UNIVERSITÀ DEGLI STUDI DI PALERMO

Meccanica dei Materiali e delle Strutture
Vol. VI (2016), no.1, pp. 74-81
ISSN: 2035-679X
Dipartimento di Ingegneria Civile, Ambientale, Aerospaziale, Dei Materiali
DICAM

EIGENFUNCTION EXPANSION BASED GALERKIN APPROACHES FOR STOCHASTIC FINITE ELEMENT ANALYSIS

S. E. Pryse* and S. Adhikari*

College of Engineering, Swansea University
Bay Campus, Fabian Way, Swansea, Wales, SA1 8EN
e-mail: s.adhikari@swansea.ac.uk

Key words: Stochastic differential equations, Error minimisation, Stochastic finite element method, eigendecomposition, Galerkin, Reduced methods, Eigenfunctions

Abstract. *A novel approach is suggested to compute the response of discretized stochastic elliptic partial differential equations by utilising the stochastic finite element analysis method. The mathematical form of the approach is established by projecting random scalars onto a random basis. By implementing an eigendecomposition of a system's stiffness matrix, the random scalars and random basis are computed. Computational reduction is achieved by approximating the random eigensolutions and by only including dominant terms. Two novel error minimisation techniques have been proposed in order to lower the error introduced by the approximations and the truncation: i) A weak Galerkin approach, ii) A strong Galerkin approach. The proposed methods are applied to analyse the bending of a stochastic cantilever beam. The results obtained through the proposed approaches are compared with those obtained by using direct Monte Carlo Simulations and by using polynomial chaos.*

1 INTRODUCTION

Uncertainties can substantially affect the analysis of physical structures. These uncertainties can occur in the properties of the material, in the geometry or boundary conditions of the structure or in the applied loads [1]. In order to represent the uncertainties that occur in physical systems, a stochastic finite element method [SFEM] can be applied. In this work, a stochastic elliptic partial differential equation is considered

$$-\nabla^2\left[a(\boldsymbol{x},\omega)\,\nabla^2\boldsymbol{u}(\boldsymbol{x},\omega)\right]=p(\boldsymbol{x}) \qquad \boldsymbol{x}\ in\ \mathfrak{D} \tag{1}$$

with the associated Dirichlet condition

$$\boldsymbol{u}(\boldsymbol{x},\omega)=0 \qquad \boldsymbol{x}\ in\ \mathfrak{D} \tag{2}$$

The spatial dimension under consideration is a bounded domain $\mathfrak{D}\in\mathbb{R}^d$ with piecewise Lipschitz boundary $\partial\mathfrak{D}$, where d is less than four. $(\Omega,\mathcal{F},B)$ is a probability space where $\omega\in\Omega$ is a sample point from the sampling space Ω, $\mathcal{F}$ is the complete σ-algebra over the subsets of Ω and B is the probability measure. In Equation (1) $a:\mathbb{R}^d\times\Omega\to\mathbb{R}$ is a random field [2], which can be viewed as a set of random variables indexed by $\boldsymbol{x}\in\mathbb{R}^d$. We assume the random field $a(\boldsymbol{x},\omega)$ to be stationary and square integrable. $a(\boldsymbol{x},\omega)$ is also able to model different physical quantities. Through combining SFEM with a Karhunen-Loève expansion, the discretized equivalent of Equation (1) can be been obtained. The details of obtaining the

discretized equivalent of Equation (1) has been omitted, but can be located in numerous textbooks including [3]. For the case of a static cantilever beam experiencing a deterministic applied force, the discretized equivalent can take the following form

$$\left[\boldsymbol{A}_0 + \sum_{i=1}^{M} \xi_i(\omega)\, \boldsymbol{A}_i\right] \boldsymbol{u}(\omega) = \boldsymbol{A}(\omega)\, \boldsymbol{u}(\omega) = \boldsymbol{f} \tag{3}$$

where $\boldsymbol{A}_0 \in \mathbb{R}^{n\times n}$ represents a deterministic, positive definite, symmetric matrix. $\boldsymbol{A}_{\mathrm{i}} \in \mathbb{R}^{n\times n}$ are random symmetric matrices for $\mathrm{i} = 1, 2, \ldots, \mathrm{M}$, $\boldsymbol{u}(\omega) \in \mathbb{R}^n$ the response vector, $\xi_i(\omega)$ a set of random variables and $\boldsymbol{f} \in \mathbb{R}^n$ the deterministic input force vector. For the remainder of this paper we will assume $\xi_i(\omega)$ to be Gaussian random variables. We aim to produce a new solution approach to solve the discretised equivalent of Equation (1).

2 MOVIVATION

The exact solution to the set of stochastic linear equations given above can be obtained through a direct Monte Carlo Simulations [DMCS] approach by solving the following expression for each sample.

$$\boldsymbol{u}(\omega) = \boldsymbol{A}(\omega)^{-1}\, \boldsymbol{f} \tag{4}$$

However, DMCS can be seen as a computationally expensive method, especially if there is a large number of stochastic linear equations to be solved. In order to avoid the use of DMCS, alternative methods have been explored. The response of Equation (3) can be represented through summing products of random scalars and deterministic vectors

$$\boldsymbol{u}(\omega) = \sum_{j=1}^{M_1} a_j(\omega)\, \boldsymbol{g}_j \tag{5}$$

Equation (5) can be considered as the polynomial chaos method [PC] where $a_j(\omega)$ would correspond to the polynomial chaoses, and $\boldsymbol{g}_j$ would correspond to unknown deterministic vectors. The full details of this method can be found in [3]. Due to the vector in Equations (5) being deterministic, we aim to acquire a solution where both the scalars and vectors are random.

$$\boldsymbol{u}(\omega) = \sum_{j=1}^{M_2} a_j(\omega)\, \boldsymbol{g}_j(\omega) \tag{6}$$

We aim to see if Equation (6) can incorporate the full stochastic nature of Equation (3). Therefore the aim of this paper is to obtain an expression for the response of Equation (3) that is of the same form as Equation (6).

3 INTRODUCING THE METHOD

In order to implement our aim, the random eigenvalue problem is initially considered

$$\boldsymbol{A}(\omega)\, \boldsymbol{\phi}_k(\omega) = \lambda_k(\omega)\, \boldsymbol{\phi}_k(\omega); \qquad k = 1, 2, \ldots n \tag{7}$$

For convenience, the matrices of the random eigenvalues and eigenvectors of $\boldsymbol{A}(\omega)$ are defined as follows

$$\boldsymbol{\Lambda}(\omega) = \text{diag}\,[\lambda_1(\omega), \lambda_2(\omega), \dots, \lambda_n(\omega)] \in \mathbb{R}^{n\times n} \tag{8}$$

$$\boldsymbol{\Phi}(\omega) = [\boldsymbol{\phi}_1(\omega), \boldsymbol{\phi}_2(\omega), \dots, \boldsymbol{\phi}_n(\omega)] \in \mathbb{R}^{n\times n} \tag{9}$$

The random eigenvalues are arranged in ascending order so $\lambda_1(\omega) < \lambda_2(\omega) < \cdots < \lambda_n(\omega)$. The corresponding eigenvectors are arranged in the same order. Due to the orthogonality of $\boldsymbol{\Phi}(\omega)$, we can deduce that $\boldsymbol{\Phi}(\omega)^{-1} = \boldsymbol{\Phi}(\omega)^{\mathrm{T}}$. Thus the following identities can be defined (ω has been omitted for notational convenience)

$$\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{A}\,\boldsymbol{\Phi} = \boldsymbol{\Lambda}; \quad \boldsymbol{A} = \boldsymbol{\Phi}^{-\mathrm{T}}\boldsymbol{\Lambda}\,\boldsymbol{\Phi}^{-1} \quad \text{and} \quad \boldsymbol{A}^{-1} = \boldsymbol{\Phi}\,\boldsymbol{\Lambda}^{-1}\,\boldsymbol{\Phi}^{\mathrm{T}} \tag{10}$$

Using these identities, the response of Equation (3) can be expressed as

$$\boldsymbol{u}(\omega) = \boldsymbol{\Phi}\,\boldsymbol{\Lambda}^{-1}\,\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{f} = \sum_{j=1}^{n} \frac{\boldsymbol{\phi}_j^{\mathrm{T}}(\omega)\,\boldsymbol{f}}{\lambda_j(\omega)}\,\boldsymbol{\phi}_j(\omega) \tag{11}$$

Equation (11) is of the same form as Equation (6) where $\frac{\boldsymbol{\phi}_j^{\mathrm{T}}(\omega)\,\boldsymbol{f}}{\lambda_j(\omega)}$ corresponds to the scalar term $a_j(\omega)$, $\boldsymbol{\phi}_j(\omega)$ corresponds to the vector term $\boldsymbol{g}_j(\omega)$ and n corresponds to M_2. In this particular method, $\boldsymbol{\phi}_j(\omega)$ forms a complete orthogonal basis. Therefore, it can be concluded that the response of Equation (3) can be expressed in the same form as Equation (6) where we have random scalars projected onto a stochastic basis. For the remainder of this paper, this method has been labelled [SP].

4 COMPUTATIONAL REDUCTION

4.1 Approximating the random eigenvalues and eigenvectors

Approximating the random eigenvalues and eigenvectors may improve the calculation cost and there are numerous methods of doing so. Due to its low computational cost, a first order perturbation approach for obtaining the random eigenvalues and eigenvectors has been explored. Solutions of different perturbation methods are obtained by truncating the Taylor series expansion. Due to its efficiency and ease, the first order perturbation method has been used. The *j*th random eigenvalue and its corresponding random eigenvector is given by

$$\lambda_j \approx \lambda_{j_0} + \sum_{k=1}^{M}\left(\frac{\partial\lambda_j}{\partial\xi_k}\right)\xi_k(\omega) \quad \text{and} \quad \boldsymbol{\phi}_j \approx \boldsymbol{\phi}_{j_0} + \sum_{k=1}^{M}\left(\frac{\partial\boldsymbol{\phi}_j}{\partial\xi_k}\right)\xi_k(\omega) \tag{12}$$

where $\xi_k(\omega)$ is a set of Gaussian random variables with mean zero and unit variance. By differentiating the eigenvalue equation with respect to ξ_k, pre-multiplying with $\boldsymbol{\phi}_j^{\mathrm{T}}$ and utilising that $\boldsymbol{\phi}_j^{\mathrm{T}}\,\boldsymbol{\phi}_j = 1$, the derivatives of the eigenvalues and eigenvectors can be expressed as

$$\frac{\partial\lambda_j}{\partial\xi_k} = \boldsymbol{\phi}_{j_0}^{\mathrm{T}}\,\frac{\partial\boldsymbol{A}}{\partial\xi_k}\boldsymbol{\phi}_{j_0} \quad \text{and} \quad \frac{\partial\boldsymbol{\phi}_j}{\partial\xi_k} = \sum_{i=1\neq j}^{N}\frac{\boldsymbol{\phi}_{i_0}^{T}\,\frac{\partial\boldsymbol{A}}{\partial\xi_k}\boldsymbol{\phi}_{j_0}}{\lambda_{j_0} - \lambda_{i_0}}\,\boldsymbol{\phi}_k \tag{13}$$

where $\frac{\partial\boldsymbol{A}}{\partial\xi_k} = \boldsymbol{A}_k$. The derivative of the eigenvectors requires all the deterministic eigenvalues and eigenvectors to be known. However care is needed with this approach as all the eigenvalues need to be unique and the coefficient of variation needs to be of a moderate

value. For the case of repeated eigenvalues, the general method of approximating the response of Equation (3) continues to be valid. However a different approach would be needed whilst approximating the eigensolutions to that given in this section.

4.2 Truncation

The series given in Equation (11) can be truncated after a certain amount of terms. The high terms of the summation have a relatively low value due to the eigenvalues being ordered ascendingly; this allows the low valued terms to be discarded whilst retaining the dominant terms in the series. Consequently Equation (11) can be truncated as follows

$$\boldsymbol{u}(\omega) \approx \sum_{j=1}^{t} \frac{\boldsymbol{\phi}_j^{\mathrm{T}}(\omega)\, \boldsymbol{f}}{\lambda_j(\omega)}\, \boldsymbol{\phi}_j(\omega) \tag{14}$$

where $\lambda_j(\omega)$ and $\boldsymbol{\phi}_j(\omega)$ represent the random eigenvalues and the random eigenvectors and t corresponds to the number of terms retained in the summation. A low-cost MCS is performed in order to obtain the full response of $\boldsymbol{u}(\omega)$.

5 ERROR MINIMISATION

The approximations and truncation introduced in Section 4 introduces error into the calculation. This has motivated an error minimisation appraoch, and as a consequence, two Galerkin approaches are considered: (a) a weak Galerkin approach [SPWG] (b) a strong Galerkin approach [SPSG]. The SPWG approach is initially considered.

5.1 A weak Galerkin approach [SPWG]

For this approach, the solution vector is modified to take the following form

$$\tilde{\boldsymbol{u}}(\omega) \approx \sum_{j=1}^{t} c_j \left(\frac{\boldsymbol{\phi}_j^{\mathrm{T}}(\omega)\, \boldsymbol{f}}{\lambda_j(\omega)} \right) \boldsymbol{\phi}_j(\omega) \tag{15}$$

where $\lambda_j(\omega) \in \mathbb{R}$ and $\boldsymbol{\phi}_j(\omega) \in \mathbb{R}^{N\times N}$ represent the approximated random eigenvalues and eigenvectors, $\boldsymbol{f} \in \mathbb{R}^{N\times N}$ the deterministic applied force and $c_j \in \mathbb{R}^t$ are deterministic constants which need to be determined. The residual vector for this the new approach is defined as

$$\boldsymbol{r}_1(\omega) = \boldsymbol{A}(\omega)\tilde{\boldsymbol{u}}(\omega) - \boldsymbol{f} \tag{16}$$

By making the residual orthogonal to a basis function, the deterministic scalars c_j can be computed. As Equation (15) can be viewed as a projection onto a subset of the random eigenvectors, the residual can be made orthogonal to the same subset of random eigenvectors.

$$\boldsymbol{r}_1(\omega) \perp \boldsymbol{\phi}_k(\omega) \qquad \forall \;\; k = 1, 2, \ldots, t \tag{17}$$

Here $\langle v(\omega), w(\omega)\rangle = \int_{\omega} v(\omega) w(\omega) P(d\omega)$ defines the inner product norm. By using this condition and the expression for the residual, one has

$$\mathbb{E}\left\{ \boldsymbol{\phi}_k^{T}(\omega) \left[\mathbf{A}(\omega) \left(\sum_{j=1}^{t} c_j \left(\frac{\boldsymbol{\phi}_j^{\mathrm{T}}(\omega)\, \boldsymbol{f}}{\lambda_j(\omega)} \right) \boldsymbol{\phi}_j(\omega) \right) - \boldsymbol{f} \right] \right\} = 0 \tag{18}$$

where $\mathbb{E}\{\blacksquare\}$ donates the expected value. For notational convenience, we can define $\beta_j(\omega) = \left(\frac{\boldsymbol{\phi}_j^{\mathrm{T}}(\omega)\,\boldsymbol{f}}{\lambda_j(\omega)}\right)$, thus it can be shown that Equation (18) can take the following form

$$\mathbb{E}\left\{\sum_{j=1}^{t} \boldsymbol{\phi}_k^T(\omega)\,\mathbf{A}(\omega)\,\boldsymbol{\phi}_j(\omega)\,\beta_j\,(\omega)\,c_j\right\} = \mathbb{E}\{\boldsymbol{\phi}_k(\omega)\}\boldsymbol{f} \tag{19}$$

By defining the vector $\boldsymbol{c} = [c_1, c_2, \dots, c_t]^{\mathrm{T}}$, Equation (19) can be simplified to

$$\mathbb{E}\{\boldsymbol{Z}(\omega)\}\,\boldsymbol{c} = \mathbb{E}\{\boldsymbol{y}(\omega)\} \qquad j,k = 1,2,\dots,t \tag{20}$$

where $\boldsymbol{Z}_{kj} = \boldsymbol{\phi}_k^T(\omega)\,\mathbf{A}(\omega)\,\boldsymbol{\phi}_j(\omega)\,\beta_j(\omega)$; $\forall\, j,k = 1,2,\dots,t$, and $\boldsymbol{y}(\omega) = \boldsymbol{\phi}_k(\omega)\,\boldsymbol{f}$; $\forall\ k = 1,2,\dots,t$. Therefore by solving the set of linear equations given by Equation (20), an explicit closed form for the unknown coefficients can be obtained. The arising expected values can be computed by using low-order fast Monte Carlo Simulations.

5.2 A strong Galerkin approach [SPSG]

In a similar manner to the previous approach, the solution vector can be modified to take the following form

$$\acute{\boldsymbol{u}}(\omega) \approx \sum_{j=1}^{t} d_j(\omega)\left(\frac{\boldsymbol{\phi}_j^{\mathrm{T}}(\omega)\,\boldsymbol{f}}{\lambda_j(\omega)}\right)\boldsymbol{\phi}_j(\omega) \tag{21}$$

where $\lambda_j(\omega) \in \mathbb{R}$ and $\boldsymbol{\phi}_j(\omega) \in \mathbb{R}^{N\times N}$ represent the approximated random eigenvalues and eigenvectors, $f \in \mathbb{R}^{N\times N}$ the deterministic applied force. Contrary to the previous approach, $d_j(\omega) \in \mathbb{R}$ are unknown constants that need to computed for each realisation. The residual vector for this approach is defined as

$$\boldsymbol{r}_2(\omega) = \boldsymbol{A}(\omega)\acute{\boldsymbol{u}}(\omega) - \boldsymbol{f} \tag{22}$$

By making the residual orthogonal to a basis function, $d_j(\omega)$ can be computed. By using the same analogy as seen in the previous approach, the residual can be made orthogonal to the subset of random eigenvectors. This results in the following expression

$$\boldsymbol{\phi}_k^T\,(\omega)\left[\mathbf{A}\,(\omega)\left(\sum_{j=1}^{t} d_j(\omega)\left(\frac{\boldsymbol{\phi}_j^{\mathrm{T}}(\omega)\,\boldsymbol{f}}{\lambda_j(\omega)}\right)\boldsymbol{\phi}_j(\omega)\right) - \boldsymbol{f}\right] = 0 \tag{23}$$

For notational convenience, we can define $\beta_j(\omega) = \left(\frac{\boldsymbol{\phi}_j^{\mathrm{T}}(\omega)\,\boldsymbol{f}}{\lambda_j(\omega)}\right)$. Therefore, the above expression can be manipulated to give

$$\left(\sum_{j=1}^{t} \boldsymbol{\phi}_k^T(\omega)\,\mathbf{A}(\omega)\,\boldsymbol{\phi}_j(\omega)\,\beta_j\,(\omega)\,d_j(\omega)\right) = \boldsymbol{\phi}_k(\omega)\,\boldsymbol{f} \tag{24}$$

where $d_j(\omega)$ would be computed for each realisation. By defining the vector $\boldsymbol{d}(\omega) = [d_1(\omega), d_2(\omega), \dots, d_t(\omega)]^{\mathrm{T}}$, Equation (15) can be simplified to

$$\boldsymbol{Z}(\omega)\,\boldsymbol{d}(\omega) = \boldsymbol{y}(\omega) \qquad j,k = 1,2,\dots,t \tag{25}$$

where $\boldsymbol{Z}_{kj} = \boldsymbol{\phi}_k^T(\omega)\,\mathbf{A}(\omega)\,\boldsymbol{\phi}_j(\omega)\,\beta_j(\omega); \quad \forall\, j,k = 1,2,\dots,t,$ and $\boldsymbol{y}(\omega) = \boldsymbol{\phi}_k(\omega)\,\boldsymbol{f};$ $\forall\ k = 1,2,\dots,t.$ The number of equations that need to be solved in order to calculate the unknown vector $\boldsymbol{d}(\omega)$ corresponds to the value of t. Therefore, similarly to the weak Galerkin approach, the lower the dimension of the reduced system, the fewer the number of equations that need to be solved.

6 APPLICATION

Thus far we have discussed five different methods for computing or approximating the response of Equation (3): i) DMCS, ii) SP, iii) SPWG, iv) SPSG and v) PC. These five methods are now applied to compute the bending of a static Euler-Bernoulli cantilever beam of length 1.00 m. The cross-section of the beam is a rectangle of length 0.04 m and height 0.005 m and a 1.00 N deterministic vertical point load is applied at the free end of the beam.

The system has been discretized into a 100 elements by using SFEM. Consequently, the dimension of the corresponding determinant matrix is 200 × 200. For the deterministic case, the Young's modulus is E = 200 × 10^9 Nm^{-2} thus corresponding to a steel beam. The deterministic second moment of area (moment of inertia) of the beam is 4.1667 × 10^{-11} m^4. The bending rigidity, EI, can be assumed to be a stationary Gaussian random field

$$EI\,(x,\omega) = \overline{EI}\,\big(1 + a(x,\omega)\big) \tag{26}$$

where $\overline{EI} = 83.33$ Nm^2. The function $a(x,\omega)$ represents a stationary Gaussian field with zero mean, with x being the coordinate direction of the length of the beam. The standard deviation is given by $0.18 \times \overline{EI}$, and the covariance function by

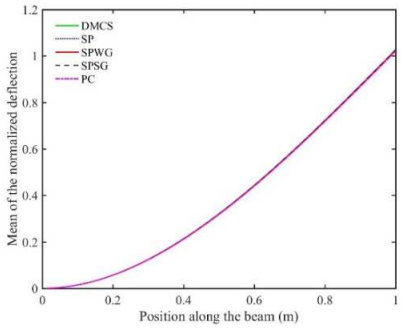
$$C_a(x_1,x_2) = \sigma_a^2 e^{(|x_1 - x_2|)/\,\mu_a} \tag{27}$$

where μ_a is the correlation length (0.50 m) and $\sigma_a$the standard deviation; The KL expansion of the system has been truncated to include two terms. The five listed methods have been simulated 10,000 times. For the SP, SPWG and SPSG methods, 4 terms have been retained in their respective equations, and a 4th order polynomial chaos approach is used for the PC method.

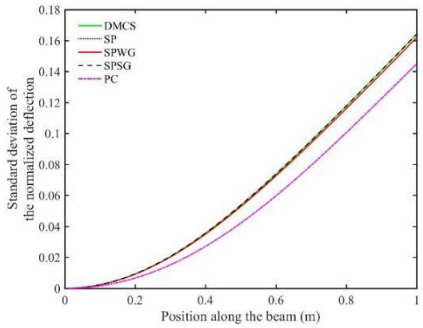
Figures 1a and 1b illustrate the mean and standard deviation of the normalised vertical displacement at all nodes of the beam. It is apparent that all the approximation methods captures the mean of the DMCS method pretty well. The SPSG method best captures the standard deviation of the DMCS method, whilst both the SP and SPWG methods considerably out performs the PC approach. The percentage error of the vertical displacement arising when using the SP, SPWG, SPSG and the PC methods in place of the DMCS method is illustrated in Figures 1c and 1d. The percentage error is given by

$$\epsilon_{\%} = 100 \times \frac{|DMCS - CM|}{DMCS} \tag{28}$$

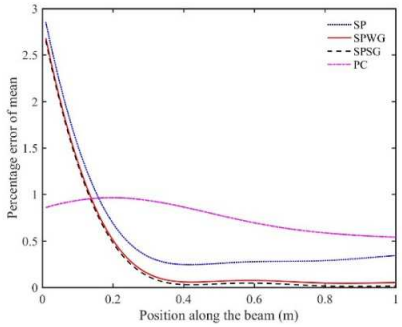
where *DMCS* indicates the solution of the direct approach, and *CM* the solution of the comparable methods. Barring the initial 0.20m of the bar, the percentage error of the mean and standard deviation of the SP, SPWG and SPSG methods are considerably lower than that of the PC method. When comparing the SPWG and SPSG methods it is apparent that the percentage error of the mean is always slightly lower when the SPSG method is used. However when comparing the percentage error of the standard deviation, barring the initial 0.30m of the bar, the SPSG method considerably outperforms the SPWG method.



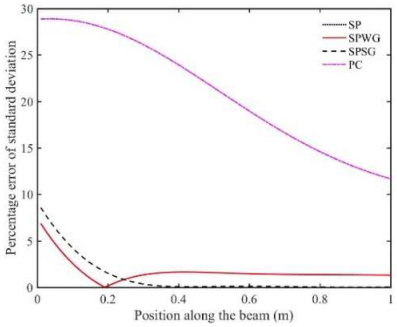
(a) Mean of the vertical displacement



(b) Standard deviation of the vertical displacement



(c) Percentage error of the mean



(d) Percentage error of the standard deviation

The L^2 relative error for the mean and standard deviation of the response of the cantilever beam is considered. The L^2 relative error of the mean is defined as

$$\epsilon_{L^2} = \frac{\|\boldsymbol{\mu}_{DMCS} - \boldsymbol{\mu}_{CM}\|_2}{\|\boldsymbol{\mu}_{DMCS}\|_2} \tag{29}$$

where $\boldsymbol{\mu}_{DMCS}$ denotes the mean of the response vector obtained by using the DMCS method and $\boldsymbol{\mu}_{CM}$ the mean of the response vector of the comparable methods. The expression for the L^2 relative error of the standard deviation takes a similar form to that of Equation (29). Table 1 displays the L^2 relative error for the mean and the standard deviation for different truncation values of Equations (11), (15) and (21) and for the PC approach.

It is apparent that both the SPWG and SPSG methods produces lower relative errors than the SP method for the mean. When considering the L^2 relative error for the standard deviation, the SPWG method does not significantly lower the error. However a substantial drop in the relative error is seen when the SPSG method is used. As the relative errors for both the mean and standard deviation decreases as the number of terms in the summations increase, it is apparent that the truncation value can be chosen in line with an acceptable level of error. The CPU times for the five methods is given in Table 2.

Table 1: L^2 error arising in the mean and standard deviation of the response for each of the methods

	Relative error in the mean				**Relative error in the standard deviation**			
Number of terms	**SP**	**SPWG**	**SPSG**	**PC**	**SP**	**SPWG**	**SPSG**	**PC**
2	0.0063	0.0058	0.0058	-	0.0092	0.0092	0.0029	-
4	0.0036	9.59×10^{-4}	9.07×10^{-4}	-	0.0101	0.0101	0.0010	-
5	0.0025	6.37×10^{-4}	5.54×10^{-4}	-	0.0102	0.0101	5.14 × 10-4	-
10	0.0025	3.28×10^{-5}	9.54×10^{-5}	-	0.0102	0.0102	9.88 × 10-5	-
-	-		-	0.0017	-		-	0.0635

Table 2: Computational time

Method	**DMCS**	**SP**	**SPWG**	**SPSG**	**PC**
CPU (sec)	10.21	0.76	3.25	3.24	0.84

Regarding CPU, the SP method outperforms all the other methods. Although that using the SPWG and SPSG methods is faster than the DMCS method, the PC method is faster. However when the order of the PC method or the number of degrees of freedom is sufficiently increased, the CPU of the PC method will substantially increase and in turn, will be higher than the SPWG and SPSG methods. Generally it can be concluded that the SPSG method outperforms the SPWG method as it produces less error in a similar CPU time. However the SPWG method does not need as much storage capacity as the SPSG method.

7 CONCLUSION

An approach has been suggested to calculate the response of discretized stochastic elliptic partial differential equations. Through utilising the stochastic finite element method and the random eigenvalue problem, it has been proven that the solution can be represented through projecting random scalar onto a random basis. Due to the high computational cost associated with calculating the exact solution, a reduced approach is proposed where random eigenvalues and eigenvectors are approximated and low valued terms are discarded. Two novel multiplicative Galerkin error minimisation approaches have been presented. i) a weak Galerkin approach, ii) a strong Galerkin approach. The proposed methods have been used to analyse the bending of a static cantilever beam. The solutions obtained through the proposed methods have been compared with those obtained through direct MCS and through using a polynomial chaos method. Although both error minimisation approaches lowers the error of the mean, it is only the strong Galerkin error minimisation approach that substantially lowers the error arising in the standard deviation. Further work would focus on efficient ways of computing random eigenfunctions and performing model-order reduction.

8 ACKNOWLEDGEMENTS

The authors acknowledge the financial support received from Engineering Research Network Wales (one of three Sêr Cymru National Research Networks).

BIBLIOGRAPHY

[1] G. Maymon, *Structural Dynamics and Probabilistic Analysis for Engineers*, Butterworth-Heinemann, (2008).

[2] E. VanMarcke, *Random Fields: Analysis and Synthesis*, MIT Press, Cambridge MA, (1983).

[3] R. G. Ghanem and P. D. Spanon, *Stochastic Finite Elements: A Spectral Approach (revised edition)*, Dover Publications Inc., (2012).