-order expansions. By means of example, using SMOLPACK [34] with a level 6 sparse quadrature on the current eight-dimensional problem would lead to a NISP sample having on the order of $6,000$ realizations. This would clearly support the low-order expansions afforded by stochastic preconditioning, with a number of realizations about one order of magnitude smaller than with MC, and nearly two orders of magnitude smaller than the present fully tensorized Gauss quadrature.

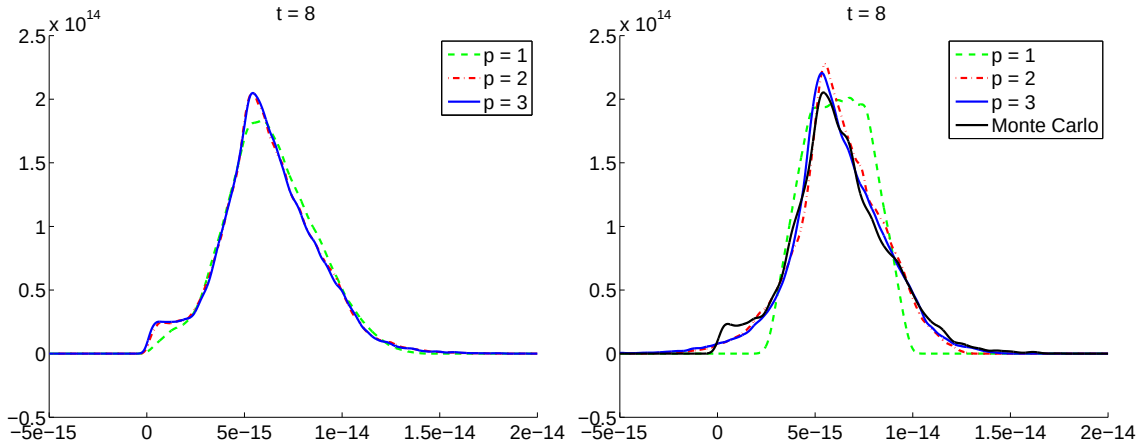

Figure 14: PDFs of $[H]$ at time $t = 8$. Left: preconditioned NISP at different PC orders as indicated. Right: direct NISP method with the same orders. Also shown on the right is the PDF of $[H]$ generated with Monte-Carlo sampling.

7. Generalized preconditioners

So far we have demonstrated that preconditioning of realizations prior to projection potentially allows for a significant reduction in order of the PC representation and thus of computational efforts. This was demonstrated for 6 of the 7 species present in the hydrogen oxidation problem, for which simple scaling transformations were applied. It is clear that the complexity reduction achieved by the preconditioned projection depends on the appropriateness of the underlying transformations. In particular, we have seen that the transformation has to incorporate two essential features: it should ensure a) the proper definition of the scaling factors for all realizations and b) yield PC representation with rapid spectral decay at all times of interest for the transformed variable. It is then legitimate to ask whether any dynamical system possesses a transformation satisfying those two requirements. Clearly, as far as simple scaling transformations are considered, the answer is negative and the case of OH, discussed below, is an example of such breakdown. This fact does not, however, imply that the preconditioning has limited scope, but merely that one should be careful when choosing the transformations on which the preconditioning would be based. In the following, we illustrate these issues in light of the behavior for the stochastic OH concentration.

7.1. Random dynamics of OH

Figure 15 shows individual realizations of $X^1(t, \boldsymbol{\xi}) = [OH](t, \boldsymbol{\xi})$. We observe that for most of the realizations shown the dynamics are non-monotonic; the concentration grows, reaches a maximum, and subsequently decays to its asymptotic value. For these realizations, the dynamics are fairly similar to the case of HO_2 and H shown in Figure 1, suggesting the use of a similar scaling transformations with scaling factors defined as in Table 2. However, a closer inspection of Figure 15(a) reveals that some realizations have different dynamics, specifically, a fast initial rise of the concentration followed by a very slow evolution toward the equilibrium value. For these realizations, the appropriate definition of the scaling factors is more likely to be that of monotonically increasing dynamics.

As a result of these different dynamics, the transformations used for the other species are not amenable to the OH case. The previous transformation for non-monotonic dynamics cannot be used as some realizations do not exhibit an intermediate maximum. On the other hand, the transform designed previously for monotonically increasing concentration is not suited to non-monotonic realizations and yields an ineffective preconditioning.

One can think of two alternatives to address this issue. First, one could use different transformation types, depending on $\boldsymbol{\xi}$; this corresponds to the general idea of local approximation based on the partitioning of Ω^*, as used for instance in adaptive intrusive methods. This idea, although attractive, will not be pursued here because of the need for efficient strategies to construct the partition of the random parameter domain. Alternatively, one can think of more general transformations having almost surely well defined scaling factors

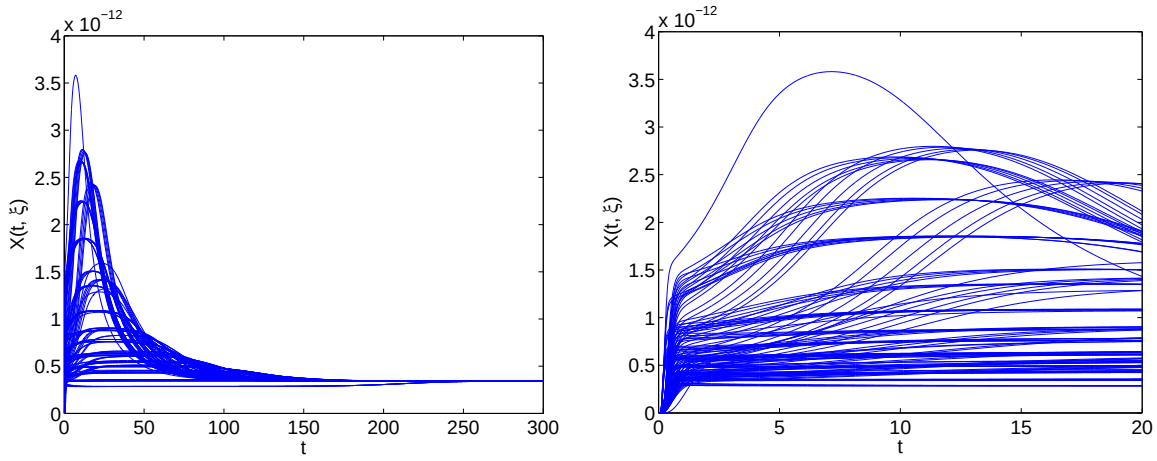

Figure 15: Individual realizations of $X^1(t,\boldsymbol{\xi})$: (a) long time behavior, $0 \le t \le 300$, (b) short time behavior, $0 \le t \le 20$. Note that in order to enhance the appearance of the plots, not all realizations are shown.

and yielding transformed variables with tight PC spectra. Being more aligned with the preconditioning of the other species, this latter alternative is followed below.

Since this section is only concerned with $X^1 = [OH]$, we drop in the following the superscript in X^1 and Y^1, and simply use X and Y.

7.2. Stochastic transformation for OH

The objective is to define a transformation such that the random dynamics remain essentially similar (in phase) for almost every $\boldsymbol{\xi} \in \Omega^*$. We address this issue by considering an *affine* scaling transformation as follows. We begin by defining a *deterministic* reference dynamics, X^{ref} through

$$X^{ref}(t) = X^{\infty}\big(1 - \exp(-\lambda t)\big), \quad \lambda = 1/100,$$

with X^{∞} the deterministic steady state OH concentration (uniquely determined by deterministic equilibrium constants and deterministic initial concentrations). This reference ensures that $X(t,\boldsymbol{\xi}) - X^{ref}(t)$ has almost surely a non-monotonic dynamics consisting in an increasing stage, followed by a decaying stage: $X(t,\boldsymbol{\xi}) - X^{ref}(t)$ present almost surely a local maximum. This, in turn, ensures well-defined scaling factors $\hat{c}$ and $\hat{t}$ as

$$\begin{cases} \hat{c}(\boldsymbol{\xi}) = \max\limits_{t} \big(X(t,\boldsymbol{\xi}) - X^{ref}(t)\big), \\ \hat{t}(\boldsymbol{\xi}) = \text{ time taken for } \big(X(t,\boldsymbol{\xi}) - X^{ref}(t)\big) \text{ to return to } \frac{1}{2}\hat{c}(\boldsymbol{\xi}). \end{cases}$$

With these scaling factors, we define the transformation Φ according to

$$\Phi\big[X(t,\boldsymbol{\xi}); \hat{c}\big] = \frac{1}{\hat{c}(\boldsymbol{\xi})}\big(X(t,\boldsymbol{\xi}) - X^{ref}(t)\big), \tag{24}$$

and consider the scaled variable $Y\big(\tau(t,\boldsymbol{\xi}),\boldsymbol{\xi}\big) = \Phi\big[X(t,\boldsymbol{\xi})\big]$ where $\tau(t,\boldsymbol{\xi}) = t/\hat{t}(\boldsymbol{\xi})$.

Figure 16(a) shows scaled realizations, Y, corresponding to the same selection in Figure 15. We see that the affine scaling "synchronizes" the realizations which all experience the same type of dynamics. The variability in the scaled variables, at fixed scaled-time τ, can be appreciated from Figure 16(b). We observe that the relative variability is well controlled except at small values τ, where large variation is evident. As further discussed in Section 8, this is due to our use of a *global scaling* strategy which is not expected to generally address complex behavior involving a wide range of timescales.

7.3. Projection of the scaled variable and recovery

Using the definition of the transformation in (24), the projection of the scaled variable can proceed following Algorithm 2. We briefly examine the suitability of the preconditioning by analyzing the distribution

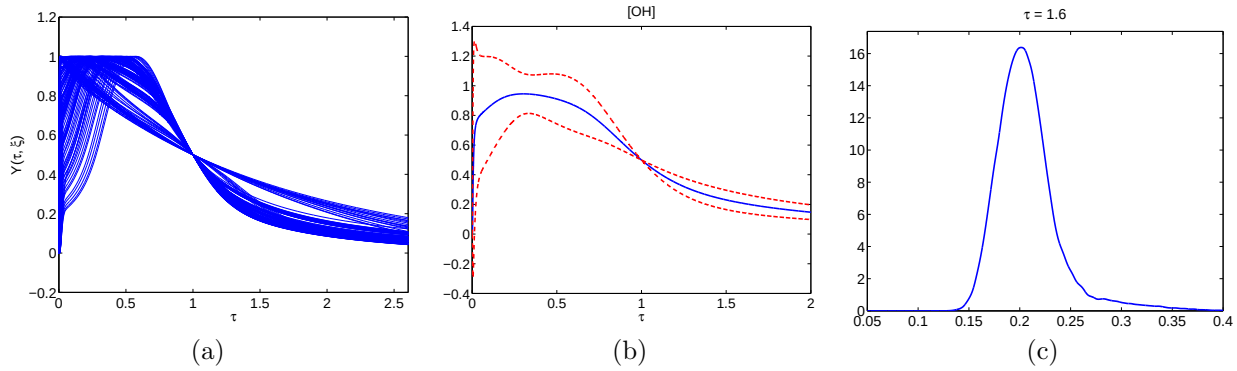

Figure 16: (a) Selected realizations of the scaled variable $Y(\tau, \boldsymbol{\xi})$; (b) mean of the scaled variable Y with ± 2 standard deviation bounds, versus the scaled time τ; (c) PDF of Y at $\tau = 1.6$.

of the recovered solution $\tilde{X}$. A sample of this exercise is shown in Figure 17, where we compare PDFs of the reconstructed solution, for different order expansions, with a PDF obtained through Monte Carlo sampling. Intermediate times, $t = 14$ and $t = 18$, are selected at which the unscaled realizations exhibit a wide range of variation (Fig. 15). The results indicate close agreement between all estimates provided, indicating both the transformation and reconstruction are performing adequately, at least for the intermediate times selected.

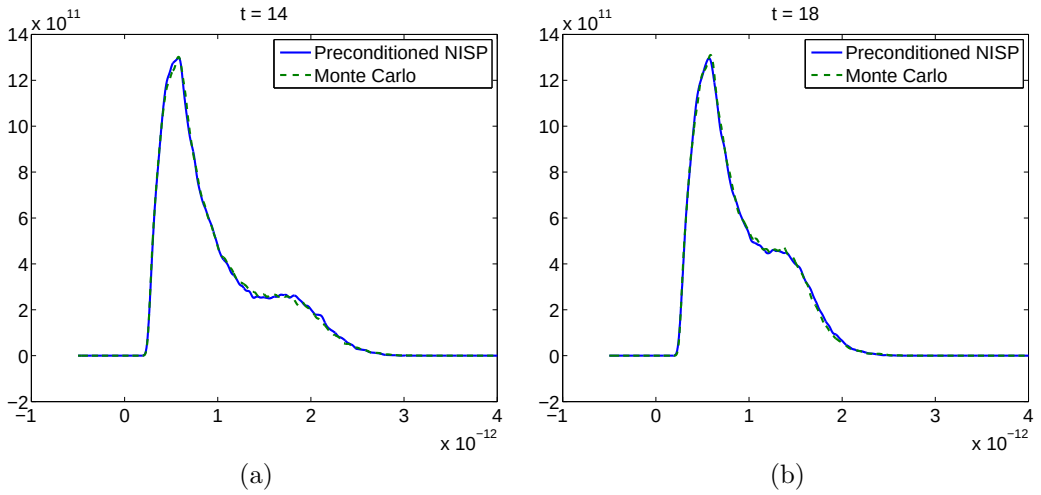

Figure 17: PDFs of the recovered state variable OH compared with its true distribution obtained via Monte Carlo simulations: (a) at $t = 14$ and (b) at $t = 18$.

One should note, however, that in the present case where complex dynamics are observed, the preconditioner is not optimal, and not as effective as in the examples of the previous section. An example concerns the the fact that the scaled realizations (Fig. 16(a)) do not exhibit as sharp a collapse as for the remaining concentrations. Another example concerns the PDF of Y provided in Figure 16(c), which exhibit a well defined shoulder around $Y = 0.27$, followed by an extended tail. Thus, the scaling in the present case not as effective in simplifying the distribution of Y as in the examples of the previous section.

8. Summary and discussion

This paper has focused on exploring the potential of a non-intrusive stochastic preconditioning formalism to enable efficient low-order PC representation of a complex stochastic dynamical system. The approach is illustrated for the case of a stiff oxidation problem with large uncertainties in rate parameters. The system had been known to exhibit a complex dependence on the stochastic rate parameters leading to large variations in the response of individual species and in the corresponding response timescales.

A simplified, multiscale, preconditioning approach was considered that is based on judiciously exploiting a coarse computational database of individual realizations. The database is used to define appropriate characteristic time and concentration scales for each species and each of the realizations. A NISP approach is then used to express these *global* characteristic scales in terms of appropriate, low-order PC expansions, and the latter are used to scale individual realizations. NISP is also used to project the resulting stretched realizations onto the PC basis. This leads to a reprensentation of the response of the stochastic system in terms of an appropriate combination of the PC expansions for the preconditioners and for the stretched variables. Computational experiments conducted using a coarse computed database showed that the preconditioning approach can lead to low-order PC representations with rapid spectral decay, and that the latter may be efficiently sampled to recover the distribution of the original variables in the physical timescale.

Despite their limited scope, the present experiences naturally lead to a number of basic observations, and motivate further extensions. We conclude this paper with listing a few of these:

1. The computations and analyses discussed above relied heavily on the availability of a computed database of model realizations, which in particular enabled consideration of different scalings and projection approaches, and systematic study of their associated properties. For the present problem, construction of the realization database was relatively straightforward, requiring moderate amount of storage and a small CPU overhead. Extension of the present non-intrusive methodology to larger-scale would naturally benefit from the availability of large databases, and of efficient schemes for storing, indexing, retrieving, interpolating and otherwise transforming individual realization data.
2. The implementations above relied on simple, linear or affine, scaling transformations in order to (approximately) synchronize between different realizations and consequently define scaled variables having variances that are significantly smaller than those of the unscaled variables. Once these transformations were selected, the database could be exploited in an automatic fashion in order to determine the stretching parameters suitable for each realization, and consequently conduct the associated NISP. In more general settings, it may be worthwhile to consider larger families of transformations, and to determine "optimal" sets of stretching parameters. Thus, it may be natural to approach the scaling of realizations as an optimization problem, constrained possibly by the requirement that the PC representation of the optimal transformation be an invertible stochastic quantity. This would in particular ensure that the original variables can be recovered in the physical time domain. By building libraries of suitable transformations, such optimization approaches would offer the promise of a fully automated process for determining optimal preconditioners.
3. In the present implementations, a simplified approach was adopted for the purpose of representing the stochastic preconditioners and the preconditioned solution, i.e. the scaled variables. Specifically, for both quantities, PC expansions were used that have the same truncation order. This is evidently not an essential requirement of the present methodology, as expansions of different truncation order may be readily used for the preconditioner and for the preconditioned solution. Though not considered in the present work, such mixed order approaches may provide additional means of controlling the properties of the overall representation, particularly to ensure that the latter can be well represented on the underlying grid of stochastic realizations. Another possibility that has not been considered here, and that may be worthwhile to explore, concerns the use of mixed basis representations for the preconditioner and the preconditioned solution. Depending on the nature of the preconditioner, such mixed representations may appear akin to enriched basis concepts that have been recently advanced for PC representations [13]. Specifically, a formal link between stochastic preconditioning and enriched basis representations may be made evident if, instead of the usual, additive, enriched basis construction, one relies on a factorized construction in which the enriching functions appears as factors of a PC expansion. The resulting representations would evidently share many similarities with the preconditioning constructs, though they would not automatically incorporate multiscale time stretching.
4. A simplified approach to preconditioning the stochastic solution was adopted that was based on the introduction of *global* time scales for individual components of the solution. While this simplified approach appeared to be suitable for the examples treated, it should be emphasized that the resulting representation may not be necessarily optimal at all times. For instance, in both early stages of the ignition process and at very large time, a highly efficient local description of the stochastic solution may evidently be sought in terms of the Jacobian of the stochastic solution. Consequently, it may appear advantageous to seek extensions of the present formalism, in which preconditioned representations are

localized over finite time intervals. Another avenue consists of exploring links between global timescales derived from observations of specific realizations, and local stochastic timescales, derived for instance from computational singular perturbation [18, 19, 20, 39] analysis of the stochastic dynamical system. For instance, the local timescales may be suitably averaged and consequently used as means to construct or refine global transformations. Alternatively, local timescales may be exploited to adaptively define and construct local representations.

5. An additional possibility would involve combining (locally or globally) preconditioned representations with adaptive representation concepts in the space of random parameters. This subject has attracted much recent interest, and several approaches have been proposed, including multi-resolution representations [22, 24, 25], domain decomposition methods [24], and sparse adaptive quadratures [11, 33, 34]. The potential incorporation of preconditioned representations into such constructions is currently being pursued.
6. In most of the computations above, we have relied on a NISP approach based on a coarse fully-tensored Gauss quadrature. Combined with the fact that individual realizations could be efficiently generated, this enabled us to demonstrate the advantages of the stochastic preconditioning in a setting where quadrature errors are naturally controlled. For higher dimensional problems, however, or in situations where individual realizations require substantial effort to generate, alternative approaches should be explored that enable minimization of the number of realizations. Various alternatives can be readily implemented for this purpose, including sparse grid methods [11, 33, 34], or compressed sensing concepts [8, 7, 27]. Implementation of these methods to further enhance the performance of preconditioned representations is the focus of ongoing work.

Acknowledgment

Research supported by the U.S. Department of Energy, Office of Advanced Scientific Computing Research under Award DE-SC0001980 (AA, OMK), by Office of Naval Research under Award N00014-10-1-0498 (AA, OMK, MI), and by the French National Research Agency Grant ANR-08-JCJC-0022 (OLM). HNN acknowledges the support of the US Department of Energy (DOE), Office of Basic Energy Sciences, Division of Chemical Sciences, Geosciences, and Biosciences. Sandia National Laboratories is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the US DOE under contract DE-AC04-94-AL85000.

References

[1] M. Abramowitz and I.A. Stegun. *Handbook of mathematical functions*. Dover, New York, 9 edition, 1972.
[2] K.E. Atkinson. *An introduction to numerical analysis*. John Wiley & Sons Inc., New York, second edition, 1989.
[3] I. Babuška, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal.*, 45(3):1005–1034, 2007.
[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] Z. I. Botev. Kernel density estimation using Matlab. Available at http://www.mathworks.us/matlabcentral/fileexchange/authors/27236.
[6] R. H. Cameron and W. T. Martin. The orthogonal development of non-linear functionals in series of fourier-hermite functionals. *Ann. Math.*, 48:385–392, 1947.
[7] E. Candès and T. Tao. Near optimal signal recovery from random projections: universal encoding strategies. *IEEE Trans. Inform. Theory*, 52:5406–5425, 2006.
[8] D. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52:1289–1306, 2006.
[9] R. Tempone F. Nobile and C.G. Webster. An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.*, 46(5):2411–2442, 2008.
[10] B. Ganapathysubramanian and N. Zabaras. Sparse grid collacation schemes for stochastic natural convection problems. *J. Comput. Phys.*, 225:652–685, 2007.
[11] T. Gerstner and M. Griebel. Numerical integration using sparse grids. *Numer. Algorithms*, 18:209–232, 1998.
[12] R.G. Ghanem and P.D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Dover, 2002. 2nd edition.
[13] D. Ghosh and R. Ghanem. Stochastic convergence acceleration through basis enrichment of polynomial chaos expansions. *Int. J. Num. Methods Eng.*, 73:162–184, 2008.
[14] Svante Janson. *Gaussian Hilbert Spaces*. Cambridge University Press, 1997.
[15] O. Kallenberg. *Foundations of Modern Probability*. Springer, second edition, 2002.
[16] A. Keese. *Numerical Solution of Systems with Stochastic Uncertainties : A General Purpose Framework for Stochastic Finite Elements*. PhD thesis, Tech. Univ. Braunschweigh, 2004.

[17] 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.
[18] S.H. Lam and D.A. Goussis. Understanding complex chemical kinetics with computational singular perturbation. *Symposium (International) on Combustion*, 22(1):931 – 941, 1989.
[19] S.H. Lam and D.A. Goussis. A study of homogeneous methanol oxidation kinetic using CSP. *Proc. Comb. Inst.*, 24:113–120, 1992.
[20] S.H. Lam and D.A. Goussis. The CSP method for simplifying kinetics. *International Journal of Chemical Kinetics*, 26(4), 1994.
[21] O.P. Le Maître and O.M. Knio. *Spectral Methods for Uncertainty Quantification With Applications to Computational Fluid Dynamics*. Scientific Computation. Springer, 2010.
[22] 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.
[23] O.P. Le Maître, L. Mathelin, O.M. Knio, and M.Y. Hussaini. Asynchronous time integration for polynomial chaos expansion of uncertain periodic dynamics. *Discrete and Continuous Dynamical Systems*, 28(1):199–226, 2010.
[24] 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. Phys.*, 197(2):502–531, 2004.
[25] O.P. Le Maître, H.N. Najm, P.P. Pebay, R.G. Ghanem, and O. M. Knio. Multi-resolution-analysis scheme for uncertainty quantification in chemical systems. *SIAM Journal on Scientific Computing*, 29(2):864–889, 2007.
[26] X. Ma and N. Zabaras. An adaptative hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *J. Comp. Phys.*, 2009. in press.
[27] L. Mathelin and K.A. Gallivan. A compressed sensing approach for partial differential equations with random input data. *submitted*, 2010.
[28] L. Mathelin and M. Hussaini. A stochastic collocation algorithm for uncertainty analysis. Technical Report NASA/CR-2003-212153, NASA Langley Research Center, 2003.
[29] L. Mathelin and O.P. Le Maître. Uncertainty quantification in a chemical system using error estimated-based adaptation. *Th. Comp. Fluid Dyn.*, in press.
[30] I.M. Babuška M.K. Deb and J.T. Oden. Solution of stochastic partial differential equations using galerkin finite element techniques. *Comp. Meth. App. Mech; Eng.*, 190(48):6359–6372, 2001.
[31] H. Najm, B. Debusschere, Y. Marzouk, S. Widmer, and O. Le Maître. Uncertainty quantification in chemical systems. *Int. J. Num. Eng.*, 80(6):789–814, 2009.
[32] A.W. Naylor and G.R. Sell. *Linear Operator Theory in Engineering and Science*. Springer, 1982.
[33] K. Petras. On the smolyak cubature error for analytic functions. *Advances in Computational Mathematics*, 12:71–93, 2000.
[34] K. Petras. Fast calculation of coefficients in the smolyak algorithm. *Num. Algo.*, 26:93–109, 2001.
[35] B.D. Phenix, J.L. Dinaro, M.A. Tatang, J.W. Tester, J.B. Howard, and G.J. McRae. Incorporation of parametric uncertainty into complex kinetic mechanisms: Application to hydrogen oxidation in supercritical water. *Combustion and Flame*, 112:132–146, 1998.
[36] M.T. Reagan, H.N. Najm, B.J. Debusschere, O.P. Le Maître, O.M. Knio, and R.G. Ghanem. Spectral stochastic uncertainty quantification in chemical systems. *Combustion Theory and Modelling*, 8:607–632, 2004.
[37] 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 Flame*, 132:545–555, 2003.
[38] S.A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Akad. Nauk SSSR*, 4:240–243, 1963.
[39] M. Valorani, D.A. Goussis, F. Creta, and H.N. Najm. Higher order corrections in the approximation of low-dimensional manifolds and the construction of simplified problems with the CSP method. *J. Comput. Phys.*, 209:754–786, November 2005.
[40] X. Wan and G.E. Karniadakis. An adaptative multi-element generalized polynomial chaos method for stochastic differential equations. *J. Comput. Phys.*, 209:617–642, 2005.
[41] X. Wan and G.E. Karniadakis. Multi-element generalized polynomial chaos for arbitrary propability measures. *SIAM J. Sci. Comp.*, 28(3):901–928, 2006.
[42] S. Wiener. The Homogeneous Chaos. *Amer. J. Math.*, 60:897–936, 1938.
[43] D. Williams. *Probability with martingales*. Cambridge Mathematical Textbooks. Cambridge University Press, Cambridge, 1991.
[44] D. Xiu and J.S. Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.*, 27(3):1118–1139, 2005.
[45] D.B. Xiu and G.E. Karniadakis. The Wiener-Askey Polynomial Chaos for stochastic differential equations. *SIAM J. Sci. Comput.*, 24:619–644, 2002.