Fast Algorithm for Computing Karhunen-Loève Expansion

Qian-Yong Chen[*]

Abstract

Karhunen Loève expansion has been proven to be an efficient way to approximate second-order random processes. But numerical computation of the eigen-pairs can be costly, especially for high spatial dimension problems with short correlation length. In this paper, we consider the stationary correlation function $C(\mathbf{x_1}, \mathbf{x_2})$ (i.e., C only depends on the difference $(\mathbf{x_1} - \mathbf{x_2})$), on regular domains, e.g., rectangular domain for 2D problems. We propose an algorithm that is based on the separation of x- and y- variables in the correlation function. Such a separation is automatic for correlation functions like the squared exponential correlation, and will be achieved through the singular value decomposition when the correlation function is not separable by nature. The storage requirement is linear, and the computation cost will be in the order of $O(M\sqrt{M})$, where M is the total number of Legendre modes.

Keywords: Karhunen Loève expansion, SVD

[*] Department of Mathematics and Statistics, University of Massachusetts Amherst (qchen@math.umass.edu).

1 Introduction

Uncertainty arises in many physical and engineering applications, due to either intrinsic probabilistic nature or lack of data. No matter what the source of uncertainty or randomness is, such random effects in the inputs will produce stochastic solutions as outputs. The impact of such uncertainty should be modeled and analyzed, and there has been an intensive research in this aspect in recent years [9, 8, 10, 21, 3, 2, 26, 5, 25, 11, 20, 17]. A common approach is to treat each variable with uncertainty as a random variable (or random field), and rewrite the system as a stochastic system. The methods to study such a system include the sampling based methods (such as the Monte-Carlo methods), the perturbation methods, the operator based methods, the methods based on moment equations, and the recently developed generalized polynomial chaos methods. Or, one may take the classical approach based on stochastic calculus and model the problem with stochastic differential equation (SDE) in which the inputs are idealized processes, such as Wiener processes, Poisson processes etc.

There are various techniques to simulate the second-order random processes (i.e., with finite second moment) for both Gaussian and non-Gaussian processes, although the methods for the former are much better established. We are more interested in the methods that approximate a random function (field, or process) in a similar way of approximating a deterministic function. Depending on the random variables used in the approximation, there are roughly three kinds of approximations: *point discretization*, *average discretization*, and *series expansion methods* [7, 14, 16, 23].

In point discretization, the random variables at discrete positions (i.e., $u(x_1, \omega), u(x_2, \omega), \ldots$) are used to approximate the random field. For example, one could partition the whole domain $\mathcal{D}$ into non-overlapping elements, and approximate the random field on each element by a single random variable $u(x_i, \omega)$ with x_i belonging to the element. This corresponds to the piece-wise constant approximation for deterministic functions. High order piece-wise approximations can also be achieved in a similar manner, e.g., by using the shape functions typically defined in the Finite Element Method (FEM), then it becomes one of the series expansion methods. In average discretization, the random variables on any individual points are not used. Instead, the averages of the random field $u(x, \omega)$ over non-overlapping elements are used in the approximation.

Taking a different approach, the series expansion method writes the random field as a series expansion over a complete set of deterministic functions. The Karhunen-Loève (KL) expansion [15], Orthogonal Series expansion (OSE) [27], the expansion optimal linear estimation (EOLE) method [14] all fall into this category. For approximation or simulation purpose, a truncated expansion will be used. The efficiency of these methods are typically characterized by the number of random variables needed in the truncated expansion in order to achieve a prescribed accuracy.

In particular, the Karhunen Loève (KL) expansion [15] is a spectral representation of random fields. Its discrete version is often called the principal component analysis (PCA). It can be applied to both stationary and non-stationary random fields, although the algorithm in this paper only works for stationary fields. The Karhunen Loève expansion is optimal among all possible linear representations of random processes in the sense of the mean-square error. (Some nonlinear approximations of random processes were studied for data compression in [6].)

However, the computation of the KL expansion could be expensive, specially for high dimension problems with small correlation length. One major computation cost is from the eigen-solver, many of which nowadays can take either an explicit matrix or a subroutine doing matrix-vector multiplication. Inspired by the fast Gauss transform in [24], in this paper we develop a fast algorithm that takes use of the fact that x- and y- variables can be separated in the correlation function. The overall storage requirement is linear $O(M)$, versus $O(M^2)$ if the matrix is formed explicitly. The matrix-vector multiplication involved in the eigen-solver is in the order of $O(M\sqrt{M})$, instead of $O(M^2)$ for standard matrix-vector multiplication. For the 2D rectangular domain, our method is more efficient than the piece-wise method with axis-parallel grids for which the fast Fourier transform (FFT) can be applied to carry out the matrix-vector multiplication.

Some other recent work in the computation of the KL expansion are [27, 12, 18, 9, 1, 13, 22]. This list only includes a very small portion of the papers we came across and is incomplete by all means. It is noteworthy to point out that a fast matrix-vector multiplication was achieved by the Fast Multipole Methods in [22], which is mostly favorable when the covariance function has low regularity. The situation is similar for the method [13] based on the hierarchical matrix technique. Our method, on the other hand, has the most advantage when the covariance kernel has high regularity, as shown by the results in Sec. 4. To certain extent, the algorithm proposed in this paper is a complement to the two fast algorithms mentioned above.

The rest of the paper is organized as follows. In Sec. 2, we briefly review the Karhunen-Loève expansion. Our new algorithm based on the separation of x- and y- variables is given in Section 3. Then in section 4, we apply the proposed method to the Squared Exponential, Rational Quadratic, and Matérn correlations, and make some concluding remarks.

2 Karhunen-Loève Expansion

For any second order random field $u(\mathbf{x},\omega)$ (i.e., the second moment is finite) with $\mathbf{x}$ being the spatial or time coordinates defined over the domain $\mathcal{D}$, the Karhunen Loève expansion takes the form

$$u(\mathbf{x},\omega) = \bar{u}(\mathbf{x}) + \sum_{k=1}^{+\infty} \sqrt{\lambda_k}\, f_k(\mathbf{x})\xi_k(\omega), \tag{1}$$

where $\{\xi_k(\omega)\}$ is a set of uncorrelated random variables with zero mean and unit variance. $\bar{u}(\mathbf{x})$ is the mean of $u(\mathbf{x},\theta)$; λ_k and $f_k(\mathbf{x})$ are the k^{th} eigenvalue and eigen-function of the covariance function $C(\mathbf{x_1},\mathbf{x_2})$, i.e.,

$$\int_{\mathcal{D}} C(\mathbf{x_1},\mathbf{x_2}) f_k(\mathbf{x_1}) d\mathbf{x_1} = \lambda_k f_k(\mathbf{x_2}). \tag{2}$$

Moreover, the uncorrelated random variables $\{\xi_k(\omega)\}$ have an explicit formula

$$\xi_k(\omega) = \frac{1}{\sqrt{\lambda_k}} \int_{\mathcal{D}} \left(u(\mathbf{x},\omega) - \bar{u}(\mathbf{x})\right) f_k(\mathbf{x}) d\mathbf{x}, \tag{3}$$

a consequence of the fact that the eigenfunctions $\{f_k(\mathbf{x})\}$ are orthonormal.

In numerical simulations, a truncated Karhunen Loève expansion

$$u_P(\mathbf{x},\omega) = \bar{u}(\mathbf{x}) + \sum_{k=1}^{P} \sqrt{\lambda_k}\, f_k(\mathbf{x})\xi_k(\omega), \tag{4}$$

is always used. Then the numerical covariance function is

$$\tilde{C}(\mathbf{x_1},\mathbf{x_2}) = \sum_{k=1}^{P} \tilde{\lambda}_k \tilde{f}_k(\mathbf{x_1})\; \tilde{f}_k(\mathbf{x_2}), \tag{5}$$

where $\tilde{\lambda}_k$ and $\tilde{f}_k$ denote the numerically computed eigenvalues and eigen-functions. In order to obtain the truncated KL expansion, one need to: 1) *Compute the eigenvalues and eigenfunctions from Eq.* (2); 2) *Determine the distribution of the uncorrelated random variables* $\{\xi_k\}$. In this paper, we focus on the computation of the eigen-pairs. Without loss of generality, the random field $u(\mathbf{x}, \omega)$ is assumed to have zero mean and unit variance at any point $\mathbf{x}$. (Any nonzero mean or other variance can be easily recovered by rescaling and shifting.)

2.1 Computation of Eigenvalues and Eigen-Functions

For most cases, the analytical formula for the eigenvalues and eigen-functions are not available, leaving numerical method the only choice to solve the eigen problem Eq. (2). In order to numerically solve Eq. (2), one will need to define the approximation to the eigen-function $\{f_k(\mathbf{x})\}$, the covariance function $C(\mathbf{x_1}, \mathbf{x_2})$, and to the equation itself. Due to the concern of stability and other issues, the Galerkin approach is often taken to obtain the approximation for Eq. (2), i.e., the whole equation will be projected to a functional space while requiring the equality still holds.

As for the eigen-functions, the approximation can be either global or piece-wise. It can be in the form of nodal or modal approximation - one choice may have advantage over the other - depending on the situation. The basis functions can be polynomials, wavelets etc. Note that the basis functions are often called shape functions in a piece-wise nodal approximation. There are also several options for the approximation of the covariance function $C(\mathbf{x_1}, \mathbf{x_2})$: **exact, induced or direct approximation**.

- When using the exact covariance function, the Galerkin procedure leads to a generalized eigenvalue problem

  $$\mathbf{C} \cdot \mathbf{V} = \mathbf{M} \cdot \mathbf{V} \cdot \mathbf{\Lambda}, \tag{6}$$

  where

  $$\mathbf{C}_{i,j} = \int_{\mathcal{D}} \int_{\mathcal{D}} C(\mathbf{x_1}, \mathbf{x_2}) \phi_i(\mathbf{x_1})\ \phi_j(\mathbf{x_2}) d\mathbf{x_1} d\mathbf{x_2}, \qquad \text{and} \qquad \mathbf{M}_{i,j} = \int_{\mathcal{D}} \phi_i(\mathbf{x})\ \phi_j(\mathbf{x}) d\mathbf{x}. \tag{7}$$

  Here $\{\phi_i(\mathbf{x})\}$ are the basis used in the approximation of the eigen-functions. The matrix $\mathbf{M}$ is often called the mass matrix. Each column of the matrix $\mathbf{V}$ is an eigenvector, and $\mathbf{\Lambda}$ is a diagonal matrix with eigenvalues on its diagonals. The procedure is computationally expensive if the basis functions are not very carefully chosen, as each element of the matrix $\mathbf{C}$ is an integral over the domain $\mathcal{D} \times \mathcal{D}$.

- If an induced approximation is taken for the covariance function, an approximation (global or local) will be first assumed for $u(\mathbf{x}, \omega)$, and $C(\mathbf{x_1}, \mathbf{x_2})$ is then derived according to its definition. The biggest advantage of using the induced approximation is that when a nodal approximation is used for $u(\mathbf{x}, \omega)$, the expensive computation of the integrals over $\mathcal{D} \times \mathcal{D}$ can be avoided. Let $\{\mathbf{z_1}, \ldots, \mathbf{z_N}\}$ be all the nodes used in the approximation of u. Then

$$u(\mathbf{x}, \omega) = (\phi_1(\mathbf{x}), \ldots, \phi_N(\mathbf{x})) \begin{pmatrix} u(\mathbf{z_1}, \omega) \\ \vdots \\ u(\mathbf{z_N}, \omega) \end{pmatrix} \Longrightarrow C(\mathbf{x_1}, \mathbf{x_2}) = (\phi_1(\mathbf{x_1}), \ldots, \phi_N(\mathbf{x_1}))\hat{\mathbf{C}} \begin{pmatrix} \phi_1(\mathbf{x_2}) \\ \vdots \\ \phi_N(\mathbf{x_2})) \end{pmatrix},$$

where $\hat{\mathbf{C}}$ is simply the discrete correlation matrix with $\hat{\mathbf{C}}_{i,j} = C(\mathbf{z_i}, \mathbf{z_j})$. With the induced approximation, the general eigenvalue problem now becomes:

$$\left(\mathbf{M} \cdot \hat{\mathbf{C}} \cdot \mathbf{M}\right) \cdot \mathbf{V} = \mathbf{M} \cdot \mathbf{V} \cdot \mathbf{\Lambda}. \tag{8}$$

- The proposed algorithm is based on the direct approximation of the covariance function for which the details will be given in the next section.

No matter which choice is taken, finding the eigen-pair for the operator in Eq. (2) turns into a matrix eigen-problem, the computation cost for which can be measured by the flops needed in the matrix-vector multiplication (for the matrix on the left side).

3 Fast Algorithm for Computing Karhunen Loève Expansion

The main idea of our algorithm is to separate x- and y- variables, since the correlation only depends on the difference $(\mathbf{x_1} - \mathbf{x_2})$. The fast calculation of the matrix-vector product is then possible. When the correlation is not separable by nature, a singular value decomposition (SVD) procedure will be applied to write the correlation as a sum of separable functions.

3.1 Direct Approximation of Correlation Function for Stationary Processes

Assume the domain $\mathcal{D} = [-1, 1] \times [-1, 1]$, and $\mathbf{x_i} = (x_i, y_i), i = 1, 2$. Define $r_x = (x_1 - x_2)/2$ and $r_y = (y_1 - y_2)/2$, so that $r_x, r_y \in [-1, 1]$. Let N_x and N_y be the number of modes used in x- and y- directions. Any stationary correlation function can be categorized as either separable or non-separable function.

Case I: A correlation is separable if it can be written as a product of two terms, each of which is a function of r_x or r_y only, but not both. For example, the squared exponential (SE) correlation is separable:

$$\begin{aligned} C(\mathbf{x_1}, \mathbf{x_2}) &= \exp(-\frac{|\mathbf{x_1} - \mathbf{x_2}|^2}{2l^2}) = \exp(-\frac{(x_1 - x_2)^2}{2l^2}) \ \exp(-\frac{(y_1 - y_2)^2}{2l^2}) \\ &= \exp(-\frac{2r_x^2}{l^2}) \ \exp(-\frac{2r_y^2}{l^2}). \end{aligned} \tag{9}$$

where l is the correlation length.

Let $F(r_x)$ be the term including only r_x. Approximate it with Chebyshev series $\sum_{i=0}^{N_x} f_i T_i(r_x)$. The coefficients can be obtained by either interpolation or L^2 projection. Then we rewrite each Chebyshev basis as a function of x_1 and x_2 as

$$T_i(r_x) = T_i(\frac{x_1 - x_2}{2}) = \sum_{k+j \le i} a_{k,j}^{(i)} T_k(x_1) T_j(x_2), \qquad i = 0, 1, \ldots, N_x,$$

where the coefficients $a_{k,l}^{(i)}$ can be easily computed by using the orthogonality of Chebyshev polynomials, combined with Gauss-Chebyshev quadrature. The last step is to change the basis to orthonormal Legendre polynomials as the weight function is simply one in the original eigenvalue problem. Apply the same procedure to the term that includes only r_y. In the end, the approximated correlation function takes the form

$$C(\mathbf{x_1}, \mathbf{x_2}) \approx \left(\sum a_{i,j}^{(*)} \bar{P}_i(x_1) \bar{P}_j(x_2) \right) \cdot \left(\sum a_{i,j}^{(**)} \bar{P}_i(y_1) \bar{P}_j(y_2) \right), \tag{10}$$

where $\bar{P}_i$ is the orthonormal Legendre polynomial with degree i. Denote the matrices in Eq. (10) as $\mathbf{A}_1$ and $\mathbf{A}_2$ - both of size $(N_x + 1)$-by-$(N_y + 1)$

$$\mathbf{A}_1 = \left(a_{i,j}^{(*)} \right) \qquad \mathbf{A}_2 = \left(a_{i,j}^{(**)} \right)$$

Now approximate the eigen-function $f(x, y)$ with orthonormal Legendre series $\sum f_{i,j} \bar{P}_i(x) \bar{P}_j(y)$. Reorder the basis functions $\{\bar{P}_i(x)\bar{P}_j(y)\}$ as $\{\Phi_1, \Phi_2, \ldots, \Phi_{(N_x+1)(N_y+1)}\}$ such that $\Phi_I = \bar{P}_{I_s(I)}(x)\bar{P}_{J_s(I)}(y)$. The indices I_s and J_s are

$$\begin{aligned} I_s &= 0, \ldots, N_x, \quad 0, \ldots, N_x, \quad \ldots, \quad 0, \ldots, N_x, \\ J_s &= 0, \ldots, 0, \quad 1, \ldots, 1, \quad \ldots, \quad N_y, \ldots, N_y, \end{aligned}$$

where the pattern repeats $(N_y + 1)$ and $(N_x + 1)$ times respectively. With the above ordering of the basis functions, the eigenvalue problem after the Galerkin procedure becomes

$$\mathbf{A} \cdot V = V \cdot \Lambda, \qquad \text{with} \qquad \mathbf{A} = \mathbf{A}_1(I_s, I_s) .* \ \mathbf{A}_2(J_s, J_s), \tag{11}$$

where $.*$ refers to the element-by-element product of two matrices.

Case II: For non-separable correlation functions, we start with an approximation of the correlation function as

$$C(\mathbf{x_1}, \mathbf{x_2}) = C(r_x, r_y) \approx (T_0(r_x), \ldots, T_{N_x}(r_x)) \ B \begin{pmatrix} T_0(r_y) \\ \vdots \\ T_{N_y}(r_y) \end{pmatrix},$$

then compute the singular value decomposition of the matrix B and discard the insignificant terms. Now the approximated correlation is

$$C(\mathbf{x_1}, \mathbf{x_2}) \approx \sum_{j=1}^{S} \sigma_j \left[(T_0(r_x), \ldots, T_{N_x}(r_x)) U_j \right] \cdot \left[V_j^T \begin{pmatrix} T_0(r_y) \\ \vdots \\ T_{N_y}(r_y) \end{pmatrix} \right],$$

in which $\{U_j\}$ and $\{V_j\}$ are the corresponding columns and rows of the left and right matrices in the singular value decomposition. Note that each term in the summation is separable, for which the technique described in **Case I** can be readily applied. The final eigenvalue problem becomes

$$\left(\sum_{j=1}^{S} \mathbf{A}^{(j)} \right) \cdot V = V \cdot \Lambda. \tag{12}$$

The computational cost will be proportional to the value of S, which is found to be small for all the cases tested. For a couple of worst situations, S can be as large as 12, if all the terms with singular value slightly bigger than machine zero are kept.

3.2 Fast Matrix-Vector Multiplication

The matrix $\mathbf{A}$ in Eq. (11) (or $\mathbf{A}^{(\mathrm{j})}$ in Eq. (12) for non-separable correlation functions) has a block structure. More precisely it has $(N_y + 1) \times (N_y + 1)$ blocks, with each block being a $(N_x + 1) \times (N_x + 1)$ matrix. In particular, the (i, j) block of the $\mathbf{A}$ is $\mathbf{A}_1 \cdot a_{i,j}^{(**)}$.

Dividing a column vector f of size $(N_x + 1)(N_y + 1)$ into $f = (f_0, f_1, \ldots, f_{Ny})^T$, with each f_i being a vector of size (N_x+1), the matrix-vector product $\mathbf{A}f$ can be written as

$$\mathbf{A}f = \begin{pmatrix} a^{(**)}_{0,0}\mathbf{A}_1 f_0 + a^{(**)}_{0,1}\mathbf{A}_1 f_1 + \ldots + a^{(**)}_{0,N_y}\mathbf{A}_1 f_{N_y} \\ \ldots \\ a^{(**)}_{N_y,0}\mathbf{A}_1 f_0 + a^{(**)}_{N_y,1}\mathbf{A}_1 f_1 + \ldots + a^{(**)}_{N_y,N_y}\mathbf{A}_1 f_{N_y} \end{pmatrix}.$$

Clearly, the products $\mathbf{A}_1 f_0, \mathbf{A}_1 f_1, \ldots, \mathbf{A}_1 f_{N_y}$ can be computed once and saved. Plus the matrices $\mathbf{A}_1$ and $\mathbf{A}_2$, the memory requirement is in the order of $O(N_x N_y)$. (*We assumed $N_x = O(N_y)$ for the seek of estimates of memory requirement and computation cost.*) As for the computation complexity, we need to

- Compute $\mathbf{A}_1 f_i$ for all $i = 0, \ldots, N_y$ that needs $O((N_y+1)(N_x+1)^2)$ flops,
- And then compute $a^{(**)}_{i,j} \cdot (\mathbf{A}_1 f_j)$ for all $0 \le i, j, \le N_y$, for which the amount of flops needed is $O((N_x+1)(N_y+1)^2)$.

Let $M = (N_x+1)(N_y+1)$ be the size of the matrix, then the required storage is linear $O(M)$ and the total flops is of the order $O(M\sqrt{M})$.

4 Numerical Results and Discussions

Three types of correlation functions picked from [19] were considered:

Squared exponential: $$C(\mathbf{x_1}, \mathbf{x_2}) = \exp\left(-\frac{r^2}{2l^2}\right),$$

Rational quadratic: $$C(\mathbf{x_1}, \mathbf{x_2}) = \left(1 + \frac{r^2}{2\alpha l^2}\right)^{-\alpha},$$

Matérn: $$C(\mathbf{x_1}, \mathbf{x_2}) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}\, r}{l}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}\, r}{l}\right),$$

with the domain $[0,10] \times [0,10]$. In the above formula, $r = |\mathbf{x_1} - \mathbf{x_2}|$, l is the correlation length, and K_ν is the modified Bessel function. In the following tests, we let $\alpha = 4$ for the rational quadratic correlation, and $\nu = 1.5, 2.5$ (the two most interesting cases for machine learning) for the Matérn correlation. Note that the regularity of the Matérn correlation increases as the value of ν gets bigger, with two extreme cases: the exponential correlation $\exp(-r/l)$ when $\nu = 0.5$ and the squared exponential correlation when $\nu \to +\infty$.

Our algorithm was compared against the approach that uses **a piece-wise approximation for the eigenfunctions and an induced approximation for the correlation function**. For the latter, a uniform triangular mesh with axis-parallel grids was used so that the discrete correlation matrix $\hat{\mathbf{C}}$ is a *block Toeplitz matrix*. More precisely, the square domain is first partitioned into smaller squares of the same size, each of which is then divided into two triangles. The eigen-function is then approximated on each triangle by a degree p polynomial (with uniform nodes). Because of the Toeplitz structure, the complexity is $O(M^{3/2}\log M)$ when employing the fast Fourier transform for the matrix-vector multiplication. (Again we assume that $N_x = O(N_y)$ when estimating the computation cost.)

Same as in [1], we only considered the global relative error for the variance:

$$\frac{1}{|\mathcal{D}|}\int_{\mathcal{D}}\left|\frac{\tilde{C}(\mathbf{x},\mathbf{x}) - C(\mathbf{x},\mathbf{x})}{C(\mathbf{x},\mathbf{x})}\right| d\mathbf{x} = \left|1 - \frac{\sum_{k=1}^{P}\tilde{\lambda}_k}{|\mathcal{D}|}\right|. \tag{13}$$

Of course, when numerically computing the KL expansion, only the enough accurate eigenvalues and eigenfunctions will be included in the KL expansion, and in the above error formula as well.

The CPU time needed to achieve a prescribed accuracy was used as the main indicator of the algorithm's efficiency.

Step I: For either method, we find the largest mesh size (or smallest matrix size) that leads to the prescribed accuracy for the global variance in Eq. (13), by the following iterations:

- Input: prescribed accuracy ϵ, polynomial order p on each element, and the correlation function.

- Procedure: Start with a very coarse mesh, and compute enough eigenvalues and eigen-functions. (The solver irbleigs [4] was used.) Using the eigen-pairs from a very fine mesh as the 'exact' solution, compute the relative error for the eigenvalues and keep in the KL expansion the terms whose corresponding error is less than $\epsilon/10$. Then compute the variance error as shown in (13). Stop if it is less than ϵ. Otherwise, refine the mesh and repeat the above steps.

The above description is for the piece-wise algorithm. For our new algorithm, one only needs to increase the number of modes instead of refining the mesh.

Step II: We then rerun both methods with the largest mesh size (or smallest orthonormal Legendre modes) that leads to the prescribed accuracy, and record the CPU time. For fair comparison, multi-threading is now disabled. All the computations were done with 64-bit Matlab on Dell T7500N Workstation with Intel(R) Xeon(R) CPU X5680 @3.33GHz.

The results when the correlation length is one and five are given in Table 1 and 2. As shown in Tab. 1, the algorithm proposed in this paper outperforms the piece-wise method when the correlation length is not large. It is particularly true when the correlation function is analytic (i.e., the Squared Exponential correlation). Another observation is that the piece-wise quadratic approximation is always much more efficient than the piece-wise linear approximation. But there is no gain at all when the order is increased to the piece-wise cubic approximation if the correlation gets rough, e.g., the Matérn correlation with $\nu = 1.5$. For completeness, two eigen-functions are included in Fig. 1 for the Matérn correlation with $\nu = 1.5$.

However, the proposed algorithm performs not as good as the piece-wise method when the correlation length is very large. *Note that correlation length* $= 5$ *is unreasonably large in practice for a domain size* $[0, 10] \times [0, 10]$. Nevertheless, it indicates that a lower order piece-wise approximation can yield accurate eigenvalues if very few them need to be computed.

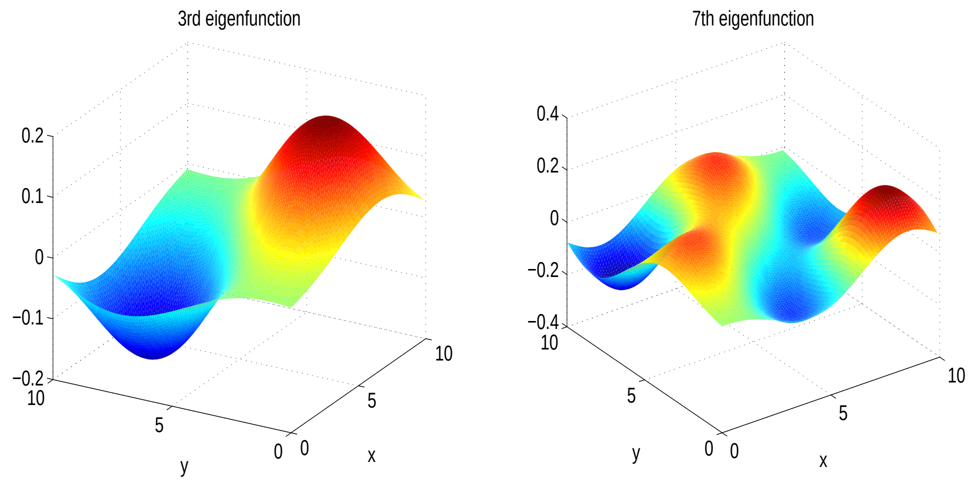
Table 1: CPU time for computing the KL expansion with prescribed accuracy $\epsilon = 0.1$. The last column is from the new algorithm. Correlation length = 1. SE: Squared Exponential; RQ: Rational Quadratic.

	$p = 1$	$p = 2$	$p = 3$	Legendre
SE	389.76	30.85	14.67	2.28
RQ	29.31	1.67	1.04	1.06
Matérn ($\nu = 2.5$)	36.82	2.54	2.43	2.17
Matérn ($\nu = 1.5$)	108.63	5.83	8.19	5.70

Table 2: CPU time for computing the KL expansion with accuracy $\epsilon = 0.1$. Correlation length = 5.

	$p = 1$	$p = 2$	$p = 3$	Legendre
SE	0.533	0.075	0.046	0.117
RQ ($\alpha = 4$)	0.112	0.022	0.026	0.218
Matérn ($\nu = 2.5$)	0.245	0.048	0.031	0.688
Matérn ($\nu = 1.5$)	0.282	0.057	0.059	0.535

Acknowledgment: The author's work was supported in part by the National Science Foundation under the grant DMS-1016047.


3rd eigenfunction

7th eigenfunction

Figure 1: Example eigen-functions for Matérn correlation, $\nu = 1.5$. Correlation length = 1.

References

[1] D. L. Allaix, V. I. Carbone, and G. Mancini. Adaptive discretization of 1D homogeneous random fields. In M. et al., editor, *Safety, Reliability and Risk Analysis: Theory, Methods and Applications*, pages 51–65. Taylor & Francis Group, London, 2009.

[2] I. Babuška, F. Nobile, and R. Tempone. A stochastic variational multiscale method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal.*, 45(3):1005–1034, 2007.

[3] I. Babuška, R. Tempone, and G. E. Zouraris. Galerkin finite element approximations of stochastic elliptic differential equations. *SIAM J. Numer. Anal.*, 42:800–825, 2004.

[4] J. Baglama, D. Calvetti, and L. Reichel. Irbl: An implicitly restarted block Lanczos method for large-scale Hermitian eigenproblems. *SIAM J. Sci. Comput*, 24:1650–1677, 2003.

[5] Q.-Y. Chen, D. Gottlieb, and J. S. Hesthaven. Uncertainty analysis for the steady-state flows in a dual throat nozzle. *J. Comput. Phy.*, 204:378–398, 2005.

[6] A. Cohen and J.-P. D'Ales. Nonlinear approximation of random functions. *SIAM J. Appl. Math.*, 57(2):518–540, 1997.

[7] O. Ditlevsen. Dimension reduction and discretization in stochastic problems by regression methods. In F. Casciati and B. Roberts, editors, *Mathematical Models for Structual Reliability Analysis, CRC Mathematical Modelling Series,2, 51-138*, 1996.

[8] R. G. Ghanem. Stochastic finite elements for heterogeneous media with multiple random non-Gaussian properties. *ASCE J. Eng. Mech.*, 125(1):26–40, 1999.

[9] R. G. Ghanem and P. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Springer, Berlin, 1991.

[10] M. Grigoriu. Simulation of stationary non-Gaussian translation processes. *J. Eng. Mech.*, 124(2):121–126, 1998.

[11] T. Hou, W. Luo, B. Rozovskii, and H. M. Zhou. Wiener chaos expansions and numerical solutions of randomly forced equations of fluid mechanics. *J. Comput. Phys.*, 217:687–706, 2006.

[12] S. P. Huang, S. T. Quek, and K. K. Phoon. Convergence study of the truncated Karhunen-Loève expansion for simulation of stochastic processes. *Int. J. Numer. Meth. Engng.*, 52:1029–1043, 2001.

[13] B. N. Khoromskij, A. Litvinenko, and H. G. Matthies. Application of hierarchical matrices for computing the Karhunen-Loève expansion. *Computing*, 84:49–67, 2009.

[14] C.-C. Li and A. Der Kiureghian. Optimal discretization of random fields. *J. Eng. Mech.*, 119(6):1136–1154, June 1993.

[15] M. Loève. *Probability Theory*. Springer-Verlag, New York, 4th edition, 1977.

[16] G. Matthies, C. Brenner, and C. Guedes Soares. Uncertainties in probabilistic numerical analysis of structures and solids - stochastic finite elements. *Struct. Safety*, 19(3):283–336, 1997.

[17] W. L. Oberkampf, J. C. Helton, and K. Sentz. Mathematical representation of uncertainty. In *AIAA Non-Deterministic Approaches Forum*, Seattle, WA, April 2001. Paper No. 2001-1645.

[18] K. K. Phoon, H. W. Huang, and S. T. Quek. Comparison between Karhunen-Loève and wavelet expansion for simulation of Gaussian processes. *Comput. Struct.*, 82:985–991, 2004.

[19] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. the MIT Press, 2006.

[20] G. I. Schuëller (Editor). A state-of-the-art report on computational stochastic mechanics. *Probabilistic Engineering Mechanics*, 12(4):197–321, 1997. IASSAR report.

[21] C. Schwab and R. A. Todor. Sparse finite elements for elliptic problems with stochastic data. *Numer. Math.*, 95:707–734, 2003.

[22] C. Schwab and R. A. Todor. Karhunen-Loève approximation of random fields by generalized fast multipole methods. *J. Comput. Phy.*, 217:100–122, 2006.

[23] B. Sudret and A. Der Kiureghian. Stochastic finite element methods and reliability: A state-of-the-art report. Report UCB/SEMM-2000/08, Department of Civil & Environmental Engineering, University of California, Berkeley, November 2000.

[24] J. Tausch and A. Weckiewicz. Multidimensional fast Gauss transforms by Chebyshev expansions. *SIAM J. Sci. Comput.*, 31(5):3547–3565, 2009.

[25] X. Wan and G. E. Karniadakis. An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *J. Comput. Phys.*, 209(2):617–642, 2005.

[26] D. Xiu and G. E. Karniadakis. Modeling uncertainty in flow simulations via generalized polynomial chaos. *J. Comput. Phy.*, 187:137–167, 2003.

[27] J. Zhang and B. Ellingwood. Orthogonal series expansion of random fields in reliability analysis. *J. Eng. Mech.*, 120(12):2660–2677, December 1994.