# Karhunen-Loève Decomposition with Polynomial Chaos Surrogate

K. Sargsyan
Sandia National Laboratories, Livermore, CA

December 21, 2019

Consider a function $f(x;\lambda)$ where $x$ is the design condition, e.g. location, time, or simply enumerating outputs of the model $f()$. The uncertain parameters are denoted by $\lambda \in \mathbb{R}^d$, where $d$ is the dimensionality of the input parameter space. Typically, we are given $N_x \times S$ forward model evaluations $f_{ij} = f(x_i, \lambda_j)$, often called *training* runs with $i = 1, \dots, N_x$ and $j = 1, \dots, S$.

The goal is to build a cheap-to-evaluate surrogate model $f_c(x, \lambda)$ that approximates $f(x, \lambda)$ over a given range (typically a hypercube) of $\lambda$, for all the design conditions of interest $x_i$. That is, $f(x_i, \lambda) \approx f_c(x_i, \lambda)$ for $i = 1, \dots, N_x$. Jumping ahead, that surrogate model is the right hand side of Eq. (1) (naïve) or Eq. (4) (KL-based).

The naïve approach is to build a separate surrogate for each of the $N_x$ conditions, i.e.

$$f(x_i;\lambda) \approx \sum_{k=0}^{K} c_{ik}\Psi_k(\lambda) \tag{1}$$

but this requires $N_x$ surrogate constructions. Often, when $x$ corresponds to a regional grid, $N_x$ can be a large number, making this quite inefficient.

Instead, in cases when there are strong correlations in $x$-dimension (this is very often the case, again, when $x$ denotes a spatio-temporal dimension), it is better to first reduce the dimensionality via Karhunen-Loève decomposition in the following way.

$$f(x;\lambda) \approx \bar{f}(x) + \sum_{m=1}^{M} \xi_m(\lambda)\sqrt{\mu_m}\phi_m(x) \tag{2}$$

where $\bar{f}(x)$ is the mean field, $\xi_m(\lambda)$ are KL coefficients, and $\sqrt{\mu_m}\phi_m(x)$ are KL modes scaled by the square-root of eigenvalues.

Now, given training samples $f_{ij} = f(x_i, \lambda_j)$, the KL decomposition produces training samples $\xi_{mj} = \xi_m(\lambda_j)$ for $j = 1, \dots, S$, so one needs to construction PC surrogate for each of the $M$ KL coefficients only. Usually, $M \ll N_X$, hence the computational efficiency.

Let's plug in the PC expansions for

$$\xi_m(\lambda) \approx \sum_{k=0}^{K} b_{mk}\Psi_k(\lambda) \tag{3}$$

into the KL form (2). We obtain

$$\begin{aligned} f(x;\lambda) &\approx \bar{f}(x) + \sum_{m=1}^{M}\sum_{k=0}^{K} b_{mk}\Psi_k(\lambda)\sqrt{\mu_m}\phi_m(x) \\ &= \bar{f}(x) + \sum_{k=0}^{K}\left[\sum_{m=1}^{M} b_{mk}\sqrt{\mu_m}\phi_m(x)\right]\Psi_k(\lambda) \\ &= \sum_{k=0}^{K}\left[\bar{f}(x)\delta_{k,0} + \sum_{m=1}^{M} b_{mk}\sqrt{\mu_m}\phi_m(x)\right]\Psi_k(\lambda) \end{aligned} \tag{4}$$

which is essentially the same as (1), only reorganized a bit (the mean field is absorbed into the 0-th order coefficient via Kronecker delta $\delta_{k,0}$).

One can compute PC sensitivities for each eigenmode using (3), or for each $x_i$ using (4) (similar to (1)). Note that the mean field will not affect sensitivities.

Also note that some bookkeeping is required in case when PCs are computed via Bayesian compressed sensing, since different variables may have different set of bases (in the above computation, we used all bases and enumerated all of them from $k = 0$ to $K$).