Multi–Element Stochastic Spectral Projection for High Quantile Estimation

Jordan Ko & Josselin Garnier

Laboratoire de Probabilités et Modèles Aléatoires, Université Paris VII
75205 Paris Cedex 13, France

Abstract

We investigate quantile estimation by multi–element generalized Polynomial Chaos (gPC) metamodel where the exact numerical model is approximated by complementary metamodels in overlapping domains that mimic the model's exact response. The gPC metamodel is constructed by the non–intrusive stochastic spectral projection approach and function evaluation on the gPC metamodel can be considered as essentially free. Thus, large number of Monte Carlo samples from the metamodel can be used to estimate α–quantile, for moderate values of α. As the gPC metamodel is an expansion about the means of the inputs, its accuracy may worsen away from these mean values where the extreme events may occur. By increasing the approximation accuracy of the metamodel, we may eventually improve accuracy of quantile estimation but it is very expensive. A multi–element approach is therefore proposed by combining a global metamodel in the standard normal space with supplementary local metamodels constructed in a bounded domain about the design points corresponding to the extreme events. To improve the accuracy and to minimize the sampling cost, sparse–tensor and anisotropic–tensor quadrature are tested in addition to the full–tensor Gauss quadrature in the construction of local metamodels; different bounds of the gPC expansion are also examined. The global and local metamodels are combined in the multi–element gPC (MEgPC) approach and it is shown that MEgPC can be more accurate than Monte Carlo or importance sampling methods for high quantile estimations for input dimensions roughly below $N = 8$, a limit that is very much case– and α–dependent.

Key words: quantile estimation, generalized Polynomial Chaos, metamodels, multi–element refinement, multiple design points, numerical quadrature, Sparse–tensor quadrature, anisotropic quadrature

1. Introduction

The estimation of high quantile is a challenging numerical topic and has received a considerable amount of attention in many research disciplines [1]. Let us consider a numerical model of a complex system $y = f(\boldsymbol{x})$ where $\boldsymbol{x}$ is a N–tuple vector and $f : \mathbb{R}^N \to \mathbb{R}$ is a deterministic numerical model. The numerical model is expensive in terms of the computation time required for a single run. We are interested in the case where its input is a random vector $\boldsymbol{X} = (X_1, ..., X_N)$; in this context, the output $Y = f(\boldsymbol{X})$ is a random scalar. The α–quantile of Y is the level y_α such that the probability that Y takes a value lower than y_α is α:

$$y_\alpha = \inf\{y; F(y) \geq \alpha\}, \tag{1}$$

where $F(y)$ is the cumulative distribution function (cdf) of Y, *i.e.* $F(y) = \mathbb{P}(Y \leq y)$. To accurately estimate high quantiles with the Monte Carlo (MC) method, a large number of samples is required. When the numerical models are computationally expensive to evaluate, the large MC samples required to accurately estimate high quantiles may render this approach impractical.

Different approaches have been developed to improve the accuracy of α–quantile estimation when the number of evaluations on the complete model is limited. As we assume no prior knowledge of the function in

question, the estimation of high quantiles usually requires the following steps. First, y_α is roughly estimated. Second, design points, which are the most likely inputs $\boldsymbol{\xi}_\alpha$ satisfying $f(\boldsymbol{\xi}_\alpha) = y_\alpha$, are sought. Third, sampling refinements are performed near the design points. High computational cost arises especially in the first step due to the accuracy required in the preliminary quantile estimation and in the second step due to the multiple function evaluations to estimate values, gradients, and Hessians of the complete model. Importance sampling (IS) is a refined Monte Carlo sampling strategy that can be used in the third step to concentrate MC inputs near $\boldsymbol{\xi}_\alpha$; we will compare this strategy with our metamodel-based approach. Note also that, within the context of structural reliability, First-Order Reliability Method (FORM) and Second-Order Reliability Method (SORM) have been developed to estimate quantiles but their errors are difficult to estimate [2, 3, 4].

In the current study, we investigate quantile estimation by multi–element generalized Polynomial Chaos (gPC) metamodels. The gPC has recently been applied to uncertainty quantification studies [5, 6, 7]. By approximating the complete model with an accurate spectral metamodel, different statistical measures such as mean, variance and Sobol' sensitivity indices can be readily computed [7]. Large number of MC samples can be rapidly evaluated on the metamodel to estimate the quantile of the model solution. In addition, the metamodel can be used in the design point search algorithm to estimate the values of the inputs corresponding to the solution quantiles and the metamodel approach is valid even for complex nonlinear complete models with multiple design points. In comparison to the multiple design point algorithm proposed in [8], the current approach based on gPC metamodels do not require numerical modifications to $f(\boldsymbol{x})$ in order to locate all design points.

As the gPC metamodels are spectral expansions about the means of the inputs, their accuracy may worsen away from these mean values where the extreme events may occur. By increasing the global approximation level, we may eventually increase the accuracy of the global metamodel away from the mean but it can be very expensive. Thus, a multi–element approach is used by combining the global metamodel with supplementary local metamodels centered at the design points. Similar multi–element approaches have been used previously although the local refinement goal was to reduce the contribution to the global variance from each local element [9, 10, 11, 12].

The current approach is similar to the parametric study of the random inputs in [13, 14] with the exception that the complete models are replaced by their metamodels. Indeed metamodels have been used previously to improve the cost of quantile estimation. Bucher & Bourgund have used metamodels constructed from polynomial expansions without cross terms [15] and polynomial chaos metamodels were used in [16]. Despite the availability of the metamodel, these studies used conventional gradient descent algorithms to determine the design points, which increases the sampling costs significantly. Moreover, polynomials with infinite supports are used to construct localized refinements around the design points [16]; the accuracy of the expansion can be improved if appropriate polynomials with finite supports are used.

The methodology used in the current study is outlined in Fig. 1. Quantile estimation methods are described in Sec. 2. The construction of the gPC metamodel is detailed in Sec. 3. Once the preliminary estimation of the α–quantile of Y is obtained from the global gPC metamodel, the Lagrange multiplier method is used to search for the design points and the details applied to the metamodels of this approach are outlined in Sec. 4. Sec. 5 illustrates how the local gPC metamodels in the multi–element approach are combined with the global metamodel to estimate the α–quantile. Finally, the proposed methods are applied to some examples in Sec. 6.

2. Quantile Estimation

The current state of art for quantile estimation in the context of failure probability is documented in details in [17]. Key methods and their application to quantile estimation are presented in the following section.

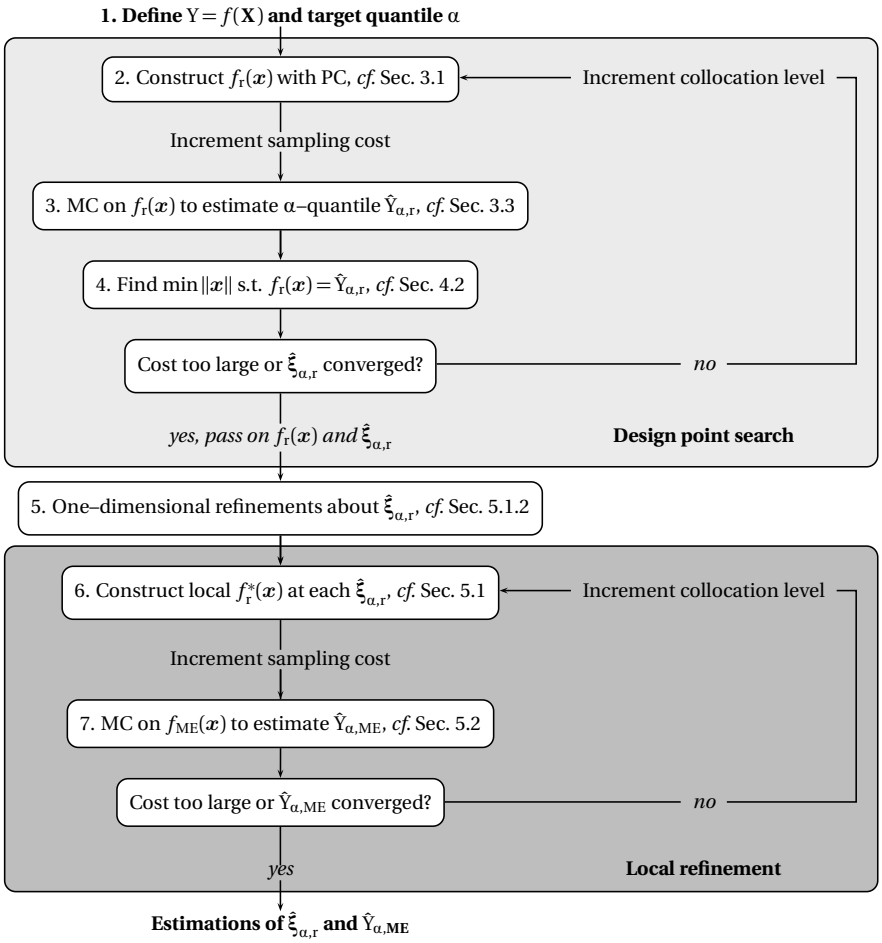

Figure 1: The methodology used to estimation the α–quantile of the solution output, Y.

2.1. Standard Monte Carlo method

The cdf $F(y)$ of Y is the function $y \to F(y) = \mathbb{P}(f(\boldsymbol{X}) \in (-\infty, y])$ whose integral form is

$$F(y) = \mathbb{E}[\mathbf{1}_{(-\infty,y]}(f(\boldsymbol{X}))] = \int_{\mathbb{R}^N} \mathbf{1}_{(-\infty,y]}(f(\boldsymbol{x}))\rho(\boldsymbol{x})d\boldsymbol{x}, \tag{2}$$

where $\rho(\boldsymbol{x})$ is the probability density function (pdf) of $\boldsymbol{X}$. Monte Carlo (MC) method and its variants are the traditional approaches to approximate this integral [18]. The cdf can be estimated from Z MC samples as

$$\hat{F}(y) \;=\; \frac{1}{Z}\sum_{i=1}^{Z} \mathbf{1}_{(-\infty,y]}(Y_i), \tag{3}$$

where $Y_i = f(\boldsymbol{\zeta}_i)$ and $\boldsymbol{\zeta}_i$ are independently generated according to the input pdf $\rho(\boldsymbol{x})$ and $\hat{}$ denotes that the solution is an empirical estimate. The empirical α–quantile can be estimated from $\hat{Y}_\alpha = \inf\{y; \hat{F}(y) \geq \alpha\}$ which gives

$$\hat{Y}_\alpha = Y_{(\lceil \alpha Z \rceil)}, \tag{4}$$

where $\{Y_{(i)}\}_{i=1}^{Z}$ is the ordered set of the Z MC samples $\{Y_i\}_{i=1}^{Z}$ such that $Y_{(1)} \leq Y_{(2)} \leq ... \leq Y_{(Z)}$. The empirical quantile $\hat{Y}_\alpha$ converges to y_α as $Z \to \infty$ and it is asymptotically normal provided F is continuously differentiable at y_α and $F'(y_\alpha) > 0$ [19]. When Z is fixed a non-asymptotic confidence interval for the α–quantile y_α can be estimated by the Wilks estimation technique [20]. The probability of y_α falling within the $[\alpha + \epsilon_1, \alpha + \epsilon_2]$–empirical quantile of $\{Y_i\}_{i=1}^{Z}$ is larger than a prescribed level γ, *i.e.*

$$\mathbb{P}(y_\alpha \in [Y_{(\lceil Z(\alpha+\epsilon_1)\rceil)}, Y_{(\lceil Z(\alpha+\epsilon_2)\rceil)}]) \;\geq\; \gamma, \tag{5a}$$

provided that we take

$$\epsilon_1 = \max\{\epsilon : C_\alpha(Z,\epsilon) \leq \frac{1-\gamma}{2}\} \quad \text{and} \quad \epsilon_2 = \min\{\epsilon : C_\alpha(Z,\epsilon) \geq 1 - \frac{1-\gamma}{2}\}, \tag{5b}$$

where

$$C_\alpha(Z,\epsilon) = \sum_{j=0}^{\lfloor Z(1-\alpha-\epsilon)\rfloor} \binom{Z}{j} (1-\alpha)^j \alpha^{Z-j}, \tag{5c}$$

for $\epsilon_1 < 0 < \epsilon_2$. For high quantiles, *i.e.* $\alpha \sim 1$, Z needs to be large to reduce the length of the confidence interval [21, 19]; therefore, the MC approach may not be practical in cases where the function $f(\boldsymbol{x})$ is computationally expensive to evaluate.

2.2. Monte Carlo with Very Large Samples

The size of the MC sample required to obtain high quantiles with reasonable confidence interval can be large and the size of the solution vector $(Y_j)_{j=1}^{Z}$ may exceed the memory limit of the computer. We propose a method that circumvents the numerical difficulty in sorting and storing a very large vector. Let Z^* be the size of the largest allowable vector. If $\boldsymbol{Y} = (Y_j)_{j=1}^{Z^*}$ are Z^* MC samples, only the subset $\tilde{\boldsymbol{Y}} = \{\boldsymbol{Y} : Y_j \geq \hat{Y}_{\alpha^-}\}$ contributes to the computation of $\hat{Y}_\alpha$, where $\hat{Y}_{\alpha^-}$ is chosen such that $\hat{Y}_{\alpha^-} < y_\alpha$. An appropriate and conservative choice of $\hat{Y}_{\alpha^-}$ is the quantile at a lower α, *e.g.* $\hat{Y}_{\alpha^-} = \inf\{y; \hat{F}(y) \geq \alpha^-\}$ where $\alpha^- < \alpha$.

We repeat MC simulations n times where $n = \max\{n : \sum_{i=1}^{n} |\tilde{\boldsymbol{Y}}^i| < Z^*\}$. Here $\boldsymbol{Y}^i$ is the i–th subset of size Z^*, $\tilde{\boldsymbol{Y}}^i$ is defined as above and $|\tilde{\boldsymbol{Y}}^i|$ is the size of $\tilde{\boldsymbol{Y}}^i$. The cumulative subset from the n sets of simulations, $\tilde{\boldsymbol{Y}} = \cup_{i=1}^{n} \tilde{\boldsymbol{Y}}^i$, allows us to compute the quantile from nZ^* MC outputs without having to store and sort the entire set. In addition, $\hat{Y}_{\alpha^-}$ is determined from the first set, *i.e.*

$$\hat{Y}_{\alpha^-} = Y^1_{(\lceil \alpha^- Z^* \rceil)},$$

where $Y^1_{(\lceil . \rceil)}$ is the first ordered subset of the MC results. We need to make sure that α^- is sufficiently close to α to minimize $|\tilde{\boldsymbol{Y}}^i|$ but sufficiently smaller than α to ensure that $\hat{Y}_{\alpha^-} < y_\alpha$.

Accordingly, the empirical estimate of $\hat{F}$ in Eq. (3) for $y > y_{\alpha^-}$ is rewritten as

$$\hat{F}(y) \;=\; 1 - \frac{1}{nZ^*} \sum_{k=1}^{|\tilde{\boldsymbol{Y}}|} \mathbf{1}_{(y,+\infty)}(\tilde{Y}_k), \tag{6}$$

where $\tilde{\boldsymbol{Y}} = (\tilde{Y}_k)_{k=1}^{|\tilde{\boldsymbol{Y}}|}$. The empirical estimate of the α-quantile y_α is thus

$$\hat{Y}_\alpha = \tilde{Y}_{(\lceil |\tilde{\boldsymbol{Y}}| - nZ^*(1-\alpha) \rceil)}, \tag{7}$$

where $(\tilde{Y}_{(k)})_{k=1}^{|\tilde{\boldsymbol{Y}}|}$ is the ordered set of the samples $\tilde{\boldsymbol{Y}}$.

2.3. Importance Sampling

The convergence of the MC method can be improved by reducing the variance in the MC approximation error. Importance sampling is a variance reduction technique that is particularly convenient for quantile

estimation. The idea is that, instead of estimating integral (2) with a sample taken with the pdf $\rho(\boldsymbol{x})$, importance sampling uses an importance pdf, $\bar{\rho}(\boldsymbol{x})$, that favors a certain region of interest of the input parameters (which is the one that contributes to integral (2) in our case). The method is based on the representation of the cdf $F(y)$ as an expectation with respect to a random variable $\bar{\boldsymbol{X}}$ with the pdf $\bar{\rho}(\boldsymbol{x})$:

$$F(y) = \mathbb{E}[\mathbf{1}_{(-\infty,y]}(f(\boldsymbol{X}))] = \int \mathbf{1}_{(-\infty,y]}(f(\boldsymbol{x}))\frac{\rho(\boldsymbol{x})}{\bar{\rho}(\boldsymbol{x})}\bar{\rho}(\boldsymbol{x})d\boldsymbol{x} = \bar{\mathbb{E}}\Big[\mathbf{1}_{(-\infty,y]}(f(\bar{\boldsymbol{X}}))\frac{\rho(\bar{\boldsymbol{X}})}{\bar{\rho}(\bar{\boldsymbol{X}})}\Big], \tag{8}$$

where $\bar{\mathbb{E}}$ is the expectation with respect to the distribution of $\bar{\boldsymbol{X}}$ with the pdf $\bar{\rho}(\boldsymbol{x})$. A natural estimator of the cdf $F(y)$ is in this context

$$\hat{F}(y) = \frac{1}{Z}\sum_{i=1}^{Z}\mathbf{1}_{(-\infty,y]}(f(\bar{\boldsymbol{\zeta}}_i))\frac{\rho(\bar{\boldsymbol{\zeta}}_i)}{\bar{\rho}(\bar{\boldsymbol{\zeta}}_i)}, \tag{9}$$

where $\bar{\boldsymbol{\zeta}}_i$ are generated according to the pdf $\bar{\rho}(\boldsymbol{x})$. In fact, as shown in [22] it is better (in the point of view of the asymptotic variance of the quantile estimator) to introduce an alternative estimator of the cdf of Y:

$$\hat{F}(y) = 1 - \frac{1}{Z}\sum_{i=1}^{Z}\mathbf{1}_{(y,+\infty)}(f(\bar{\boldsymbol{\zeta}}_i))\frac{\rho(\bar{\boldsymbol{\zeta}}_i)}{\bar{\rho}(\bar{\boldsymbol{\zeta}}_i)}, \tag{10}$$

which follows from the representation

$$F(y) = 1 - \mathbb{E}[\mathbf{1}_{(y,\infty)}(f(\boldsymbol{X}))] = 1 - \bar{\mathbb{E}}\Big[\mathbf{1}_{(y,\infty)}(f(\bar{\boldsymbol{X}}))\frac{\rho(\bar{\boldsymbol{X}})}{\bar{\rho}(\bar{\boldsymbol{X}})}\Big],$$

and to estimate the α-quantile by the standard formula $\hat{Y}_\alpha = \inf\{y; \hat{F}(y) \geq \alpha\}$. The estimator $\hat{Y}_\alpha$ converges to y_α as $Z \to \infty$ and it is asymptotically normal whatever the importance pdf $\bar{\rho}$ (provided the support of ρ is included in the support of $\bar{\rho}$ and $\mathbb{E}[\rho^2(\boldsymbol{X})/\bar{\rho}^2(\boldsymbol{X})] < \infty$). However the convergence rate and the asymptotic variance of the estimator strongly depend on the choice of the importance pdf and it is possible to reduce the variance dramatically by a proper choice (see [23, Section V.1] for a general introduction to importance sampling and [22] for a specific discussion for quantile estimation). For the case in which $\boldsymbol{X}$ is a standard Gaussian vector, a rather efficient choice is to take $\bar{\rho}(\boldsymbol{x}) = \rho(\boldsymbol{x}+\boldsymbol{\xi}_\alpha)$ where $\boldsymbol{\xi}_\alpha$ is the design point associated with the α–quantile. Of course the design point is not known so one has to to take $\bar{\rho}(\boldsymbol{x}) = \rho(\boldsymbol{x}+\hat{\boldsymbol{\xi}}_\alpha)$ where $\hat{\boldsymbol{\xi}}_\alpha$ is an estimator of the design point (such estimators are introduced in Section 4). If there are several design points, then a mixture of shifted Gaussian pdfs can be used.

In the context of the Monte Carlo sampling with a very large number of samples outlined in the previous section, importance sampling can also be used as follows. The first set $\boldsymbol{Y}^0$ of size Z^* is sampled by a standard Monte Carlo technique. The estimator $\hat{Y}_{\alpha^-}$ of the α^-–quantile is defined as in (6): $\hat{Y}_{\alpha^-} = Y^0_{(\lceil \alpha^- Z^* \rceil)}$. Additionally one computes an estimator $\hat{Y}_\alpha$ of the α–quantile and an estimator $\hat{\boldsymbol{\xi}}_\alpha$ of the associated design point. Next we carry out n sets of size Z^* of importance sampling simulations $\boldsymbol{Y}^1, \ldots, \boldsymbol{Y}^n$ with the pdf $\bar{\rho}(\boldsymbol{x}) = \rho(\boldsymbol{x}+\hat{\boldsymbol{\xi}}_\alpha)$. Here $n = \max\{n : \sum_{i=1}^n |\tilde{\boldsymbol{Y}}^i| < Z^*\}$ and $\tilde{\boldsymbol{Y}}^i = \{Y_j \in \boldsymbol{Y}^i : Y_j \geq \hat{Y}_{\alpha^-}\}$. We introduce the set $\tilde{\boldsymbol{Y}} = \cup_{i=1}^n \tilde{\boldsymbol{Y}}^i$ and we denote by $\tilde{Y}_k$ its elements $\tilde{\boldsymbol{Y}} = \{\tilde{Y}_k, k = 1...|\tilde{\boldsymbol{Y}}|\}$. Here we need also to record the corresponding weights $\tilde{w}_k = \rho(\tilde{\boldsymbol{\zeta}}_k)/\bar{\rho}(\tilde{\boldsymbol{\zeta}}_k)$ for $k = 1...|\tilde{\boldsymbol{Y}}|$. The cdf estimator is

$$\hat{F}(y) \;=\; 1 - \frac{1}{nZ^*}\sum_{k=1}^{|\tilde{\boldsymbol{Y}}|}\mathbf{1}_{(y,+\infty)}(\tilde{Y}_k)\tilde{w}_k.$$

The final α–quantile estimator is again $\hat{Y}_\alpha = \inf\{y; \hat{F}(y) \geq \alpha\}$.

3. Generalized Polynomial Chaos

The numerical model from which quantiles are evaluated can often be complex and computationally expensive. The computation cost can be significantly reduced by the use of metamodels. Metamodels can be either a reduced numerical code of the complete model or a response surface calibrated to mimic the complete model for typical realizations of the input parameters. The latter is the focus of this Section. In the context of quantile estimation, supplementary metamodels are constructed around exceptional but most important realizations of the input parameters which determine the values of the quantiles. The metamodel is constructed with the stochastic spectral projection which involves multi-dimensional numerical integration of the complete model solution over the input domain. Its construction is outlined before post–processing methods are described.

3.1. Stochastic Spectral Projection

We assume that the components of the input $\boldsymbol{X}$ are independent; this allows us to build our approximation functional spaces with tensor-product rules and write the joint pdf as a product of the marginal pdfs, *i.e.* $\rho(\boldsymbol{x}) = \prod_{j=1}^{N} \rho_j(x_j)$. We can then consider the following multivariate orthonormal polynomials that are built as tensor products of univariate orthonormal polynomials along each random dimension with respect to the probability measure $\rho(\boldsymbol{x})d\boldsymbol{x}$: $\phi_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \phi_{\alpha_1}(x_1) \otimes \ldots \otimes \phi_{\alpha_N}(x_N)$. Here, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N) \in \mathbb{N}^N$ represents a set of multi-indices and we define the length of $\boldsymbol{\alpha}$ with the following norm: $|\boldsymbol{\alpha}| \equiv \alpha_1 + \cdots + \alpha_N$. We therefore have $\mathbb{E}[\phi_{\boldsymbol{\alpha}}(\boldsymbol{X})\, \phi_{\boldsymbol{\beta}}(\boldsymbol{X})] = \mathbb{E}[\phi^2_{\boldsymbol{\alpha}}(\boldsymbol{X})]\delta_{\boldsymbol{\alpha}\boldsymbol{\beta}}$ where δ is the Kronecker symbol.

The goal is to construct a metamodel of $f(\boldsymbol{x})$ as a polynomial expansion, *i.e.* $f_{\mathrm{r}}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}^N} f_{\boldsymbol{\alpha}}\phi_{\boldsymbol{\alpha}}(\boldsymbol{x})$ where the subscript "r" denotes that the expansion is a reduced model. The deterministic coefficients $f_{\boldsymbol{\alpha}}$ are computed from:

$$f_{\boldsymbol{\alpha}} = \frac{\mathbb{E}[f(\boldsymbol{X})\phi_{\boldsymbol{\alpha}}(\boldsymbol{X})]}{\mathbb{E}[\phi^2_{\boldsymbol{\alpha}}(\boldsymbol{X})]} = \mathbb{E}[f(\boldsymbol{X})\phi_{\boldsymbol{\alpha}}(\boldsymbol{X})], \quad \text{for } \boldsymbol{\alpha} \in \mathbb{N}^N. \tag{11}$$

The expectation used above is an inner product in the random space where

$$\mathbb{E}[f(\boldsymbol{X})\phi_{\boldsymbol{\alpha}}(\boldsymbol{X})] = \int_{\mathbb{R}^N} f(\boldsymbol{x})\phi_{\boldsymbol{\alpha}}(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}. \tag{12}$$

When the method used to approximate the polynomial expansion has a theoretical polynomial accuracy of P, the conventional stochastic spectral projection includes all polynomials whose length of the multi-indices is smaller or equal to P, *i.e.* $\{\phi_{\boldsymbol{\alpha}} \,/\, |\boldsymbol{\alpha}| \leq P\}$. We call such an expansion the *canonical P expansion*. If the computation of $f_{\boldsymbol{\alpha}}$ has a polynomial accuracy of $\boldsymbol{P}$, *i.e.* P_n accurate in each random dimension, we can extend the expansion to the $\phi_{\boldsymbol{\alpha}}$'s whose degree in *each* dimension is not greater than P_n, *i.e.* $\{\phi_{\boldsymbol{\alpha}} \,/\, \boldsymbol{\alpha} \leq \boldsymbol{P}\}$. We call such an expansion the *maximum P expansion*. This is appropriate when we want to include *all* gPC terms satisfying an isotropic P–accuracy or when the computation has an anisotropic $\boldsymbol{P}$–accuracy. The number of additional terms can be significant for large N and P.

In practice, the order P is chosen based on accuracy and cost requirements. The multi-index $\boldsymbol{\alpha}$ can be renumbered with a unique index and the cardinality of the corresponding set is given by $M = \binom{N+P}{P}$ in the canonical approach. The gPC representation with $(M+1)$ terms reads:

$$f_{\mathrm{r}}(\boldsymbol{x}) = \sum_{m=0}^{M} f_m \phi_m(\boldsymbol{x}). \tag{13}$$

When the maximum P expansion is used, m is more generally taken to be an index pointing to all the polynomials where $\boldsymbol{\alpha}\leq\boldsymbol{P}$. For a given random support, some classical polynomial's weight functions are similar to the pdfs of some random variables. This close correspondence allows an optimal polynomial basis to be chosen to represent the selected random variables. The original Wiener polynomial chaos used the Hermite polynomials as the expansion basis for the Gaussian random variables [24]. Such an expansion forms a complete basis in Hilbert space determined by their corresponding support and converges to any L^2

function in the L^2 sense when the number of terms approaches infinity according to the Cameron–Martin theorem [25]. The expansion was generalized by [26] to represent non–Gaussian random fields by the use of polynomials from the Wiener–Askey scheme [27]. For example, the Legendre polynomials' weighting function is similar to the *pdf* of the uniform distribution.

In this study, we adopt the non–intrusive approach where we perform a Galerkin projection of the stochastic solution directly onto each member of the orthogonal basis. It has the advantage not needing modifications to the existing deterministic solver and the computation of Eq. (11) is achieved through numerical quadrature. The construction of the quadrature scheme is outlined in the following section.

3.2. Numerical Quadratures

The multidimensional integral (12) is approximated as the weighted sum of function evaluations at deterministic points in the support. The numerical solver is not modified and acts as a black box from which the solutions are repeatedly sampled. The inner product in the multivariate integral (12) is approximated as the following weighted sum

$$\int_{\mathbb{R}^N} f(\boldsymbol{x})\phi_m(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x} \approx \sum_{k=1}^{Z_Q} w_k f(\boldsymbol{z}_k)\phi_m(\boldsymbol{z}_k), \tag{14}$$

where $f(\boldsymbol{x})$ is evaluated at Z_Q quadrature points, $\boldsymbol{z}_k$, each with an associated weight, w_k. The accuracy of the quadrature approximation depends on the design of the quadrature points and weights. Three types of quadrature are considered in this paper: the Gauss full–tensor quadrature, the Smolyak sparse–tensor quadrature, and the anisotropic–tensor quadrature.

The N–dimensional full quadrature is constructed from full–tensor products of one–dimensional quadratures, $\mathcal{Q}_L^1$, as

$$\mathcal{Q}_L^N[f] = \left(\mathcal{Q}_L^1 \times \cdots \times \mathcal{Q}_L^1\right) f, \tag{15}$$

where L is the quadrature level or the number of quadrature points in each dimension. The multivariate weights are the products of the corresponding one-dimensional weights and they are derived in detail in [28]. The Gauss–Hermite and Gauss–Legendre quadratures are used for the global and the local metamodels respectively. The number of quadrature points in full quadratures scales as L^N where $P = 2L - 1$ is the polynomial accuracy of the quadrature in each random dimension. The exponential growth in the quadrature cost leads to the *"curse of dimensionality"* [29] and it makes the full quadratures appropriate only in low dimensions.

For higher dimensions, the quadrature costs may be reduced by using the sparse–tensor quadrature [30, 31]. The sparse quadrature is constructed by embedding successive tensor–product rules as

$$\mathcal{Q}_L^N[f] = \sum_{|\mathbf{k}|_1 \leq N+L-1} \left(\triangle_{k_1}^1 \otimes \cdots \otimes \triangle_{k_D}^1\right) f, \tag{16}$$

where L is the quadrature level and $\triangle_k^1$ is the one–dimensional difference grid between successive quadrature formula, *i.e.* $\triangle_k^1[f] = (\mathcal{Q}_k^1 - \mathcal{Q}_{k-1}^1)f$. For the bounded support in the local metamodel, the sparse quadrature is constructed using the Clenshaw–Curtis Chebyshev rules. The quadrature abscissae are the extrema of the Chebyshev polynomials [32] and the abscissae from increasing quadrature levels are nested, *i.e.* $\mathcal{Q}_{k-1}^1 \subset \mathcal{Q}_k^1$, which significantly reduce the total number of quadrature points. For the global metamodel with infinite support, generic 1D nested quadrature rules do not exist. However, the Kronrod–Patterson (KP) rule provides an optimal way to choose m additional quadrature points that complements the existing n Gauss quadrature points with an improve accuracy of $P = 2m + n - 1$, *cf.* [33, 34, 35]. Additional roots are symmetric about the origin and are real only for extensions $m = \{1, 2, 6, 10, 16\}$ or $m = \{1, 2, 8, 10, 20\}$, *cf.* [36]. The computation of the quadrature weights w_k is described in details in [31]. In the construction of the multivariate sparse quadrature rule, these 1D quadrature points are incrementally incorporated in

the sparse–tensor product. Although quadrature cost is reduced, Nobile *et al.* [37] observe that the sparse quadrature is optimal only when the integrand is isotropic.

If the function response is dominant in only one direction, an anisotropic quadrature could be used where the numbers of quadrature points are direction–dependent. In this study, for the local metamodel around the design point $\boldsymbol{\xi}_\alpha$, we can use an anisotropic quadrature which has high quadrature levels in the direction with the most significant change in function response and $L = 2$ in all other directions with linear or no function variation. The product rule for the anisotropic quadrature is thus

$$\mathcal{Q}_L^N[f] = \left(\mathcal{Q}_{k_1}^1 \times \cdots \times \mathcal{Q}_{k_N}^1\right) f, \quad \text{where} \quad k_1 = L \quad \text{and} \quad k_2 = \cdots = k_N = 2. \tag{17}$$

Examples of the different quadrature schemes are shown in Fig. 2. Details of the quadrature construction and the computation of w_k can be found in [38].

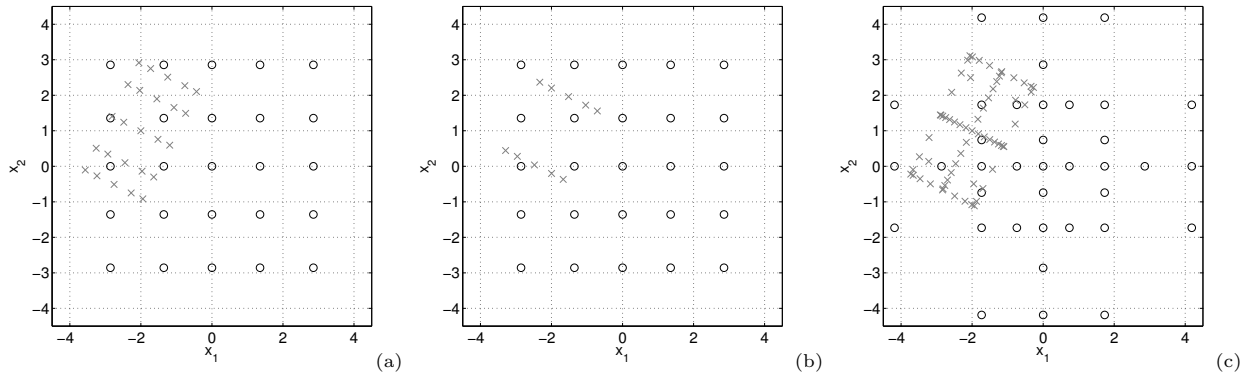

Figure 2: Different quadrature schemes used in this study are shown for the two–dimensional standard normal support and the corresponding bounded finite local support constructed on the design point at $(-2, 1)$. Full Gauss–Hermite quadrature with quadrature level $L = 5$ are plotted with circles in (a) and (b). Sparse Hermite Kronrod–Patterson quadrature with $L = 5$ is plotted with circles in (c). The gray crosses show the local quadratures in (a), (b) and (c) and they are full Gauss–Legendre, anisotropic Gauss–Legendre and sparse Clenshaw–Curtis Chebyshev quadratures, respectively. They all have a quadrature level $L = 5$. The displacement, rotation and scaling of the local quadratures are explained in Sec. 5.1.

3.3. Post Processing

Within the framework of the gPC expansion, the statistical moments are easily obtained. The mean and variance are

$$\begin{aligned} \mathbb{E}[f_{\mathrm{r}}(\boldsymbol{X})] &= f_0, \\ \sigma_f^2 = \mathrm{Var}[f_{\mathrm{r}}(\boldsymbol{X})] &= \sum_{m=1}^{M} f_m^2 \mathbb{E}[\phi_m^2(\boldsymbol{X})] = \sum_{m=1}^{M} f_m^2, \end{aligned}$$

and higher moments can be similarly derived. A sensitivity analysis of the system can also be carried out to identify which random dimensions are the most dominant onto the solution. This is achieved by calculating the partial variances from the Sobol' sensitivity indices associated with each random variable X_n as follows [39]

$$D_n = \mathrm{Var}\left[\mathbb{E}\left[f_{\mathrm{r}}(\boldsymbol{X})|X_n\right]\right].$$

In the context of the gPC expansion, they can be computed as

$$D_n = \sum_{m \in I_n} f_m^2 \mathbb{E}[\phi_m^2(\boldsymbol{X})] = \sum_{m \in I_n} f_m^2, \tag{18}$$

where I_n is the set of indices to the polynomials containing only x_n. In practice, those indices are numerically very easy to compute due to the hierarchical nature of the orthogonal polynomial basis and its tensor–like form. Moreover, the response surfaces of $f(\boldsymbol{x})$ can be easily constructed from Eq. (13) which allows one to predict the system solution at arbitrary points in the support. We can compute an empirical estimator of the cdf of $f_{\mathrm{r}}(\boldsymbol{X})$ from

$$\hat{F}_{\mathrm{r}}(y) \;=\; \frac{1}{Z_{\mathrm{MC}}} \sum_{k=1}^{Z_{\mathrm{MC}}} \mathbf{1}_{(-\infty,y]}(Y_{k,\mathrm{r}}),$$

where $Y_{k,\mathrm{r}} = f_{\mathrm{r}}(\boldsymbol{\zeta}_k)$ and $\boldsymbol{\zeta}_k$ are independent random variables generated according to the pdf $\rho(\boldsymbol{x})$. The α–quantile $y_{\alpha,\mathrm{r}}$ of $f_{\mathrm{r}}(\boldsymbol{X})$ can be approximated from the ordered set $Y_{(1),\mathrm{r}} \leq Y_{(2),\mathrm{r}} \leq \cdots \leq Y_{(k),\mathrm{r}}$

$$\hat{Y}_{\alpha,\mathrm{r}} = Y_{(\lceil \alpha Z \rceil),\mathrm{r}}, \tag{19}$$

where a large number of MC samples can be used. However, the accuracy of the quantile will be poor for very large values of α in the sense that the α–quantile $y_{\alpha,\mathrm{r}}$ of $f_{\mathrm{r}}(\boldsymbol{X})$ can be significantly different from the α–quantile y_α of $f(\boldsymbol{X})$.

4. Design Point Search

In the parameter space, the feasibility surface of the complete model is the surface $\{\boldsymbol{x} \in \mathbb{R}^N, f(\boldsymbol{x}) - y_\alpha = 0\}$. From now on, we assume the components of the random input vector are independent and identically distributed random variables with Gaussian distribution, zero mean and unit variance. For dependent random vectors not in the standard normal space, the Rosenblatt or Nataf bijective transformations can be used to transform them into the desired standard normal space [40, 41]. For independent random vectors not in the standard normal space, the inverse cumulative distribution function method, *cf.* [42], can be applied to each input random variable:

$$\tilde{x}_i = \Phi^{-1} \circ F_{X_i}(x_i), \tag{20}$$

where F_{X_i} is the cdf of the random variable X_i and Φ is the cdf of a standard Gaussian distribution. The design points, $\boldsymbol{\xi}_{\alpha_i}$, are the points on the feasibility surface $\{\boldsymbol{x} \in \mathbb{R}^N\,,\, f(\boldsymbol{x}) - y_\alpha = 0\}$ that are the closest to the origin. The subscript i is used to index multiple design points; in the rest of the paper i is omitted if only one design point is present or considered. The design points are found by minimizing an objective function subject to the feasibility constraint. Thus, defining the limit state function $g(\boldsymbol{x})$ and the objective function $r(\boldsymbol{x})$ as

$$g(\boldsymbol{x}) \;=\; f(\boldsymbol{x}) - y_\alpha, \tag{21}$$

$$r(\boldsymbol{x}) \;=\; \frac{1}{2}\boldsymbol{x}^T\boldsymbol{x}, \tag{22}$$

the design points are solutions to the following minimum constraint problem

$$\min r(\boldsymbol{x}) \quad s.t. \quad g(\boldsymbol{x}) = 0. \tag{23}$$

The design points are the optimal center points for the construction of the response surface in reliability analysis [43, 44] and the corresponding search algorithms are reviewed extensively in [4, 45]. In the following section, the gradient projection method, applied to the complete model, and the Lagrange multiplier method, applied to the gPC metamodel, are introduced.

4.1. Gradient Projection Method

The gradient projection method (GPM) described in [45] is implemented in the current study. First, the α–quantile is estimated from the complete model using Eq. (4) and the limit state function (21) is approximated by

$$g(\boldsymbol{x}) = f(\boldsymbol{x}) - \hat{Y}_\alpha. \tag{24}$$

From an initial guess that lies not far from the feasibility surface $\{\boldsymbol{x} \in \mathbb{R}^N\,,\, g(\boldsymbol{x}) = 0\}$, GPM generates a series of points, each satisfying $g(\boldsymbol{x}) = 0$, that gradually converge towards a minimum of $r(\boldsymbol{x})$. At each iteration, a new point is generated along the negative gradient of $g(\boldsymbol{x})$ which is determined using the central difference scheme from the complete model. A Newton–type correction is used to project the new point back onto the limit state surface. The iterations continue until the empirical design point, $\hat{\boldsymbol{\xi}}_\alpha$, converges.

In this study, the initial guess is taken as the input whose function solution value is the closest to $\hat{Y}_\alpha$, *i.e.*

$$\boldsymbol{x}_0 = \underset{\boldsymbol{x}\in\{\boldsymbol{\zeta}_1,\ldots,\boldsymbol{\zeta}_{Z_{\mathrm{MC}}}\}}{\operatorname{argmin}} \|f(\boldsymbol{x}) - \hat{Y}_\alpha\|, \tag{25}$$

where $\boldsymbol{\zeta}_1, ..., \boldsymbol{\zeta}_{Z_{\mathrm{MC}}}$ are the MC samples used to generate $\hat{Y}_\alpha$ in Eq. (4). In fact, as $f(\boldsymbol{x})$ is not necessarily convex, we need to repeat the design point search with sufficiently many initial guesses to ensure that all global design points are located.

4.2. gPC Lagrange Multiplier Method

The gPC metamodel $f_{\mathrm{r}}(\boldsymbol{x})$ is constructed as outlined in Sec. 3 (Eq. (13)). The value of y_α is then estimated from the global gPC metamodel using Eq. (19) and the limit state function (21) is approximated by

$$g_{\mathrm{r}}(\boldsymbol{x}) = f_{\mathrm{r}}(\boldsymbol{x}) - \hat{Y}_{\alpha,\mathrm{r}}. \tag{26}$$

The solutions of the design points involving linear and higher–order gPC metamodel will be examined separately in the following sections.

4.2.1. Linear Limit State

Let us consider the linear gPC metamodel $f_{\mathrm{r}}(\boldsymbol{x}) = f_0 + \sum_{m=1}^{N} f_m x_m$ for $f(\boldsymbol{x})$ in $\mathbb{R}^N$. The corresponding feasibility surface $\{\boldsymbol{x} \in \mathbb{R}^N\,,\, f_{\mathrm{r}}(\boldsymbol{x}) - \hat{Y}_{\alpha,\mathrm{r}} = 0\}$ is a hyperplane. The point $\boldsymbol{x}$ of this hyperplane that minimizes $\|\boldsymbol{x}\|$ is the point whose normal vector, with respect to the hyperplane, goes through $\mathbf{0}$. The normal vector of an arbitrary point is given by its gradient, *i.e.* $\boldsymbol{n} = \nabla f_{\mathrm{r}}(\boldsymbol{x})$; for the first–order gPC metamodel, the normal vector is $\boldsymbol{n} = (f_1, f_2, \ldots, f_N)$. The linear estimation of $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$ thus has the value $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}} = l\boldsymbol{n}$ and l can be determined by solving $g_{\mathrm{r}}(\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}) = 0$ which gives

$$f_0 - \hat{Y}_{\alpha,\mathrm{r}} + l \sum_{m=1}^{N} f_m^2 = 0. \tag{27}$$

Although its computation cost is not high, the linear gPC metamodel will fail to find the design points if more than one is present.

4.2.2. High–Order Limit State

The accuracy of the $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$ can be improved by increasing the accuracy of the metamodel, *i.e.* the order of gPC expansion. The Lagrange multiplier method (LMM) is an approach to find the maxima or minima of an objective function subject to limit state functions and it is used to solve Eq. (23) involving high–order gPC limit state functions $g_{\mathrm{r}}(\boldsymbol{x})$. Using the multiplier λ, we combine the limit state (26) and the objective (22) to form the Lagrange function

$$\Lambda(\boldsymbol{x}, \lambda) = r(\boldsymbol{x}) + \lambda g_{\mathrm{r}}(\boldsymbol{x}).$$

If $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$ is a minimum of the original constraint problem, then it must also be a stationary point of the Lagrange function, where gradients of $\Lambda(\boldsymbol{x},\lambda)$ are 0, *i.e.*

$$\boldsymbol{x}_n + \lambda \sum_{m=0}^{M} f_m \frac{\partial \phi_m(\boldsymbol{x})}{\partial x_n} = 0, \quad \text{for } n = 1, ..., N, \tag{28}$$

$$\sum_{m=0}^{M} f_m \phi_m(\boldsymbol{x}) = \hat{Y}_{\alpha,\mathrm{r}}. \tag{29}$$

However, not all stationary points are solutions to the minimum constraint problem (23) and they need to be verified *a posteriori*. When solved analytically, this algorithm is able to find all the stationary points of the gPC metamodel, from which the local extrema could then be determined. The analytical solution of second-order gPC limit state function with few random variables is feasible; however for multivariate and higher-order limit state functions, a numerical solution is needed.

The Newton method is used to solve the linearized version of the Lagrange function recursively [46]. Given initial guesses $\boldsymbol{x}_0$ and λ_0, successive increments in $\boldsymbol{x}$ and λ are determined from

$$\begin{aligned} \nabla\Lambda(\boldsymbol{x},\lambda)^T + \boldsymbol{H}(\boldsymbol{x},\lambda)\delta\boldsymbol{x} \quad &+\nabla g_{\mathrm{r}}(\boldsymbol{x})^T\delta\lambda \quad = 0, \\ g_{\mathrm{r}}(\boldsymbol{x}) + \nabla g_{\mathrm{r}}(\boldsymbol{x})\delta\boldsymbol{x} \quad & \quad = 0, \end{aligned}$$

where $\boldsymbol{H}(\boldsymbol{x},\lambda)$ is the Hessian in $\boldsymbol{x}$ of $\Lambda(\boldsymbol{x},\lambda)$, *i.e.* $\boldsymbol{H}(\boldsymbol{x},\lambda) = \boldsymbol{I} + \lambda\nabla^2 g_{\mathrm{r}}(\boldsymbol{x})$. The above equations can be solved in the matrix form as

$$\begin{bmatrix} \delta\boldsymbol{x} \\ \delta\lambda \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}(\boldsymbol{x},\lambda) & \nabla g_{\mathrm{r}}(\boldsymbol{x})^T \\ \nabla g_{\mathrm{r}}(\boldsymbol{x}) & 0 \end{bmatrix}^{-1} \begin{bmatrix} -\nabla\Lambda(\boldsymbol{x},\lambda)^T \\ -g_{\mathrm{r}}(\boldsymbol{x}) \end{bmatrix}.$$

Although calls to the metamodel can be considered as essentially cost free in comparison to evaluations of the complete function, for metamodels with large N and P, the Newton method can still be time consuming in practice. Therefore, we need to select the initial guesses well to minimize the number of computations. Eq. (25) is also used to determine the initial guess with $f_{\mathrm{r}}(\boldsymbol{x})$ and $\hat{Y}_{\alpha,\mathrm{r}}$ used instead of $f(\boldsymbol{x})$ and $\hat{Y}_\alpha$ in the equation.

Two levels of convergence in gPC Lagrange multiplier design points need to be considered. First, the iterative numerical search of $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$ needs to converge. This convergence criterion does not need to be very strict because the additional accuracy gained from higher decimal places in $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$ will not necessarily translate to significant improvement in the quantile estimation. Second, the values of $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$ from successive quadrature level L and different gPC order P need to agree. Since the metamodels are polynomial approximations of the complete function, there could be numerical critical points where $f_{\mathrm{r}}(\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}) = \hat{Y}_{\alpha,\mathrm{r}}$ but $f(\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}})$ is very different from $\hat{Y}_{\alpha,\mathrm{r}}$. These numerical critical points can be easily identified with an additional function evaluation on the complete model and they will not be recurring at all L and P tested. In addition, they are likely to be further away from the origin of the standard normal space than the physical critical points as the physical design points should have the smallest modulus.

4.3. Cost Comparison

The cost of the gPC LMM depends only on the number of quadrature points required to construct the gPC metamodel since both $\hat{Y}_{\alpha,\mathrm{r}}$ and $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$ are computed from the gPC metamodel. In GPM, the cost consists of that of MC estimation of y_α and that of gradient projection search for $\boldsymbol{\xi}_\alpha$. The number of MC sample, Z_{MC}, required to ensure that y_α lies within the confidence interval $[\hat{Y}_{\alpha+\epsilon_1}, \hat{Y}_{\alpha+\epsilon_2}]$ with a probability γ can be determined from Eqs. (5) and by inspection of these equations, Z_{MC} increases roughly by $\mathcal{O}(10)$ for each decrease of $\mathcal{O}(10^{-1})$ in $1-\alpha$ for fixed γ and ϵ. With $\hat{Y}_\alpha$ and initial guesses $\boldsymbol{x}_0$ determined from Eq. (25), we search the design point with the GPM whose estimated cost is $n_l(N+1)$ where $n_l \approx 5-20$ [47]. As GPM

can only find local design points near $\boldsymbol{x}_0$, several initial guesses may be required to ensure that all global design points are located. Therefore, the total cost is

$$Z_{total} \sim Z_{\mathrm{MC}} + mn_l(N+1). \tag{30}$$

where m is the number of initial guesses.

Using these parameters, the design point search cost for different methods at different dimensions are shown for $\alpha = 99\%$ and 99.9% in Fig. 3. The cost for gPC LMM with full and sparse quadratic metamodels are shown with squared and circled lines. GPM with $\gamma = 50\%$ (dashed) and $\gamma = 90\%$ (dashed–dotted) for $\epsilon_{1,2} = \pm(1-\alpha)/2$ are shown with lower and upper bounds of $n_l = 5$ and $n_l = 20$ for one initial guess, *i.e.* $m = 1$. Taking the lower cost bound ($n_l = 5$ and $\gamma = 50\%$), the full quadratic metamodel is as costly as GPM between $N = 5$ and 7. For sparse quadrature metamodels, this limit is higher. This N–limit increases with increasing α, γ and n_l. Although the range of applicable N is very much problem-dependent, a conservative estimate would place the limit of effective–N of gPC LMM at about $N = 7$.

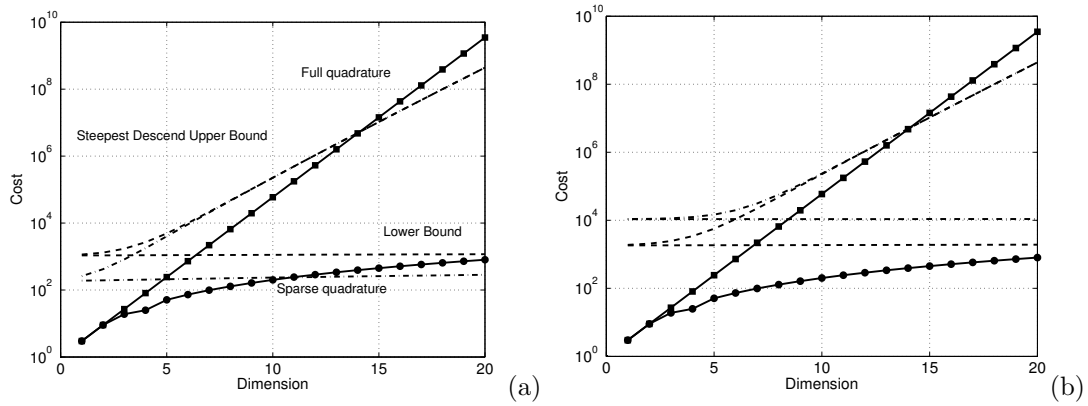

Figure 3: Cost comparison of different methods used to determine the design points for (a) $\alpha = 99\%$ and (b) $\alpha = 99.9\%$. The squared and circled lines are the cost of the gPC LMM with full and sparse quadratures with quadratic polynomial accuracy. The dashed (resp. dashed–dotted) lines denote cost of the GPM with the complete model when we look for 50% (resp. 90%) probability that the y_α falls within $[\alpha \pm \epsilon]$–empirical quantile estimator where $\epsilon = (1-\alpha)/2$.

5. Local Refinement

Once the design points $\hat{\boldsymbol{\xi}}_{\alpha_i,r}$ are determined from the global metamodel, the function response in their neighbourhoods can be refined. The following sections describe the two steps of this refinement procedure, namely the definition of the local refinement domain D_{β_i} for the i–th design point $\hat{\boldsymbol{\xi}}_{\alpha_i,r}$ and the construction of the multi–element gPC metamodel.

5.1. Local Refinement Domain Definition

A local metamodel is constructed on a N–dimensional hypercube enclosing each design point. The location, orientation, and size of the hypercube affect the accuracy of the local metamodel. The definition of the local refinement domain can be represented by a transformation operator $\boldsymbol{x} = \mathbf{T}_i(\boldsymbol{x}^*)$ where a point in the uniform bounded support $\boldsymbol{x}^* \in [-1,1]^N$ is mapped to the local domain $\boldsymbol{x} \in D_{\beta_i}$. In addition to a shift by $\hat{\boldsymbol{\xi}}_{\alpha_i,\mathrm{r}}$, the transformation contains a scaling matrix $\mathbf{S}_i$ and a rotation matrix $\mathbf{R}_i$, *i.e.* $\boldsymbol{x} = \hat{\boldsymbol{\xi}}_{\alpha_i,\mathrm{r}} + \mathbf{R}_i\mathbf{S}_i\boldsymbol{x}^*$. The subscript i denotes that the transformation is associated with the i–th design point but it will be omitted in this section for the clarity in notation. The following sections outline the constructions of the scaling and the rotation matrices.

5.1.1. New Orthogonal Basis

Since $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$ is a design point, the feasibility surface $f_{\mathrm{r}}(\boldsymbol{x}) - \hat{Y}_{\alpha,\mathrm{r}} = 0$ is orthogonal to the direction $\hat{\boldsymbol{\xi}}'_{\alpha,\mathrm{r}} = \hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}/\|\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}\|$ at $\boldsymbol{x} = \hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$. Therefore the function $f_{\mathrm{r}}(\boldsymbol{x})$ is slowly varying in the directions orthogonal to $\hat{\boldsymbol{\xi}}'_{\alpha,\mathrm{r}}$. Motivated by this remark and by the possibility to use an anisotropic quadrature, we decide that instead of having the hypercube aligned along the axes of the standard normal space, it should be aligned along $\hat{\boldsymbol{\xi}}'_{\alpha,\mathrm{r}}$ and its transverse directions by applying the rotation matrix $\mathbf{R}$ to the original canonical basis $\{\boldsymbol{e}_1, ..., \boldsymbol{e}_N\}$. $\mathbf{R}$ can be determined using Gram–Schmidt orthonormalization, *i.e.* $(\boldsymbol{d}_1, \boldsymbol{d}_2, ..., \boldsymbol{d}_N) = \mathbf{R}(\boldsymbol{e}_1, ..., \boldsymbol{e}_N)$ where $\boldsymbol{d}_1 = \hat{\boldsymbol{\xi}}'_{\alpha,\mathrm{r}}$ and $(\boldsymbol{d}_n)_{n=2}^N$ are its transverse directions.

5.1.2. One–Dimensional Refinements

After the new orthogonal basis is determined, one–dimensional gPC expansions of the function responses are constructed about $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$ along $\boldsymbol{d}_1 = \hat{\boldsymbol{\xi}}'_{\alpha,\mathrm{r}}$ and its transverse directions. These metamodels are 1D expansions of the form of Eq. (13) and are used to examine the function response in detail before the N–dimensional local refinement is carried out. The quadrature costs of 1D refinement studies scale roughly as NL which is low compared to the cost of full or sparse quadratures.

The 1D refinements can give additional information about the area near the design point in different ways. First, they can determine if the design point is numerical, *i.e.* $f_{\mathrm{r}}(\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}) = \hat{Y}_{\alpha,\mathrm{r}}$ but $f(\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}) \not\approx \hat{Y}_{\alpha,\mathrm{r}}$. Second, they can reveal if the function response is odd or even in each orthogonal direction. If one of the parities is dominant in one particular direction, the local metamodel should only be carried out at the corresponding quadrature levels. Third, the 1D refinements demonstrate if the function response is anisotropic amongst the transverse directions. Partial variances are good quantitative measures of anisotropy. The anisotropic quadrature described in Sec. 3.2 can be used to decrease the quadrature cost if the function is indeed anisotropic. Last, the size of D_β can also be adjusted according to the 1D refinement results. By plotting the response of the 1D metamodel, one can readily observe if the lengths of D_β in each orthogonal direction includes sufficient zones of interest in the complete function. In addition, the response of the global metamodel along the same orthogonal directions can be compared with the 1D refinement to identify zones in need of local refinement. Detailed examples shown in Sec. 5.1.4 describe how the 1D refinement can be used to determine the size of D_β.

5.1.3. Refinement Zone Dimensions

The scaling matrix $\mathbf{S}$ is a diagonal matrix that determines the size of D_β where S_{nn} is the scaling factor in the direction $\boldsymbol{d}_n$. First we define the length of D_β in the direction $\boldsymbol{d}_1 = \hat{\boldsymbol{\xi}}'_{\alpha,\mathrm{r}}$ to be $2l_1$ and place $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$ at the mid–point along $\hat{\boldsymbol{\xi}}'_{\alpha,\mathrm{r}}$. As the purpose of the local refinement is to improve the metamodel accuracy away from the input means, *i.e.* origin of the standard normal space, we need to ensure that D_β is sufficiently far from the origin so that $l_1 < \|\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}\|$. For the first iteration of the local metamodel, $l_1 = 1$ is used although the one–dimensional refinements described in the previous section can be used to determine a better value of l_1 when $\hat{Y}_{\alpha,\mathrm{r}}$ become more accurate at later refinement stages.

Next the lengths $(l_n)_{n=2}^N$ of D_β in its transverse directions $(\boldsymbol{d}_n)_{n=2}^N$ are determined either according to the 1D refinement results or as a function of l_1. If the 1D refinement shows that the value of the function near the design point is close to $\hat{Y}_{\alpha,\mathrm{r}}$, we define the transverse lengths using the following approach: We consider the hyperplane that is perpendicular to $\hat{\boldsymbol{\xi}}'_{\alpha,\mathrm{r}}$ and goes through $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}} - l_1\hat{\boldsymbol{\xi}}'_{\alpha,\mathrm{r}}$ and we ensure that the pdfs of points on the edge of the hyperplane have at most the same value as the pdf of the design point, *i.e.* that the distance to the edge of the hyperplane is at most $\|\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}\|$. From the centre of the hyperplane, located at $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}} - l_1\hat{\boldsymbol{\xi}}'_{\alpha,\mathrm{r}}$, the edges of the hyperplane are located at $\boldsymbol{x}_n = \hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}} - l_1\hat{\boldsymbol{\xi}}'_{\alpha,\mathrm{r}} + l_n\boldsymbol{d}_n$. Therefore, l_n are determined by $\|\boldsymbol{x}_n\| = \|\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}\|$ for $n = 2$ to N. If the transverse l_n is too large, the number of inputs having less probability than the inputs contributing to $\hat{Y}_{\alpha,\mathrm{r}}$ increases. If l_n is too small, inputs with sufficiently large contribution to $\hat{Y}_{\alpha,\mathrm{r}}$ are missed. If the 1D refinement shows that the value of the function near the design point is not close to $\hat{Y}_{\alpha,\mathrm{r}}$, the approach in the following section is used.

5.1.4. Examples

A Gaussian and a hyperbolic tangent function for $N = 5$ are used to illustrate how 1D refinements can be used to determine the size of D_β, *cf.* Fig. 4. The results are plotted with $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$ as the origin along the axes $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$. The exact solution is plotted in solid dashed lines. The responses of quadratic global metamodels are plotted in gray lines and the gray circles denote the values of $\hat{Y}_{\alpha,\mathrm{r}}$. The 1D local quartic (with Z=5 points) refinements are shown in solid black lines and the quadrature evaluations used to construct the 1D refinements are in black dots.

We want to define $\boldsymbol{l} = (l_n)_{n=1}^N$ such that regions with y values close to y_α are included in D_β. Although the approximation from the global metamodel is poor near the design point, the 1D local refinements resolve very accurately the function response using only few additional quadrature samples. Therefore, we define the refinement zone such that 1D refinement regions with y values larger than the best estimation of y_α are included in D_β. Before the local metamodels are constructed, the global $\hat{Y}_{\alpha,\mathrm{r}}$ (gray circles in Fig. 4) is used as the best estimation of y_α. With additional local metamodel, this boundary can be updated accordingly (gray triangles representing the updated estimation $\hat{Y}_{\alpha,\mathrm{ME}}$ of y_α).

In the Gaussian example, *cf.* Fig. 4.(a), the initial choice of $l_1 = 1$ includes sufficient areas about $\hat{Y}_{\alpha,\mathrm{r}}$. The estimation of the transverse l_n outlined in Sec. 5.1.3 includes regions whose y values are much smaller than $\hat{Y}_{\alpha,\mathrm{r}}$; therefore, transverse l_n is limited at 1.5 by examining the one–dimensional refinement values of y. With the first multi–element estimation of $\hat{Y}_{\alpha,\mathrm{ME}}$ (gray triangle), the refined region with y value larger than the estimated quantile becomes even smaller; thus, transverse l_n is decreased to 1. In the hypertangent example, *cf.* Fig. 4.(b), a value of $l_n = 1$ is sufficient in all transverse directions $\boldsymbol{d}_n$ because all y values are above $\hat{Y}_{\alpha,\mathrm{r}}$ since the responses in these directions are flat. Therefore, we use the estimation of transverse l_n outlined in Sec. 5.1.3 to ensure that enough probable inputs are included in D_β.

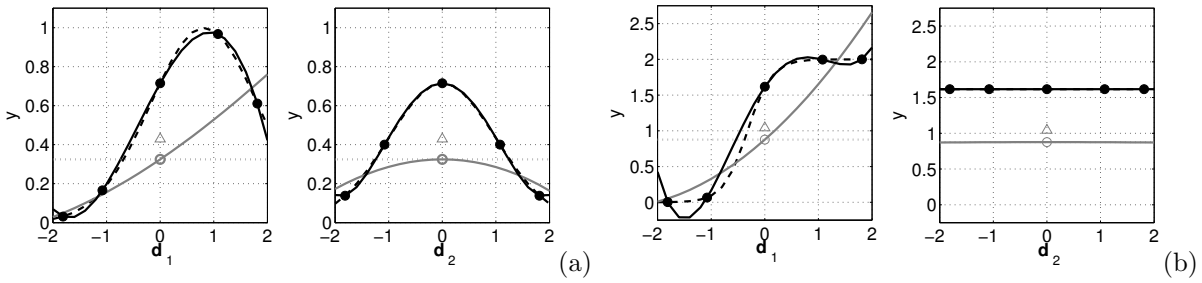

Figure 4: The one–dimensional local refinement about the design point for (a) a Gaussian and (b) a hypertangent function. The responses from the global metamodels (quadratic) and the one–dimensional metamodels (quartic) are plotted in black and gray lines, respectively. The gray circles and triangles show the estimated y_α from the global metamodel and the multi–element metamodel. The black dashed line is the exact function response and the black dots are the complete function evaluations used to construct the one–dimensional refinements metamodels.

Examples of hypercubes D_β in 2D and 3D are shown in Fig. 5. The values of $\boldsymbol{\xi}_\alpha$ are $(-2, 1)$ and $(2, 2, 2)$ in the 2D and 3D case, respectively. In the 2D illustration, an 1×1 square (dashed) is placed at the origin. In addition, a scaled, rotated and translated local domain in the new orthogonal axis is plotted in black lines. The size of D_β is determined with the scaling rules outlined in Sec. 5.1.3 with $l_1 = 1$. For the 3D case, the local domain is plotted in black as well and no scaling is used to make the illustration clearer. In addition, two additional refinement regions with the same $\boldsymbol{\xi}'_\alpha$ but with different transverse axes are also shown with gray dashed lines in Fig. 5.(b). However, it appears that the orientation of the transverse axes is not important. The local metamodels are constructed on these local bounded domains using the stochastic spectral projection approach outlined in Sec. 3.

5.2. Multi–Element gPC

We have a global gPC metamodel which has a support in the standard normal space, $D = \mathbb{R}^N$. Within D, we define local refinement domains, D_{β_i} centered around $\hat{\boldsymbol{\xi}}_{\alpha_i,\mathrm{r}}$ where local gPC metamodels are constructed.

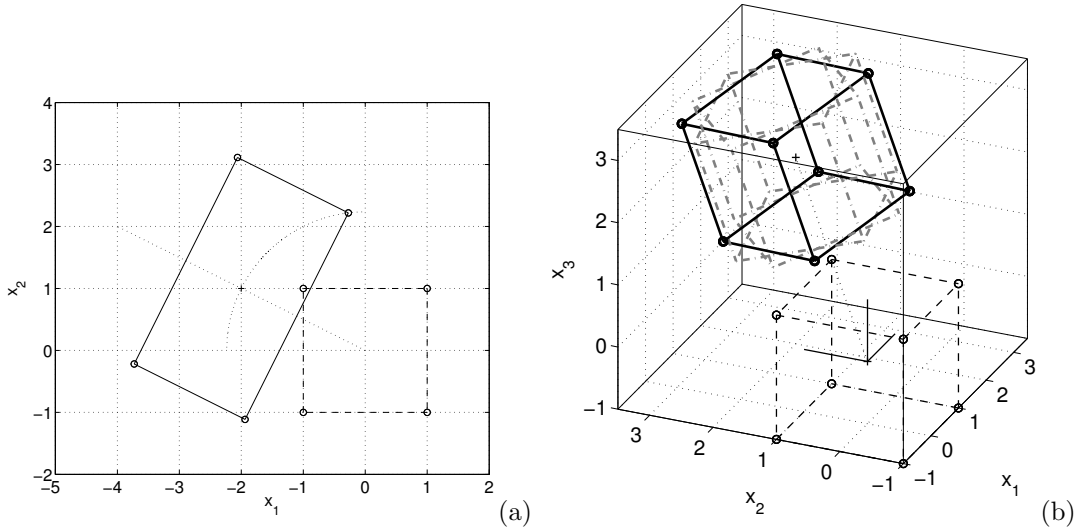

Figure 5: The local refinement zone definition in (a) 2D and (b) 3D. The dashed–lines mark the unit–length bounded support centered about the origin and the black solid lines mark the bounded support that is translated, rotated and scaled according to the location of the design point. The scaling of the local zone follows the rules outlined in Sec. 5.1.3 in (a) while no scaling is used in (b) to facilitate the illustration. In (b), the gray dashed–dotted lines outline several possible orientations of the rotated support.

Thus, we have a multi–element approach where the standard normal space is decomposed as follows:

$$D_{\mathrm{ME}} = \begin{cases} D_{\mathrm{global}} = D \setminus D_{\mathrm{local}}, & \text{domain of global gPC,} \\ D_{\mathrm{local}} = \cup_{i=1}^{N_\beta} D_{\beta_i}, & \text{domains of local refinement about } \hat{\boldsymbol{\xi}}_{\alpha_i,\mathrm{r}}, \text{ for } i = 1, ..., N_\beta. \end{cases} \tag{31}$$

where D_{β_i} is the i–th hypercube built in Sec. 5.1, centered at $\hat{\boldsymbol{\xi}}_{\alpha_i,\mathrm{r}}$ and N_β is the number of design points. The local domain is the union of the refinement domains D_{β_i}. While the global metamodel is used in D_{global}, the local metamodel is used in each D_{β_i}. We represent the local metamodel for $\boldsymbol{x} \in D_{\beta_i}$ as

$$f^*_{\mathrm{r},i}(\boldsymbol{x}) \quad = \quad \sum_{m=0}^{M_i^*} f^*_{m,i} \psi_m(\mathbf{T}_i^{-1}(\boldsymbol{x})), \quad \text{for} \quad i = 1, ..., N_\beta, \tag{32}$$

where $f^*_{m,i} = \mathbb{E}\left[f(T_i(\boldsymbol{X}^*))\psi_m(\boldsymbol{X}^*)\right]$ and the gPC truncation in D_{β_i} is denoted by M_i^*. We take $\boldsymbol{X}^*$ as a random vector with a uniform pdf in $[-1,1]^N$ and $\psi_m(\boldsymbol{x}^*)$ the Legendre polynomial.

Thus, the final multi–element gPC (MEgPC) metamodel is

$$f_{\mathrm{ME}}(\boldsymbol{x}) = \begin{cases} \sum_{m=0}^{M} f_m \phi_m(\boldsymbol{x}), & \text{if } \boldsymbol{x} \in D_{\mathrm{global}}, \\ \sum_{m=0}^{M_i^*} f^*_{m,i} \psi_m(\mathbf{T}_i^{-1}(\boldsymbol{x})), & \text{if } \boldsymbol{x} \in D_{\beta_i}. \end{cases} \tag{33}$$

When MC samples are taken on Eq. (33), we use the inverse function to determine if the MC point lies in D_{β_i}, *i.e.* $\boldsymbol{x} \in D_{\beta_i}$ if $\mathbf{T}_i^{-1}(\boldsymbol{x}) \in [-1,1]^N$. Because Eq. (33) is not computationally expensive to evaluate, a large number of MC points can be used to determine the α–quantiles. The α–quantile y_α can now be approximated with the MEgPC metamodel with Z MC samples as

$$\hat{Y}_{\alpha,\mathrm{ME}} = Y_{(\lceil \alpha Z \rceil),\mathrm{ME}}, \tag{34}$$

where $Y_{k,\mathrm{ME}} = f_{\mathrm{ME}}(\boldsymbol{\zeta}_k)$ and $Y_{(1),\mathrm{ME}} \leq Y_{(2),\mathrm{ME}} \leq \cdots \leq Y_{(k),\mathrm{ME}}$ is the ordered set. The Monte Carlo method

with very large sample size, outlined in Section 2.2, can also be used. The random inputs $\boldsymbol{\zeta}_k$ are independent random variables generated according to the pdf $\rho(\boldsymbol{x})$.

In the above approach, the design points $\hat{\boldsymbol{\xi}}_{\alpha_i,\mathrm{r}}$ are determined from $\hat{Y}_{\alpha,\mathrm{r}}$ of the global metamodel. Clearly, the value of $\hat{\boldsymbol{\xi}}_{\alpha_i,\mathrm{r}}$ can be refined after the first multi–element quantile estimation by solving the minimum constraint problem (23) with an updated limit state function based on the local metamodel, *i.e.* $g(\boldsymbol{x}) = f^*_{\mathrm{r},i}(\boldsymbol{x}) - \hat{Y}_{\alpha,\mathrm{ME}}$. The new design point estimator $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{ME}}$ will likely be slightly more accurate.

6. Examples

A Gaussian-like and a hyperbolic tangent functions are used to validate the MEgPC approach in the next two sections. First, the accuracy in the design points determined by GPM and gPC LMM are first compared at selected random dimension N. Second, the convergences of errors in the empirical α–quantile estimated with different numerical methods are demonstrated at selected N. Finally, the different quantile estimation methods are compared at different N values. After the MEgPC approach is validated, it is applied to a flooding model in the third section.

6.1. Gaussian peak

We examine the quantile of the output of a function that is the superposition of Gaussian peaks in the following form

$$f(\boldsymbol{x}) = \sum_{i=1}^{N_\alpha} \prod_{n=1}^{N} \exp\left(-\frac{(x_n - \mu_{n,i})^2}{2\sigma_{n,i}^2}\right), \tag{35}$$

where the N–dimensional input are independent and identically distributed random variables following the Gaussian distribution with mean zero and variance one. $\boldsymbol{\sigma}_i = (\sigma_{n,i})_{n=1}^N$ controls the width of the i-th peak centered at $\boldsymbol{\mu}_i = (\mu_{n,i})_{n=1}^N$. N_α is the number of peaks included in the function.

6.1.1. Design Point Search

Validation For $N = 2$ and $N_\alpha = 1$ (single peak), the pseudo–exact 99%–quantile y_α is determined from 1×10^9 MC points and the pseudo–exact design point $\boldsymbol{\xi}_\alpha$ is found with an iterative algorithm to locate $f(\boldsymbol{\xi}_\alpha) = y_\alpha$ along the direction $\boldsymbol{\mu}' = \boldsymbol{\mu}/\|\boldsymbol{\mu}\|$. Then, LMM is first validated by examining the convergence of $\|\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}\|$ and $\hat{\boldsymbol{\xi}}'_{\alpha,\mathrm{r}}$ for $\boldsymbol{\sigma} = (1,1)$ and $\boldsymbol{\mu} = (2,-1)$. Due to the low dimensionality, only the full quadrature is used for the validation case. The error of $\|\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}\|$ is normalized with $\|\boldsymbol{\xi}_\alpha\|$ and the convergence of $\hat{\boldsymbol{\xi}}'_{\alpha,\mathrm{r}}$ is determined from $\hat{\boldsymbol{\xi}}^T_{\alpha,\mathrm{r}}\boldsymbol{\xi}_\alpha/\|\boldsymbol{\xi}_\alpha\|^2$ whose value would be 1 if the two vectors overlap exactly. Two different numerical solutions are examined, *cf.* Fig. 6. First, the convergences $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$ determined from linear gPC constraints (27) constructed with quadrature levels $L = 2$ to $L = 7$ are evaluated. While $\hat{\boldsymbol{\xi}}'_{\alpha,\mathrm{r}}$ converges spectrally with increasing L, $\|\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}\|$ does not converge due to both approximation errors in $\hat{Y}_{\alpha,\mathrm{r}}$ and $f_\mathrm{r}(\boldsymbol{x})$. Therefore, linear gPC metamodels should be omitted. Second, the gPC metamodels from $L = 2$ to 14 and $P = L - 1$ are used as the limit state function for the gPC LMM. With increasing L, the convergence of $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$ towards its exact value is clearly observed in both $\hat{\boldsymbol{\xi}}'_{\alpha,\mathrm{r}}$ and $\|\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}\|$.

Single Peak The accuracies of the gPC LMM and GPM are compared. Eq. (35) with $N_\alpha = 1$ (single peak) and $N = 5$ is examined in detail with $\sigma_n = 1$ and $\mu_n = 2/\sqrt{N}$ for $n = 1, ..., N$ (so that $\|\boldsymbol{\mu}\| = 2$).

For GPM, the initial guess $\hat{Y}_\alpha$ is estimated from $f(\boldsymbol{x})$ with sufficient MC points to obtain 90% probability that y_α lies within the confidence interval $[\hat{Y}_{\alpha-\epsilon}, \hat{Y}_{\alpha+\epsilon}]$ where $\epsilon = (1-\alpha)/2$. Eq. (25) is used to determine the initial guess from the MC samples; in GPM, only one initial guess is used. Due to randomness in the initial guess of $\hat{Y}_\alpha$, the ensemble average of the error convergences in $\hat{\boldsymbol{\xi}}_\alpha$ from 100 independent realizations is computed; only the mean is shown in Fig. 7 as the standard deviation is small. It is clear that the initial MC sampling is expensive and leads to the large initial cost at the start of the $\hat{Y}_\alpha$ convergence. In comparison,

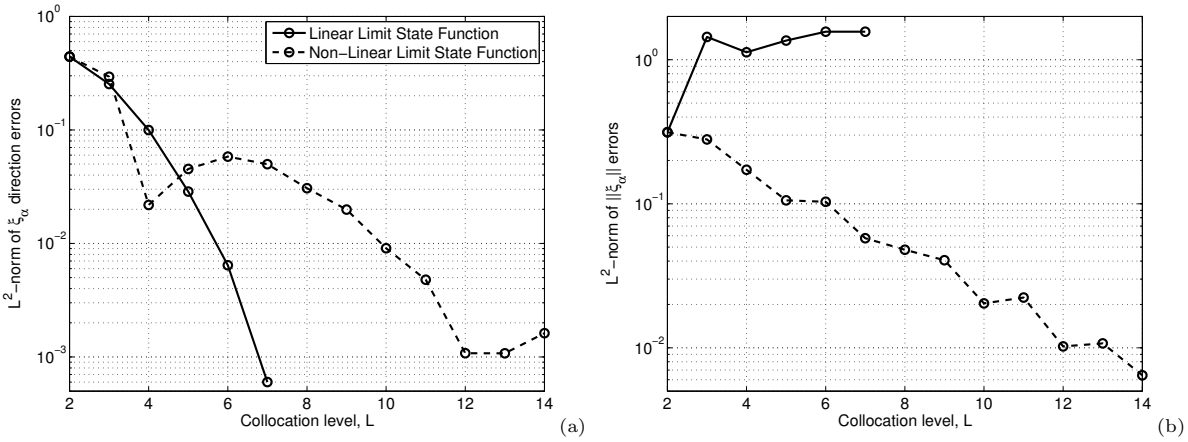

Figure 6: The convergence of the design point direction and distance from two numerical methods whose gPC limit state functions are constructed with increasing quadrature level.

the convergence of the error in $\hat{\boldsymbol{\xi}}_\alpha$ in the GPM step is rapid, resulting in a nearly vertical line in the log scale. It is within the range estimated by Eq. (30).

In the gPC LMM, the full Gauss–Hermite and sparse Hermite Kronrod–Patterson sparse quadratures are used to construct $f_{\rm r}(\boldsymbol{x})$ from which $\hat{Y}_{\alpha,{\rm r}}$ is determined from 1×10^6 MC samples and Eq. (25) is also used to determine the initial guesses from these MC samples. The convergences of $\hat{\boldsymbol{\xi}}_{\alpha,{\rm r}}$ errors for the full and sparse quadratures from $L = 3$ to 7 are shown with squared and circled lines in Fig. 7. The number of samples shown is the cumulative quadrature cost from all the refinements, *e.g.* cost at $L = 5$ is total of quadrature points from $L = 3$ to 5.

The convergences of the $\hat{\boldsymbol{\xi}}_{\alpha,{\rm r}}$ errors with increasing L for the full gPC metamodel is continuous but it is less so for the sparse quadrature cases. The error convergence depends on the accuracy of the y_α estimation as well as the metamodel resolution of $f(\boldsymbol{x})$ about the true design point. As the accuracies of the full and sparse $\hat{Y}_{\alpha,{\rm r}}$ are comparable, *cf.* Fig. 9, the irregular sparse quadrature convergence is likely due to the inaccuracy of $f_{\rm r}(\boldsymbol{x})$. The sparse quadrature has a lower ratio of off–axis quadrature points in comparison to the full quadrature and this likely influences the metamodel resolution near the design point which lies off the axes, *cf.* Fig. 2. With increasing α, the region of interest moves further away from the means of the inputs which leads to deterioration in the $f_{\rm r}(\boldsymbol{x})$ accuracy near the design point; therefore, for the same quadrature level, the accuracy of $\hat{\boldsymbol{\xi}}_{\alpha,{\rm r}}$ worsens with increasing α. For example, at the higher quantile $\alpha = 99.99\%$, the sparse quadrature fails to accurately estimate the design point at the lower quadrature levels. Although the normalized error in $\hat{\boldsymbol{\xi}}_{\alpha,{\rm r}}$ appears to be large at about $\mathcal{O}(10^{-1})$, the refinement zone of size $\mathcal{O}(1)$ is large enough to ensure that $\boldsymbol{\xi}_\alpha$ is included in the local support. In comparison to GPM, the gPC LMM can accurately estimate $\hat{\boldsymbol{\xi}}_{\alpha,{\rm r}}$ while avoiding the initial MC cost.

Multiple Peaks Multiple peaks could exist in the original function response or they could arise because of the transformation of correlated inputs to the standard normal space [8]. One advantage of the gPC LMM is that a single metamodel is used to locate all design points; in contrast several initial guesses are needed for GPM to ensure that all design points are found, which increases the sampling cost. In this section, $\hat{\boldsymbol{\xi}}_{\alpha,{\rm r}}$ convergence for multiple Gaussian–like peaks is examined for different dimensions. All the peak maxima are located at $\|\boldsymbol{\mu}_i\| = 2$ with the same width $\boldsymbol{\sigma}_i = (1, ..., 1)$, which makes the problem more difficult than if a peak is more probable, *i.e.* closer to $\mathbf{0}$. The location of the i–th peak, $\boldsymbol{\mu}_i$, is defined as $\mu_{i,n} = 2/\sqrt{N}$ for $n = 1...N\backslash\{i\}$ and $\mu_{i,n} = -2/\sqrt{N}$ if $n = i$. The latter definition ensures that the peaks are located in different quadrants. The pseudo exact design points $\boldsymbol{\xi}_{\alpha_i}$ are determined in the same manner as in the single peak case.

The average L^2–norm error convergences of $\hat{\boldsymbol{\xi}}_{\alpha_i,r}$ with respect to $\boldsymbol{\xi}_{\alpha_i}$ for $N = 3$ cases with double and trip peaks ($N_\alpha = 2$ or 3) are shown in Fig. 8. An error of 10 is assigned when the gPC LMM fails to locate

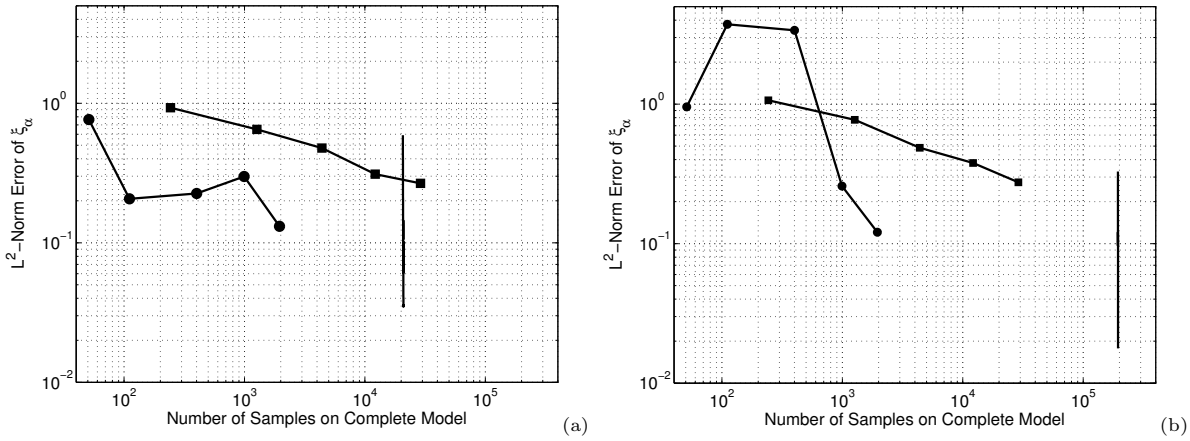

Figure 7: The L^2–norm error convergences for different design point search algorithms for $N = 5$ for $\alpha = 99.9\%$ and $\alpha = 99.99\%$. The squared and circled lines are the convergence from successive full and sparse quadratures refinement for the gPC estimations of $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$. Collocation levels from $L = 3$ and $L = 7$ are tested. The solid lines show the convergence in the design point search from GPM whose target quantile was determined from sufficient MC samples such that there is a 90% probability that y_α falls within $[\alpha \pm \epsilon]$–empirical quantile estimator where $\epsilon = (1 - \alpha)/2$.

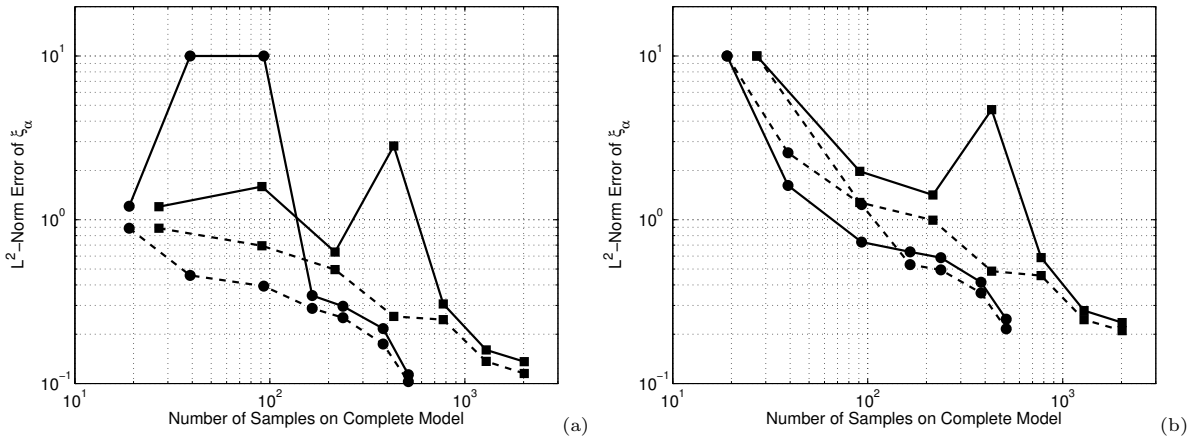

Figure 8: The L^2–norm error convergence of $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$ determined by the gPC LMM for $N = 3$ for (a) double and (b) triple peaks. The errors for $\alpha = 99.9\%$ and 99.99% are plotted in dashed and solid lines respectively. The squared and circled lines are the convergences from full and sparse quadratures from $L = 3$ and $L = 9$. When the algorithm fails to determine all the design points, an arbitrary error of 10 is assigned.

all the design points. Convergence results from $\alpha = 99.9\%$ and 99.99% are shown and the computations of the design points at the lower quantiles are more accurate in most cases. Although the sparse quadrature convergence is more irregular, it is in general more accurate than the full quadrature at the same number of samples. The same test is repeated for higher dimensions but the gPC LMM is not very accurate in locating multiple design points above $N = 4$. While the sparse quadrature fails to locate multiple design points for $N = 4$, full quadrature requires quadrature level of $L \geq 5$ ($Z_Q = 624$) and $L \geq 7$ ($Z_Q = 2401$) to locate double and triple Gaussian peaks, respectively. For low N, the gPC LMM is therefore efficient in locating multiple design points and the full metamodels give more consistent convergence in the $\hat{\boldsymbol{\xi}}_{\alpha_i,r}$ values, albeit at a slightly higher quadrature cost. The gPC LMM is also more efficient than GPM in locating multiple design points for $N \leq 4$. In GPM, the number of MC points required to estimate the initial $\hat{Y}_\alpha$ depends solely on the probability and size of the confidence interval; thus, it remains large even for low N. Moreover, multiple initial guesses are required to ensure that all the design points are located and these two factors

combined make GPM much less cost efficient than the gPC LMM. In addition, the gPC LMM design point estimators are more robust than these of GPM as the convergence of the GPM solutions are very sensitive to the algorithm's parameters whose optimal values are dependent on the test functions.

In the presence of multiple design points, der Kiureghian & Dakessian labelled the point closest to the origin as the global design point and the other ones as the local design points [8]. They suggest that the difficulties in multiple design point problems arise due to the misidentification of local design points as the global design point and the effects that the local design points may have on the global FORM/SORM results. In the context of the proposed method, both issues can be easily avoided. Firstly, the accuracy of the gPC metamodel worsens with increasing $\|\boldsymbol{x}\|$; therefore, the global design point with the smaller $\|\boldsymbol{x}\|$ will be resolved before the local design points whose $\|\boldsymbol{x}\|$ are smaller. Secondly, the separate local metamodels are constructed in different local domains about the respective estimated design points and they are not point refinements as in the case of FORM/SORM. Even if the local domain contains design points not identified by gPC LMM, their effects will be individually resolved by the local metamodel.

6.1.2. Quantile Estimation

Local metamodels are constructed at the design points found. The convergences of the L^2–norm errors of the α–quantile estimated from MC, IS and MEgPC approaches are computed with respect to the pseudo–exact value of y_α determined from $f(\boldsymbol{x})$ with 1×10^9 MC samples. The means from the MC and IS estimators $\hat{Y}_\alpha$ at selected Z determined from 500 independent series of simulations are shown in Fig. 9. The standard deviations are not shown but their magnitudes are roughly 25% of the means for both MC and IS. The sample distribution for the IS quantile is as follows: $Z/2$ are used to estimate the initial $\hat{Y}_\alpha$ from $f(\boldsymbol{x})$, at most $Z/4$ sample are used to determine $\hat{\boldsymbol{\xi}}_\alpha$ with GPM and the remaining samples are used for the importance sampling evaluation of Eq. (9). Although altering the sample distribution affects the accuracy of $\hat{Y}_\alpha$, we found that it is better to ensure that the initial $\hat{Y}_\alpha$ has a high accuracy. As expected, the convergence rate for the MC quantile is $\sim 1/\sqrt{Z_{\mathrm{MC}}}$. The accuracy of the IS method converges faster and is slightly more accurate than the MC method for high number of samples. The convergence of $\hat{Y}_{\alpha,\mathrm{r}}$ from the full and sparse global metamodels versus cumulative sampling costs are shown in squared and circled lines, respectively. The global $\hat{Y}_{\alpha,\mathrm{r}}$ are less accurate than both the MC and IS methods in all cases; clearly, the local refinement approach is needed and the accuracies of different local refinement approaches are examined in detail below.

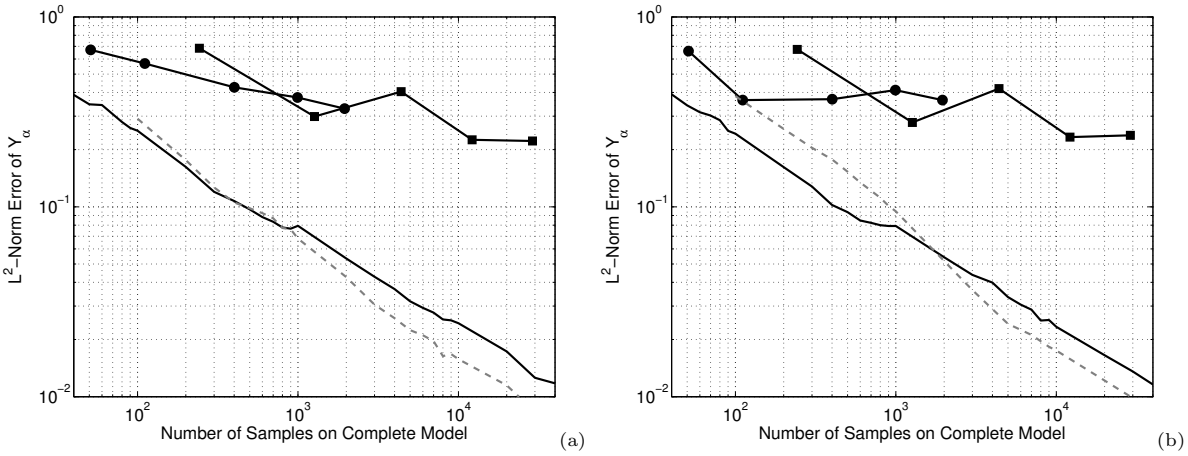

Figure 9: Convergences in the L^2–norm error of $\hat{Y}_\alpha$ in $N = 5$ for (a) $\alpha = 99.9\%$ and (b) 99.99%. The black line is the mean convergence of 500 series of Monte Carlo simulations. The gray line is the mean convergence of 500 series of importance sampling simulations whose design points are determined from the GPM. The squared and circled lines are the convergences of the global gPC $\hat{Y}_\alpha$ for the full and sparse quadratures, respectively.

Different Local Refinements Different types of local metamodels are compared against the MC convergence in Fig. 10. For clarity, only the MC convergences of $\hat{Y}_\alpha$ are shown. The accuracy of the sparse

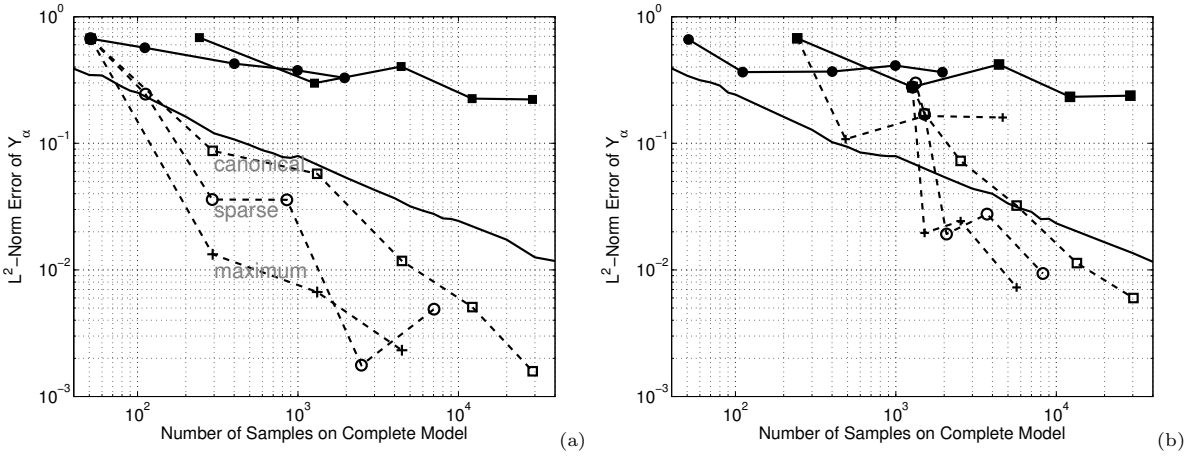

Figure 10: Details of the local gPC supplementary refinement for (a) $\alpha = 99.9\%$ and (b) $\alpha = 99.99\%$. The solid squared and circled lines are the convergences of the global gPC $\hat{Y}_\alpha$ with full and sparse quadratures. The dashed lines are the supplementary local refinements where the squares and circles represent the local metamodels constructed from full and sparse quadratures, respectively. In addition, the crosses represent the full quadrature case with maximum P expansion. The convergence of the MC $\hat{Y}_\alpha$ in the solid line is shown as a reference.

local metamodels is comparable to these of the full local metamodels; in the cases shown, the sparse $\hat{Y}_{\alpha,\mathrm{ME}}$ are more accurate than their full counterparts for the same quadrature cost. Moreover, the maximum P expansion from the full metamodel, *cf.* Sec. 3.1, increases the accuracy of $\hat{Y}_{\alpha,\mathrm{ME}}$ without additional sampling costs. The maximum P expansion increases the number of expansion terms significantly especially for high N; for example, at $P = 4$, M increases from 15 to 45 for $N = 2$ in contrast to $N = 5$ which increases from 252 to 3125. These additional expansion terms improve the accuracy of the gPC metamodel and $\hat{Y}_{\alpha,\mathrm{ME}}$. For $\alpha = 99.99\%$, only the design point from higher order full quadrature $\hat{Y}_{\alpha,\mathrm{r}}$ were sufficiently accurate to determine the locations of the local refinements. When the accuracies of estimators of global $\hat{Y}_{\alpha,\mathrm{r}}$ and $\hat{\boldsymbol{\xi}}_{\alpha_i,\mathrm{r}}$ are low, local refinement does not lead to significant improvement in the multi–element quantile estimator By comparing the convergences of the local sparse quadrature $\hat{Y}_{\alpha,\mathrm{ME}}$ (dashed line with crosses in Fig. 10(b)) from $L = 3$ and $L = 4$, it is clear that, for a given sample budget, it is better to maximize the accuracy of $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$ before local refinement is carried out.

6.1.3. Target Cost Study

In the following section, we compare the MEgPC against MC and IS from $N = 1$ to $N = 7$ for $\alpha = 99.9\%$ and $\alpha = 99.99\%$. A sample budget needs to be imposed in order to compare the different methods and we arbitrarily define a linear increase in the sample budget with N where $Z = 100N$. Different combinations of the global and local metamodels are also used to compute $\hat{Y}_{\alpha,\mathrm{ME}}$. Full and sparse global metamodels are denoted with solid and dashed lines with symbols; their full and sparse local metamodels are shown with crosses and circles, *i.e.* solid line with circles are full global metamodel with sparse local metamodel refinements. Only the maximum P expansion results are shown as they are more accurate than their canonical P expansion counterparts. The MC and IS accuracy estimators are determined from 500 independent series of simulations. In all cases, most of the sample budget is concentrated on the global metamodel before local refinement is carried out. In all cases, the MEgPC approach does not use all the sampling budget due to its fixed and deterministic quadrature costs. Further work is required to determine how the leftover points can be used to improve the accuracy of the quantile estimation.

For the given cost budget, the IS $\hat{Y}_\alpha$ is more accurate than the MC $\hat{Y}_\alpha$ at higher α and lower N. At higher N, the improvement from importance sampling is not significant because of the large number of MC samples. For $\alpha = 99.9\%$, the MEgPC $\hat{Y}_{\alpha,\mathrm{ME}}$ is more accurate than the MC and IS $\hat{Y}_\alpha$ in most cases and the accuracy of the full and sparse local metamodels are comparable. For the maximum P expansion

approach, the improvement in accuracy increases with higher N; this is likely due to the fact that additional expansion terms are included with higher N as demonstrated in the previous Section. The comparison of the accuracies in $\hat{Y}_{\alpha,\mathrm{ME}}$ from different methods also depend on the sample budget. For example, the 99.9%–quantile at $N = 6$ is estimated from 462 quadrature points, much smaller than the budget limit of 600; the next refinement, requiring 878 points, would give an error of 3.01×10^{-2}. For the $\alpha = 99.99\%$ case, the accuracy of the MEgPC approach becomes inferior to MC and IS above $N = 4$ for the sample budget imposed because of the global metamodel's inadequacy in locating the design points. For example, in the $N = 5$ case $\hat{Y}_{r,\mathrm{ME}}$ error is decreased to the order of $\mathcal{O}(10^{-2})$ if the sample budget is increased above 1000.

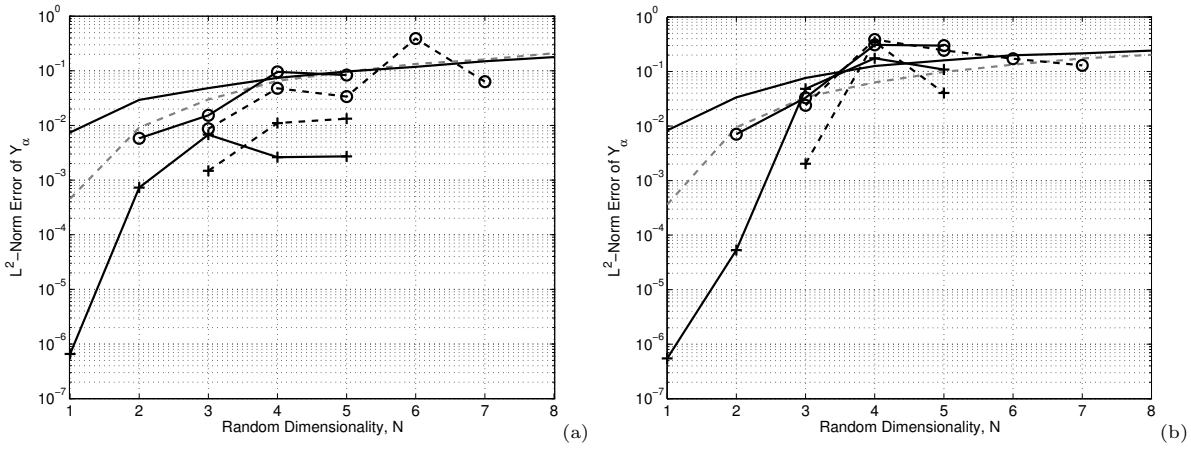

Figure 11: Quantile estimation as a function of N for different methods at (a) $\alpha = 99.9\%$ and (b) $\alpha = 99.99\%$. The solid black and dashed gray lines are the mean error convergences of 500 series of MC and IS simulations. The solid and dashed lines with symbols are multi–element results with full and sparse global metamodels. The symbols denote the supplementary local refinements where the crosses and circles represent the local metamodels constructed from full (maximum P expansion) and sparse quadratures, respectively.

6.2. Hypertangent function

The hypertangent function resembles closely a reliability response surface where two states of a system are connected by a continuous transition. Such a response is modelled by a hypertangent function as:

$$f(\boldsymbol{x}) = 1 + \tanh\left(\sum_{n=1}^{N} \sigma_n (x_n - \mu_n)\right), \tag{36}$$

where $\boldsymbol{\mu} = (\mu_n)_{n=1}^{N}$ controls the location of the ridge and $\boldsymbol{\sigma} = (\sigma_n)_{n=1}^{N}$ its orientation.

6.2.1. Design Point Search

The design points for $\alpha = 99\%$ and 99.9% and $N = 5$ are determined with GPM and the gPC LMM. The same methodology as in Sec. 6.1.1 is used. The convergences for the errors in the design point search for $\|\boldsymbol{\mu}\| = 2.5/\sqrt{N}$ and $\sigma_n = 1$ for $n = 1...N$ are shown in Fig. 12. The dominant cost in GPM is still the number of MC samples required to compute the initial estimate of $\hat{Y}_\alpha$. However, its convergence from its initial guess is rapid; due to the shape of the function, the design point can also be readily found using a simple steepest descend approach. For the gPC LMM $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$, the errors from four quadrature levels ($L = 3$ to 6) are shown for both the full and sparse metamodels. Better convergences are observed in the full quadrature case than the sparse case. As the hypertangent function is anisotropic with one dominant direction, the sparse quadrature provides a poorer function approximation, leading to the difference in $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$ convergence. However, the gPC LMM is more cost efficient than GPM in determining the design point; when the full quadrature is used, it is also more accurate for a given number of samples.

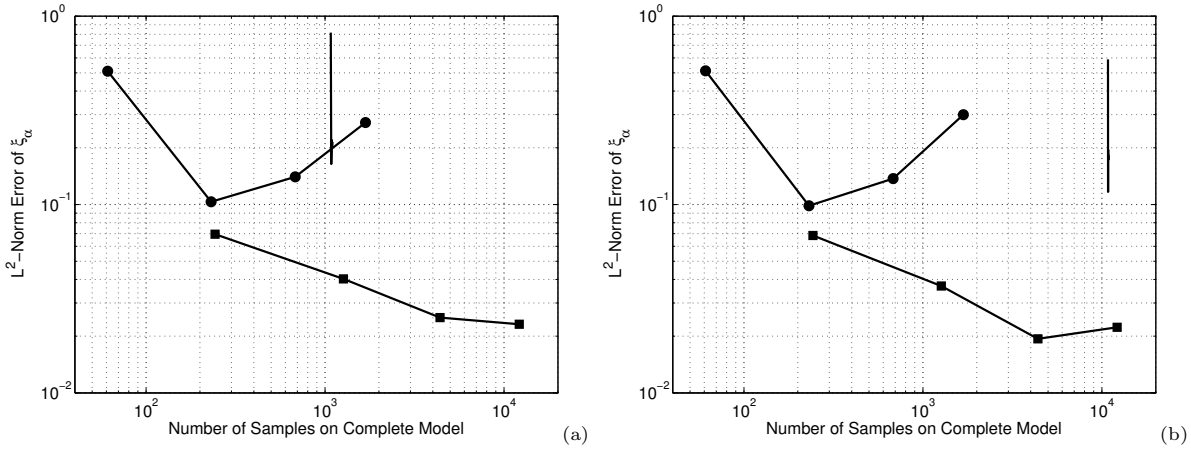

Figure 12: The L^2–norm error convergences for different design point search algorithms for $N = 5$ for $\alpha = 99\%$ and $\alpha = 99.9\%$. The squared and circled lines are the convergence from successive full and sparse quadratures refinement for the gPC estimations of $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$. Collocation levels from $L = 3$ and $L = 6$ are tested. The solid lines show the convergence in the design point search from GPM whose target quantile was determined from sufficent MC samples such that there is a 90% probability that y_α falls within $[\alpha \pm \epsilon]$–empirical quantile estimator where $\epsilon = (1 - \alpha)/2$.

6.2.2. Quantile Estimation

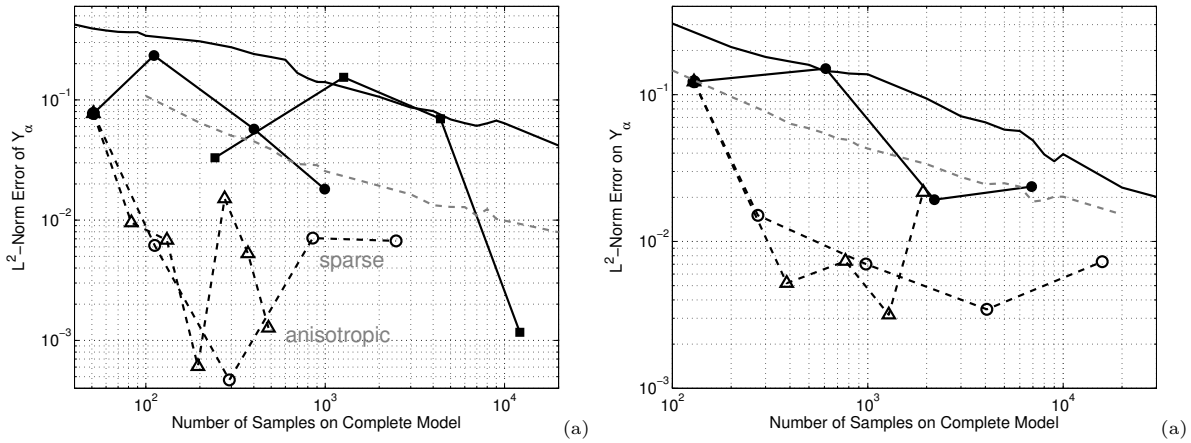

Figure 13: Convergences in the L^2–norm error of $\hat{Y}_\alpha$ in $N = 5$ and $N = 8$ for 99.99%. The solid black and gray dashed lines are the means of the $\hat{Y}_\alpha$ convergences determined from 100 series of MC and IS simulations. The squared and circled lines are the convergences of the global gPC $\hat{Y}_{\alpha,\mathrm{r}}$ for the full and sparse quadratures, respectively. The dashed lines are their supplementary local refinements where the triangles and circle represent the local metamodels constructed from anisotropic and sparse quadratures, respectively.

The convergences of the L^2–norm errors of 99.9% and 99.99%-empirical quantiles at $N = 5$ are compared in Fig. 13. Same $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ as the previous section are used and the same methodology in Sec. 6.1.2 is followed to compute the quantile estimators from the MC, IS and MEgPC approach. Only the means of the MC and IS error convergences are shown and their standard deviations are roughly 25% of the means for both MC and IS. At the sample size range examined, the IS quantiles are more accurate than their MC counterparts.

Quantile estimates $\hat{Y}_{\alpha,\mathrm{r}}$ are obtained from the gPC global metamodel and the $\alpha = 99.99\%$ results are shown for $N = 5$ and 8. Without local refinements, the values of $\hat{Y}_{\alpha,\mathrm{r}}$ are already more accurate than some of their MC and IS counterparts. For low quantile at $\alpha = 99\%$ and 99.9% (not shown), the global gPC $\hat{Y}_{\alpha,\mathrm{r}}$ are also more accurate than the MC and IS $\hat{Y}_\alpha$. Before the local metamodels are constructed, the one–dimensional

refinements, *cf.* Fig. 4.(b), are conducted about the design point and they reveal a strong dominance of the function response in the $\hat{\boldsymbol{\xi}}'_{\alpha,\mathrm{r}}$–direction. Such strong anisotropic response may be economically approximated by the anisotropic quadrature, *cf.* Sec. 2 and Fig. 2. As the sparse local metamodels have similar results to the anisotropic metamodels, only the full and anisotropic $\hat{Y}_{\alpha,\mathrm{ME}}$ are plotted in Fig. 13 and they show that both metamodels improve significantly the accuracy of the quantile estimators. The anisotropic $\hat{Y}_{\alpha,\mathrm{ME}}$ have convergence rates comparable to these of the full $\hat{Y}_{\alpha,\mathrm{ME}}$; however, the anisotropic $\hat{Y}_{\alpha,\mathrm{ME}}$ convergences are more irregular where oscillations and slight divergence can be observed in the $N = 8$ and $\alpha = 99.99\%$ case. Nevertheless, in all cases, local metamodels improve the quantile estimators especially for very large α and make the MEgPC approach more accurate than MC or IS.

6.2.3. Target Cost Study

We now compare the MEgPC against the MC and IS methods from $N = 1$ to $N = 8$ for $\alpha = 99.9\%$ and $\alpha = 99.99\%$. We impose the same arbitrary sample budget of $Z = 100N$ and the same methodology in Sec. 6.1.3 is followed. The normalized errors of MC and IS $\hat{Y}_\alpha$ are compared against those from MEgPC $\hat{Y}_{\alpha,\mathrm{ME}}$ in Fig. 14.

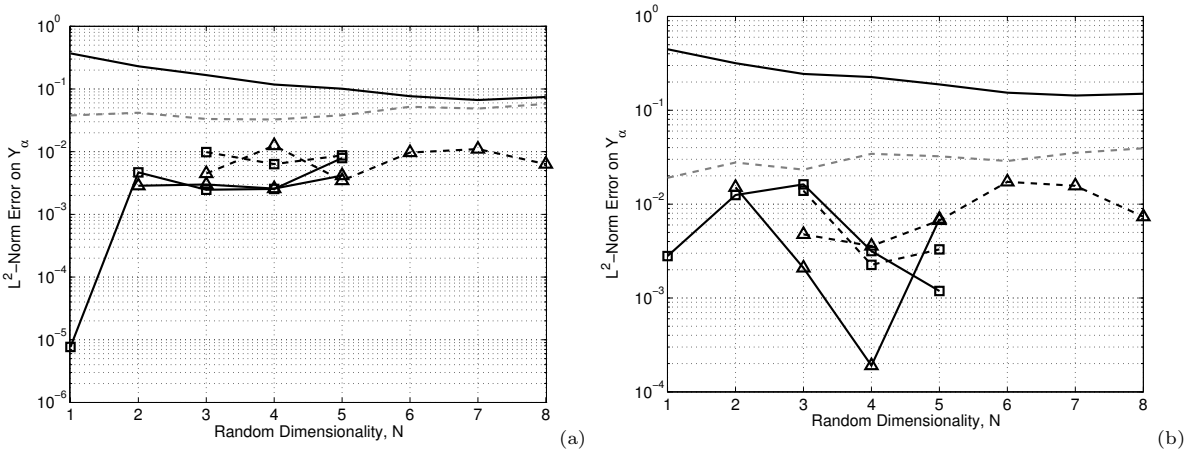

Figure 14: Quantile estimation as a function of N for different methods at (a) $\alpha = 99.9\%$ and (b) $\alpha = 99.99\%$. The solid black and dashed gray lines are the mean error convergences of 500 series of MC and IS simulations. The solid and dashed lines with symbols are multi–element results with full and sparse global metamodels. The symbols denote the supplementary local refinements where the squares and triangles represent the local metamodels constructed from full and anisotropic quadratures, respectively.

For $\alpha = 99\%$ (not shown), MC and IS have roughly the same accuracy between $N = 3$ to $N = 5$; with increasing α, IS $\hat{Y}_\alpha$ becomes more accurate than the MC $\hat{Y}_\alpha$. In the two quantiles shown, the IS $\hat{Y}_\alpha$ are more accurate than the MC $\hat{Y}_\alpha$, especially at lower N. In comparison, MEgPC $\hat{Y}_{\alpha,\mathrm{ME}}$ are more accurate than MC and IS $\hat{Y}_\alpha$ at all values of α tested. In particular, the anisotropic quadratures perform better in most cases than the sparse quadratures (not shown) which is in turn more accurate than the full quadratures. Having a low number of quadrature points allows local metamodel of higher quadrature level to be constructed; thus allowing one to better observe the convergence in the estimated quantile. In the case of anisotropic functions, the anisotropic quadrature is the most economical quadrature scheme to capture the function response.

6.3. Flooding function

We now consider a practical problem related to a risk of flood. The risk of flood can be defined as a first passage problem when the level of a watercourse exceeds the height of a dyke. While the water level H of a watercourse can be calculated with the Navier–Stokes equations using the appropriate boundary conditions, a simplified analytic model was introduced in [48]. The approximation involves the water course

flow Q, the water course bed friction K_s, the upstream water height Z_m, and downstream water height Z_v. The water course flow is treated as a random variable with Gumbel distribution, a typical approximation of extreme value theory in natural phenomenon. The water course bed friction is treated as a Gaussian random variable to account for the cumulative effects of different physical uncertainties. The triangular distribution are used to model the upstream and downstream water heights where the random variables are bounded with a maximum likelihood at a certain value. In addition, the watercourse length and width are taken as constant $L = 5000$ and $B = 300$, respectively. The watercourse height H can then be approximated using the closed formula

$$H = \left(\frac{Q}{K_s B \sqrt{\frac{Z_m - Z_v}{L}}} \right)^{3/5} . \tag{37}$$

The parameters of the random variables are listed in Table 1. These random variables are transformed into the standard normal space using the inverse cdf method.

Table 1: The parameters used to defined the random variables in the watercourse height equation (37)

Random variable	Distribution	Parameters	D_n
Q ($m^3 s^{-1}$)	Gumbel	μ=1013, β=558	6.46×10^{-1}
K_s ($m^{1/3} s^{-1}$)	Gaussian	μ=30, σ=7.5	1.88×10^{-2}
Z_v (m)	Triangular	a=49, b=50, c=51	3.83×10^{-3}
Z_m (m)	Triangular	a=54, b=55, c=56	3.83×10^{-3}

The quantiles of the output H are determined from 1×10^9 MC samples and the empirical quantiles from different methods are determined using the methodology outlined in Sec. 6.1.3. The convergences of the quantiles at $\alpha = 99.9\%$ and $\alpha = 99.99\%$ are shown in Fig. 15. The accuracy of IS $\hat{Y}_\alpha$ is better than the MC $\hat{Y}_\alpha$ for both cases. Since $N = 4$ is not high, the full quadrature is used to construct the global gPC metamodel and the accuracies of the global gPC $\hat{Y}_{\alpha,\mathrm{r}}$ are comparable to the IS results for both cases. The partial variances, D_n, are computed from the global quadratic gPC metamodel with Eq. (18) and are listed in Table 1. From the global metamodel, it can already be observed that the function global dependence in Q is strong and this observation is further verified locally through the one–dimensional refinement studies after the design point is identified. Therefore, the anisotropic quadrature is used to construct the local metamodel in addition to the full quadrature.

In all cases, the errors in $\hat{Y}_{\alpha,\mathrm{ME}}$ are significantly reduced with local refinement. The improvement from the full and anisotropic metamodels are comparable for a given number of samples although the convergence of the anisotropic quadrature is more irregular. Indeed the quantile convergence is similar to those observed in Fig. 13 and the watercourse height, H, has an anisotropic response. In addition to the results observed in the previous section, we can conclude that the MEgPC approach provides quantile estimators that are at least one order of magnitude more accurate than the MC or IS counterparts when the complete model has an anisotropic response.

7. Conclusion

A multi–element gPC (MEgPC) metamodel is used to estimate high quantiles of solutions of multivariate numerical models which are computationally expensive to evaluate. The multi–element metamodel consists of a global gPC metamodel in the standard normal domain and local gPC metamodels in bounded domains centered at design points corresponding to the quantile. The metamodels were constructed using a non–intrusive method where the numerical solver is sampled on fixed and deterministic quadrature points; full–tensor, anisotropic–tensor, and Smolyak sparse–tensor quadratures are tested. Three numerical models, namely a Gaussian–like function, a hypertangent function, and a flooding model are used to validate the

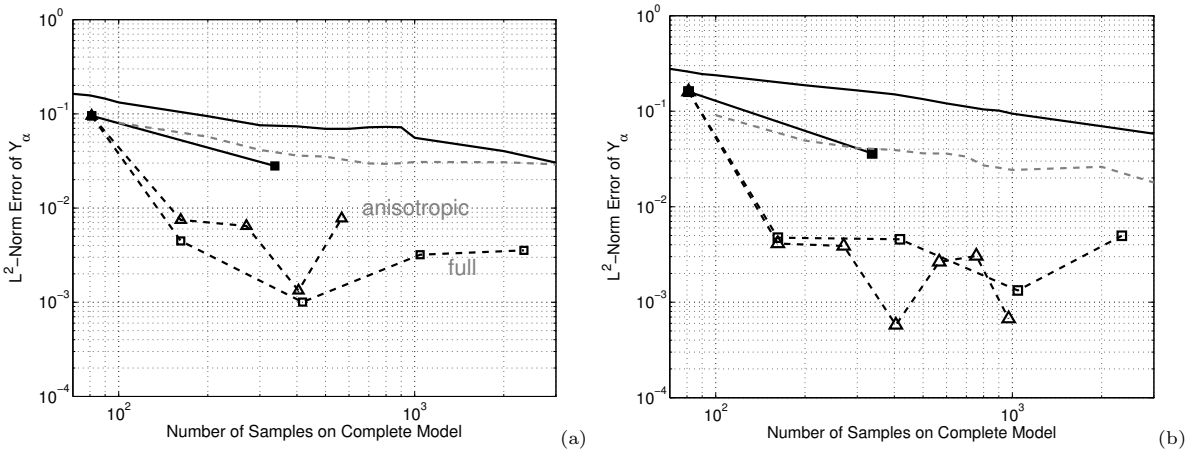

Figure 15: Convergence in the L^2–norm errors of $\hat{Y}_\alpha$ for (a) $\alpha = 99.9\%$ and (b) $\alpha = 99.99\%$. The solid black and dashed gray lines are the means of 100 series of MC and IS simulations. The squared lines are the convergences of the global gPC $\hat{Y}_{\alpha,\mathrm{r}}$ with full quadratures. The dashed lines are the supplementary local refinements where the squares and triangles represent the local metamodels constructed from full and anisotropic quadratures, respectively.

numerical method proposed in this study. The global gPC metamodel is used to estimate the initial α–quantile, $\hat{Y}_{\alpha,\mathrm{r}}$, and to determine the location of the design points $\hat{\boldsymbol{\xi}}_{\alpha,\mathrm{r}}$ with the Lagrange multiplier method. In comparison to the conventional approach where the initial quantile estimation and the subsequent design point search occur separately with costly evaluations of the complete model, the gPC Lagrange multiplier method accurately estimate the design points at a fraction of the computation cost. The gPC Lagrange multiplier method is also examined in numerical examples with multiple Gaussian–like peaks and is able to accurately locate all design points for random dimension $N \leq 4$. In all three test cases, the local gPC metamodels significantly improve the accuracy of the quantile estimators. For the Gaussian–like function, the maximum P expansion improves the accuracy of the gPC approximation with the conventional canonical P expansion when a full–tensor quadrature is used. Using one–dimensional metamodels constructed about the design points, it is readily revealed that the both the hypertangent function and the empirical flooding model have an anisotropic response in the bounded refinement domain near the design point. Consequently, an anisotropic quadrature scheme is used to reduce the quadrature cost; indeed the anisotropic results are comparable to their full–tensor or the sparse–tensor counterparts in both cases.

For all methods tested, the accuracy depends on the complete function response as well as the number of evaluations available on the complete model. Therefore, a target cost study is conducted to examine the effect of sampling budgets on the accuracy of the different methods for different types of complete model. The gPC metamodel cost depends on the number of quadrature points which is fixed and deterministic; therefore, the MEgPC approach may not use the entire sampling budget. In contrast, the entire sampling budget is used for the Monte Carlo and the importance sampling methods. Given a sample budget, it is better to allocate more sampling points to the global gPC metamodel such that the design points sought are as accurate as the cost constraint allows. For the Gaussian–like function, the MEgPC $\hat{Y}_{\alpha,\mathrm{ME}}$ is superior to the MC and IS $\hat{Y}_\alpha$ for $\alpha = 99.9\%$ and $\alpha = 99.99\%$ up to $N = 5$ and comparable up to $N = 7$. For the hypertangent function and the flooding model, the MEgPC approach is approximately one order of magnitude more accurate than the MC and IS α–quantile estimators for both $\alpha = 99.9\%$ and $\alpha = 99.99\%$ at all N values examined (from $N = 1$ to $N = 8$). In the context of reliability analysis, it can be asserted that the MEgPC approach proposed can significantly increase the accuracy of high quantile estimation, especially for functions are of the hypertangent type. Without additional complete function evaluations, the MEgPC metamodel can also be used for sensitivity analysis and parameter optimization of the complete model; these two additional features are not attainable with MC or IS.

References

References

[1] D. G. Robinson, A survey of probabilistic methods used in reliability, risk and uncertainty analysis: analytical techniques I, Technical Report SAND98-1189, Sandia National Laboratories, 1998.
[2] H. Madsen, S. Krenk, N. Lind, Methods of structural safety, Dover, Mineola, 2006.
[3] O. Ditlevsen, H. Madsen, Structural reliability methods, Wiley, 1996.
[4] R. Rackwitz, Reliability analysis - a review and some perspectives, Structural Safety 23 (2001) 365–395.
[5] R. Ghanem, P. Spanos, Stochastic Finite Elements: A Spectral Approach, Springer, Berlin, 1991.
[6] D. Xiu, D. Lucor, C.-H. Su, G. Karniadakis, Stochastic modeling of flow-structure interactions using Generalized Polynomial Chaos, Journal of Fluids Engineering 124 (2001) 51–59.
[7] J. Ko, D. Lucor, P. Sagaut, Sensitivity of two-dimensional spatially developing mixing layers with respect to uncertain inflow conditions, Physics of Fluids 20 (2008) 077102–20.
[8] A. der Kiureghian, T. Dakessian, Multiple design points in first and second-order reliability, Structural Safety 20 (1998) 37–49.
[9] X. Wang, G. E. Karniadakis, An adaptive multi-element generalized polynomial chaos method for stochastic differential equations, Journal of Computational Physics 209 (2005) 617–642.
[10] X. Wan, G. E. Karniadakis, Multi-element generalized polynomial chaos for arbitrary probability measures, SIAM Journal of Scientific Computing 28 (2006) 901–928.
[11] J. Foo, X. Wan, G. E. Karniadakis, The multi-element probabilistic collocation method (ME-PCM): Error analysis and applications, Journal of Computational Physics 227 (2008) 9572–9595.
[12] J. L. Meitour, D. Lucor, J.-C. Chassaing, Prediction of stochastic limit cycle oscillations using an adaptive polynomial chaos method, Journal of Aeroelasticity and Structural Dynamics 2 (2010) 3–22.
[13] M. Barbato, Finite element response sensitivity, probabilitic response and reliability analyses of structrual systems with applications to Earthquake, Ph.D. thesis, Department of Structural Engineering, University of California at San Diego, La Jolla, CA, USA, 2007.
[14] Q. Gu, Finite element response sensitivity and reliability analyses of soil-foundation structure-interaction systems, Ph.D. thesis, Department of Structural Engineering, University of California at San Diego, La Jolla, CA, USA, 2008.
[15] C. G. Bucher, U. Bourgund, A fast and efficient response surface approach for structural reliability problems, Structural Safety (1990) 57–66.
[16] M. Paffrath, U. Wever, Adapted polynomial chaos expansion for failure detection, Journal of Computational Physics 226 (2007) 263–281.
[17] M. M. Zuniga, Méthodes stochastiques pour l'estimation contrôlée de faibles probabilités sur des modèles physiques complexes. Application au domaine nucléaire., Ph.D. thesis, Université Paris VII, 2010.
[18] G. S. Fishman, Monte Carlo. Concepts, algorithms, and applications, Springer, New York, 1996.
[19] H. A. David, H. N. Nagaraja, Order Statistics, Wiley, Hoboken, 3rd edition, 2003.
[20] S. Wilks, Mathematical Statistics, John Wiley, New York, 1962.
[21] R. R. Bahadur, A note on quantiles in large samples, The Annals of Mathematical Statistics 37 (1966) 577–580.
[22] P. W. Glynn, Importance sampling for monte carlo estimation of quantiles, in: Proceedings of the Second International Workshop on Mathematical Methods in Stochastic Simulation and Experimental Design, pp. 180–185.
[23] S. Asmussen, P. W. Glynn, Stochastic simulation, algorithms and analysis, New York, 2007.
[24] N. Wiener, The homogeneous chaos, American Journal of Mathematics 60 (1938) 897–936.
[25] R. Cameron, W. Martin, The orthogonal development of nonlinear functionals in series of Fourier–Hermite functionals, Annals of Mathematics 48 (1947) 385–392.
[26] D. Xiu, G. E. Karniadakis, The Wiener-Askey polynomial chaos for stochastic differential equations, SIAM Journal of Scientific Computing 24 (2002) 691–644.
[27] R. Askey, J. Wilson, Some basic hypergeometric orthogonal polynomials that generalize Jacobi polynomials, volume 54, American Mathematics Society, Providence, Rhode Island, 1985.
[28] W. Gautschi, Construction of gauss-christoffel quadrature formulas, Math. Comp. 22 (1968) 251–270.
[29] I. Babuška, F. Nobile, R. Tempone, A stochastic collocation method for elliptic partial differential equations with random input data, SIAM J. Numer. Anal. 45 (2007) 1005–1034.
[30] S. A. Smolyak, Interpolation and quadrature formulas for tensor products of certain classes of functions, Dokl. Akad. Nauk SSSR 4 (1963) 240–243.
[31] T. Gerstner, M. Griebel, Numerical integration using sparse grids, Numerical Algorithms 118 (1998) 209–232.
[32] C. W. Clenshaw, A. R. Curtis, A method for numerical integration on an automatic computer, Numerische Mathematik 2 (1960) 197–205.
[33] A. S. Kronrod, Nodes and Weights of Quadrature Formulas, Consultants Bureau, New York, 1965.
[34] T. N. L. Patterson, The optimal addition of points to quadrature formulae, Mathematics of Computation 22 (1968) 847–856.
[35] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. Flannery, Numerical Recipes in C, Cambridge University Press, Cambridge, 2002.
[36] A. Genz, B. D. Keister, Fully symmetric interpolatory rules for multiple integrals over infinite regions with Gaussian weight, Journal of Computational and Applied Mathematics 71 (1996) 299–309.

[37] F. Nobile, R. Tempone, C. G. Webster, A sparse grid stochastic collocation method for partial differential equations with random input data, SIAM Journal on Numerical Analysis 46 (2008) 2309–2345.
[38] J. Ko, Applications of the generalized polynomial chaos to the numerical simulations of stochastic shear flow, Ph.D. thesis, Université Pierre et Marie Curie, 2009.
[39] I. M. Sobol', Sensitivity estimate for non-linear mathematical models, Mathematical modelling and computational experiments 1 (1993) 407–414.
[40] M. Rosenblatt, Remarks on a multivariate transformation, The Annals of Mathematical Statistics 23 (1952) 470–472.
[41] A. Nataf, Détermination des distributions dont les marges sont données, Comptes rendus de l'Académie des Sciences de Paris 255 (1962) 42–43.
[42] J. E. Gentle, Random Number Generation and Monte Carlo Methods, Springer-Verlag, New York, 2003.
[43] T.-J. Yao, Y. Wen, Response Surface Method for Time-Variant Reliability Analysis, Technical Report, University of Illinois Engineering Experiment Station. College of Engineering. University of Illinois at Urbana-Champaign, 1993.
[44] K. M. Carley, N. Y. Kamneva, J. Reminga, Response Surface Methodology: CASOS Technical Report, Technical Report, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, 2004.
[45] P.-L. Liu, A. der Kiureghian, Optimization algorithms for structural reliability, Structural Safety (1991) 161–177.
[46] D. G. Luenberger, Y. Ye, Linear and Nonlinear Programming, Springer, New York, 3 edition, 2008.
[47] B. Sudret, Metamodels for rare event probability calculation, in: OPUS Workshop 4 "Uncertainty propagation, rare quantile and extreme failure proability estimation".
[48] M. M. Zuniga, J. Garnier, E. Remy, E. de Rocquigny, Adaptative directional stratification for controlled estimation of the probability of a rare event, Reliability Engineering & System Safety 96 (2011) 1691–1712.