

Part 1

Subsection (a)

(i)

$Y(x, \omega)$ is a stationary Gaussian process in 1 dimension, with x as the dimensional variable. Consequently, the mean is independent of position ($\mu \neq \mu(x)$) and the covariance is only a function of the displacement.

$$\mathbb{E}\{Y(x, \omega)\} = \mu_Y \quad (1)$$

$$\mathbb{Cov}\{x_1, x_2\} = \sigma_Y^2 \exp \left[-\frac{1}{p} \left(\frac{|x_1 - x_2|}{L} \right)^p \right] \quad (2)$$

where $L = 0.3$, $\mu_Y = 1.0$, $p = 1.0$, $\sigma_Y^2 = 0.3$.

We develop a Karhunen-Loève (KL) expansion of $Y(x, \omega)$, as given by:

$$Y(x, \omega) = \mu_Y(x) + \sum_{i=1}^{\infty} \sqrt{\lambda_i} \psi_i(x) Z_i(\omega) \quad (3)$$

As $Y(x, \omega)$ is a Gaussian process, then $Z_i(\omega) \sim N(0, 1)$, and all $Z_i(\omega)$ for $i \in [0, \infty)$ are jointly Gaussian and independent.

Using a finite number of eigenvectors, the KL expansion can be truncated and modeled with a finite sum, given by:

$$Y^m(x, \omega) = \mu_Y(x) + \sum_{i=1}^m \sqrt{\lambda_i} \tilde{\psi}_i(x) Z_i(\omega) \quad (4)$$

To find the eigenvectors $\tilde{\psi}_i(x)$ and the eigenvalues λ_i , we employ the Nyström method. Through eigendecomposition, the Nyström method calculates the eigenvectors and eigenvalues at selected quadrature points, using quadrature points and weight generated from the Gauss-Legendre quadrature rule. Here, the Gauss-Legendre quadrature rule based on *qrule.m* is used to generate required quadrature points and weights (x_k and w_k). The points and weights are renormalized to the $[0, 1]$ domain, given that *qrule.m* is based on the $[-1, 1]$ interval.

$$\sum_{k=1}^m \mathbb{Cov}(x, x_k) \tilde{\psi}_i(x_k) w_k = \tilde{\lambda}_i \tilde{\psi}_i(x) \quad (5)$$

In matrix form, the equation can be expressed as:

$$\mathbf{C} \mathbf{W} \boldsymbol{\psi}_i = \tilde{\lambda}_i \boldsymbol{\psi}_i \quad (6)$$

where $\mathbf{C} \in \mathbb{R}$ is the matrix of covariances, $\mathbf{W} \in \mathbb{R}^{m \times m}$ is a diagonal matrix of quadrature weights, and $\boldsymbol{\psi}_i \in \mathbb{R}^m$ is the vector of eigenvectors.

To ensure that the coefficient matrix is symmetric, the eigenvalue problem is re-expressed as:

$$\mathbf{W}^{1/2} \mathbf{C} \mathbf{W}^{1/2} \boldsymbol{\psi}'_i = \tilde{\lambda}_i \boldsymbol{\psi}'_i \quad (7)$$

where $\boldsymbol{\psi}'_i = \mathbf{W}^{1/2} * \boldsymbol{\psi}_i$.

We solve for $\boldsymbol{\psi}'_i(x_k)$ and $\tilde{\lambda}_i(x_k)$ using the *eig* function on MATLAB. To obtain eigenvectors at non-quadrature points, interpolation between the quadrature point is performed using:

$$\tilde{\psi}_i(x) = \frac{1}{\lambda_i} \sum_{k=1}^m \text{Cov}(x, x_k) \tilde{\psi}_i(x_k) w_k \quad (8)$$

Finally, the KL expansion of $Y^m(x, \omega)$ is given by:

$$Y^m(x, \omega) = \mu_Y(x) + \sum_{i=1}^m \sqrt{\lambda_i} \tilde{\psi}_i(x) Z_i(\omega) \quad (9)$$

The error of the truncation in L_2 is given by:

$$\mathbb{E} [\|Y - Y^m\|_{L^2}^2] = \sum_{i=m+1}^t \lambda_i \quad (10)$$

To estimate this error, we set the upper limit to a very large number ($t = 10^8$), and compute the sum from the $(m+1)^{th}$ to the highest order eigenvalue. The results are summarized in Fig. 1. From the figure, we see that the truncation error from the KL expansion decreases very quickly with the stochastic dimension. A residual of lower than 1 % is attained with approximately $n = 50$. This is mathematically reasonable as the Gaussian quadrature algorithm converges to the real integrand with increasing n at spectral rates. However, spectral convergence rates were not observed for the eigenvalues. This causes the higher order eigenvalues to decrease quickly with the number of dimensions, resulting in the observed outcome.

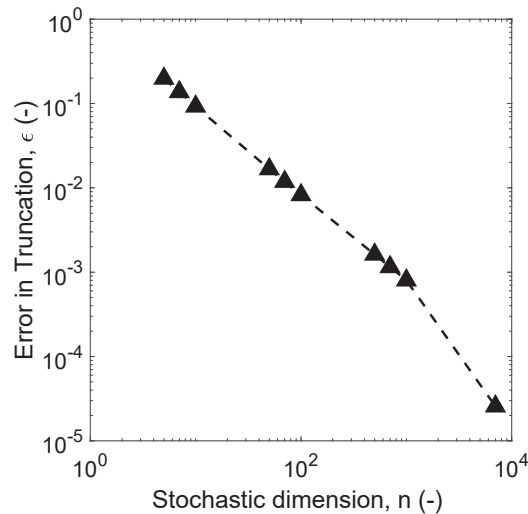


Figure 1: Plot of the truncation error as a function of the stochastic dimension, n . The truncation error converges algebraically with n , attaining a residual of ~ 0.01 with $n = 100$.

(ii)

Here, we use the KL expansion of Y to compute a total-degree polynomial chaos expansion of $k(x, \omega)$. The polynomial degree is denoted by p , while the number of stochastic dimension is denoted by n .

$$k(x, \omega) = \exp[Y(x, \omega)] \quad (11)$$

$$= \sum_{\|\vec{\alpha}\|_1 < p} C_{\vec{\alpha}} \text{He}_{\vec{\alpha}}(x, Z_1, \dots, Z_n) \quad (12)$$

To obtain multidimensional $C_{\vec{\alpha}}$, we generalize the single dimensional derivation presented for C_{α} in the lecture notes. With 1 random variable ($n = 1$), the following derivations were obtained:

$$\langle \text{He}_{\alpha_i}(x), \text{He}_{\alpha_i}(x) \rangle = \alpha_i! \quad (13)$$

$$\langle k(x, \omega), \text{He}_{\alpha_i}(x) \rangle = \int_{-\infty}^{\infty} \exp(\mu_Y + \sigma_Y Z_i) \text{He}_{\alpha_i}(x) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{Z_i^2}{2}\right) dZ_i \quad (14)$$

$$= \exp\left(\mu_Y + \frac{\sigma_Y^2}{2}\right) \sigma_Y^{\alpha_i} \quad (15)$$

Therefore, C_{α_i} can be computed by:

$$C_{\alpha_i} = \exp\left(\mu_Y + \frac{\sigma_Y^2}{2}\right) \frac{\sigma_Y^{\alpha_i}}{\alpha_i!} \quad (16)$$

For multiple random variables, the formulation for $C_{\vec{\alpha}}$ is derived as:

$$C_{\vec{\alpha}} = \frac{\langle k(x, \omega), \text{He}_{\vec{\alpha}}(x) \rangle}{\langle \text{He}_{\vec{\alpha}}(x), \text{He}_{\vec{\beta}}(x) \rangle} \quad (17)$$

$$\langle \text{He}_{\vec{\alpha}}(x), \text{He}_{\vec{\beta}}(x) \rangle = \int \dots \int \left(\prod_{i=1}^n \text{He}_{\alpha_i}(Z_i) \right) \left(\prod_{j=1}^n \text{He}_{\beta_j}(Z_j) \right) f_{\vec{Z}}(\vec{z}) d\vec{z} \quad (18)$$

$$= \prod_{i=1}^n \int_{-\infty}^{\infty} \text{He}_{\alpha_i}(z_i) \text{He}_{\beta_i}(z_i) f_{Z_i}(z_i) dz_i \quad (19)$$

$$= \prod_{i=1}^n \alpha_i! \quad (20)$$

$$\langle k(x, \omega), \text{He}_{\vec{\alpha}}(x) \rangle = \int \dots \int \exp\left[\mu_Y + \sum_{i=1}^n \sqrt{\lambda_i} \tilde{\psi}_i(x) Z_i\right] \prod_{i=1}^n \text{He}_{\alpha_i}(z_i) f_{\vec{Z}}(\vec{z}) d\vec{z} \quad (21)$$

$$= \exp(\mu_Y) \prod_{i=1}^n \int_{-\infty}^{\infty} \exp[\sqrt{\lambda_i} \tilde{\psi}_i(x) Z_i] \text{He}_{\alpha_i} f_{Z_i}(z_i) dz_i \quad (22)$$

$$= \exp(\mu_Y) \prod_{i=1}^n \left[\exp\left(\frac{\lambda_i \tilde{\psi}_i^2(x)}{2}\right) (\sqrt{\lambda_i} \tilde{\psi}_i(x))^{\alpha_i} \right] \quad (23)$$

(iii)

We first explore how the mean and variance of $k(x, \omega)_{PCE}$ varies with n and p , before quantifying them with a metric. Here, 50 Monte Carlo samples are taken to calculate the statistical moments. As seen in Fig. 9, the expected value of $k(x, \omega)_{PCE}$ appears to converge towards the true value e^1 with increasing n and p . In fact, the expected value of $k(x, \omega)_{PCE}$ lies within 1 standard deviation of the true value with $n = 3$. On the other hand, for a fixed number of Monte Carlo samples and polynomial degree, the variance of the PCE decrease with n to a minimum at $n = 6$, before increasing with n . A likely explanation might be that the ratio of PCE terms relative to the number of Monte Carlo samples increases quickly with n , and the individual randomness of the Z_i in the PCE expansion is additive, propagating to yield a larger variance.

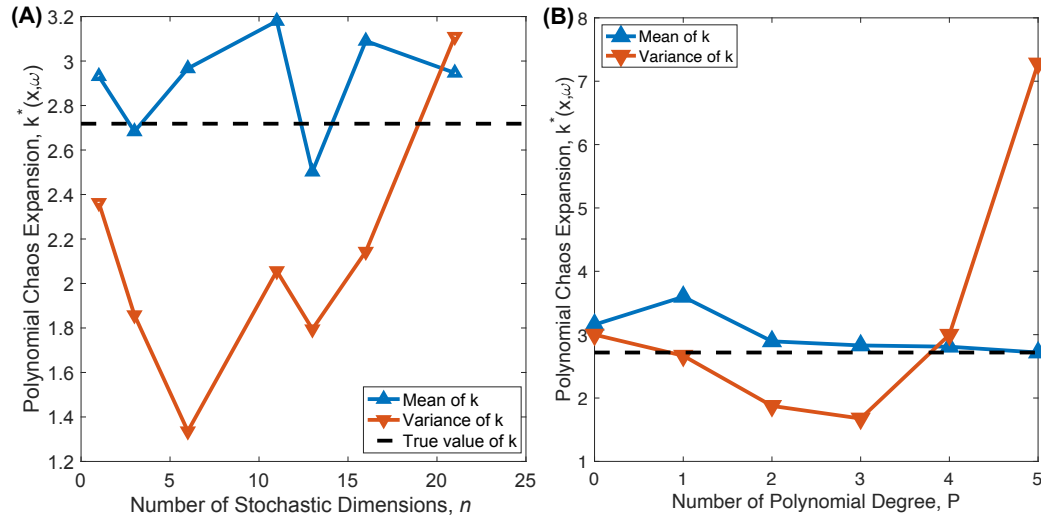


Figure 2: Plot of expected value and variance of $k(x, \omega)_{PCE}$, as a function of n and p . Here, 50 Monte Carlo samples are used to calculate the respective statistical moments.

We analyze the convergence of the polynomial chaos expansion (PCE) of the log-normal random variable. In general, for a given number of random variables in the PCE formulation, the expected value of $k(x, \omega)_{PCE}$ converges to the true value (here, the true value is the exponential of the mean value of Y) with increasing polynomial degrees of freedom. Further, for a given value of p , we see that the variance of the $k(x, \omega)_{PCE}$ appears to increase with the number of random variable n . Thus, to reduce the bias and obtain accurate means, we require a sample size much larger n values.

To quantify the error, we define the L^2 -type error as:

$$\delta_{L^2} = \|k(x, \omega)_{PCE} - k_{true}\|_{L^2} \quad (24)$$

$$= \int_B (k(x, \omega)_{PCE} - k_{true})^2 f_{\vec{Z}} d\vec{Z} \quad (25)$$

$$= \mathbb{E}_{\Omega} \{(k(x, \omega)_{PCE} - k_{true})^2\} \quad (26)$$

Here, $k_{true} = e^1$, the expected value of the actual log-normal random variable $k(x, \omega)$. The plot of the L^2 error is depicted in Fig. 3. Similar to the observations in Fig. 9, the L^2 -error converges very quickly with p . Also, it is worth noting that the higher the number of stochastic dimensions (n), the larger the polynomial degree required for convergence. This is mathematically reasonable

as the more terms in the PCE is required to capture the variance induced by the increasing degree of stochastic freedom, leading to the observed outcome.

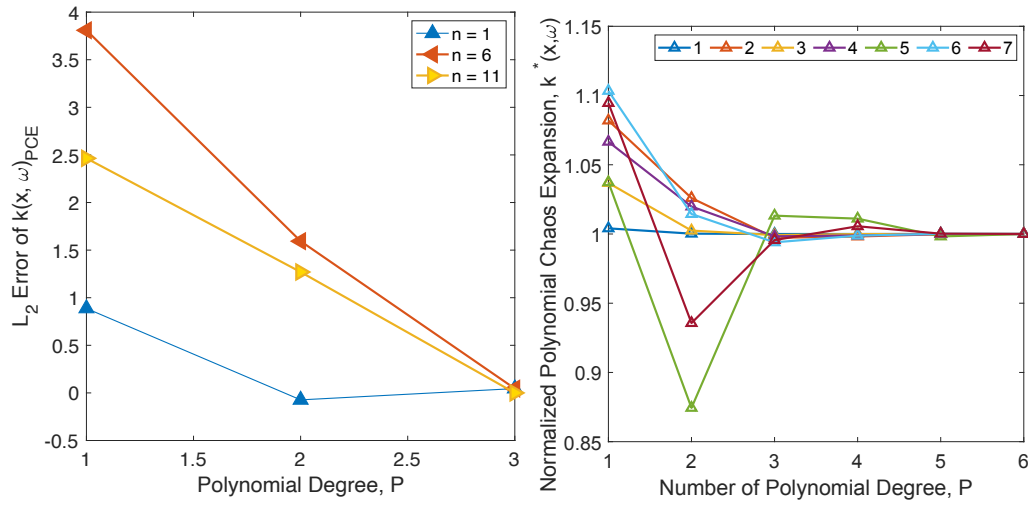


Figure 3: (A) Plot of the L_2 error as a function of n and p . (B) Convergence of the expected value with n and p , using 50 Monte Carlo points. The legend illustrates the color scheme used for the number of stochastic dimensions.

Subsection (b)

In this section, we plot realization of $k(x, \omega)$ based on three formulations, namely the PCE, KL and the untruncated expansions. Using 100 Monte Carlo samples, 5 stochastic dimensions and 3 polynomial degrees, the probability density of the solution, u , at three different locations is presented in Fig. 4, 5 and 6. Across the distributions, the distributions of the untruncated and KL expansions of Y appear to be very similar, as compared to the distribution from the PCE of Y . Unlike the untruncated and KL expansions, the PCE method yielded a narrower probability density distribution, causing more points to be more centered around the expected value. The distribution of the probability density of the KL expansion of Y is very similar to the untruncated method. This is mathematically reasonable. Using Monte Carlo sampling, we observed that the expected value and variance from 50 samples of the KL expansion of Y is within 5% of the original values (mean = 1, variance = 0.3). Therefore, with 5 stochastic dimensions, the samples indicate that the KL expansion of Y behaves very similar to the untruncated Y , leading to the almost similar behaviors in the probability density. The deviation of the PCE method from the untruncated and KL expansion methods likely occurred due to the numerical approximation from a finite truncation of expansion.

The mean field of the three methods for the $[0,1]$ domain is presented in Fig. 7. We observe that the mean value at $x = 0$ is consistently at 1, by virtue of the Dirichlet boundary condition, and the gradient at $x = 0$ is fixed across all three methods, by virtue of the Neumann boundary condition. In general, the untruncated Y method yielded solutions that are higher in expected value across the entire domain as compared to the other two methods. However, we also observe that the KL expansion and PCE methods yielded very similar expected value for a given n and p . These values are approximately within 10 % of the untruncated estimates, demonstrating that the

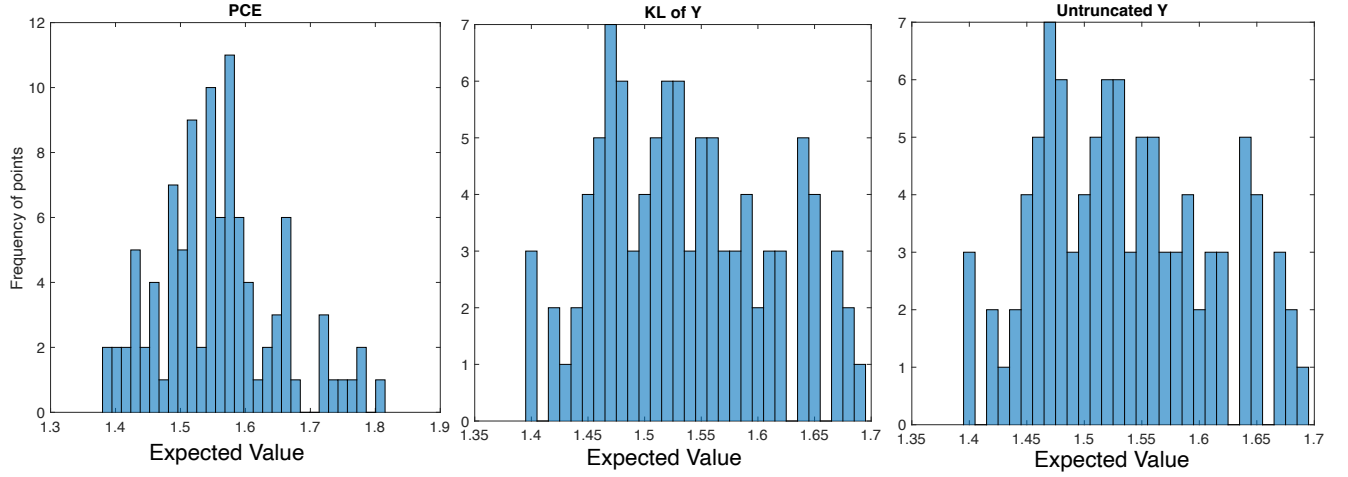


Figure 4: Plot of the probability density function of $k(x, \omega)$ as a function of the expected value for (A) polynomial chaos expansion, (B) Karhunen-Loeve expansion and (C) untruncated expansion at $x = 0.5$.

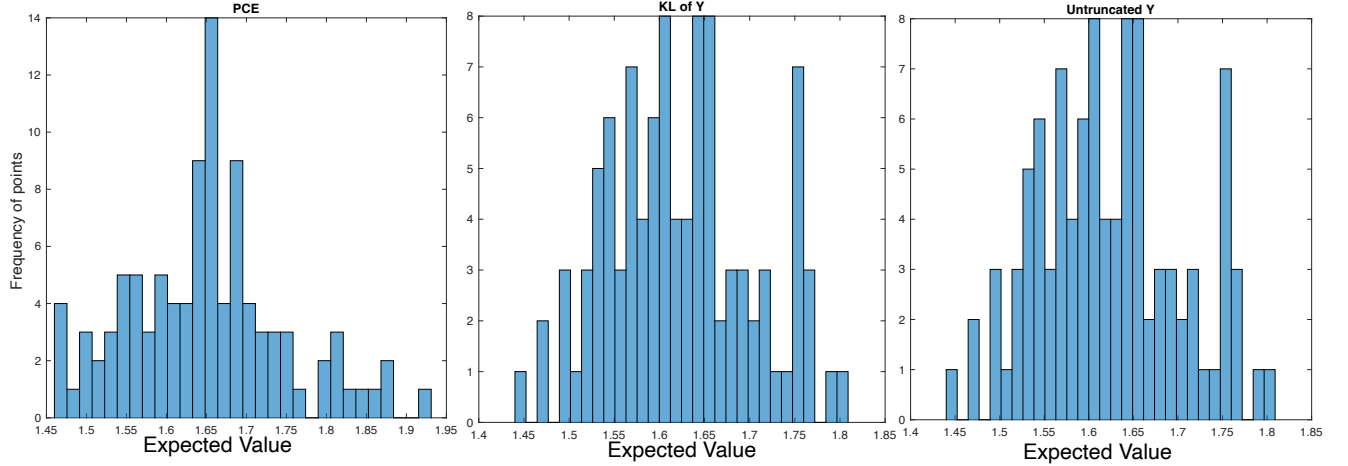


Figure 5: Plot of the probability density function of $k(x, \omega)$ as a function of the expected value for (A) polynomial chaos expansion, (B) Karhunen-Loeve expansion and (C) untruncated expansion at $x = 0$.

polynomial approximations are viable in estimating the log-normal random variable with finite number of expansion terms. The plot suggests that the aliasing and truncation errors lead to an underprediction of the local solution $u(x)$. The higher the polynomial degree, and the higher the number of stochastic dimensions, the smaller the difference in the methods, as evident from the convergence plot in Fig. 3.

The covariance field of the solution u can be estimated using the equation:

$$\text{Cov}(x_1, x_2) = \mathbb{E}\{(\vec{u}_{sol} - \mathbb{E}[\vec{u}_{sol}]) * (\vec{u}_{sol} - \mathbb{E}[\vec{u}_{sol}])^T\} \quad (27)$$

The derived covariance field for the three methods is presented in Fig. 8. Comparing across the three methods, it appears that the truncated methods (KL and PC expansions) lead to a less smooth covariance field. Despite that, the covariance of the field converges to 0 for $x = 1$ for all three methods. This is expected as the Dirichlet boundary condition at $x = 1$ causes the local solution value ($u(x = 1)$) to be a deterministic variable, leading to no covariances with the other

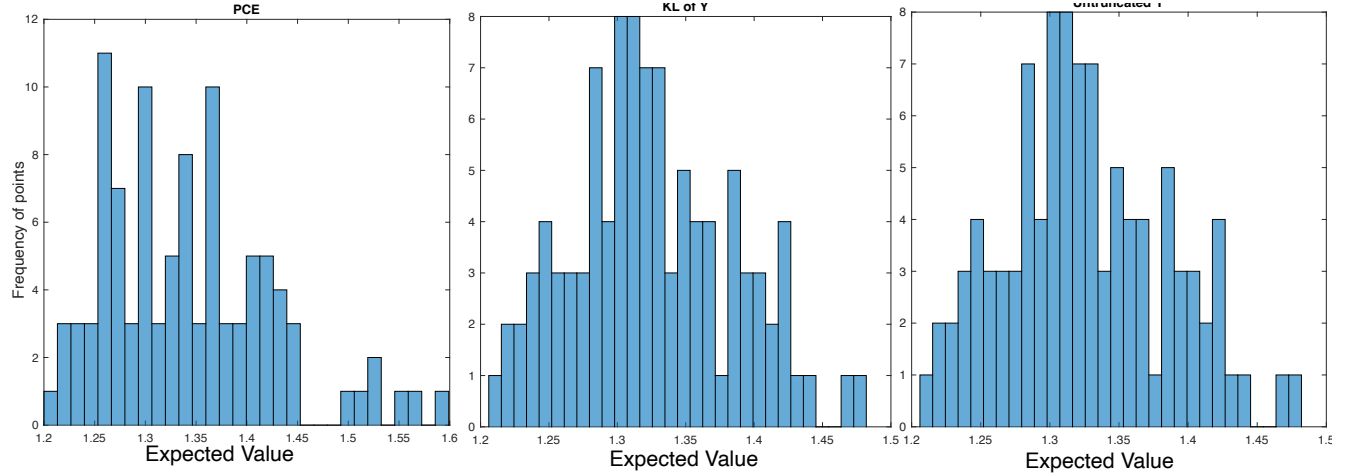


Figure 6: Plot of the probability density function of $k(x, \omega)$ as a function of the expected value for (A) polynomial chaos expansion, (B) Karhunen-Loeve expansion and (C) untruncated expansion at $x = 0.75$.

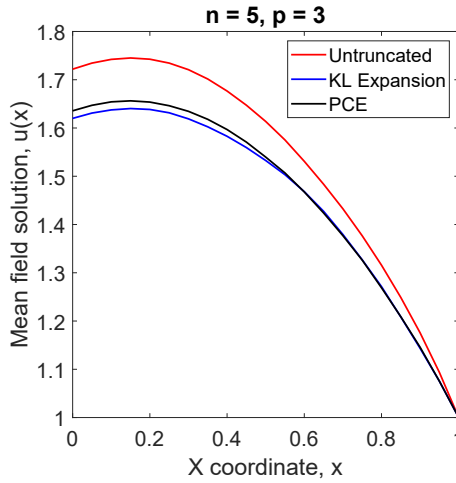


Figure 7: Plot of mean field as a function of x , for all three methods.

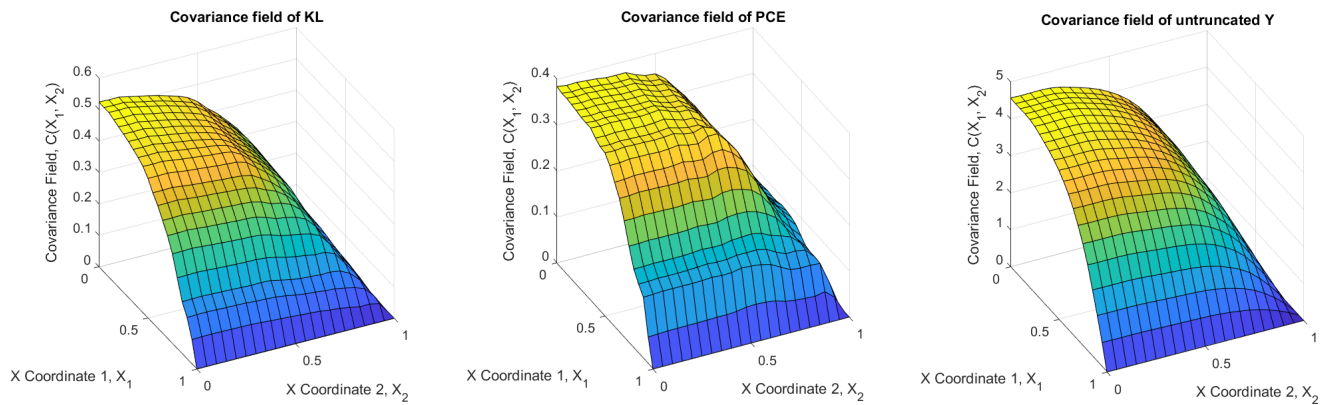


Figure 8: Plot of covariance field as a function of x , for all three methods.

parameters. The covariance between x_1 and x_2 increases as x_1 or x_2 tends towards 0, indicating a stronger correlation. This suggests that the randomness of the solution is consistently largest

at $x = 0$, and that any variation in its local value will have a large impact on the surrounding points. Between the three methods, the truncated fields (KL and PC expansions) have a weaker magnitude in their covariance fields as compared to the untruncated Y . This implies that the error in the stochastic approximation lowers the correlation between the different points in the domain, leading to more independent behavior. Therefore, the polynomial approximation is unable to fully replicate the inter-connectedness of the local solution fields.

Finally, a few realizations of the solution u for the three methods are provided in the following figure for illustration. Consistent with the earlier plots, the order of the scatter of the solution fields is consistent with the earlier plots.

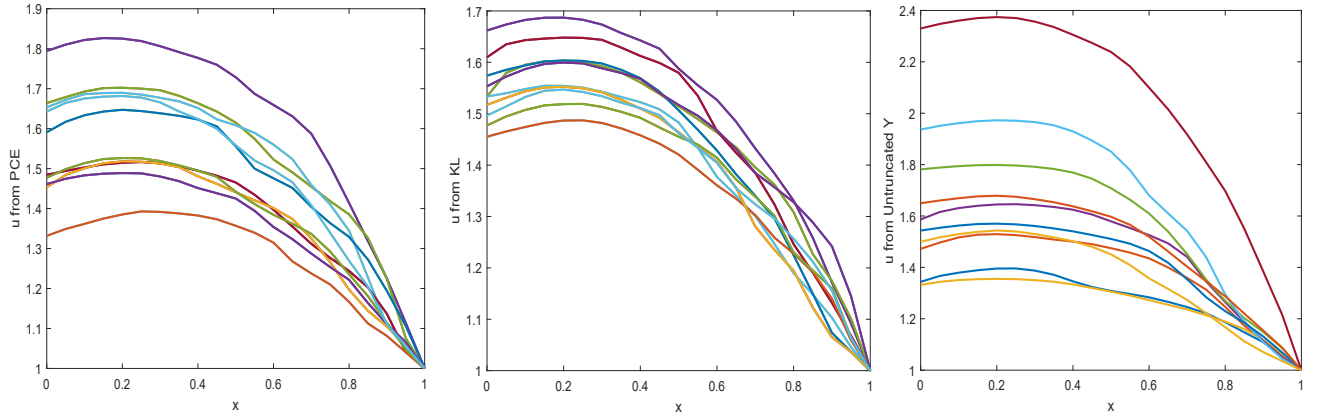


Figure 9: Several realizations of the solution field as a function of x for the three methods.

Subsection (c)

Here, we conduct a global sensitivity analysis for $u(x = 0.5)$, using Saltelli's method, for k and F . A total of $2 * 10^6$ Monte Carlo sample points were used to perform the sensitivity analysis. The first 10^6 samples are used to construct matrix \mathbf{A} , and the remaining 10^6 samples are used for matrix \mathbf{B} . Matrix \mathbf{C}_i is obtained by substituting the i -th row of the matrix \mathbf{B} with the i -th row of the matrix \mathbf{A} .

$$\mathbf{A} = \begin{bmatrix} F^1 & \vec{k}^1 \\ \vdots & \vdots \\ F^M & \vec{k}^M \end{bmatrix} \quad (28)$$

$$\mathbf{B} = \begin{bmatrix} F^{M+1} & \vec{k}^{M+1} \\ \vdots & \vdots \\ F^{2M} & \vec{k}^{2M} \end{bmatrix} \quad (29)$$

$$\mathbf{C}_1 = \begin{bmatrix} F^1 & \vec{k}^{M+1} \\ \vdots & \vdots \\ F^M & \vec{k}^{2M} \end{bmatrix} \quad (30)$$

$$\mathbf{C}_2 = \begin{bmatrix} F^{M+1} & \vec{k}^1 \\ \ddots & \\ \ddots & \\ F^{2M} & \vec{k}^M \end{bmatrix} \quad (31)$$

The matrices are then passed into the provided ODE solver to generate the model outputs, $y_A = \text{ODE-solver}(\mathbf{A})$, $y_B = \text{ODE-solver}(\mathbf{B})$ and $y_C = \text{ODE-solver}(\mathbf{C}_i)$. Then, f_0^2 is calculated as the mean of y_A . The first order and total order sensitivities are calculated using the following expressions:

$$S_i = \frac{\frac{1}{M} y_A^T y_{C_i} - f_0^2}{\frac{1}{M} y_A^T y_A - f_0^2} \quad (32)$$

$$S_{T_i} = 1 - \frac{\frac{1}{M} y_B^T y_{C_i} - f_0^2}{\frac{1}{M} y_A^T y_A - f_0^2} \quad (33)$$

From our simulations, the calculated S_F and S_k are 0.0873 and 0.8504, and the calculated S_{T_F} and S_{T_k} are 0.1919 and 0.9362, respectively. The main sensitivity index for k is significantly higher than that of F , approximately one order of magnitude greater to be precise. This implies that the model is generally much more sensitive to changes in the k values as compared to F . Further, the main sensitivities appear to dominate, accounting for approximately 90 % of the total sensitivity indices for k . This implies that changes in F impact the values of k to a small degree. On the contrary, the main sensitivity index of F is only approximately 50 % of the total sensitivity index, implying that changes in values of k has a sizeable effect on the results of F .

Part 2

Subsection 2.1.1

(a)

Here, we use a least-square polynomial approach to approximate the solution to $u(x = 0.5)$, using a total degree polynomial expansion of degree p . There are five inputs random variables, namely a normalized \bar{F} ($\bar{F} = \frac{F - \mu_F}{\sigma_F}$), and 4 stochastic random variable dimensions for the Karhunen-Loeve approximation of k . The least squares polynomial approximation is obtained by solving the optimization problem with Monte Carlo simulations:

$$u^* = \operatorname{argmin}_u \mathbb{E}_X \{ ||u_{ODE}(X) - u(X)||^2 \} \quad (34)$$

$$= \operatorname{argmin}_u \frac{1}{M} \sum_{i=1}^M |u_{ODE}(X^i) - u(X^i)|^2 \quad (35)$$

where X^i are sampled identically and independently from the standard Gaussian distribution.

To be precise, we are looking for a vector of coefficients, c_i such that the approximation can be done with the following linear combination of Hermite polynomials. Here, the expansion follows that of a total-degree polynomial approximation, similar to the PCE expansion in Part 1. The coefficients, however, are obtained from the minimization of an equivalent least square problem instead.

$$k(x, \omega) = \exp[Y(x, \omega)] \quad (36)$$

$$= \sum_{||\vec{\alpha}||_1 < p} C_{\vec{\alpha}} \operatorname{He}_{\vec{\alpha}}(x, Z_1, \dots, Z_5) \quad (37)$$

$$\vec{c}_s^* = \operatorname{argmin}_c ||\mathbf{V}\vec{c} - \vec{u}_{ODE}||^2 \quad (38)$$

where \mathbf{V} is the $M * n$ Vandermonde matrix, \vec{u}_{ODE} is the vector of outputs from the ODE solver, and \vec{c} is the coefficient vector. When \mathbf{V} has full column rank, the minimization can be re-expressed equivalently as:

$$\mathbf{G}\vec{c}_s^* = \mathbf{V}^T \vec{u}_{ODE} \quad (39)$$

where $\mathbf{G} = \mathbf{V}^T \mathbf{V}$. The linear system can be solved quickly with a linear system solver (e.g. Gauss-Seidel).

(b)

Here, we explore the convergence rate of the least square polynomial approximation to the actual results from the original ODE solver. To ensure valid comparison, as hinted in the question, we kept the number of stochastic dimensions constant at 5 for this entire section. The only variable model parameters are the total degree of the polynomial (p) and the number of Monte Carlo sample points (M).

In general, when the total number of polynomial expansion terms exceeds the number of Monte Carlo samples (in this case, $n_{total} > p$), the linear system of equation in Eq. 39 is under-constrained.

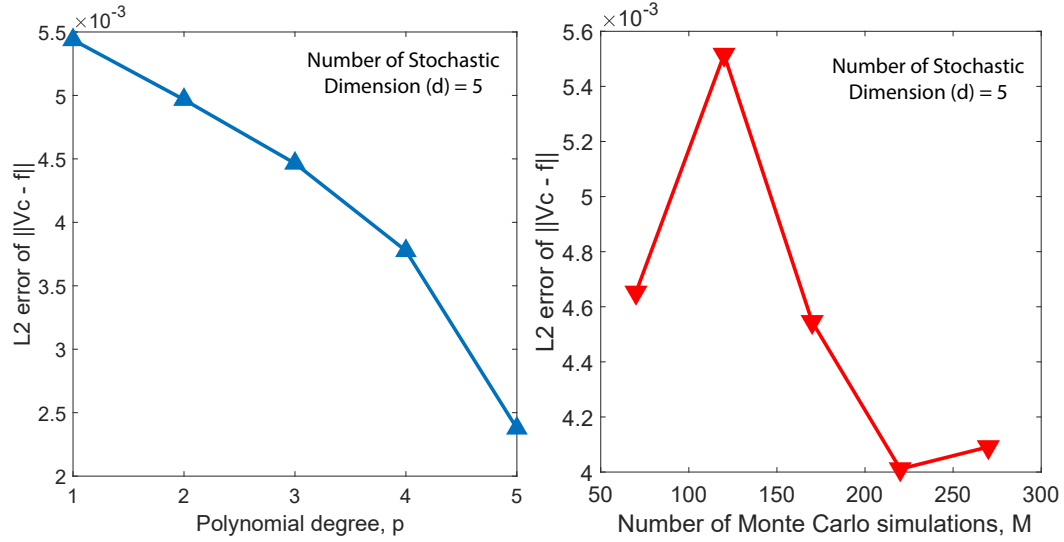


Figure 10: Plot of mean field as a function of x , for all three methods.

In that case, there will be infinitely number of solution vector \vec{c}_s that will yield a perfect regression to Eq. 39. Therefore, to avoid the triviality of such under-constrained systems, we investigate the relationship of the approximation error with p and M when the number of Monte Carlo samples exceed the number of polynomial expansion coefficients.

Here, we define the L_2 truncation error based on Eq. 35, and plot the error as a function of p and M in Fig. 10, for $u(x = 0.5)$. By inspection, we observe that the total error of the least square approximation reduces very quickly with total polynomial degree, for a constant number of Monte Carlo points. It attains a very low error in the order of 10^{-3} with 1 total degree polynomial expansion. The relationship between the least square error and the number of Monte Carlo points appears to be more nuanced. We see that the error rises with the number of Monte Carlo points to a maximum at approximately $M = 110$, before decreasing with increasing M . We believe that this makes mathematical sense. As M decreases toward the limit of $n_{terms} = M$, there will be more fitting parameters for a given set of constraints. From an optimization perspective, there will be a larger solution space to derive an optimal optimum, leading to a decrease in the residual error for $M < 110$. As M increases from 110, the variance of $u(x = 0.5)$ decreases with increasing Monte Carlo simulation (asymptotically unbiased). Therefore, the improved estimation of the expected value of $u(x = 0.5)$ likely leads to better approximation of the coefficient vector \vec{c}_s , causing the error to fall with increasing M past the maximum point. Overall, for the scheme to be stable, we want to pick a large number M such that the coefficient vector is consistent and the linear solver is not under-determined. For $d = 5, p = 3$, our simulations in Fig. 10 suggest that the optimal M lies after $M \simeq 100$.

Subsection 2.1.2

(a)

In this section, we use the least-square polynomial approximation presented in the previous subsection to approximate the solution $u(x)$ at selected points along the domain. Here, to allow for a good balance between the number of regression coefficients (\vec{c}_s) and the number of degrees of freedom (controlled by the number of Monte Carlo samples), we pick M sufficiently large such that it is always at least two times greater than the number of coefficients. This is based on our findings from the previous section, where the L2-error decays rapidly with M past $M \simeq 100$, for 50 polynomial expansion terms.

The probability density of the solution $u(x)$ at $x = 0$, $x = 0.5$ and $x = 0.75$ is presented in Fig. 11, using 100 Monte Carlo samples. Overall, the distributions appear to be slightly skewed, but the shape of the distribution is fairly similar across all three positions. Compared to the Monte Carlo results obtained in Part 1, the variance of the distribution from the least square approximation appears to be much smaller, and the span of the expected values is much narrower. From my intuition, the variance reduction observed for the least-square occurs due to the regression method in the least-squares approximation. The regression inherently uses a deterministic method to solve for the coefficient vectors, sidestepping the inherent stochasticity of the model. The mean field using the least-square polynomial approximation is depicted in Fig. 12. The mean

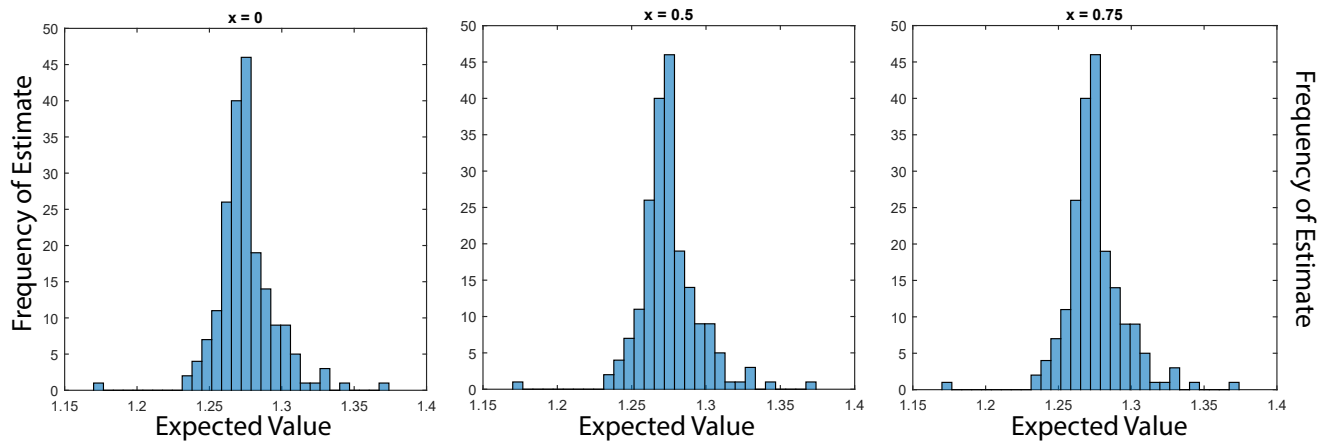


Figure 11: Plot of the probability density function of $u(x, \omega)$ from the least square polynomial approximation as a function of the expected value for (A) $x = 0$, (B) $x = 0.5$ and (C) $x = 0.75$.

field is obtained by stitching together the results from 50 local least-square polynomial approximations, each evaluated with 100 Monte Carlo samples. Compared to the Monte Carlo results in Part 1, we see that the general trend of u with x is identical; the mean field increases with x to a maximum at around $x \simeq 0.2$, before decreasing to the Dirichlet boundary condition at $x = 1$.

However, as evident from its non-smooth nature, the local estimations from the least-square approximation are not smooth with x . Although the plot is a collection of only 50 scatter points, the positions of the respective points at $0 < x < 0.5$ appear to be very disordered, unlike the previous results presented in Part 1. This occurs because the coefficient vectors of the different x positions are regressed independently of each other. Consequently, the independent nature of the

respective positions imply that the mean of the adjacent points does not affect the mean value of its neighbours. This causes a scatter of the mean value to occurs, leading to the observed figure.

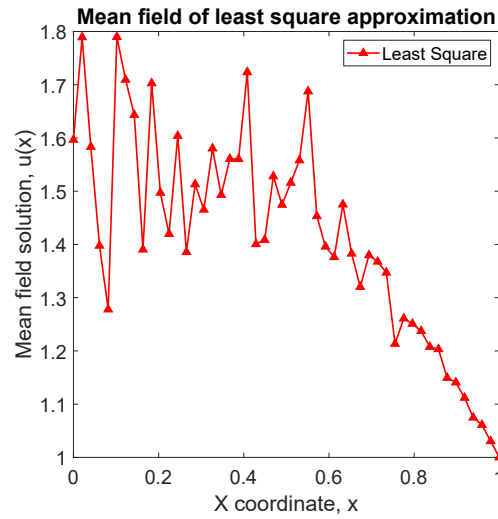


Figure 12: Plot of mean field as a function of x , for the least squares method.

Next, we use the Monte Carlo results and the derived mean field to calculate the covariance field from the least squares approximation. Similar to Part 1, we used Eq. 27 to determine the covariance field. The resultant covariance field is plotted in Fig. 13.

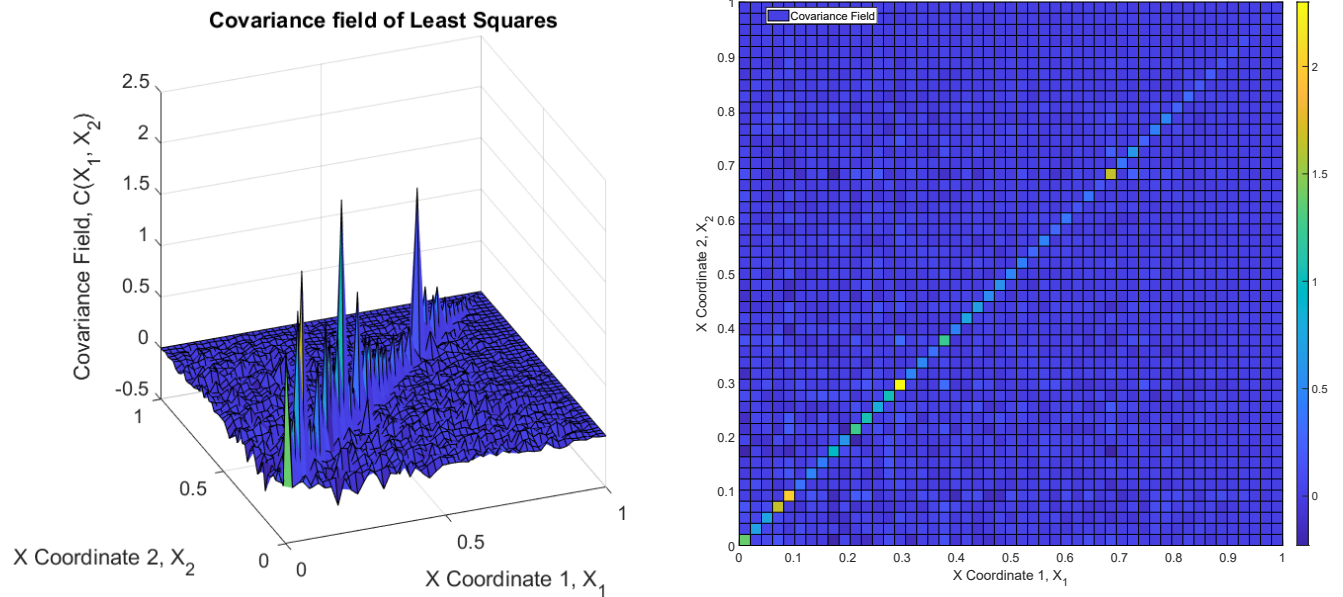


Figure 13: Covariance field based on the least-square polynomial approximation method.

The results presented in the covariance field look very different from similar plots in Part 1. Unlike the prior plots, the covariance field from the least-square approximation elucidated that the variance between different positions is close to 0, as evident by the blue spots in the heat map.

As discussed in the previous paragraphs, the least-square polynomial approximations at a given position are solved independently of other positions. Consequently, we expected that the results of u at different positions to be independent. This is corroborated by the covariance field, indicating that the correlation between different x_i 's is very small, while the covariance is dominated by the diagonal terms (the variance of x_i 's).

A few realizations of the solution field is provided in Fig. 14. Unlike the case in Part 1, the realizations here are very discontinuous, leading to the jagged appearance in the figure. However, they all cluster around the mean field value that is presented in the earlier figures.

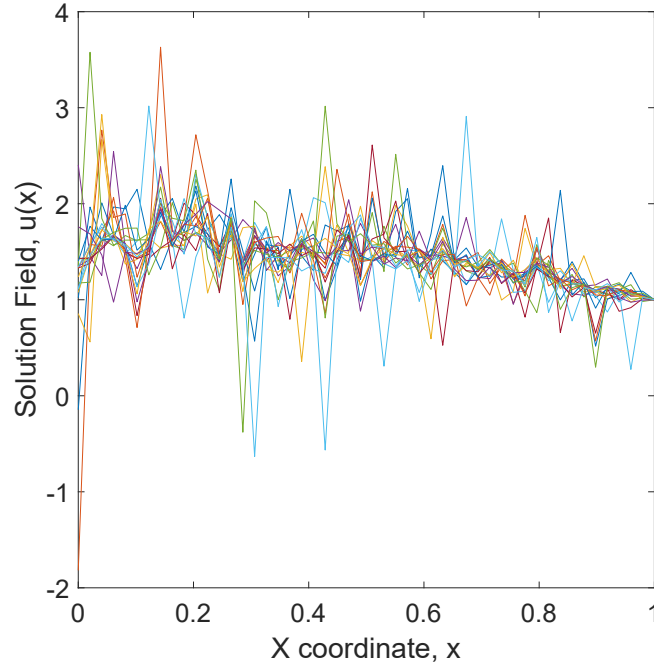


Figure 14: Realizations of the solution field for different Monte Carlo simulation points.

(b)

Here, we are interested in the eigenvalues of the direct Karhunen-Loeve expansion of the solution field $u(x, \omega)$. The finite truncated expansion can be written as:

$$u^m(x, \omega) = \mu_u(x) + \sum_{i=1}^m \sqrt{\lambda_i} \tilde{\psi}_i(x) Z_i(\omega) \quad (40)$$

where λ_i and $\tilde{\psi}_i(x)$ are the eigenvalues and eigenvectors evaluated at a particular position. The eigenvalues are evaluated at the quadrature points, and can be solved using the Nystrom method, as depicted in Eq. 41.

$$\sum_{k=1}^m \text{Cov}(u(x), u(x_k)) \tilde{\psi}_i(x_k) w_k = \tilde{\lambda}_i \tilde{\psi}_i(x) \quad (41)$$

Unlike the Karhunen-Loeve expansion of Y presented in Part 1, unfortunately, we do not have an explicit function for the covariance between two points. Therefore, we use the covariance field for

the untruncated Y presented in Fig. 8 (right) to estimate the covariance.

Numerically, we use *qrule.m* to generate a large number of quadrature points (in this case, we pick $m = 100$) within the interval $[0,1]$. Then, we solve for the covariance field, using the untruncated $Y(x, \omega)$ method that is presented in Part 1, to obtain the covariance matrix of the respective quadrature points. Then, we combine the covariance matrix with the algebra of the Nystrom method in Eq. 41 to solve for the 100 eigenvalues, using the *eig* function on MATLAB.

The normalized eigenvalues of $Y(x, \omega)$ and $u(x, \omega)$ are presented in Fig. 15. Our results indicate that both sets of eigenvalues decay very quickly with increasing number of dimensions, which is consistent with the theoretical underpinnings of the truncation of the Karhunen-Loeve expansion. Relative to the largest eigenvalue, it appears that the decay of the eigenvalues is more rapid for $u(x, \omega)$ than $Y(x, \omega)$. This indicates that the error of truncation decays much quicker for $u(x, \omega)$ than $Y(x, \omega)$. The exact mechanism for this difference is unclear to me by inspection, but mathematically, it arises from the difference in the covariance fields between $u(x, \omega)$ and $Y(x, \omega)$. The covariance between points x_1 and x_2 is smaller for $u(x, \omega)$ than the predicted values from Eq. 2. Consequently, based on my speculation, from a mathematical perspective, the expansion will require fewer terms to capture the smaller covariance of $u(x, \omega)$. This leads to the largest eigenvalue of $u(x, \omega)$ to be significantly larger than the smallest one, leading to a sharper decrease with increasing dimensions observed in the figure.

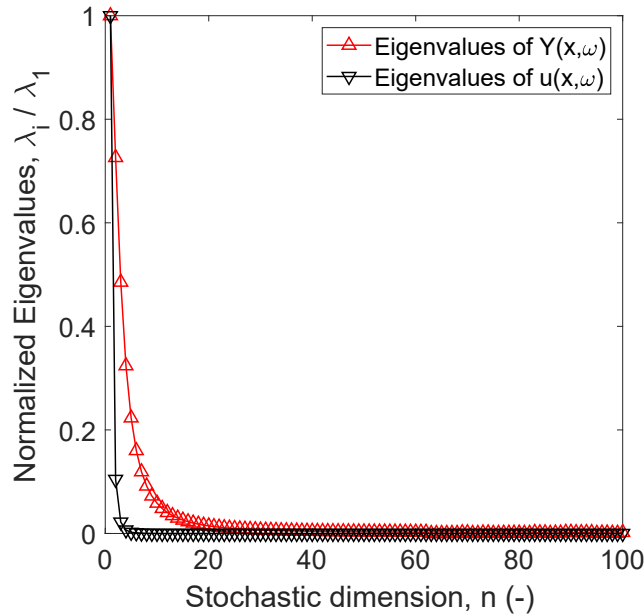


Figure 15: Comparison between the normalized eigenvalues of $Y(x, \omega)$ and $u(x, \omega)$.

(c)

Here, we consider the polynomial chaos expansion of $u(x = 0.5)$. We are interested to use the coefficients from the polynomial chaos expansion of $u(x = 0.5)$ to deduce the main and total effect sensitivity indices for F and k respectively.

$$u(x = 0.5, \omega) = \sum_{\|\vec{\alpha}\|_1 < p} C_{\vec{\alpha}} \text{He}_{\vec{\alpha}}(x, k_1, \dots, k_n, F) \quad (42)$$

Taking the variance of the PCE:

$$\text{Var} \{u(x = 0.5, \omega)\} = \sum_{\|\vec{\alpha}\|_1 < p} C_{\vec{\alpha}}^2 \|\text{He}_{\vec{\alpha}}(x, k_1, \dots, k_n, F)\|_{L_2}^2 \quad (43)$$

Given that $\text{Var}\{\text{He}\} = \mathbb{E}\{\text{He}^2\} = \|\text{He}_{\vec{\alpha}}(x, k_1, \dots, k_n, F)\|_{L_2}^2$, as the expected value for $\text{He}(x, \omega)$ is 0 (standard Gaussian).

We divide Eq. 43 by $\text{Var} \{u(x = 0.5, \omega)\}$, leading to the following expression:

$$1 = \sum_{\|\vec{\alpha}\|_1 < p} \frac{C_{\vec{\alpha}}^2 \|\text{He}_{\vec{\alpha}}(x, k_1, \dots, k_n, F)\|_{L_2}^2}{\text{Var} \{u(x = 0.5, \omega)\}} \quad (44)$$

Compared to the high dimensional model representation, we can associate the terms of Eq. 44 to the respective sensitivity indices. Here, we group the n dimensions of \vec{k} to be a block parameter, with F being the second parameter, to be consistent with our method in Part 1.

$$1 = \sum_i \frac{\text{Var}\{u_i(X_i)\}}{\text{Var}\{u\}} + \sum_i \sum_{j>i} \frac{\text{Var}\{u_{i,j}(X_i, X_j)\}}{\text{Var}\{u\}} \quad (45)$$

Here, there are only two terms. The expression can be simplified to become:

$$1 = \frac{\text{Var}\{u_1(\vec{k})\}}{\text{Var}\{u\}} + \frac{\text{Var}\{u_2(F)\}}{\text{Var}\{u\}} + \frac{\text{Var}\{u_{1,2}(\vec{k}, F)\}}{\text{Var}\{u\}} \quad (46)$$

$$= S_1 + S_2 + S_{1,2} \quad (47)$$

Comparing the PCE to the HDMR formulation, it is apparent that the sum of terms in the PCE that are univariate functions (of either \vec{k} or F) will be either S_1 or S_2 , the main sensitivity effects. Similarly, the sum of terms that are multivariate functions of \vec{k} and F will be the cross terms, represented by $S_{1,2}$. From this, the total sensitivity index of \vec{k} and F can be calculated with the equations:

$$S_{T_{\vec{k}}} = S_1 + S_{1,2} \quad (48)$$

$$S_{T_F} = S_2 + S_{1,2} \quad (49)$$

In the PCE, if $\alpha_F = 0$ in a particular polynomial, then it is not a function of F as the Hermite polynomial of F for $\alpha = 0$ is 1. S_1 , the main sensitivity effect coefficient for \vec{k} , is then the sum of all terms where $\alpha_F = 0$. Likewise, S_2 is then the sum of all terms where $\alpha_{\vec{k}} = 0$. $S_{1,2}$ will be the sum of all the remaining terms.

Like before, the $\|\text{He}_{\vec{\alpha}}(x, k_1, \dots, k_n, F)\|_{L_2}^2$ is calculated as the product of the elements of $\vec{\alpha}$:

$$\|\text{He}_{\vec{\alpha}}(x, k_1, \dots, k_n, F)\|_{L_2}^2 = \langle \text{He}_{\vec{\alpha}}(x), \text{He}_{\vec{\alpha}}(x) \rangle \quad (50)$$

$$= \prod_{i=1}^n \alpha_i! \quad (51)$$

We also adopted the same formula to compute the coefficients C_α as that presented in Part 1. We estimate the variance of u at $x = 0.5$ with vanilla Monte Carlo, using 10,000 independent sample points with the untruncated $Y(x, \omega)$. Lastly, the main and total effect sensitivities for \vec{k} and F are calculated with Eq. 48 and 49.

Our simulations indicate that:

$$S_F = 0.223 \quad (52)$$

$$S_{\vec{k}} = 0.6192 \quad (53)$$

$$S_{F, \vec{k}} = 0.1578 \quad (54)$$

$$S_{T_F} = 0.3808 \quad (55)$$

$$S_{T_{\vec{k}}} = 0.7770 \quad (56)$$

Our results here suggest that the sensitivity of the results $u(x = 0.5)$ is stronger towards \vec{k} than F , which aligns with our predictions made earlier with Saltelli's method. However, the influence from F is a lot stronger with the HDMR formulation than the Saltelli's method. Assuming that Saltelli's method is more accurate, given that no truncation is required, the PCE formulation overestimates the influence of F on the $u(x = 0.5)$.