Proceedings of the IMAC-XXVII
February 9-12, 2009 Orlando, Florida USA

COUPLING POLYNOMIAL CHAOS EXPANSIONS WITH GAUSSIAN PROCESS EMULATORS: AN INTRODUCTION

F.A. DiazDelaO*
School of Engineering
Swansea University, U. K.

S. Adhikari†
School of Engineering
Swansea University, U. K.

ABSTRACT

The present document introduces some ideas that could eventually lead to the use of Gaussian process emulators as an alternative to reduce the computational cost of the polynomial chaos expansion method. A simple stochastic mechanics problem is solved with the aid of an emulation algorithm. The results and challenges in adopting the proposed coupling are discussed.

1 INTRODUCTION

Unless uncertainty is incorporated, most mathematical models of complex engineering systems can be considered to be incomplete, no matter how sophisticated the constitutive models or powerful the employed computational tools. The intrinsic randomness of the material or the loads involved may be such that deterministic modeling delivers a highly inaccurate representation of the physical reality. Since the finite element method has been extensively tested in the context of deterministic engineering mechanics, one natural extension of this approach is to include random parameters in the partial differential equations governing the system's response. The governing stochastic partial differential equations can be discretized using the stochastic finite element method [1–14]. For further details, refer to the review papers [15–20]. In general, the resulting equilibrium equation is

$$\mathbf{K}(\mathbf{x},\theta)\mathbf{u}(\mathbf{x},\theta) = \mathbf{f}(\mathbf{x},\theta) \tag{1}$$

where $\mathbf{K}(\mathbf{x},\theta) \in \mathbb{R}^{N\times N}$ is a random matrix, $\mathbf{u}(\mathbf{x},\theta) \in \mathbb{R}^{N}$ is the random response vector, $\mathbf{f}(\mathbf{x},\theta) \in \mathbb{R}^{N}$ is the (possibly) random forcing vector, and N is the number of degrees of freedom in the finite element mesh. The arguments $\mathbf{x}$ and θ correspond to the spatial and random dimensions, respectively.

Several methods to approximate the solution of Eq. (1) and consequently calculate the statistics of the response $\mathbf{u}(\mathbf{x},\theta)$ are available in the literature. These include Monte Carlo simulation techniques [21, 22], perturbation methods through Taylor series [1, 3, 23], expansion methods through Neumann series [6, 24] and other analytical methods [25–29]. Despite their theoretical appeal, their implementation presents at least one of the following disadvantages: a) lack of the geometrical appeal presented by the finite element method; b) limited applicability due to restrictive analytical constraints; c) non-guaranteed convergence of the Talyor and Neumann series involved; d) potentially high computational cost.

Aiming to tackle some of the above mentioned disadvantages, Ghanem and Spanos [2] proposed the *polynomial chaos expansion method*, also referred to simply as *polynomial chaos*. Later this approach was extended by several authors [11, 13, 14, 30–32]. It essentially consists in representing each component of the random displacement vector $\mathbf{u}(\mathbf{x},\theta)$ as a series of orthogonal polynomials $\{\Psi_j(\theta)\}_{j=0}^{\infty}$ in the standard normal variables $\{\xi_k(\theta)\}_{k=1}^{\infty}$, such that

$$\mathbf{u}(\mathbf{x},\theta) = \sum_{j=0}^{P-1} \mathbf{c}_j(\mathbf{x})\Psi_j(\theta) \tag{2}$$

*Postgraduate student. School of Engineering, Swansea University, Singleton Park, Swansea SA2 8PP, UK. Email: diazdelao@yahoo.com.mx
†Professor of Aerospace Engineering. School of Engineering, Swansea University, Singleton Park, Swansea SA2 8PP, UK. Email: s.adhikari@swansea.ac.uk

where each $\mathbf{c}_j$ is a deterministic vector in $\mathbb{R}^n$. Equation (2) is then substituted in (1), and $\mathbf{K}(\mathbf{x}, \theta)$ is discretized using the following truncated Karhunen-Loève expansion.

$$\mathbf{K}(\mathbf{x}, \theta) = \sum_{i=0}^{M} \mathbf{K}_i(\mathbf{x}) \xi_i(\theta) \tag{3}$$

where each $\mathbf{K}_i(\mathbf{x})$ is assembled from elementary matrices as in the deterministic finite element method. The approximate solution is obtained by minimizing the underlying residual

$$\mathcal{R}(\mathbf{x}, \theta) = \sum_{i=0}^{M} \sum_{j=0}^{P-1} \mathbf{K}_i \mathbf{c}_j(\mathbf{x}) \xi_i(\theta) \Psi_j(\theta) - \mathbf{f}(\mathbf{x}, \theta) \tag{4}$$

by means of a projection onto the space spanned by $\{\Psi_j(\theta)\}_{j=0}^{P-1}$.

Ghanem and Spanos [2] claim that polynomial chaos is more versatile than the series representation methods in the sense that it is less restrictive and offers better convergence and geometrical clarity. However, the problem of computational cost remains. As shown by Sudret and Der Kiureghian [9], the number of polynomials P in the expansion grows very rapidly for small increases in the order of the polynomials p, and in the number of terms in the Karhunen-Loève expansion M, namely

$$P(p, M) = \sum_{k=0}^{p} \binom{M + k - 1}{k} \tag{5}$$

This pitfall has an important implication, pointed out by Debusschere et al. [33]. Unless high order polynomials are employed, the accuracy of the polynomial chaos representation may be inaccurate and unstable, that is, produce non-physical values for the parameters modeled. Unfortunately, the use of high values of p may be computationally intractable. Additionally, in an engineering system with a large number of degrees of freedom, the computation time can rapidly become prohibitively expensive. In this setup, a *Gaussian process emulator* can be an inexpensive surrogate model of a computer code implementation, also known as simulator, of polynomial chaos.

2 GAUSSIAN PROCESS EMULATORS

Let $\eta(\cdot)$ be a computer code, or simulator, such that for every pair $\mathbf{x}$, it returns the output $\mathbf{y}(\mathbf{x})$. If the simulator is computer intensive, it can only be evaluated at a limited number of inputs. This complexity allows $\eta(\cdot)$ to be regarded as a random variable. Consequently, assume that it admits the following representation

$$\eta(\cdot) = \mathbf{h}(\cdot)^T \boldsymbol{\beta} + Z(\cdot) \tag{6}$$

where $\mathbf{h}(\cdot)$ is a vector of known functions, $\boldsymbol{\beta}$ is a vector of unknown coefficients, and $Z(\cdot)$ is a stochastic process with mean zero and covariance function $Cov(\cdot, \cdot)$.

An emulator is a statistical approximation to the simulator. Not only does it approximate $\eta(\cdot)$, it provides a probability distribution for it. Broadly speaking, emulation works in the following way: A small and carefully selected set of code runs $\{\mathbf{x}_\ell, \eta(\mathbf{x}_\ell)\}_{\ell=1}^{n}$ is treated as training data used to update a prior distribution of the simulator. Following Haylock and O'Hagan [34], this prior distribution takes the form of a Gaussian stochastic process, namely

$$\eta(\cdot) | \boldsymbol{\beta}, \sigma^2 \sim N(\mathbf{h}(\cdot)^T \boldsymbol{\beta}, \sigma^2 C(\cdot, \cdot)) \tag{7}$$

where the correlation function $C(\cdot, \cdot)$ is such that

$$C(\mathbf{x}, \mathbf{x}') = e^{-(\mathbf{x} - \mathbf{x}')^T \mathbf{B} (\mathbf{x} - \mathbf{x}')} \tag{8}$$

and $\mathbf{B}$ is a diagonal positive definite matrix of smoothness parameters.

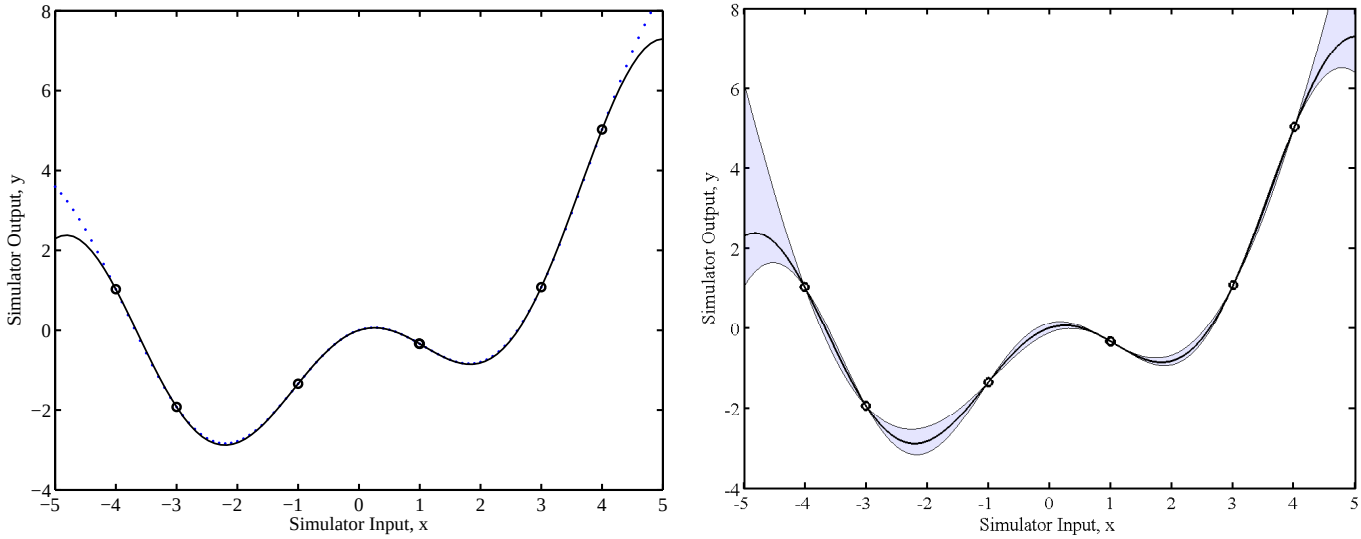
Definition. Gaussian stochastic process: Let $\mathcal{X} \subseteq \mathbb{R}^d$. Then $Z(\mathbf{x})$ for $\mathbf{x} \in \mathcal{X}$ is a Gaussian stochastic process if for any $L \geq 1$ and any choice $\{\mathbf{x}_1, \ldots, \mathbf{x}_L\} \subseteq \mathcal{X}$, the vector $[Z(\mathbf{x}_1), \ldots, Z(\mathbf{x}_L)]^T$ has a multivariate normal distribution.

After conditioning on the training data and updating using standard integration techniques [34], the mean of the resulting posterior distribution provides a fast approximation to the output of the simulator at any untried input, whereas it returns the known value of the simulator at each of the initial runs. Such posterior distribution is

$$\eta(\cdot)|\mathbf{y}, \sigma^2 \sim N(m^{**}(\cdot), \sigma^2 C^{**}(\cdot,\cdot)) \tag{9}$$

where $\mathbf{y} = [\mathbf{y}_1 = \eta(\mathbf{x}_1), \ldots, \mathbf{y}_n = \eta(\mathbf{x}_n)]^T$. Very conveniently, this posterior is also a Gaussian process distribution. For the explicit expressions of $m^{**}(\cdot)$ and $C^{**}(\cdot,\cdot)$, and the estimation of σ, please refer to the appendix.

For illustration of the above, take a trivial one-dimensional simulator and suppose for a moment it is computer intensive. Figure 1(a) depicts the case when 6 training runs (the circles) are used. The predictive mean of the emulator (the dots) approximates the real values of the simulator (the solid line) at several untried inputs throughout the input domain. On the other hand, it returns the exact value of the simulator at each of the training points. Figure 1(b) shows upper and lower bounds of two standard deviations for the predictive mean. Note how the uncertainty is equal to zero in each of the training runs, as it would be expected, since the emulator reproduces the simulator's output at these points.



(a) Predictive mean using 6 training runs.

(b) Uncertainty about the predictive mean.

Fig. 1: Approximation to the simulator $\mathbf{y} = 0.5\mathbf{x} - \mathbf{x}\sin(\mathbf{x})$ using the predictive mean of a Gaussian process emulator and uncertainty about this mean; (-): output of the simulator, (o): training runs, $(\cdots)$: emulator's predictive mean.

3 COUPLING

Gaussian process emulators have already been implemented in a number of scientific fields. To name a few, Challenor et al. [35] emulated what they consider to be a moderately complex climate model. Rougier [36] presented another application to a climate model. Bates et al. [37] emulated a model of a complete revolution of a piston's shaft. Haylock and O'Hagan [34] applied emulators to a model of doses to organs of the body after ingestion of a radioactive substance. It is important to note that the simulators involved in these studies are deterministic, as they always produce the same output **y** given the same input **x**. The process is summarized below.

Emulation Algorithm

1. Select n design points $\mathbf{x}_1, \ldots, \mathbf{x}_n$.
2. Obtain the vector of observations $\mathbf{y} = [\mathbf{y}_1 = \eta(\mathbf{x}_1), \ldots, \mathbf{y}_n = \eta(\mathbf{x}_n)]^T$.

3. Update the prior distribution (7), which contains subjective information, by adding the objective information **y** and thus obtaining the distribution (9). This will enable the calculation of the predictive mean $m^{**}(\cdot)$, given the data **y**. As already mentioned, such mean is a fast approximation of $\eta(\mathbf{x})$ for any **x**.

Since a polynomial chaos simulator is not deterministic, the algorithm above should not be applied directly. An idea by Kleijnen [38] and Kleijnen and van Beers [39] is to use m repetitions of the simulator at each of the n design points $\mathbf{x}_1, \ldots, \mathbf{x}_n$. That way, if $\mathbf{y}_{\ell k}$ denotes the ℓ-th realization at the k-th design point, then the mean response $\overline{\mathbf{y}}_\ell = \frac{1}{m}\sum_{k=1}^{m} \mathbf{y}_{\ell k}$ and the variance of the response $\mathbf{S}_\ell = \frac{1}{m-1}\sum_{k=1}^{m}(\overline{\mathbf{y}}_\ell - \mathbf{y}_{\ell k})^2$ are used to standardize the simulator's output with the transformation

$$\widetilde{\mathbf{y}}_\ell = \frac{\mathbf{y}_\ell - \overline{\mathbf{y}}_\ell}{\sqrt{\mathbf{S}_\ell / m^2}} \tag{10}$$

The algorithm above is then applied to the detrended training data $\{\mathbf{x}_\ell, \widetilde{\mathbf{y}}_\ell\}_{\ell=1}^{n}$.

It can be shown [9] show that the polynomial chaos expansion method eventually casts the system in Eq.(1) as a system of the form

$$\mathcal{K} \cdot \mathcal{U} = \mathcal{F} \tag{11}$$

where each of the P components of $\mathcal{U} = [\mathcal{U}_0, \ldots, \mathcal{U}_{P-1}]^T$ is N-dimensional and consequently the global stiffness matrix $\mathcal{K}$ is of size $NP \times NP$. As the order of the polynomials or the number of terms in the Karhunen-Loève expansion increase, the solution of the system (11) becomes increasingly expensive. The problem is then to solve for only a small subset of nodes such that the corresponding solutions will be used as input for the emulation algorithm.

4 NUMERICAL INVESTIGATION

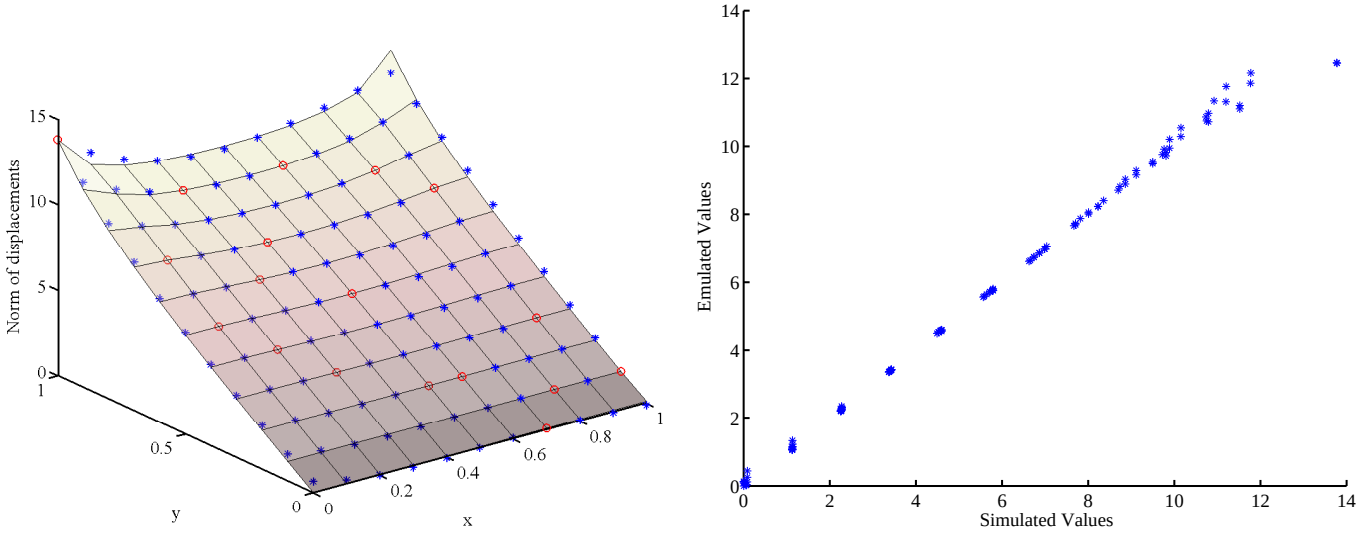
A polynomial chaos simulator in Matlab [9] was used to solve a problem found in Ghanem and Spanos [2]. The problem involves a cantilever thin square plate subject to a uniform in-plane tension along the free edge. The modulus of elasticity is modeled as a Gaussian random process with known mean and known exponential covariance structure. The aim was to calculate the norm of the displacement of each node in the finite element model of the plate. Such norm can be calculated by, for example, the Euclidean distance between the original and final positions of a given node. The plate was assumed to coincide with the set $[0, 1] \times [0, 1]$. Various experiments were run, varying the number of nodes in the finite element mesh, the order of the polynomial chaos expansion, and the order of the Karhunen-Loève expansion. The following table illustrates how quickly can the method become computationally intractable, even for the seemingly simple case of 25 nodes. The table shows the number of nodes (each with two degrees of freedom), the time in seconds employed to assemble the global stiffness matrix $\mathcal{K}$ and its size for a given order of the polynomial chaos expansion. The numbers in parenthesis indicate the case when Matlab returned an out-of-memory message.

No. nodes	p	Secs. Global	Size Global
25	2	0.73	750^2
25	4	14.2	$3{,}500^2$
25	6	301.44	$10{,}500^2$
25	8	1,722.73	$24{,}750^2$
25	10	7,420.65	($50{,}050^2$)
64	2	4.57	$1{,}920^2$
64	4	294.28	$8{,}960^2$
64	6	2,937.46	$26{,}880^2$
64	8	16,409.48	($63{,}360^2$)
121	2	30.27	$3{,}630^2$
121	4	1,289.50	$16{,}940^2$
121	6	12,464.53	($50{,}820^2$)

TABLE 1: Computation time (seconds) of the simulator, for different orders of the polynomial chaos expansion and increasing number of nodes. The order of the Karhunen-Loève expansion is 4.

Given the norms of the displacements of a small number of nodes, an emulator was constructed to infer the corresponding norms in the rest of the nodes. The nodes were selected by generating a Latin hypercube [40] in $[0, 1] \times [0, 1]$ and taking the closest node to each point generated, aiming to spread evenly the set of "design nodes" across the plate. The number of design

nodes was chosen to be around 15% of the total number of nodes. Figure 2(a) shows the norms of the node displacements as a function of the nodes' location, for the case of $p = 4$, $M = 4$ and 121 nodes. The circles correspond to the displacements of the design nodes, whereas the stars are emulated norms of displacements for the rest of the nodes. Figure 2(b) shows a comparison between the emulated norms of displacements and their actual value, that is, obtained by running the simulator.



(a) Norms of the displacements.

(b) Comparison between emulated and simulated norms.

Fig. 2: Norms of the displacements; (o): corresponding to design nodes, ($\star$):emulated norm of displacement. The scatterplot compares each of the emulated norms with its actual, or simulated value.

5 CONCLUSIONS

The coupling of the polynomial chaos expansion method with Gaussian process emulators was applied to a simple problem in the context of the stochastic finite element method. Several approaches were combined in an attempt to provide an inexpensive alternative to the use polynomial chaos expansion in structural dynamics. Good correspondence between the emulated and the actual data was found for a particular example of the problem. However, many important challenges remain before it can be claimed that Gaussian process emulators are a viable option to improve the computation time of the polynomial chaos expansion method. First, the idea of detrending the data with the transformation (10) may present a disadvantage. It is assumed that it is possible to obtain m repetitions of the simulator in each of the design points, although the convenience of doing this relies completely in the computational cost of the simulator. Thus, it could be the case that m is too small to start with. Second, suitable measures should be developed to assess the adequacy of the emulator. Even for the case of deterministic simulators, this is an area in current development. Finally, the problem of solving the system (11) for a relatively small number of nodes in order to use these solutions as design points has to be looked at carefully. Recently, Sarkar et al. [41] have developed a stochastic domain decomposition technique that could prove suitable for this purpose. Solving reduced versions of system (11) is appealing although it is necessary to determine its computational cost compared against the traditional polynomial chaos expansion method.

ACKNOWLEDGEMENTS

FADO gratefully acknowledges the support of the Engineering and Physical Sciences Research Council (EPSRC) for the award of a studentship through an Ideas Factory grant and the Consejo Nacional de Ciencia y Tecnología (CONACYT) for the award of a scholarship from the Mexican Government. SA gratefully acknowledges the support of EPSRC through the award of an Advanced Research Fellowship and The Leverhulme Trust for the award of the Philip Leverhulme Prize.

APPENDIX

In this section the process of updating the prior distribution (7) to obtain the posterior distribution (9) is outlined. Define $\mathbf{H} = [\mathbf{h}(\mathbf{x}_1), \ldots, \mathbf{h}(\mathbf{x}_n)]^T$ and $\mathbf{A} \in \mathbb{R}^{n \times n}$ with $\mathbf{A}_{\ell j} = C(x_\ell, x_j)\ \forall \ell, j \in \{1, \ldots, n\}$. Thus,

$$\mathbf{y}|\boldsymbol{\beta}, \sigma^2 \sim N(\mathbf{H}\boldsymbol{\beta}, \sigma^2 \mathbf{A}) \tag{12}$$

To incorporate the objective information $\mathbf{y}$ and obtain the distribution of $\eta(\cdot)|\mathbf{y}, \boldsymbol{\beta}, \sigma^2$, use of the following result, given in Krzanowski [42].

Theorem. Let $\mathbf{z} \in \mathbb{R}^n$ be a random vector such that $\mathbf{z} \sim N(\boldsymbol{\mu}, \Sigma)$. Partition $\mathbf{z}$ as $(\mathbf{z}_1, \mathbf{z}_2)^T$, where $\mathbf{z}_1 \in \mathbb{R}^p$ and $\mathbf{z}_2 \in \mathbb{R}^{n-p}$. Consequently, partition $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)^T$ and $\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$, so that $\mathbf{E}[\mathbf{z}_j] = \boldsymbol{\mu}_j$ and $Cov(\mathbf{z}_j, \mathbf{z}_k) = \Sigma_{jk}$. Then, $\mathbf{z}_1|\mathbf{z}_2 \sim N(\widetilde{\boldsymbol{\mu}}, \widetilde{\Sigma})$, where $\widetilde{\boldsymbol{\mu}} = \boldsymbol{\mu}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{z}_2 - \boldsymbol{\mu}_2)$ and $\widetilde{\Sigma} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$.

From this result, it follows that

$$\eta(\cdot)|\mathbf{y}, \boldsymbol{\beta}, \sigma^2 \sim N(m^*(\cdot), \sigma^2 C^*(\cdot, \cdot)) \tag{13}$$

where

$$m^*(x) = \mathbf{h}(x)^T\boldsymbol{\beta} + \mathbf{t}(x)\mathbf{A}^{-1}(\mathbf{y} - \mathbf{H}\boldsymbol{\beta}) \tag{14}$$

$$C^*(\mathbf{x}, \mathbf{x}') = C(\mathbf{x}, \mathbf{x}') - \mathbf{t}(\mathbf{x})^T A^{-1}\mathbf{t}(\mathbf{x}') \tag{15}$$

$$\mathbf{t}(\mathbf{x}) = [C(\mathbf{x}, \mathbf{x}_1), \ldots, C(\mathbf{x}, \mathbf{x}_n)]^T \tag{16}$$

Removing the conditioning on $\boldsymbol{\beta}$ using standard integration techniques [34], obtain the posterior distribution

$$\eta(\cdot)|\mathbf{y}, \sigma^2 \sim N(m^{**}(\cdot), \sigma^2 C^{**}(\cdot, \cdot)) \tag{17}$$

where

$$m^{**}(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T\widehat{\boldsymbol{\beta}} + \mathbf{t}(\mathbf{x})\mathbf{A}^{-1}(\mathbf{y} - \mathbf{H}\widehat{\boldsymbol{\beta}}) \tag{18}$$

$$C^{**}(\mathbf{x}, \mathbf{x}') = C^*(\mathbf{x}, \mathbf{x}') + (\mathbf{h}(\mathbf{x})^T - \mathbf{t}(\mathbf{x})^T\mathbf{A}^{-1}\mathbf{H})(\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H})^{-1}(\mathbf{h}(\mathbf{x}')^T - \mathbf{t}(\mathbf{x}')^T\mathbf{A}^{-1}\mathbf{H})^T \tag{19}$$

$$\widehat{\boldsymbol{\beta}} = (\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{A}^{-1}\mathbf{y} \tag{20}$$

To estimate σ in (9), let q be the rank of $\mathbf{H}$. Then calculate

$$\widehat{\sigma}^2 = \frac{\mathbf{y}^T(\mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{H}(\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H})\mathbf{H}^T\mathbf{A}^{-1})\mathbf{y}}{n - q - 2} \tag{21}$$

REFERENCES

[1] **Shinozuka, M.** and **Yamazaki, F.**, *Stochastic finite element analysis: an introduction*, *Stochastic structural dynamics: Progress in theory and applications*, edited by Ariaratnam, S. T., Schuëller, G. I. and Elishakoff, I., Elsevier Applied Science, London, 1998.

[2] **Ghanem, R.** and **Spanos, P.**, Stochastic Finite Elements: A Spectral Approach, Springer-Verlag, New York, USA, 1991.

[3] **Kleiber, M.** and **Hien, T. D.**, The Stochastic Finite Element Method, John Wiley, Chichester, 1992.

[4] **Manohar, C. S.** and **Adhikari, S.**, *Dynamic stiffness of randomly parametered beams*, Probabilistic Engineering Mechanics, Vol. 13, No. 1, pp. 39–51, January 1998.

[5] **Manohar, C. S.** and **Adhikari, S.**, *Statistical analysis of vibration energy flow in randomly parametered trusses*, Journal of Sound and Vibration, Vol. 217, No. 1, pp. 43–74, 1998.

[6] **Adhikari, S.** and **Manohar, C. S.**, *Dynamic analysis of framed structures with statistical uncertainties*, International Journal for Numerical Methods in Engineering, Vol. 44, No. 8, pp. 1157–1178, 1999.

[7] **Adhikari, S.** and **Manohar, C. S.**, *Transient dynamics of stochastically parametered beams*, ASCE Journal of Engineering Mechanics, Vol. 126, No. 11, pp. 1131–1140, November 2000.

[8] **Haldar, A.** and **Mahadevan, S.**, Reliability Assessment Using Stochastic Finite Element Analysis, John Wiley and Sons, New York, USA, 2000.

[9] **Sudret, B.** and **Der-Kiureghian, A.**, *Stochastic Finite Element Methods and Reliability*, Tech. Rep. UCB/SEMM-2000/08, Department of Civil & Environmental Engineering, University Of California, Berkeley, November 2000.

[10] **Gupta, S.** and **Manohar, C. S.**, *Dynamic stiffness method for circular stochastic Timoshenko beams*, Journal of Sound and Vibration, Vol. 253, No. 10, pp. 1051–1085, 2002.

[11] **Nair, P. B.** and **Keane, A. J.**, *Stochastic reduced basis methods*, AIAA Journal, Vol. 40, No. 8, pp. 1653–1664, 2002.

[12] **Elishakoff, I.** and **Ren, Y. J.**, Large Variation Finite Element Method for Stochastic Problems, Oxford University Press, Oxford, U.K., 2003.

[13] **Sachdeva, S. K., Nair, P. B.** and **Keane, A. J.**, *Comparative study of projection schemes for stochastic finite element analysis*, Computer Methods in Applied Mechanics and Engineering, Vol. 195, No. 19-22, pp. 2371–2392, 2006.

[14] **Sachdeva, S. K., Nair, P. B.** and **Keane, A. J.**, *Hybridization of stochastic reduced basis methods with polynomial chaos expansions*, Probabilistic Engineering Mechanics, Vol. 21, No. 2, pp. 182–192, 2006.

[15] **Matthies, H. G., Brenner, C. E., Bucher, C. G.** and **Soares, C. G.**, *Uncertainties in probabilistic numerical analysis of structures and solids - Stochastic finite elements*, Structural Safety, Vol. 19, No. 3, pp. 283–336, 1997.

[16] **Manohar, C. S.** and **Ibrahim, R. A.**, *Progress in structural dynamics with stochastic parameter variations: 1987 to 1998*, Applied Mechanics Reviews, ASME, Vol. 52, No. 5, pp. 177–197, May 1999.

[17] **Matthies, H. G.** and **Bucher, C.**, *Finite elements for stochastic media problems*, Computer Methods in Applied Mechanics and Engineering, Vol. 168, No. 1-4, pp. 3–17, 1999.

[18] **Manohar, C. S.** and **Gupta, S.**, *Modeling and evaluation of structural reliability: Current status and future directions*, *Research reviews in structural engineering, Golden Jubilee Publications of Department of Civil Engineering, Indian Institute of Science, Bangalore*, edited by Jagadish, K. S. and Iyengar, R. N., University Press, 2003.

[19] **Keese, A.**, *A review of recent developments in the numerical solution of stochastic PDEs (stochastic finite elements)*, Tech. Rep. Informatikbericht 2003-6, Institute of Scientific Computing, Department of Mathematics and Computer Science, Technische Universitat Braunschweig, Brunswick,, June 2003, uRL: http://opus.tu-bs.de/opus/volltexte/2003/504/.

[20] **Matthies, H. G.** and **Keese, A.**, *Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations*, Computer Methods in Applied Mechanics and Engineering, Vol. 194, No. 12-16, pp. 1295–1331, 2005.

[21] **Hurtado, J. E.** and **Barbat, A. H.**, *Monte Carlo techniques in computational stochastic mechanics*, Archives of Computational Methods in Engineering, Vol. 5, No. 1, pp. 3–29, 1998.

[22] **Papadrakakis, M.** and **Papadopoulos, V.**, *Robust and efficient methods for stochastic finite element analysis using Monte Carlo simulation*, Computer Methods in Applied Mechanics and Engineering, Vol. 134, No. 3-4, pp. 325–340, 1996.

[23] **Elishakoff, I., Ren, Y. J.** and **Shinozuka, M.**, *Improved Finite-Element Method for Stochastic Problems*, Chaos Solitons & Fractals, Vol. 5, No. 5, pp. 833–846, 1995.

[24] **Yamazaki, F., Shinozuka, M.** and **Dasgupta, G.**, *Neumann expansion for stochastic finite element analysis*, ASCE Journal of Engineering Mechanics, Vol. 114, No. 8, pp. 1335–1354, 1988.

[25] **Muscolino, G., Ricciardi, G.** and **Impollonia, N.**, *Improved dynamic analysis of structures with mechanical uncertainties under deterministic input*, Probabilistic Engineering Mechanics, Vol. 15, No. 2, pp. 199–212, 2000.

[26] **Impollonia, N.** and **Muscolino, G.**, *Static and dynamic analysis of non-linear uncertain structures*, Meccanica, Vol. 37, No. 1-2, pp. 179–192, 2002.

[27] **Falsone, G.** and **Impollonia, N.**, *A new approach for the stochastic analysis of finite element modelled structures with uncertain parameters*, Computer Methods in Applied Mechanics and Engineering, Vol. 191, No. 44, pp. 5067–5085, 2002.

[28] **Falsone, G.** and **Impollonia, N.**, *A new approach for the stochastic analysis of finite element modelled structures with uncertain parameters (vol 191, pg 5067, 2002)*, Computer Methods in Applied Mechanics and Engineering, Vol. 192, No. 16-18, pp. 2187–2188, 2003.

[29] **Impollonia, N.** and **Ricciardi, G.**, *Explicit solutions in the stochastic dynamics of structural systems*, Probabilistic Engineering Mechanics, Vol. 21, No. 2, pp. 171–181, 2006.

[30] **Xiu, D. B.** and **Karniadakis, G. E.**, *The Wiener-Askey polynomial chaos for stochastic differential equations*, Siam Journal on Scientific Computing, Vol. 24, No. 2, pp. 619–644, 2002.

[31] **Xiu, D. B.** and **Karniadakis, G. E.**, *Modeling uncertainty in flow simulations via generalized polynomial chaos*, Journal of Computational Physics, Vol. 187, No. 1, pp. 137–167, 2003.

[32] **Wan, X. L.** and **Karniadakis, G. E.**, *Beyond wiener-askey expansions: Handling arbitrary pdfs*, Journal of Scientific Computing, Vol. 27, No. 3, pp. 455–464, 2006.

[33] **Debusschere, B., Najm, H., Pebay, P., Knio, O., Ghanem, R.** and **LeMaitre, O.**, *Numerical challenges in the use of polynomial chaos representations for stochastic processes*, SIAM Journal on Scientific Computing, Vol. 22, No. 2, pp. 698–719, 2005.

[34] **Haylock, R.** and **O'Hagan, A.**, Bayesian Statistics 5, chap. On inference for outputs of computationally expensive algorithms with uncertainty on the inputs, Oxford University Press, Oxford, UK, 1996.

[35] **Challenor, P., Hankin, R.** and **Marsh, R.**, Avoiding Dangerous Climate Change, chap. Achieving robust design from computer simulations, Cambridge University Press, Cambridge, UK, 2006.

[36] **Rougier, J.**, *Probabilistic inference for future climate using an ensemble of climate model evaluations*, Climatic Change, Vol. 81, pp. 247–264, 2007.

[37] **Bates, R., Kennet, R., Steinberg, D.** and **Wynn, H.**, *Achieving robust design from computer simulations*, Quality Technology & Quantitative Management, Vol. 3, pp. 161–177, 2006.

[38] **Kleijnen, J.**, *Kriging metamodeling in simulation: a review*, Tech. Rep. 2007-13, Tilburg University, 2007.

[39] **Kleijnen, J.** and **van Beers, W.**, *Robustness of kriging when interpolating in random simulation with heterogeneous variances: some experiments*, European Journal of Operational Research, Vol. 165, pp. 826–834, 2005.

[40] **McKay, M., Conover, W.** and **Beckman, R.**, *A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code*, Technometrics, Vol. 21, pp. 239–245, 1979.

[41] **Sarkar, A., Benabbou, N.** and **Ghanem, R.**, *Domain decomposition of stochastic PDEs: theoretical formulations*, International Journal for Numerical Methods in Engineering, 2008, to appear.

[42] **Krzanowski, W.**, Principles of Multivariate Analysis, Oxford University Press, Oxford, UK, 2000.

NOMENCLATURE

n	Degrees of freedom in the finite element mesh
$\mathbf{x}$	Spatial coordinate
θ	Elementary random event
$\mathbf{K}$	Random matrix in $\mathbb{R}^{n\times n}$
$\mathbf{u}$	Random response vector in $\mathbb{R}^n$
$\mathbf{f}$	Possibly random forcing vector in $\mathbb{R}^n$
$\xi_k(\theta)$	Standard normal variable
$\Psi_j(\theta)$	Orthogonal polynomial in $\{\xi_k(\theta)\}_{k=1}^{\infty}$
$\mathbf{c}_j$	Deterministic vector in the polynomial chaos expansion
P	Number of polynomials in the polynomial chaos expansion
$\eta(\cdot)$	Simulator
$\mathbf{h}(\cdot)$	Vector of known functions in the mean of $\eta(\cdot)$

$\boldsymbol{\beta}$ Vector of coefficients in the mean of $\eta(\cdot)$
$Z(\cdot)$ Gaussian stochastic process
$C(\cdot,\cdot)$ Covariance function of $Z(\cdot)$
σ Standard deviation parameter
$\mathbf{y}$ Data vector
$\sim$ Distributed as
$\mathcal{K}$ Dynamic stiffness matrix. Polynomial chaos expansion linear system.
$\mathcal{U}$ Displacement vector. Polynomial chaos expansion linear system.
$\mathcal{F}$ Forcing vector. Polynomial chaos expansion linear system.