Numerical Karhunen–Loève expansion of a covariance function on a multidimensional domain via the finite elements method

S. Recek[a,b,*], M. Lemaire[b], A. Millard[a]

[a]*Commissariat à l'Energie Atomique, Gif-sur-Yvette, 91191, France*
[b]*LaMI – UBP & IFMA, EA 3867 / FR CNRS 2856, Aubière, 63175, France*

Abstract

This paper presents the implementation of a numerical Karhunen–Loève expansion method in a finite elements code. The procedure allows the discretization of a stochastic field in the form of a truncated series expansion including deterministic functions and random variables.

Keywords: Stochastic finite elements; Stochastic fields; Stochastic processes; Spectral expansion; Karhunen–Loève expansion; Covariance

1. Introduction

One of the major difficulties associated with stochastic analysis is the necessity to deal with random processes, which become more and more necessary to account for model uncertainties. A continuous random process $\mathbf{F}(\{\mathbf{x}\}, \omega)$ is defined as an indexed set of random variables, the index belonging to some uncountable set (e.g. the space coordinates for a random field). A natural way to treat such mathematical object is to express them in term of a limited set of random variables. For a random field, it consists, for example, in associating a random variable to nodes of a finite element (FE) model and then use the elements' shape functions to interpolate the random field on the entire domain. However such an approach can lead to a large number of random variables, which are numerically expensive to deal with, either in Monte-Carlo simulation or in stochastic analysis.

A less expensive approach is to expand the stochastic field in a Fourier-type series [1]. Let us consider a random field $\alpha(\{\mathbf{x}\}, \omega)$ of a given mean $\bar{\alpha}(\{\mathbf{x}\})$ and a covariance function:

$$\mathbf{C}_\alpha(\{\mathbf{x}\},\{\mathbf{y}\}) = \mathbf{E}\Big[(\alpha(\{\mathbf{x}\},\omega) - \bar{\alpha}(\{\mathbf{x}\}))(\alpha(\{\mathbf{y}\},\omega) - \bar{\alpha}(\{\mathbf{y}\}))\Big] \quad (1)$$

The Karhunen–Loève expansion of the process consists in:

$$\alpha(\{\mathbf{x}\},\omega) = \bar{\alpha}(\{\mathbf{x}\}) + \sum_{\mathrm{n}=1}^{\infty} \sqrt{\lambda_\mathrm{n}}\varsigma_\mathrm{n}(\omega)\phi_\mathrm{n}(\{\mathbf{x}\}) \quad (2)$$

Where λ_n and $\phi_n(\{\mathrm{x}\})$ are the eigenvalues and eigenfunctions of the covariance function, i.e. the solutions of the following Fredholm integral equation:

$$\int_\mathbf{D} \mathbf{C}_\alpha(\{\mathbf{x}_1\},\{\mathbf{x}_2\})\phi_\mathrm{n}(\{\mathbf{x}_1\})\mathbf{d}\{\mathbf{x}_1\} = \lambda_\mathrm{n}\phi_\mathrm{n}(\{\mathbf{x}_2\}) \quad (3)$$

and $\varsigma_n(\omega)$ are random variables defined by:

$$\varsigma_n(\omega) = \frac{1}{\sqrt{\lambda_\mathrm{n}}}\int_\mathbf{D} \alpha(\{\mathbf{x}\},\omega)\phi_\mathrm{n}(\mathbf{s})\mathbf{d}\{\mathbf{x}\} \quad (4)$$

If the random field $\alpha(\{\mathbf{x}\}, \omega)$ is Gaussian then these variables follow a normal probability density function (PDF). Convergence study of truncated (2) series were investigated by Huang et al. [2]. Such a decomposition is the first step for stochastic finite elements procedures in case of Gaussian random processes [3,4].

Analytical solutions of Eq. (3) are known for a small number of covariance functions in one-dimensional domains. In [1] Gahnem et al. presented a method to solve numerically Eq. (3) (for any domain and feasible function) that was very well-suited to finite elements (FE) code, even if it could be used with any base of function [5]. In this paper, we propose an

* Corresponding author. Tel.: +33 1 69 089039; Fax: +33 1 69 088684; E-mail: recek@semt2.smts.cea.fr

Computational Fluid and Solid Mechanics 2005
K.J. Bathe (Editor)

implementation in the FE code CAST3M [6] and test it on one-dimensional analytical solution cases.

2. Theory

A numerical solution of Eq. (3) can be attained through a Galerkin-type procedure [1]. Let $\{\mathbf{h_i}(\{\mathbf{x}\})\}$ be a complete base of function in the Hilbert space H of continuous functions. Each eigenfunction ϕ_k ({x}) may be represented as:

$$\phi_{\mathbf{k}}(\{\mathbf{x}\}) = \sum_{\mathbf{i}=1}^{\infty} \mathbf{d_i^{(k)}} \mathbf{h_i}(\{\mathbf{x}\}) \tag{5}$$

The difference ε_k between the two members of Fredholm equation (3) resulting from the truncature after the N^{th} term for the k^{th} eigenfunction is:

$$\varepsilon_{\mathbf{k}}(\{\mathbf{x}_1\}) = \sum_{\mathbf{i}=1}^{\mathbf{N}} \mathbf{d_i^{(k)}} \left[\int_{\mathbf{D}} \mathbf{C}_\alpha(\{\mathbf{x}_1\},\{\mathbf{x}_2\}) \mathbf{h_i}(\{\mathbf{x}_2\}) \mathbf{d}\{\mathbf{x}_2\} - \lambda_{\mathbf{k}} \mathbf{h_i}(\{\mathbf{x}_1\}) \right] \tag{6}$$

Requiring the error to be orthogonal to the approximating functions for a k given yields to:

$$\left\langle \varepsilon_{\mathbf{k}}(\{\mathbf{x}_1\}), \mathbf{h_j}(\{\mathbf{x}_1\}) \right\rangle = 0 \quad \mathrm{j} = 1, \ldots, \mathrm{N} \tag{7}$$

Thus:

$$\sum_{\mathbf{i}=1}^{\mathbf{N}} \mathbf{d_i^{(k)}} \left[\int_{\mathbf{D}} \left(\int_{\mathbf{D}} \mathbf{C}_\alpha(\{\mathbf{x}_1\},\{\mathbf{x}_2\}) \mathbf{h_i}(\{\mathbf{x}_2\}) \mathbf{d}\{\mathbf{x}_2\} \right) \mathbf{h_j}(\{\mathbf{x}_1\}) \mathbf{d}\{\mathbf{x}_1\} - \lambda_{\mathbf{k}} \int_{\mathbf{D}} \mathbf{h_i}(\{\mathbf{x}_1\}) \mathbf{h_j}(\{\mathbf{x}_1\}) \mathbf{d}\{\mathbf{x}_1\} \right] = 0 \tag{8}$$

This equation allows us to compute the k^{th} eigenvalue and eigenfunction. We can obtain a more friendly form by using two matrices [C] and [B]:

$$\mathbf{C_{ij}} = \int_{\mathbf{D}} \left(\int_{\mathbf{D}} \mathbf{C}_\alpha(\{\mathbf{x}_1\},\{\mathbf{x}_2\}) \mathbf{h_i}(\{\mathbf{x}_2\}) \mathbf{d}\{\mathbf{x}_2\} \right) \mathbf{h_j}(\{\mathbf{x}_1\}) \mathbf{d}\{\mathbf{x}_1\} \tag{9}$$

$$\mathbf{B_{ij}} = \int_{\mathbf{D}} \mathbf{h_i}(\{\mathbf{x}\}) \mathbf{h_j}(\{\mathbf{x}\}) \mathbf{d}\{\mathbf{x}\} \tag{10}$$

This leads to the following matricial form of the equation for the k^{th} mode:

$$[\mathbf{C}]\left\{\mathbf{d^{(k)}}\right\} - \lambda_{\mathbf{k}} [\mathbf{B}]\left\{\mathbf{d^{(k)}}\right\} = 0 \tag{11}$$

or equivalently

$$[\mathbf{B}]\left\{\mathbf{d^{(k)}}\right\} - \frac{1}{\lambda_{\mathbf{k}}} [\mathbf{C}]\left\{\mathbf{d^{(k)}}\right\} = 0 \tag{12}$$

At this point the problem becomes very similar to a classical dynamic problem in FEM. Given a system having a stiffness matrix [K] and a mass matrix [M], the k^{th} pulsations and modal vectors are found by solving the equation:

$$[\mathbf{K}]\left\{\mathbf{q^{(k)}}\right\} - \omega_{\mathbf{k}}^2 [\mathbf{M}]\left\{\mathbf{q^{(k)}}\right\} = 0 \tag{13}$$

Therefore, the only new piece of code to be added is the computation of the matrices [C] and [B]. The reason of inverting the Eq. (11) is that most FE software give the smallest eigenpulsations first, whereas in our expansion we are looking for the greatest eingenvalues. Then to use the built-in procedure one has to make the following correspondences:

Data:

Matrix [K] is replaced by matrix [B],
Matrix [M] is replaced by matrix [C],
No boundary condition.

Results:

The k^{th} returned pulsation ω_k gives the eigenvalue λ_k according to the relation

$$\lambda_{\mathrm{k}} = \frac{1}{\omega_{\mathrm{k}}^2} \tag{14}$$

The eigenvector $\{\mathbf{q}^{(\mathrm{k})}\}$ returned by the code gives us the eigenfunction $\phi_k(\{x\})$ using the n shape functions $N_i(\{x\})$ (n being the number of nodes in the model)

$$\phi_{\mathbf{k}}(\{\mathbf{x}\}) = \sum_{\mathbf{i}=1}^{\mathbf{n}} \mathbf{N_i}(\{\mathbf{x}\}) \mathbf{q_i^{(k)}} \tag{15}$$

3. Computation of matrices [C] and [B] using finite elements

Using the shape functions $\mathbf{N_i}(\{\mathbf{x}\})$ as base functions in expression (9) leads us to:

$$\mathbf{C_{ij}} = \int_{\mathbf{D}} \left[\int_{\mathbf{D}} \mathbf{C}(\{\mathbf{x}_1\},\{\mathbf{x}_2\}) \mathbf{N_i}(\{\mathbf{x}_2\}) \mathbf{d}\{\mathbf{x}_2\} \right] \mathbf{N_j}(\{\mathbf{x}_1\}) \mathbf{d}\{\mathbf{x}_1\} \tag{16}$$

Let us consider the first level of integration in expression (9):

$$\mathbf{f_i}(\{\mathbf{x}_1\}) = \int_{\mathbf{D}} \mathbf{C}(\{\mathbf{x}_1\},\{\mathbf{x}_2\}) \mathbf{N_i}(\{\mathbf{x}_2\}) \mathbf{d}\{\mathbf{x}_2\} \tag{17}$$

The shape function having a compact support, the integration domain may be reduced to the elements

surrounding the node i. Denoting Di such a domain and De_k the element k domain leads to:

$$\mathbf{f_i}(\{\mathbf{x}_1\}) = \sum_{\mathbf{k}_1 \in \mathbf{Di}} \left(\int_{\mathbf{De_{k1}}} \mathbf{C}(\{\mathbf{x}_1\}, \{\mathbf{x}_2\}) \mathbf{N_i}(\{\mathbf{x}_2\}) \mathbf{d}\{\mathbf{x}_2\} \right) \tag{18}$$

Then, using Gauss integration:

$$\mathbf{f_i}(\{\mathbf{x}_1\}) = \sum_{\mathbf{k}_1 \in \mathbf{Di}} \sum_{ig=1}^{n_{Gauss}} \mathbf{C}\left(\{\mathbf{x}_1\}, \left\{\mathbf{x}_{ig}^{k_1}\right\}\right) \mathbf{N_i}\left(\left\{\mathbf{x}_{ig}^{k_1}\right\}\right) w_{ig} \det([J_{k_1}]) \tag{19}$$

We can follow the same scheme for the variable $\{x_1\}$:

$$\mathbf{C_{ij}} = \int_{\mathbf{Dj}} \mathbf{f_i}(\{\mathbf{x}_1\}) \mathbf{N_j}(\{\mathbf{x}_1\}) \mathbf{d}\{\mathbf{x}_1\} \tag{20}$$

$$\mathbf{C_{ij}} = \sum_{k_2 \in Dj} \sum_{jg=1}^{n_{Gauss}} \mathbf{f}\left(\left\{\mathbf{x}_{jg}^{k_2}\right\}\right) \mathbf{N_j}\left(\left\{\mathbf{x}_{jg}^{k_2}\right\}\right) w_{jg} \det([J_{k_2}]) \tag{21}$$

Then, substituting f by its expression (19):

$$\mathbf{C_{ij}} = \sum_{k_2 \in Dj} \sum_{jg=1}^{n_{Gauss}} \left[\sum_{k_1 \in Di} \sum_{ig=1}^{n_{Gauss}} \mathbf{C}\left(\left\{\mathbf{x}_{jg}^{k_2}\right\}, \left\{\mathbf{x}_{ig}^{k_1}\right\}\right) \mathbf{N_i}\left(\left\{\mathbf{x}_{ig}^{k_1}\right\}\right) w_{ig} \det([\mathbf{J}_{k_1}]) \right] \mathbf{N_j}\left(\left\{\mathbf{x}_{jg}^{k_2}\right\}\right) w_{jg} \det([\mathbf{J}_{k_2}]) \tag{22}$$

Where:
Di is the set of element surrounding the node i,
$\mathbf{N_i}$ is the shape function associated with the node i,
n_{Gauss} is the number of Gauss points in each element,
$\{\mathbf{x}_{ig}^{k_1}\}$ is the vector of the coordinates of the ig^{th} Gauss point of the element k_1,
$\mathbf{w}_{ig}$ is the weight of the ig^{th} Gauss point,
$[\mathbf{J}_{k_1}]$ is the jacobian matrix of the element k_1.

On the other hand, the computation of the matrix [B] was more classical: expression (10) is nothing else than the mass matrix of the system for a unitary density and one degree of freedom per node.

4. Implementation

The previous scheme was implemented in the finite element software CAST3M. Only a few procedures had to be added to the code in order to compute both matrices [C] and [B].

The main difference between these procedure and most classical FE procedures is the impossibility to use elementary matrices for [C], due to the double level of integration in (9). Usually in FEM, integrations are performed in each element and then summed to give the global integration on the whole domain. It was not possible here because a global integral had to be computed first (expression (19)) before integrating again (expression (21)). However, thanks to the compact support of shape functions, these integrations can be limited to the elements surrounding both inspected nodes (the set noted Di in (18)). This raised also practical concerns for internal data manipulation: a superelement including every node of the model had to be defined in order to store [C].

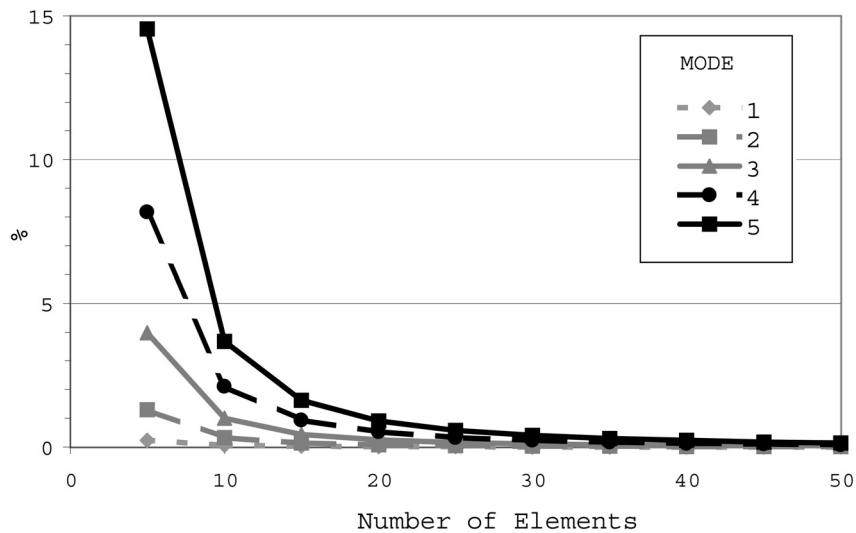

Fig. 1. Relative difference between numerical and theoretical solutions for the test beam with respect to the mesh density.

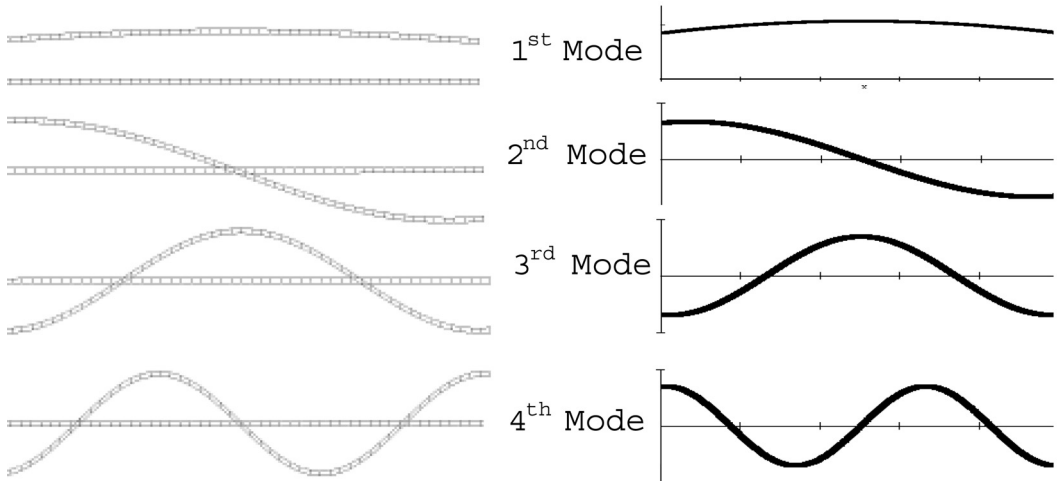

Fig. 2. Numerical and theoretical modes (left and right respectively).

5. Test case

Analytical solution of the spectral decomposition of an exponential and triangular covariance are know in the one-dimensional case. We set up the following one-dimensional exponential covariance function along a three-dimensional beam with a length equal to one meshed with linear elements:

$$\mathbf{C}_\alpha(\{\mathbf{x}_1\},\{\mathbf{x}_2\}) = \exp(-|\mathbf{x}_{11} - \mathbf{x}_{21}|) \tag{23}$$

Where $\mathbf{x}_{11}$ and $\mathbf{x}_{21}$ are the first coordinates of vectors $\{\mathbf{x}_1\}$ and $\{\mathbf{x}_2\}$ respectively.

The numerical results showed a good convergence to the analytical ones for increasing mesh density (Fig. 1). For the highest density tested, numerical eigenvalues (Table 1) and eigenfunctions (Fig. 2) were similar to those expected by theory. Noteworthy eigenvalues are inversely proportional to the cross-section area of the beam (multiplying area by ten leads to eigenvalues divided by ten), not surprisingly because we have the square of the Jacobian determinant in [C] but not in [B]. So eigenvalues corresponding to the one-dimensional case could be found either by using a unity cross-section area or by multiplying the results by its value.

Table 1
Numerical and analytical eigenvalues for the test beam with 50 elements

	Numerical	Analytical	Relative difference
1st Mode	7.388324E-01	7.388110E-01	0.0029%
2nd Mode	1.380217E-01	1.380040E-01	0.0128%
3rd Mode	4.510630E-02	4.508800E-02	0.0406%
4th Mode	2.134663E-02	2.132900E-02	0.0827%
5th Mode	1.229677E-02	1.227900E-02	0.1448%
6th Mode	7.963228E-03	7.945371E-03	0.2248%

6. Extension to tri-dimensional covariance functions

For the sake of the comparison with one-dimensional analytical solution we had to use the one-dimensional covariance function (23). However, this method is immediately usable for any feasible tri-dimensional covariance function and domain, leading to the expansion of any stationary second-order random field as long as the computational power allows it. Such a study is beyond the scope of this paper.

7. Conclusions

The method presented in this paper achieves the Karhunen–Loève expansion of a covariance function through the use of a finite element model. Such an expansion allows the discretization of random fields on the form of a set of deterministic functions (represented by their nodal values) associated with random variables, leading to either Monte-Carlo simulation or stochastic finite elements procedures.

The main limitations of this method do not come from the numerical implementation, but from the expansion itself. First the truncated Karhunen–Loève expansion may converge slowly to its exact values for some covariance functions (like the triangular ones). Second this expansion is only useful for Gaussian

random processes (else the distribution of random variables defined in (4) is unknown).

References

[1] Ghanem R, Spanos P. Stochastic finite elements: a spectral approach. Berlin: Springer-Verlag, 1991.
[2] Huang SP, Quek ST, Phoon KK. Convergence study of the truncated Karhunen–Loève expansion for simulation of stochastic processes. Int J Numer Meth Engng 2001;52:1029–1043.
[3] Ghanem R. Ingredients for a general purpose stochastic finite elements implementation. Comp Methods Appl Engrg 1999;168:19–34.
[4] Matthies HG, Bucher C. Finite elements for stochastic media problems. Comp Methods Appl Engrg 1999;168:19–34.
[5] Castrillon-Candas JE, Amaratunga K. Fast estimation of continuous Karhunen–Loève eigenfunctions using wavelets. IEEE Transaction on Signal Processing 2002;50(1):78–86.
[6] Verpeaux P, Millard A, Charras T, Combescure A. A modern approach of large computer codes for structural analysis. Proc SMIRT 10, Los Angeles, 1984.