

Document downloaded from:

<http://hdl.handle.net/10251/63714>

This paper must be cited as:

Chen Charpentier, BM.; Cortés, J.; Licea Salazar, JA.; Romero, J.; Roselló Ferragud, MD.; Santonja, F.; Villanueva Micó, RJ. (2015). Constructing adaptive generalized polynomial chaos method to measure the uncertainty in continuous models: A computational approach. *Mathematics and Computers in Simulation*. 109:113-129.
doi:10.1016/j.matcom.2014.09.002.



The final publication is available at

<http://dx.doi.org/10.1016/j.matcom.2014.09.002>

Copyright Elsevier

Additional Information

Constructing adaptive generalized polynomial chaos method to measure the uncertainty in continuous models: A computational approach

B.-M. Chen-Charpentier^a, J.-C. Cortés^{b,*}, J.-A. Licea^a, J.-V. Romero^b,
M.-D. Roselló^b, Francisco-J. Santonja^c, Rafael-J. Villanueva^b

^a*Department of Mathematics, University of Texas at Arlington, Arlington, TX
76019-0408, United States*

^b*Instituto Universitario de Matemática Multidisciplinar, Building 8G 2nd floor Access C,
Universitat Politècnica de València, 46022 Valencia (Spain)*

^c*Departamento de Estadística e Investigación Operativa, Universitat de València,
Valencia (Spain)*

Abstract

Due to errors in measurements and inherent variability in the quantities of interest, models based on random differential equations give more realistic results than their deterministic counterpart. The generalized polynomial chaos (gPC) is a powerful technique used to approximate the solution of these equations when the random inputs follow standard probability distributions. But in many cases these random inputs do not have a standard probability distribution. In this paper, we present a step-by-step constructive methodology to implement directly a useful version of *adaptive* gPC for arbitrary distributions, extending the applicability of the gPC. The paper mainly focuses on the computational aspects, on the implementation of the method and on the creation of a useful software tool. This tool allows the user to easily change the types of distributions and the order of the expansions, and to study their effects on the convergence and on the results. Several examples illustrating

*Corresponding Author. Phone number: +34-963-877000-ext.88289. Fax: +34-963-877669

Email addresses: bmchen@uta.edu (B.-M. Chen-Charpentier),
jccortes@imm.upv.es (J.-C. Cortés), juan.liceasalazar@mavs.uta.edu (J.-A. Licea),
jvromero@imm.upv.es (J.-V. Romero), drosello@imm.upv.es (M.-D. Roselló),
francisco.santonja@uv.es (Francisco-J. Santonja), rjvillan@imm.upv.es (Rafael-J. Villanueva)

the usefulness of the method are included.

Keywords: random differential equations, adaptive polynomial chaos, computing

1. Introduction and motivation

In dealing with random differential equations, a considerable number of useful methods have been developed [13, 20, 5, 4, 15, 14, 18, 21, 3, 8, 6]. Here, we are specifically interested in the generalized Polynomial Chaos method (gPC) since it has been shown to be particularly effective for a number of problems [12, 26, 9]. gPC was proposed by Xiu and Karniadakis [26] as a remarkable generalization of the pioneering work by N. Wiener [25], which is usually referred to as the *homogeneous chaos* or Hermite-PC.

The application of gPC requires that every random model parameter belongs to some of the standard probability distributions. Otherwise, it has to be approximated by some of them. In practice, this approximation is usually made by using Gaussian random variables. Although in many practical cases the previous approach can be legitimized by the Central Limit Theorem, it is desirable to investigate alternative approaches since sometimes such approximations cannot be assumed or, even if it can be assumed, the improvement of the accuracy of the results constitutes a goal itself.

The aim of this paper is to present a method that provides researchers, that do not know the foundations of gPC, a step-by-step computational approach technique to implement an adaptive gPC method to be used in random continuous models (random differential equations). In this method the random inputs do not need to belong necessarily to some of the standard probability families. This point has great interest since, in practice, most of random variables constructed to model real phenomena likely do not fit the standard distributions, and therefore the proposed method extends the applicability of the gPC. This approach requires the construction of probability density functions of input random variables that, according to gPC method, will play the same role as the weighting functions do [26]. Notice that methods to build such probability density functions from sampled data are well-developed [16, 19, 23]. Several illustrative examples including practical applications showing the usefulness of the proposed method are presented in this paper.

This paper is organized as follows. For the sake of clarity, in Section 2

we summarize gPC method focusing on its application to solve random differential equations. Section 3 is devoted to introducing a variation of gPC *adapted* to the case where every random model parameter may have probability distributions different from the standard ones. A pseudo-code algorithm to compute the main statistical functions of the solution stochastic process to random differential equations is developed in Section 4, where significant computational comments are included. In Section 5, we apply adaptive gPC to solve several illustrative examples including both, test-problems and models appearing in applications. Finally, conclusions are drawn in Section 6.

2. gPC method: a short review focusing on the solution of random differential equations

For the sake of clarity in the subsequent developments, this section explains how gPC works in dealing with the solution of random differential equations

$$\mathfrak{D}(t, \boldsymbol{\xi}(\omega); y) = f(t, \boldsymbol{\xi}(\omega)), \quad (1)$$

where \mathfrak{D} denotes a differential operator; $y = y(t, \boldsymbol{\xi}(\omega))$ is the solution stochastic process to be determined and $f(t, \boldsymbol{\xi}(\omega))$ is a forcing term. Notice that in the random differential equation (1) uncertainty is represented by $\boldsymbol{\xi}$ and it just enters through their coefficients and forcing term, although in practice it could also be considered via initial and/or boundary conditions.

Let $(\Omega, \mathfrak{F}, P)$ be a probability space, and let us consider the set L^2 whose elements are second-order real random variables, that is, real random variables ζ having finite variance, or equivalently $E[\zeta^2] < \infty$, where $E[\cdot]$ denotes the expectation operator. The set L^2 endowed with the inner product $\langle \zeta_1, \zeta_2 \rangle = E[\zeta_1 \zeta_2]$, is a Hilbert space, usually denoted by $L^2(\Omega, \mathfrak{F}, P)$ [17]. The norm inferred by the above inner product determines the mean-square convergence.

gPC is a powerful technique to represent spectrally in the random dimension random variables ζ and stochastic processes $y(t)$ in $L^2(\Omega, \mathfrak{F}, P)$. These representations are given by infinite random series defined in terms of certain orthogonal polynomial expansions $\{\Phi_i\}$ which depend on a number of random variables $\boldsymbol{\xi}(\omega) = (\xi_1(\omega), \xi_2(\omega), \dots)$, $\omega \in \Omega$,

$$\zeta = \sum_{i=0}^{\infty} \zeta_i \Phi_i(\boldsymbol{\xi}(\omega)), \quad y(t) = \sum_{i=0}^{\infty} y_i(t) \Phi_i(\boldsymbol{\xi}(\omega)). \quad (2)$$

As we pointed out in the previous section, the choice of the trial basis $\{\Phi_i\}$ is crucial in dealing with random differential equations and it is what distinguishes homogeneous chaos from gPC. In the first one, $\{\Phi_i\}$ are just the Hermite polynomials defined in terms of Gaussian random variables ξ_i . Whereas in gPC, $\{\Phi_i\}$ belong to the Wiener-Askey scheme (which contains Hermite polynomials as a particular case) and ξ_i are standard random variables, such as binomial, Poisson, beta, gamma, etc, that are linked to the orthogonal polynomial basis $\{\Phi_i\}$ [26].

For each $y(t) \in L^2(\Omega, \mathfrak{F}, P)$, assuming that $\{\Phi_i\}$ are the Hermite polynomials, the Cameron-Martin theorem guarantees that series (2) converges in the L^2 sense [7]. However, in practice, optimal convergence is limited to stochastic processes with Gaussian inputs ξ_i . Otherwise, numerical results deteriorate even for early times [11].

On the account of this fact, Xiu and Karniadakis proposed gPC focusing on the solution of random differential equations [26]. This work introduces not only an extension of Wiener's contribution but also a comprehensive approach to set the *optimal* trial basis $\{\Phi_i\}$. This set $\{\Phi_i\}$ constitutes a complete orthogonal basis in $L^2(\Omega, \mathfrak{F}, P)$ with the inner product

$$\langle \Phi_i, \Phi_j \rangle = \langle \Phi_i^2 \rangle \delta_{ij}, \quad (3)$$

where δ_{ij} is the Kronecker delta and $\langle \cdot \rangle$ denotes the ensemble average defined as follows

$$\langle g(\boldsymbol{\xi}), h(\boldsymbol{\xi}) \rangle = \int_{\text{supp}(\boldsymbol{\xi})} g(\boldsymbol{\xi}) h(\boldsymbol{\xi}) f_{\boldsymbol{\xi}}(\boldsymbol{\xi}) d\boldsymbol{\xi}, \quad (4)$$

being $f_{\boldsymbol{\xi}}(\boldsymbol{\xi})$ the joint probability density function of $\boldsymbol{\xi}$ and $\text{supp}(\boldsymbol{\xi})$ its support. Notice that in (4) $f_{\boldsymbol{\xi}}(\boldsymbol{\xi})$ plays the role of a weight function. Independence between random variables ξ_i is usually assumed in the previous development. As a consequence $f_{\boldsymbol{\xi}}(\boldsymbol{\xi})$ factorizes as the product of probability density functions $f_{\xi_i}(\xi_i)$ associated to each random variable ξ_i .

In order to keep the computations feasible in dealing with random differential equations, each random model parameter ζ as well as the solution stochastic process are represented by truncated series of the form (2),

$$\zeta \approx \sum_{i=0}^P \zeta_i \Phi_i(\boldsymbol{\xi}(\omega)), \quad y(t) \approx \sum_{i=0}^P y_i(t) \Phi_i(\boldsymbol{\xi}(\omega)), \quad (5)$$

where the number of components of random vector $\boldsymbol{\xi}(\omega)$ also needs to be truncated at a number s called the order of chaos, $\boldsymbol{\xi}(\omega) = (\xi_1(\omega), \xi_2(\omega), \dots, \xi_s(\omega))$.

Note that at this point the set $\{\Phi_i, 0 \leq i \leq P\}$ is not complete. The truncation order P is made so that all expansion polynomials up to a certain maximum degree, p , are included. This entails the following relationship between the number of terms $P + 1$ in the series expansions, the maximum degree p and the order of chaos s : $P + 1 = (p + s)!/(p!s!)$.

In this context, solving the random differential equation (1) consists of computing coefficients $y_i(t)$ appearing in (5). From the functions $y_i(t)$ one computes the expectation and the variance of the approximate solution stochastic process $y(t)$ as follows:

$$\mu_{y(t)} = \mathbb{E}[y(t)] \approx y_0(t), \quad \sigma_{y(t)}^2 = \text{Var}[y(t)] \approx \sum_{i=1}^P (y_i(t))^2 \mathbb{E}[(\Phi_i(\boldsymbol{\xi}))^2]. \quad (6)$$

To achieve this goal, first the expansion of $y(t)$ given by (5) is substituted into the random differential equation (1). Second, a Galerkin projection is done by multiplying the random differential equation by every polynomial of the expansion basis $\{\Phi_i\}$ and then, the ensemble average is taken. This leads to

$$\left\langle \mathcal{D} \left(t, \boldsymbol{\xi}; \sum_{i=0}^P y_i(t) \Phi_i(\boldsymbol{\xi}) \right), \Phi_j(\boldsymbol{\xi}) \right\rangle = \langle f(t, \boldsymbol{\xi}), \Phi_j(\boldsymbol{\xi}) \rangle, \quad j = 0, 1, \dots, P, \quad (7)$$

that corresponds to a deterministic system of $P + 1$ coupled differential equations whose unknowns are the node functions $y_i(t)$. These unknowns can be computed by standard numerical techniques such as Runge-Kutta schemes.

In [26] authors show through a simple but still illustrative example involving only one single random input ζ that exponential convergence for the mean and the variance of the solution stochastic process is retained when the orthogonal polynomial basis $\{\Phi_i\}$ is chosen in accordance with the probability distribution of ζ . Namely, if ζ has a uniform, Gaussian, beta, gamma, etc. distribution, then $\{\Phi_i\}$ should be chosen as Legendre, Hermite, Jacobi, Laguerre, etc., respectively. This proposal is based on the following key fact: the weighting functions associated to the so-called Wiener-Askey scheme, that correspond to classical families of orthogonal polynomials, are identical to the probability density functions of some standard random variables. Recently, in [10] authors provide conditions under which gPC expansions converge in the mean-square sense to the correct limit, which constitutes an extension of paper of Cameron and Martin [7] where mean-square convergence of homogeneous chaos was established.

3. An adaptive generalized Polynomial Chaos method

As we have indicated in previous sections, gPC method provides a comprehensive way to choose the *optimal* trial basis $\{\Phi_i\}$ when random model parameters belong to the standard probabilistic distributions. However clearly other distributions are also possible in practice, thus further study is demanded. The success of gPC also relies on solving the deterministic system of differential equations obtained after considering a Galerkin projection. Then it would be desirable to keep such a system as simple as possible. Bearing in mind both challenges, in the following we present a variation of gPC, referred to as adaptive gPC.

In the following, we focus on continuous mathematical models containing uncertainty into their formulation. Hereinafter, we specifically will consider models based on systems of random differential equations, which for the sake of clarity, we will denote directly without using the previous operator notation, as

$$F(t, \mathbf{y}, \dot{\mathbf{y}}) = \mathbf{0}, \quad \mathbf{y}(t_0) = \widehat{\mathbf{y}}_0, \quad (8)$$

where t is the independent variable, $F : \mathbb{R}^{2q+1} \rightarrow \mathbb{R}^q$ is a map, $\mathbf{y} = \mathbf{y}(t) = (y^1(t), y^2(t), \dots, y^q(t))^\top$ is the vector of unknown functions, $\widehat{\mathbf{y}}_0 = (y^1(t_0), y^2(t_0), \dots, y^q(t_0))^\top$ is the vector of initial conditions and $\mathbf{0} = (0, 0, \dots, 0)^\top$ denotes the null vector of size q . Here, \top denotes the transpose operator for vectors and matrices. $\{\zeta_1, \dots, \zeta_s\}$ are assumed to be the independent random input parameters in the initial value problem (i.v.p.) (8). These parameters include both the coefficients and the initial conditions. Notice that according to gPC nomenclature, in this case s is also the order of the chaos.

3.1. Fourier expansion of the model parameters

For each probability density function $f_{\zeta_i}(\zeta_i)$ associated to the random variable ζ_i , we define the following inner product whose weighting function is just $f_{\zeta_i}(\zeta_i)$

$$\langle g(\zeta_i), h(\zeta_i) \rangle_{\zeta_i} = \int_{\text{supp}(\zeta_i)} g(\zeta_i) h(\zeta_i) f_{\zeta_i}(\zeta_i) d\zeta_i, \quad 1 \leq i \leq s, \quad (9)$$

being g, h deterministic functions such that the above integral exists.

In contrast to Wiener-Askey gPC, we are going to represent both random model parameters ζ_i , $1 \leq i \leq s$, and the solution stochastic process $\mathbf{y}(t)$ in terms of a complete orthonormal polynomial basis constructed *directly* in

terms of ζ_i , $1 \leq i \leq s$. This construction is done by applying the Gram-Schmidt method with the previous inner product $\langle \cdot \rangle_{\zeta_i}$ to the canonical basis of each random model parameter ζ_i truncated at a common degree p : $\mathcal{C}_i^p = \{1, \zeta_i, (\zeta_i)^2, \dots, (\zeta_i)^p\}$, $1 \leq i \leq s$. For each i , this procedure allows us to construct the set of orthonormal polynomials

$$\Xi_i^p = \{\phi_0^i(\zeta_i), \phi_1^i(\zeta_i), \dots, \phi_p^i(\zeta_i)\}, \quad 1 \leq i \leq s, \quad (10)$$

where $\phi_0^i(\zeta_i) = 1$, $1 \leq i \leq s$. Thus each random model parameter ζ_i can be expanded in a Fourier series, i.e.,

$$\zeta_i = \alpha_0^i \phi_0^i(\zeta_i) + \alpha_1^i \phi_1^i(\zeta_i) + \dots + \alpha_p^i \phi_p^i(\zeta_i), \quad 1 \leq i \leq s. \quad (11)$$

In order to calculate the Fourier coefficients α_j^i , $1 \leq i \leq s$, $0 \leq j \leq p$, we should take into account that

- ζ_i has a non-orthonormal natural expansion as itself,
- $\phi_1^i(\zeta_i) = a_i + b_i \zeta_i$, $b_i \neq 0$.

Therefore, substituting this latter expression into (11) and comparing coefficients in both sides, one gets that the Fourier expansion of the random model parameter ζ_i is

$$\zeta_i = \alpha_0^i \phi_0^i(\zeta_i) + \alpha_1^i \phi_1^i(\zeta_i), \quad \alpha_0^i = -\frac{a_i}{b_i}, \quad \alpha_1^i = \frac{1}{b_i}, \quad 1 \leq i \leq s. \quad (12)$$

At this point several remarks can be quoted:

- To represent each random model parameter ζ_i we only have needed as starting canonical basis $\mathcal{C}_i^1 = \{1, \zeta_i\}$, $1 \leq i \leq s$, which truncation order is $p = 1$. However, in applications higher values of p should be considered in order to represent more accurately the solution stochastic process $\mathbf{y}(t)$ to random differential equation (8).
- For the sake of clarity, we have assumed that the canonical bases \mathcal{C}_i^p considered to represent each random model parameter ζ_i , $1 \leq i \leq s$, have the same truncation order. Notice that its generalization to different values, say p_i , would follow immediately from the subsequent developments.

- In many practical situations the random model parameters ζ_i do not appear directly but as arguments of non-polynomial functions, say $r(\zeta_i)$, involved into the random differential equation under study. This hinders the application of gPC method since it needs to approximate $r(\zeta_i)$ by an appropriate polynomial, which implies a loss in the approximation that could be unaffordable. Assuming that probability density function of the random variable ζ_i is known, the application of the Transformation Technique provides, under quite general conditions, a powerful method to compute, in closed form, the probability density function of $r(\zeta_i)$ in terms of the probability density function of random variable ζ_i [22]. As we shall see, this allows us to consider $r(\zeta_i)$ as a random model parameter itself rather than ζ_i .

3.2. Building the solution by adaptive polynomial chaos expansion

In Table 1 we summarize the elements as well as the notation that we have introduced at this point.

Table 1: Notation introduced for parameters, probability density functions (p.d.f.'s), inner products and orthogonal bases in order to represent the random model parameter expansions.

Parameters	p.d.f.'s	Inner products	Orthonormal bases
ζ_1	$f_{\zeta_1}(\zeta_1)$	$\langle \cdot \rangle_{\zeta_1}$	$\Xi_1^p = \{\phi_0^1, \dots, \phi_p^1\}$
...
ζ_s	$f_{\zeta_s}(\zeta_s)$	$\langle \cdot \rangle_{\zeta_s}$	$\Xi_s^p = \{\phi_0^s, \dots, \phi_p^s\}$

In our case, the random model parameter expansions correspond to $p = 1$ and they are given by:

$$\begin{aligned}
\zeta_1 &= \alpha_0^1 \phi_0^1(\zeta_1) + \alpha_1^1 \phi_1^1(\zeta_1), \\
&\vdots \\
\zeta_s &= \alpha_0^s \phi_0^s(\zeta_s) + \alpha_1^s \phi_1^s(\zeta_s).
\end{aligned} \tag{13}$$

Now, we will represent the solution stochastic process $\mathbf{y}(t)$ and the initial condition $\widehat{\mathbf{y}}_0$ in terms of a basis $\Xi = \{\Phi_j\}$ constructed from Ξ_i^p , $1 \leq i \leq s$. The elements of this basis represent multidimensional expansion polynomials and, taking into account independence between random model parameters ζ_1, \dots, ζ_s , they are built from the simple tensor product

$$\Phi_j(\boldsymbol{\zeta}) = \phi_{p_1}^1(\zeta_1) \times \dots \times \phi_{p_s}^s(\zeta_s), \tag{14}$$

for some mapping $(p_1, \dots, p_s) \rightarrow j$ starting from $(0, \dots, 0)$ that corresponds to $j = 0$. In the particular case that $p = 1$, $0 \leq p_i \leq 1$, $1 \leq i \leq s$. The tensor product above is done up to a degree r . When $p = 1$, $r \leq s$. This leads to the following representations of the solution stochastic process and its derivative

$$\mathbf{y}(t) = \sum_{j=0}^P \mathbf{y}_j(t) \Phi_j(\boldsymbol{\zeta}), \quad \dot{\mathbf{y}}(t) = \sum_{j=0}^P \dot{\mathbf{y}}_j(t) \Phi_j(\boldsymbol{\zeta}), \quad (15)$$

as well as the initial condition

$$\hat{\mathbf{y}}_0 = \sum_{j=0}^P \mathbf{y}_{0,j}(t_0) \Phi_j(\boldsymbol{\zeta}). \quad (16)$$

In general, if we want that these expansions contain all the products of the form (14) with degree less or equal than r with $0 \leq r \leq p \times s$, then the truncation order P is completely determined.

Following gPC development, now we substitute expansions (15)–(16) into the i.v.p. (8) and we obtain:

$$F \left(t, \sum_{j=0}^P \mathbf{y}_j(t) \Phi_j(\boldsymbol{\zeta}), \sum_{j=0}^P \dot{\mathbf{y}}_j(t) \Phi_j(\boldsymbol{\zeta}) \right) = \mathbf{0}, \quad \hat{\mathbf{y}}_0 = \sum_{j=0}^P \mathbf{y}_{0,j}(t_0) \Phi_j(\boldsymbol{\zeta}), \quad (17)$$

which involves random variables ζ_1, \dots, ζ_s .

3.3. Building the auxiliary system of deterministic differential equations

The goal of this section is to obtain the deterministic coefficients $\mathbf{y}_j(t)$, $0 \leq j \leq P$. To achieve this goal, we first need to consider the inner product

$$\langle g(\boldsymbol{\zeta}), h(\boldsymbol{\zeta}) \rangle_{\boldsymbol{\zeta}} = \int_{\text{supp}(\boldsymbol{\zeta})} g(\boldsymbol{\zeta}) h(\boldsymbol{\zeta}) f_{\boldsymbol{\zeta}}(\boldsymbol{\zeta}) d\boldsymbol{\zeta}, \quad f_{\boldsymbol{\zeta}}(\boldsymbol{\zeta}) = \prod_{i=1}^s f_{\zeta_i}(\zeta_i), \quad \boldsymbol{\zeta} = (\zeta_1, \dots, \zeta_s), \quad (18)$$

where, by independence of random variables ζ_i , the weighting function $f_{\boldsymbol{\zeta}}(\boldsymbol{\zeta})$ has been factorized through the probability density functions of each ζ_i , $f_{\zeta_i}(\zeta_i)$.

Now we establish a deterministic i.v.p. based on a system of $P+1$ differential equations whose unknowns are the nodes $\mathbf{y}_j(t)$, $0 \leq j \leq P$. Hereinafter,

this system will be referred to as the auxiliary system. This system is built by multiplying each equation of the random differential system (17) by all the elements of the orthonormal basis $\Xi = \{\Phi_j\}$, previously constructed and, then, taking the ensemble average $\langle \cdot \rangle_\zeta$ defined in (18). This allows us to take advantage of orthogonality in order to simplify the deterministic system of differential equations obtained in this way. This system has as initial conditions the ones obtained by multiplying (17) by $\{\Phi_j\}$ and then, taking again the ensemble average $\langle \cdot \rangle_\zeta$. This permits us to compute the coefficients $\mathbf{y}_{0,j}(t_0)$ as follows

$$\mathbf{y}_{0,j}(t_0) = \langle \widehat{\mathbf{y}}_0, \Phi_j(\zeta) \rangle_\zeta, \quad 0 \leq j \leq P. \quad (19)$$

3.4. Solving the auxiliary system and computing the mean and the standard deviation of the solution

In the previous subsection, we have constructed a deterministic i.v.p. whose unknowns are $\mathbf{y}_j(t)$, $0 \leq j \leq P$. These functions can be computed by applying a numerical scheme such as Runge-Kutta. Once they have been obtained, approximations for the mean and the variance can be computed taking into account the following relationships:

$$\langle \mathbf{y}(t) \rangle_\zeta = \mathbf{y}_0(t), \quad \Sigma_{\mathbf{y}(t)} = \sum_{i=1}^P \mathbf{y}_i(t) (\mathbf{y}_i(t))^\top \langle (\Phi_i(\zeta))^2 \rangle_\zeta. \quad (20)$$

In (20), the first expression is just the expectation of the solution stochastic process $\mathbf{y}(t)$, while the second one is the variance-covariance matrix whose diagonal represents the variance of each component $y^i(t)$, $1 \leq i \leq q$ of $\mathbf{y}(t)$.

4. Adaptive polynomial chaos algorithm

In this section we develop a pseudo-code algorithm to compute the mean and the standard deviation of the solution stochastic process of a random i.v.p. of the form (8).

The inputs of the algorithm are:

- The system of random differential equations (model): $F(t, \mathbf{y}, \dot{\mathbf{y}}) = \mathbf{0}$, with random initial condition: $\mathbf{y}(t_0) = \widehat{\mathbf{y}}_0$.
- The random model parameters including the initial conditions: $\zeta_1, \zeta_2, \dots, \zeta_s$.

- The probability density functions of the random model parameters: $f_{\zeta_1}(\zeta_1), f_{\zeta_2}(\zeta_2), \dots, f_{\zeta_s}(\zeta_s)$.
- The (common) truncation order of the polynomial expansion of each random model parameter p .

Now, we describe the algorithm:

- STEP 1. For $i = 1, \dots, s$, compute the orthonormal basis $\Xi_i^p = \{\phi_0^i(\zeta_i), \dots, \phi_p^i(\zeta_i)\}$ associated to ζ_i applying the Gram-Schmidt orthonormalization method and considering the scalar product $\langle \cdot \rangle_{\zeta_i}$ defined in (9) to the canonical basis $\mathcal{C}_i^p = \{1, \zeta_i, (\zeta_i)^2, \dots, (\zeta_i)^p\}$.
- STEP 2. For $i = 1, \dots, s$, we have that $\phi_0^i(\zeta_i) = 1$ and $\phi_1^i(\zeta_i) = a_i + b_i \zeta_i$. Thus, we compute the Fourier expansion of $\zeta_i = \alpha_0^i \phi_0^i(\zeta_i) + \alpha_1^i \phi_1^i(\zeta_i)$ using expression (12).
- STEP 3. Build the set of polynomials $\Xi = \{\Phi_j\}$ defined by (14).
- STEP 4. Define the scalar product $\langle \cdot \rangle_{\zeta}$ given by (18).
- STEP 5. Consider the truncated orthonormal expansions for the random model parameters, the solution stochastic process and its derivative and the random initial condition given by (15) and (16), respectively.
- STEP 6. Substitute the above expansions into the model to obtain the random i.v.p. (17).
- STEP 7. Obtain the auxiliary system in two phases: first, multiplying each equation of the random i.v.p. (17) by orthonormal polynomials of the basis $\Xi = \{\Phi_j\}$ and, second taking the ensemble average inferred by the inner product $\langle \cdot \rangle_{\zeta}$ constructed in STEP 4.
- STEP 8. Solve numerically the auxiliary system.
- STEP 9. Compute the mean and the standard deviation of the solution stochastic process $\mathbf{y}(t)$ taking into account expressions in (20).

We have implemented this algorithm in *Mathematica* [1].

4.1. Some computational aspects

During the implementation and when performing some tests, we realised that it is very important to calculate the Fourier expansion of the parameters as accurately as possible. In fact, in our case as the code has been implemented in *Mathematica*, we have computed these coefficients symbolically. Otherwise, we noticed that the number of coefficients of the auxiliary system increases rapidly, with most of them being very small (0 when computations were carried out symbolically), and consequently, solving the auxiliary system produces dramatic numerical errors.

Moreover, the bottleneck of this algorithm is the building of the auxiliary system (STEP 7) because we have to evaluate many inner products that may involve unbounded integrals with transcendent functions. In order to improve these computations we have considered that when multiplying each polynomial from the set $\Xi = \{\Phi_j\}$ by each equation of the random i.v.p. (17), these inner products are carried out monomial by monomial, and then, reduced to computations of the form $\langle (\zeta_1)^{i_1} \cdots (\zeta_s)^{i_s}, 1 \rangle_{\zeta}$. These computations, in some cases, can be pre-calculated with *Mathematica* in order to obtain a closed formula that allows to reduce drastically the time to build the auxiliary system.

5. Examples: a comparative case-study

In this section, we present five illustrative examples where adaptive polynomial chaos is applied. Examples 1–3 have exact solutions. They have been included to show that the proposed method is as good as Hermite-PC. More specifically, in Example 1 we consider a linear random differential equation with two random parameters and show in detail all the involved computations in order to show how the proposed method works. In Example 2, we keep studying a random linear differential equation but increased the number of random parameters. Example 3 illustrates that the method also works when dealing with random nonlinear differential equations. In Examples 1–3 we compare numerical results for the mean and variance obtained by adaptive gPC with those ones computed by other approaches, namely:

- Exact. The examples have been chosen such that exact solution for the mean and the variance are available. Thus, they act as test examples.
- Hermite-PC. It consists of obtaining the mean and the variance by applying homogeneous PC where both the random model parameters and

the solution stochastic process are represented in terms of the Hermite orthogonal polynomial basis. This approach is extensively accepted in dealing with random differential equations.

The random parameters of Examples 1–3 have standard distributions whereas the Example 4 involves a random input whose probability density function is constructed from sampled data. We show that the obtained probability density function does not follow any standard distribution, so the adaptive gPC turns out particularly useful. In this example, the numerical results provided by adaptive gPC are compared with measured data. As in practice gPC technique has interest when there is not an available solution, we end this section with Example 5. In this example, we study an epidemiological model based on a random system of differential equations with four random parameters whose exact solution is not available. Numerical comparisons against Monte Carlo sampling are carried out.

Example 1. *We deal with the random i.v.p.*

$$\dot{y}(t) + by(t) = 0, \quad y(0) = y_0, \quad (21)$$

and we will assume that

- b is a beta random variable of parameters $\alpha = \beta = 2$, i.e., $b \sim B(2; 2)$,
- $\hat{y}_0 \sim B(2; 5)$.

For the sake of clarity, in this first example we will show in detail each step of our previous development. According to previous notation random variables b and \hat{y}_0 correspond to ζ_1 and ζ_2 , respectively, and their probability density functions are

$$f_{\zeta_1}(\zeta_1) = 6(1 - \zeta_1)\zeta_1, \quad f_{\zeta_2}(\zeta_2) = 30(1 - \zeta_2)^4\zeta_2, \quad 0 < \zeta_1, \zeta_2 < 1. \quad (22)$$

We take $p = 3$ as the common degree for the two polynomial bases $\Xi_1^3 = \{\phi_0^1, \phi_1^1, \phi_2^1, \phi_3^1\}$ and $\Xi_2^3 = \{\phi_0^2, \phi_1^2, \phi_2^2, \phi_3^2\}$ (see STEP 1). These bases are defined in terms of the random model parameters ζ_1 and ζ_2 , respectively. We take $r = 3$ as the maximum degree of the multivariate polynomials $\Xi = \{\Phi_0, \Phi_1, \dots, \Phi_9\}$ (see STEP 3) defining the basis where the solution stochastic process $y(t)$ is going to be represented.

Now, we follow the algorithm developed in Section 4, step by step:

- *STEP 1. Taking the canonical basis $\mathcal{C}_1^3 = \{1, \zeta_1, (\zeta_1)^2, (\zeta_1)^3\}$, the inner product*

$$\langle g(\zeta_1), h(\zeta_1) \rangle_{\zeta_1} = \int_0^1 g(\zeta_1)h(\zeta_1)f_{\zeta_1}(\zeta_1) d\zeta_1,$$

and using Gram-Schmidt orthonormalization method, we obtain the following orthonormal basis for parameter $b = \zeta_1$,

$$\begin{aligned}\phi_0^1 &= \phi_0^1(\zeta_1) = 1, \\ \phi_1^1 &= \phi_1^1(\zeta_1) = -\sqrt{5} + 2\sqrt{5}\zeta_1, \\ \phi_2^1 &= \phi_2^1(\zeta_1) = \sqrt{14} - 5\sqrt{14}\zeta_1 + 5\sqrt{14}(\zeta_1)^2, \\ \phi_3^1 &= \phi_3^1(\zeta_1) = -\sqrt{30} + 9\sqrt{30}\zeta_1 - 21\sqrt{30}(\zeta_1)^2 + 14\sqrt{30}(\zeta_1)^3.\end{aligned}$$

Analogously, for parameter $y_0 = \zeta_2$ and the inner product

$$\langle g(\zeta_2), h(\zeta_2) \rangle_{\zeta_2} = \int_0^1 g(\zeta_2)h(\zeta_2)f_{\zeta_2}(\zeta_2) d\zeta_2,$$

its orthonormal basis is

$$\begin{aligned}\phi_0^2 &= \phi_0^2(\zeta_2) = 1, \\ \phi_1^2 &= \phi_1^2(\zeta_2) = -\frac{4}{\sqrt{5}} + \frac{14}{\sqrt{5}}\zeta_2, \\ \phi_2^2 &= \phi_2^2(\zeta_2) = \sqrt{7} - 8\sqrt{7}\zeta_2 + 12\sqrt{7}(\zeta_2)^2, \\ \phi_3^2 &= \phi_3^2(\zeta_2) = -\frac{8}{\sqrt{5}} + \frac{108}{\sqrt{5}}\zeta_2 - 72\sqrt{5}(\zeta_2)^2 + 66\sqrt{5}(\zeta_2)^3.\end{aligned}$$

- *STEP 2. Looking at $\phi_1^1(\zeta_1)$, its coefficients are $a_1 = -\sqrt{5}$ and $b_1 = 2\sqrt{5}$. Then, $\alpha_0^1 = \frac{1}{2}$, $\alpha_1^1 = \frac{1}{2\sqrt{5}}$ and the Fourier expansion of parameter $b = \zeta_1$ is*

$$b = \frac{1}{2}\phi_0^1(\zeta_1) + \frac{1}{2\sqrt{5}}\phi_1^1(\zeta_1) = \frac{1}{2} + \frac{1}{2\sqrt{5}} \left(-\sqrt{5} + 2\sqrt{5} \zeta_1 \right).$$

Analogously, the coefficients of $\phi_1^2(\zeta_2)$ are $a_2 = -\frac{4}{\sqrt{5}}$ and $b_2 = \frac{14}{\sqrt{5}}$, $\alpha_0^2 = \frac{2}{7}$, $\alpha_1^2 = \frac{\sqrt{5}}{14}$ and the expansion of parameter $\hat{y}_0 = \zeta_2$ is

$$\hat{y}_0 = \frac{2}{7}\phi_0^2(\zeta_2) + \frac{1}{14}\sqrt{5}\phi_1^2(\zeta_2) = \frac{2}{7} + \frac{1}{14}\sqrt{5} \left(-\frac{4}{\sqrt{5}} + \frac{14}{\sqrt{5}}\zeta_2 \right).$$

- *STEP 3.* The set $\Xi = \{\Phi_0 = \Phi_0(\zeta_1, \zeta_2), \dots, \Phi_9 = \Phi_9(\zeta_1, \zeta_2)\}$ is given by

$$\begin{aligned}\Phi_0 &= \phi_0^1 \phi_0^2, & \Phi_1 &= \phi_0^1 \phi_1^2, \\ \Phi_2 &= \phi_0^1 \phi_2^2, & \Phi_3 &= \phi_0^1 \phi_3^2, \\ \Phi_4 &= \phi_1^1 \phi_0^2, & \Phi_5 &= \phi_1^1 \phi_1^2, \\ \Phi_6 &= \phi_1^1 \phi_2^2, & \Phi_7 &= \phi_2^1 \phi_0^2, \\ \Phi_8 &= \phi_2^1 \phi_1^2, & \Phi_9 &= \phi_3^1 \phi_0^2.\end{aligned}$$

Note that we only use the products of polynomials whose total degree is less or equal than $r = 3$.

- *STEP 4.* Define the new inner product

$$\langle g(\zeta_1, \zeta_2), h(\zeta_1, \zeta_2) \rangle_{\zeta} = \int_0^1 \int_0^1 g(\zeta_1, \zeta_2) h(\zeta_1, \zeta_2) f_{\zeta_1}(\zeta_1) f_{\zeta_2}(\zeta_2) d\zeta_1 d\zeta_2.$$

- *STEP 5.* The expansions of the solution $y(t)$ and its derivative $\dot{y}(t)$ are

$$\begin{aligned}y(t, \zeta_1, \zeta_2) &= \sum_{j=0}^9 y_j(t, \zeta_1, \zeta_2) \Phi_j(\zeta_1, \zeta_2), \\ \dot{y}(t, \zeta_1, \zeta_2) &= \sum_{j=0}^9 \dot{y}_j(t, \zeta_1, \zeta_2) \Phi_j(\zeta_1, \zeta_2).\end{aligned}$$

- *STEP 6.* We substitute the random parameter models and unknown function expansions into the i.v.p. $\dot{y}(t) + by(t) = 0$, $y(0) = \hat{y}_0$. The resulting equation is

$$\begin{aligned}& \sum_{j=0}^9 \dot{y}_j(t, \zeta_1, \zeta_2) \Phi_j(\zeta_1, \zeta_2) \\ & + \left(\frac{1}{2} + \frac{1}{2\sqrt{5}} \left(-\sqrt{5} + 2\sqrt{5}\zeta_1 \right) \right) \sum_{j=0}^9 y_j(t, \zeta_1, \zeta_2) \Phi_j(\zeta_1, \zeta_2) = 0, \\ & \sum_{j=0}^9 y_j(0, \zeta_1, \zeta_2) \Phi_j(\zeta_1, \zeta_2) = \frac{2}{7} + \frac{1}{14} \sqrt{5} \left(-\frac{4}{\sqrt{5}} + \frac{14}{\sqrt{5}} \zeta_2 \right).\end{aligned}$$

- *STEP 7.* Now, using the inner product defined in *STEP 4*, we multiply each polynomial $\Phi_j(\zeta_1, \zeta_2)$, $j = 0, \dots, 9$, for each one of the equations

obtained in STEP 6. This leads to the auxiliary system of deterministic equations given by

$$\begin{aligned}
& \frac{1}{2}y_0(t) + \frac{1}{2\sqrt{5}}y_1(t) + \dot{y}_0(t) = 0, \\
& \frac{1}{2\sqrt{5}}y_0(t) + \frac{1}{2}y_1(t) + \sqrt{\frac{2}{35}}y_2(t) + \dot{y}_1(t) = 0, \\
& \sqrt{\frac{2}{35}}y_1(t) + \frac{1}{2}y_2(t) + \frac{1}{2}\sqrt{\frac{5}{21}}y_3(t) + \dot{y}_2(t) = 0, \\
& \frac{1}{2}\sqrt{\frac{5}{21}}y_2(t) + \frac{1}{2}y_3(t) + \dot{y}_3(t) = 0, \\
& \frac{1}{2}y_4(t) + \frac{1}{2\sqrt{5}}y_5(t) + \dot{y}_4(t) = 0, \\
& \frac{1}{2\sqrt{5}}y_4(t) + \frac{1}{2}y_5(t) + \sqrt{\frac{2}{35}}y_6(t) + \dot{y}_5(t) = 0, \\
& \sqrt{\frac{2}{35}}y_5(t) + \frac{1}{2}y_6(t) + \dot{y}_6(t) = 0, \\
& \frac{1}{2}y_7(t) + \frac{1}{2\sqrt{5}}y_8(t) + \dot{y}_7(t) = 0, \\
& \frac{1}{2\sqrt{5}}y_7(t) + \frac{1}{2}y_8(t) + \dot{y}_8(t) = 0, \\
& \frac{1}{2}y_9(t) + \dot{y}_9(t) = 0, \\
& y_0(0) = \frac{2}{7}, \quad y_1(0) = 0, \quad y_2(0) = 0, \quad y_3(0) = 0, \quad y_4(0) = \frac{\sqrt{5}}{14}, \\
& y_5(0) = 0, \quad y_6(0) = 0, \quad y_7(0) = 0, \quad y_8(0) = 0, \quad y_9(0) = 0.
\end{aligned}$$

- STEPS 8 and 9. Then, we solve the auxiliary system and compute the mean and the variance.

In Tables 2 and 3, we compare the exact values of the mean and the variance of the solution stochastic process to random i.v.p. (21) with respect to the numerical values obtained by adaptive gPC (columns μ adaptive gPC and σ^2 adaptive gPC, respectively) and PC using the Hermite polynomials as the trial basis (columns μ Hermite-PC, and σ^2 Hermite-PC, respectively). We observe that the results provided by adaptive gPC method are as good as the ones obtained by Hermite-PC.

Example 2. We deal with the random linear differential equation:

$$\dot{y}(t) + by(t) = c, \quad y(0) = \hat{y}_0, \quad (23)$$

whose exact values for the average and the variance are also available. We will assume that

- $b \sim B(2; 2)$,
- c has a uniform distribution on the interval $[2/3, 3/4]$, i.e., $c \sim U([2/3, 3/4])$,

Table 2: Comparison of the exact mean of the solution stochastic process to random i.v.p. (21) (Example 1) to the numerical values computed by adaptive gPC and Hermite-PC approaches.

t	μ exact	μ adaptive gPC	μ Hermite-PC
0.	0.2857142857	0.2857142857	0.2857142857
0.1	0.2718477865	0.2718477954	0.2718477413
0.2	0.2587835939	0.2587836030	0.2587834190
0.3	0.2464703217	0.2464703269	0.2464699379
0.4	0.2348601022	0.2348601060	0.2348594309
0.5	0.2239083330	0.2239083319	0.2239072925
0.6	0.2135734432	0.2135734447	0.2135719468
0.7	0.2038166775	0.2038166797	0.2038146312
0.8	0.1946018957	0.1946018979	0.1945991974
0.9	0.1858953868	0.1858953908	0.1858919258
1.	0.1776656974	0.1776657019	0.1776613545

Table 3: Comparison of the exact variance of the solution stochastic process to random i.v.p. (21) (Example 1) to the numerical values computed by adaptive gPC and Hermite-PC approaches.

t	σ^2 exact	σ^2 adaptive gPC	σ^2 Hermite-PC
0.	0.0255102040	0.0255102040	0.0254933498
0.1	0.0231426286	0.0231426300	0.0231268735
0.2	0.0211035898	0.0211035910	0.0210879613
0.3	0.0193424216	0.0193424220	0.0193262025
0.4	0.0178164432	0.0178164425	0.0177991206
0.5	0.0164896949	0.0164896912	0.0164709188
0.6	0.0153318747	0.0153318648	0.0153114237
0.7	0.0143174431	0.0143174203	0.0142951969
0.8	0.0134248702	0.0134248238	0.0134007870
0.9	0.0126360020	0.0126359163	0.0126101002
1.	0.0119355267	0.0119353795	0.0119078703

- $\hat{y}_0 \sim B(2; 5)$.

Following the same presentation as in the previous example, in Tables 4 and 5, we compare the exact values of the mean and the variance, respectively. Again, we observe that good results are achieved by adaptive gPC method.

Table 4: Comparison of the exact mean of the solution stochastic process to random i.v.p. (23) (Example 2) to the numerical values computed by adaptive gPC and Hermite-PC approaches.

t	μ exact	μ adaptive gPC	μ Hermite-PC
0.	0.2857142857	0.2857142857	0.2857142857
0.1	0.2027504518	0.2027504665	0.2027504104
0.2	0.1239261084	0.1239261194	0.1239259629
0.3	0.0489974095	0.0489974028	0.0489971227
0.4	-0.022263674	-0.022263687	-0.022264120
0.5	-0.090070451	-0.090070454	-0.090071055
0.6	-0.154622519	-0.154622519	-0.154623266
0.7	-0.216106707	-0.216106706	-0.216107567
0.8	-0.274697951	-0.274697949	-0.274698877
0.9	-0.330560102	-0.330560091	-0.330561029
1.	-0.383846677	-0.383846665	-0.383847522

In this example, the approximations for the mean and standard deviation of the solution can be computed for longer times using adaptive gPC. In Figure 1 (left), we show these approximations on the interval $[0, 100]$. On the right side, the exact mean and standard deviation are plotted on the same time interval.

In Figure 2 the relative error respect to the exact solution for the mean (left) and standard deviation (right) computed by adaptive gPC are represented. We observe that relative error in the variance increases over time.

Example 3. Next, we will consider a nonlinear i.v.p. based on the following random Riccati differential equation:

$$\dot{y}(t) + ay^2(t) + by(t) + c = 0, \quad y(0) = \hat{y}_0. \quad (24)$$

We will suppose that random model parameters have the following probability distributions:

Table 5: Comparison of the exact variance of the solution stochastic process to random i.v.p. (23) (Example 2) to the numerical values computed by adaptive gPC and Hermite-PC approaches.

t	σ^2 exact	σ^2 adaptive gPC	σ^2 Hermite-PC
0.	0.0255102040	0.0255102040	0.0254933498
0.1	0.0231394091	0.0231394095	0.0231236307
0.2	0.0210647289	0.0210647291	0.0210490321
0.3	0.0192155763	0.0192155761	0.0191992547
0.4	0.0175493194	0.0175493182	0.0175319036
0.5	0.0160443982	0.0160443945	0.0160256150
0.6	0.0146948275	0.0146948175	0.0146745624
0.7	0.0135058305	0.0135058076	0.0134840899
0.8	0.0124903952	0.0124903490	0.0124672668
0.9	0.0116665809	0.0116664960	0.0116421906
1.	0.0110554315	0.0110552863	0.0110298983

- $a \sim U([1/2, 2/3])$,
- $b \sim B(2; 2)$,
- $c \sim U([2/3, 3/4])$,
- $\hat{y}_0 \sim B(2; 5)$.

In Tables 6 and 7, we compare the exact values of the average and the variance, respectively. Once more, we observe that adaptive gPC method provides good approximations.

In the previous examples we have assumed each random input ζ_i follows a standard distribution, hence we have applied adaptive gPC taking as weight function into the inner product (9) the corresponding probability density function $f_{\zeta_i}(\zeta_i)$ whose analytic representation is well-known. This allowed us to compute successfully the expectation and variance of the solution. However, random parameters that do not belong to any standard statistical distributions often appear in practice, then adaptive gPC turns out particularly fruitful.

Table 6: Comparison of the exact average of the solution stochastic process to random i.v.p. (24) (Example 3) to the numerical values computed by adaptive gPC and Hermite-PC approaches.

t	μ exact	μ adaptive gPC	μ Hermite-PC
0.	0.2857142857	0.2857142857	0.2857142857
0.1	0.1980759440	0.1980759539	0.1980766519
0.2	0.1168994626	0.1168994605	0.1169005076
0.3	0.0409170822	0.0409170917	0.0409182898
0.4	-0.030899721	-0.030899714	-0.030898392
0.5	-0.099405455	-0.099405446	-0.099403949
0.6	-0.165327740	-0.165327721	-0.165325922
0.7	-0.229302586	-0.229302549	-0.229300249
0.8	-0.291901881	-0.291901818	-0.291898736
0.9	-0.353655864	-0.353655733	-0.353651529
1.	-0.415072650	-0.415072393	-0.415066619

Table 7: Comparison of the exact variance of the solution stochastic process to random i.v.p. (24) (Example 3) to the numerical values computed by adaptive gPC and Hermite-PC approaches.

t	σ^2 exact	σ^2 adaptive gPC	σ^2 Hermite-PC
0.	0.0255102040	0.0255102040	0.0254933498
0.1	0.0216489702	0.0216489712	0.0216382260
0.2	0.0188292728	0.0188292705	0.0188222912
0.3	0.0167281955	0.0167281900	0.0167232872
0.4	0.0151469745	0.0151469535	0.0151428818
0.5	0.0139655414	0.0139654797	0.0139612695
0.6	0.0131167575	0.0131166236	0.0131114709
0.7	0.0125721264	0.0125718420	0.0125650291
0.8	0.0123345580	0.0123339882	0.0123248301
0.9	0.0124362245	0.0124351284	0.0124229264
1.	0.0129405475	0.0129385258	0.0129225503

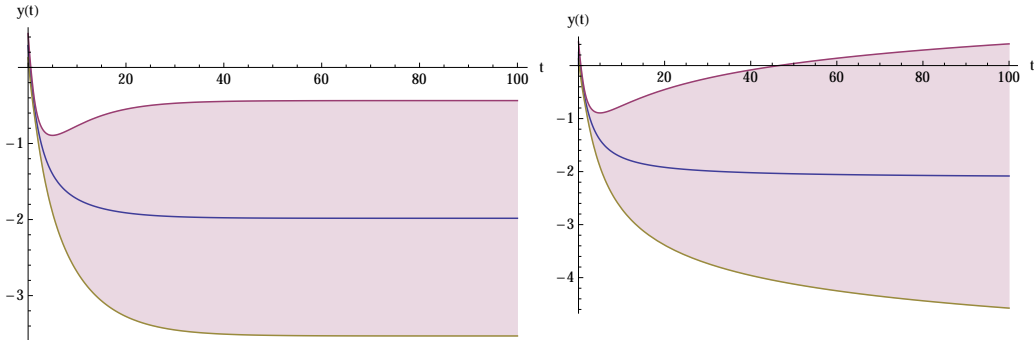


Figure 1: Mean (center line) and mean plus standard deviation (upper line) and mean minus standard deviation (lower line) on the interval $[0, 100]$ in the Example 2. Adaptive gPC approximation (left) and exact solution (right).

Example 4. *Let us consider a real problem based on the Newton's cooling law to illustrate this situation [2]. According to this thermodynamic law, when a body is heated up to the temperature T_0 and then, it is exposed to the medium temperature T_m , its temperature $T(t)$ in the instant t varies according to the i.v.p.*

$$\dot{T}(t) = K(T(t) - T_m), \quad T(0) = T_0, \quad (25)$$

where K denotes the cooling constant. When this model is applied in practical situations, the calibration of K is required. This calibration is performed after careful measurements which depend on the body composition as well as the accuracy of the measuring instruments. So, it is more realistic to assume that K is a random variable rather than a deterministic constant. Table 8 shows the values of K in a particular case.

In order to apply adaptive gPC, we first need to assign a statistical distribution to K . Kernel methods constitute a family of powerful techniques to approximate the probability density function of a random variable from sampled data that result very useful when standard distributions fail [16, 19, 23]. Many of these methods are implemented in software packages. Figure 3 shows the distribution function (left) and the probability density function (right) of the cooling random parameter K obtained from the sample collected in Table 8 by using the command `"SmoothKernelDistribution"` of Mathematica. We can see that the obtained probability density function does not seem to be any standard distribution.

Now, following the same development as in the previous examples we

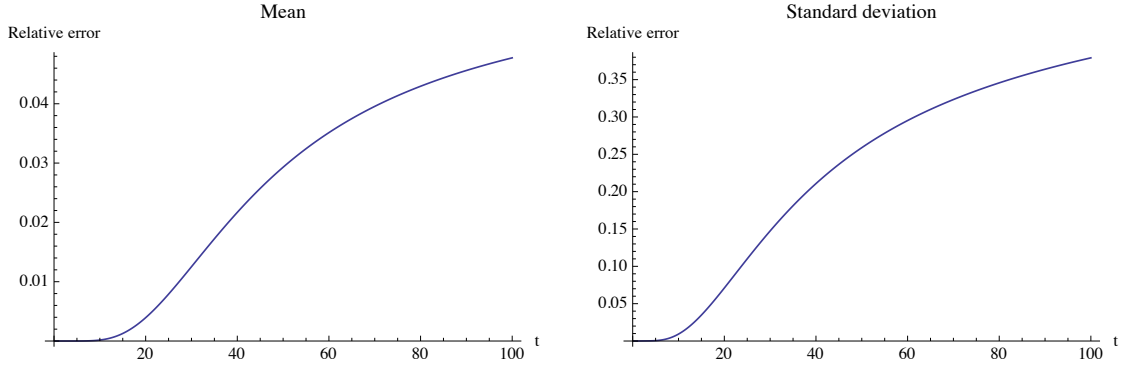


Figure 2: Relative error respect to the exact solution for the mean (left) and standard deviation (right) computed by adaptive gPC on the interval $[0, 100]$ in the Example 2. Note that scale in the vertical axis is different for each plot.

Table 8: Values of the cooling constant K in Example 4.

-0.0610092	-0.060129	-0.0565267	-0.0565262	-0.0547071	-0.0533779	-0.0523492
-0.0505862	-0.0500046	-0.0487714	-0.0484111	-0.0481034	-0.0472823	-0.0465562
-0.045926	-0.0458568	-0.0458198	-0.0453695	-0.0441748	-0.0438579	-0.0435833
-0.0433647	-0.0431569	-0.0423424	-0.0422292	-0.0418355	-0.0417759	-0.0411636
-0.0408778	-0.0413735	-0.0409582	-0.0407478	-0.0405775	-0.0406751	-0.0402963
-0.0399196	-0.0398389	-0.0397901	-0.0397549	-0.0394868	-0.0392476	-0.0392629
-0.0390598	-0.0388823	-0.0387293	-0.0388298	-0.0387068	-0.0388397	-0.0387445
-0.0386894	-0.0383818	-0.0383492	-0.0383374	-0.0383277	-0.0381127	-0.0381412
-0.0381901	-0.0382597	-0.0383306	-0.0381828	-0.0382918	-0.0381769	-0.038325
-0.0382222	-0.0384107	-0.0386236	-0.0383079	-0.038536	-0.0387903	-0.0385109
-0.0388053	-0.0385498	-0.038887	-0.038954	-0.0390134	-0.0391166	-0.0392393
-0.0400254	-0.0398597	-0.0400265	-0.0402468	-0.04009	-0.0403515	-0.0406095
-0.0408971	-0.0412168	-0.0415716	-0.0415449	-0.0424427	-0.042422	-0.0429344
-0.0429996	-0.0436032	-0.0442786	-0.0450384	-0.0458985	-0.0454253	-0.0456316

have computed approximations of the mean and standard deviation to the temperature $T(t)$ (in degrees Fahrenheit) for $0 \leq t \leq 73.5$ (in seconds). Figure 4 shows the corresponding results. In order to check the accuracy of the approximations provided by adaptive gPC, in this plot we have also represented by points the real sampled temperatures (see [2]). We observe adaptive gPC provides very good approximations.

Example 5. We will consider the SIRS (Susceptible-Infectious-Recovered-Susceptible) model for the transmission dynamics of Respiratory Syncytial Virus (RSV) proposed by Weber et al. in [24]. This model is based on the

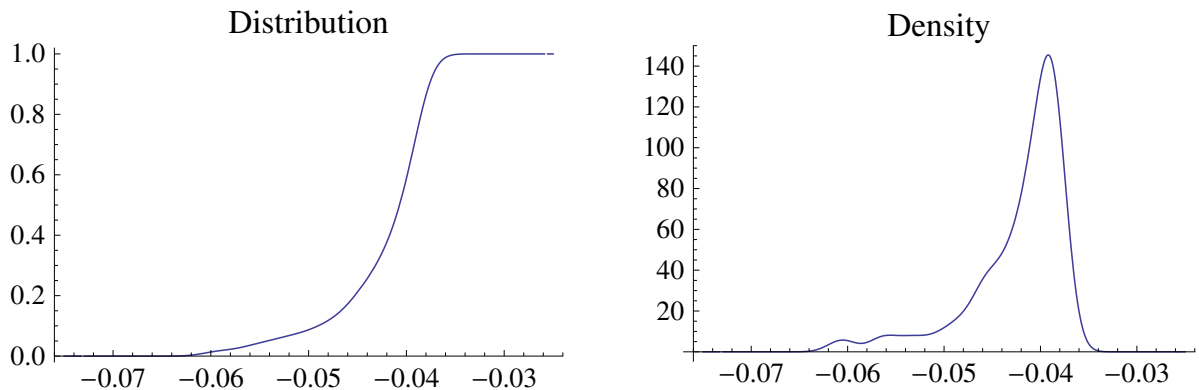


Figure 3: Distribution function (left) and probability density function (right) of the cooling random parameter K obtained with *Mathematica* using "SmoothKernelDistribution" command in the Example 4.

following non-autonomous system of differential equations

$$\begin{aligned}
 \dot{S}(t) &= \mu - \mu S(t) - \beta(t)S(t)I(t) + \gamma R(t), \\
 \dot{I}(t) &= \beta(t)S(t)I(t) - \nu I(t) - \mu I(t), \\
 \dot{R}(t) &= \nu I(t) - \mu R(t) - \gamma R(t), \\
 S(t_0) &= S_0, \\
 I(t_0) &= I_0, \\
 R(t_0) &= R_0,
 \end{aligned} \tag{26}$$

where $S(t)$, $I(t)$ and $R(t)$ are the population of susceptible, infectious and recovered, μ is the birth rate and it is supposed to be equal to the mortality rate, γ is the rate of loss of immunity, ν is the rate of loss of infectiousness, $\beta(t) = b_0(1 + b_1 \cos(2\pi t + \phi))$ is the infection transmission rate, being b_0 the average of transmission, b_1 the amplitude of the seasonal fluctuation and ϕ the seasonal phase, and t is the time in years.

In [24], the authors study the spread of RSV in Finland, obtaining the following parameter values: $\mu = 0.013$; $\gamma = 360/200 = 1.8$; $\nu = 36$; $b_0 = 44$; $b_1 = 0.36$; $\phi = 0.6$.

For this example, in order to apply adaptive gPC, let us assume that some of the previous parameters are random variables. Notice that this assumption is realistic since in practice these values are set from sampled data. Specifically we will assume that:

- $\gamma \sim U([360/210, 360/190])$,

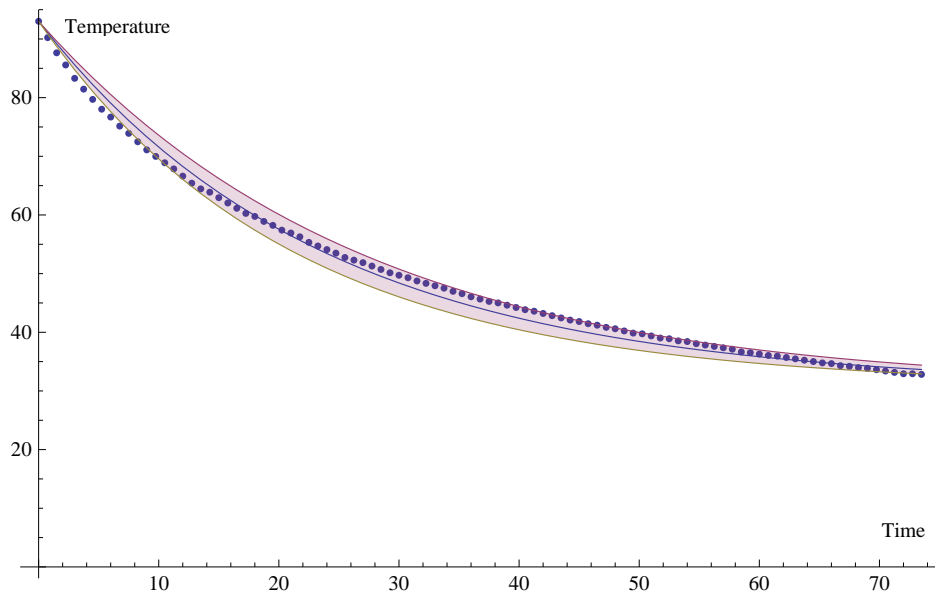


Figure 4: Approximation of the expectation (central line) and, expectation plus/minus standard deviation (upper/lower line), respectively of the temperature $T(t)$ (in degrees Fahrenheit) on the time-interval $[0, 73.5]$ (in seconds) in the Example 4. The points correspond to the real sampled temperature (see [2]).

- ν follows a Gaussian distribution of mean 36 and variance 2, i.e., $\nu \sim N(36; 2)$,
- $b_0 \sim U([42, 46])$,
- $b_1 \sim U([0.31, 0.41])$.

Notice that in this way, all the above random variables have as average the deterministic values given in [24].

Now, we will assume that 1% of total population is infectious, so initial conditions are given by $S_0 = 0.99$, $I_0 = 0.01$ and $R_0 = 0$. For our computations we have taken $p = r = 2$ and $[0, 5]$ as the time interval where the study was performed. In Figure 5 the mean of each component of the solution stochastic process together with itself minus/plus their standard deviation functions are represented. This provides us an idea of the variability of the average behaviour of each population for $0 \leq t \leq 5$. In Figure 6

the deterministic solution to the mean of the solution stochastic process are compared. We see that, in this case, both are similar.

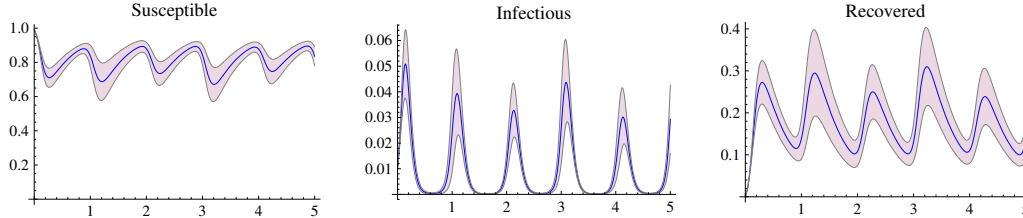


Figure 5: Mean percentage (center line), mean plus standard deviation percentage (upper line) and mean minus standard deviation percentage (lower line) functions on the time-interval $[0, 5]$ of Susceptible (left), Infectious (center) and Recovered (right) subpopulations obtained applying the adaptive gPC to the RSV model (26).

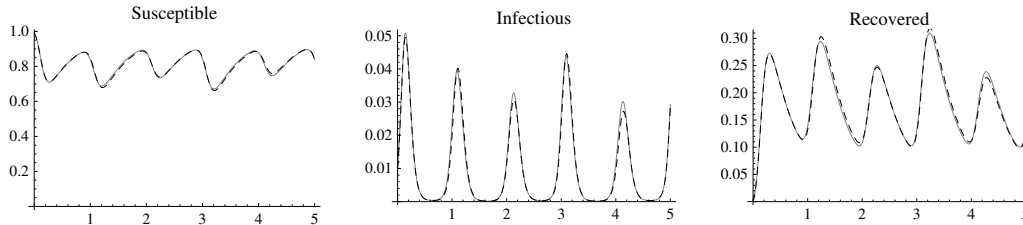


Figure 6: Comparison of the deterministic solution (dashed line) and the mean of the solution stochastic process (continuous line) on the interval $[0, 5]$ of Susceptible (left), Infectious (center) and Recovered (right) subpopulations obtained after solving the auxiliary system of differential equations associated to the RSV model (26).

In order to assess the quality of the approximations provided by adaptive gPC, the absolute error for both, the mean and the standard deviation of the solution, have been compared against Monte Carlo method using 100 000 simulations. The results are plotted in Figures 7 and 8.

All the above examples illustrate the ability of the proposed adaptive gPC method to handle models described by different types of differential equations containing randomness in several parameters with different probability distributions.

6. Conclusions

In this paper we have studied a variation of gPC method, referred to as adaptive gPC. This method is able to handle random inputs following both

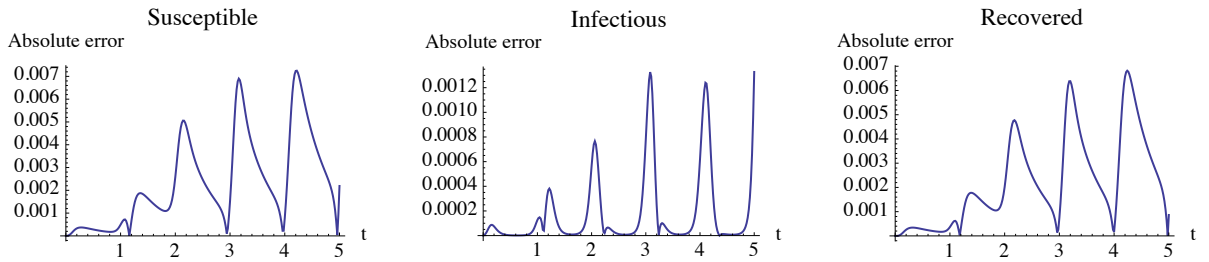


Figure 7: Absolute error of the mean of the solution to the RSV model (26) computed by adaptive gPC and Monte Carlo method using 100 000 simulations on the interval $[0, 5]$.

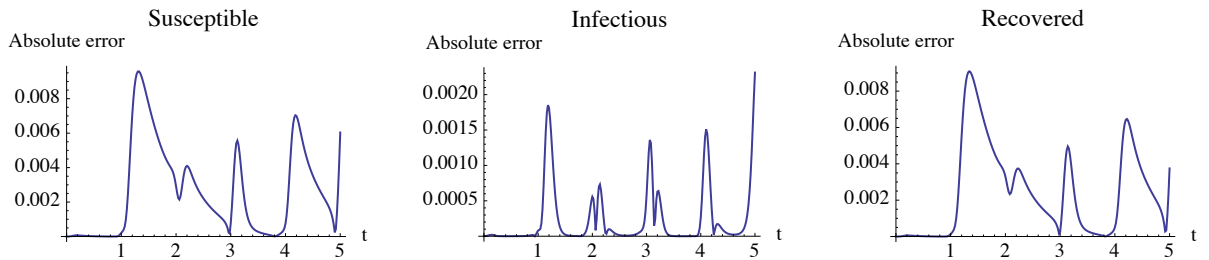


Figure 8: Absolute error of the standard deviation of the solution to the RSV model (26) computed by adaptive gPC and Monte Carlo method using 100 000 simulations on the interval $[0, 5]$.

standard and non-standard probability distributions. Under this approach, the random inputs are represented by Fourier expansion series of orthonormal polynomials built using an inner product where the weight function is precisely the probability density function of the corresponding input. This expansion only requires two terms, which facilitates solving the auxiliary system of differential equations and better numerical accuracy can be achieved. The proposed technique has been illustrated through a wide range of examples.

Finally, we want to point out that the aim of the paper is to present a method that provides researchers that do not know the foundations of gPC, a step-by-step computational approach method to implement an adaptive gPC method to be used in random continuous models.

Acknowledgements

This work has been partially supported by the Spanish M.C.Y.T. grants: MTM2009-08587 and DPI2010-20891-C02-01; FIS PI-10/01433; Universitat Politècnica de València grants: PAID06-11-2070 and PAID-00-11-2753, and Universitat de València grant: UV-INV-PRECOMP12-80708.

References

- [1] <http://www.wolfram.com/products/mathematica>, 2012.
- [2] <http://www.math.montana.edu/frankw/ccp/cases/newton/frstlook.htm>, 2012.
- [3] G. Adomian, Solving Frontier Problems of Physics: The Decomposition Method, Kluwer, Boston, 1994.
- [4] L. Arnold, Stochastic Differential Equations: Theory and Applications, John Wiley & Sons, New York, 1974.
- [5] N. Bellomo, R. Riganti, Nonlinear Stochastic Systems in Physics and Mechanics, World Scientific, Singapore, 1987.
- [6] G. Calbo, J.C. Cortés, L. Jódar, Mean square power series solution of random linear differential equations, *Comput. Math. Appl.* 59 (2010) 559–572.
- [7] R.H. Cameron, W.T. Martin, The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals, *Ann. Math.* 48 (1947) 385–392.
- [8] J.C. Cortés, L. Jódar, L. Villafuerte, Numerical solution of random differential initial value problems: Multistep methods, *Math. Meth. Appl. Sci.* 34 (2011) 63–75.
- [9] B.J. Deusschere, H.N. Najm, P.P. Pébay, O.M. Knio, R.G. Ghanem, O.P.L. Maître, Numerical challenges in the use of polynomial chaos representations for stochastic processes, *SIAM J. Sci. Comput.* 26 (2004) 698–719.

- [10] O.G. Ernst, A. Mugler, H.J. Starkloff, E. Ullmann, On the convergence of generalized polynomial chaos expansions, *ESAIM-Math. Model. Numer. Anal.-Model. Math. Anal. Numer.* 46 (2012) 317–339.
- [11] M. Gerritsma, J.B. van der Steen, P. Vos, G. Karniadakis, Time-dependent generalized polynomial chaos, *J. Comput. Phys.* 229 (2010) 8333–8363.
- [12] R. Ghanem, P.D. Spanos, *Stochastic Finite Elements: A Spectral Approach*, Dover Publications, Mineola, New York, 1991.
- [13] I.I. Gihman, A.V. Skorohod, *Stochastic Differential Equations*, Springer-Verlag, Berlin, 1972.
- [14] A. Jahedi, G. Ahmadi, Application of Wiener-Hermite expansion to non stationary random vibrations of a Duffing oscillator, *J. Appl. Mech.* 50 (1983) 436–442.
- [15] P. Kloeden, E. Platen, *Numerical Solution of Stochastic Differential Equations*, Springer, Berlin, 1999.
- [16] D.P. Kroese, T. Taimre, Z.I. Botev, *Handbok of Monte Carlo Methods*, Wiley, Hoboken, New Jersey, 2011.
- [17] M. Loève, *Probability Theory*, Van Nostrand, Princeton, New Jersey, 1963.
- [18] A.H. Nayfeh, *Problems in Perturbation*, John Wiley & Sons, New York, 1985.
- [19] J.S. Simonoff, *Smoothing Methods in Statistics*, Springer-Verlag, New York, 1996.
- [20] T.T. Soong, *Random Differential Equations in Science and Engineering*, Academic Press, New York, 1973.
- [21] P.T.D. Spanos, W.D. Iwan, The existence and uniqueness of solution generated by equivalent linearization, *Int. J. Non-Linear Mech.* 13 (1978) 71–78.
- [22] R. Walpole, R. Myers, S. Myers, *Probability and Statistics for Engineers and Scientists*, Prentice Hall, New Jersey, 6th edition, 1998.

- [23] M.P. Wand, M.C. Jones, Kernel Smoothing, Chapman & Hall, London, 1995.
- [24] A. Weber, M. Weber, P. Milligan, Modeling epidemics caused by respiratory syncytial virus (RSV), *Math. Biosci.* 172 (2001) 95–113.
- [25] N. Wiener, The homogeneous chaos, *Am. J. Math.* 60 (1938) 897–936.
- [26] D. Xiu, G. Karniadakis, The Wiener-Askey polynomial chaos for stochastic differential equations, *SIAM J. Sci. Comput.* 24 (2002) 619–644.