Available online at www.eccomasproceedia.org
Eccomas Proceedia UNCECOMP (2023) 165-182

ECCOMAS
Proceedia

UNCECOMP 2023
5th ECCOMAS Thematic Conference on
Uncertainty Quantification in Computational Science and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou (eds.)
Athens, Greece, 12-14 June 2023

MULTI-OUTPUT GAUSSIAN PROCESS SURROGATE MODELS FOR INVERSE UNCERTAINTY QUANTIFICATION IN RANDOM NEUTRONICS

Paul Lartaud[1,2], Philippe Humbert[2], and Josselin Garnier[1]

[1] Centre de Mathématiques Appliquées, Ecole Polytechnique, Institut Polytechnique de Paris
91128 Palaiseau Cedex, France
e-mail: {paul.lartaud,josselin.garnier}@polytechnique.edu

[2] CEA, DAM, DIF
F-91297, Arpajon, France
e-mail: philippe.humbert@cea.fr

Abstract. *Neutron noise analysis describes a set of techniques used in nuclear safeguards and nuclear security to identify an unknown fissile material based on the temporal correlations of the detected neutrons. The observations are often noisy due to the stochastic nature of the underlying physical processes, which makes the resolution of this inverse problem complex. Moreover, the uncertainty quantification of the estimation of the fissile material parameters has not been studied thoroughly, though it is crucial for such an ill-posed inverse problem. On top of this, the analytical direct model widely used to describe neutron correlations is based on strong physical assumptions, which are never met in a real-world scenario. Overall, the knowledge of the quality of the predictions is crucial for the decision makers.*

This work addresses dual objectives. First of all, surrogate models specifically designed for multiple correlated outputs are built to extend and improve the biased analytical model currently used, while providing a reliable quantification of the errors associated with the predictions. The surrogate models are based on multi-fidelity, multi-output Gaussian process regression. Different approaches are investigated to build the surrogate models. Their performance are evaluated with various metrics, and the coverage probabilities are estimated to guarantee the reliability of the predictions and their uncertainty quantification.

Finally, the inverse problem is solved with a Bayesian approach which takes into account the covariance of the measurements and the predictive means and covariances of the Gaussian process surrogate model. Thus, the general methodology presented in this work accounts for both sources of error, measurement noise and model bias. The resulting posterior distribution more accurately reflects the inferred knowledge about the material properties. This method is applied to a specific test case coming from a neutron multiplicity benchmark.

Keywords: Surrogate modeling, inverse problem, uncertainty quantification, Gaussian processes

ISSN:2623-3339
Peer-review under responsibility of the organizing committee of UNCECOMP 2023.
doi: 10.7712/120223.10332.19926

1 INTRODUCTION

This work falls within the scope of nuclear safeguards and non-proliferation. The general goal is to identify an unknown nuclear material based on passive, non-destructive measurements of neutron correlations [1]. More specifically, due to the strong non-linearity of the direct model, linking the material characteristics and the neutron correlation observations, the inverse problem that is considered here is ill-posed in the sense of Hadamard. Hence, it is essential to quantify the uncertainties in the resolution of the inverse problem and to provide credible regions to the decision makers, instead of a single point estimate.

In this work, the inverse problem is solved by Bayesian inference with the help of a Gaussian process surrogate model, to account for both the model error and the measurement error [2]. For this specific application, the outputs are strongly correlated. Thus we would like to build a multi-output surrogate model with improved mean predictions compared to the approximated analytical model, and reliable covariance predictions.
Different methods such as the Linear Coregionalization Model [3] and Bayesian Polynomial Chaos Expansion are investigated to build the surrogate models and their respective performance and reliability are quantified.

2 NEUTRON CORRELATIONS

2.1 The direct model

In this paragraph, a brief description of neutron correlation measurements is provided and the construction of the dataset used for the surrogate models training is detailed. Neutron noise analysis is a non destructive method allowing to infer information from a nuclear material by measuring neutron correlations. Indeed, in a fissile material, neutrons are created by different nuclear reactions, and most notably by induced fissions. Each induced fission can yield up to 10 neutrons. This neutron multiplicity affects the statistics of the counts in the neutron detector. The quantities of interest in our neutron correlation measurements are the number of multiple detections. A multiple detection refers to the simultaneous detection of multiple neutrons, within a given time window of a few milliseconds. These quantities are linked to the binomial moments of the detection statistics. Böhnel and Cifarelli [4, 5] showed that the neutron multiplication of a fissile medium could be determined by the measures of the three first moments of the detection statistics. This method is detailed in [6] and derived by Paszit in [7].

The details of the theory behind neutron noise analysis are not provided in this work. For the reader, the main aspects to keep in mind is that we are considering three distinct observations for a single neutron correlation measurement.

- The average count rate R of neutrons in the detector.
- The second asymptotic Feynman moment Y_∞.
- The third asymptotic Feynman moment X_∞.

The second and third asymptotic Feynman moments describe the average number of double and triple correlated detections. A distinction is made between accidental correlations, for which the neutrons hit the detector at the same time by chance, and are independent from one another (i.e. come from independent chains of fission), and correlated detections where the neutrons are correlated in time due to fissions in the material.

The direct model most commonly used to describe neutron correlations is the so-called point model f_p which is based on strong assumptions. It provides a link between some material characteristics x and the observations $y = (R, Y_\infty, X_\infty) = f_p(x)$. However, this direct model is strongly non-linear and inherently biased due to its underlying assumptions. That is why we would like to improve it using a surrogate model based on supervised learning and more specifically on Gaussian process regression. Let us first describe how the dataset used for training the surrogate models is created.

2.2 Creation of the dataset

Our goal here is to create a dataset from which the surrogate model can learn the direct model. This is done with the help of the Monte-Carlo neutronic code MCNP6 [8]. This code describes the full neutronic transport problem. It cannot be used as a direct model in a Bayesian inverse problem framework because of its prohibitive cost. A single run for a simple geometry can take up to roughly twenty minutes. To create the dataset, 1125 simulations are performed with MCNP6. The geometry of the problem is the following. We consider an unknown spherical object containing a mix of ^{239}Pu and ^{240}Pu metallic. The sphere is surrounded by a material containing a neutron absorber B_4C, and a neutron moderator made of polyethylene which slows down the neutron to the low energy range, where fissions occur. A cylindrical neutron detector containing ^{3}He with a polyethylene layer is placed at a given distance of the object. Finally, a thick concrete layer surrounds the geometry. For each simulation, the distances, compositions and even densities of the materials are changed randomly. Each time, neutron correlations measurements are obtained and the inputs x of the direct model are evaluated. In the point model, four inputs are considered :

- k_p the prompt multiplication factor
- ε_F the ratio of the number of detections over induced fissions.
- S the source intensity in neutrons per second.
- x_s the number of source neutrons obtained by spontaneous fissions over the total number of source neutrons.

The point model is inherently biased because the full transport problem is summarized to four parameters. Three more parameters are added to the dataset in order to provide more information to the surrogate model.

- Φ the ratio of fast over thermal neutron flux.
- ε_A the ratio of the number of parasitic captures in the moderator over induced fissions.
- J the ratio of out-going over in-going neutron current at the outer layer of the geometry.

The dimension of the input vector x is 7 and the dimension of the output vector y is 3. The dataset contains 1125 input-output pairs. In a preprocessing step, the dataset is transformed with the Box-Cox transform [9] to improve the training of the surrogate model.

3 INVERSE PROBLEM

3.1 Bayesian resolution

Our goal is to solve the inverse problem with Bayesian inference [10]. Given N independent observations of $D = 3$ outputs $\mathbf{y} = (y^{(i)})_{1 \leq i \leq N}$ with independent Gaussian noise on each observation :

$$y^{(i)} = f_p(x) + \epsilon^{(i)} \text{ with } \epsilon^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_m) \tag{1}$$

we would like to find the posterior distribution $p(x|\mathbf{y})$. The 3×3 covariance matrix $\mathbf{C}_m$ of the measurement error is assumed known. In fact, it can be estimated by a bootstrap method once the observations are provided [11]. Then the posterior distribution can be obtained up to a multiplicative constant:

$$p(x|\mathbf{y}) \propto p(x)L(\mathbf{y}|x) \tag{2}$$

where $p(x)$ is the prior distribution which quantifies the information on the parameters prior to the measurement, and $L(\mathbf{y}|x)$ is the likelihood given by :

$$L(\mathbf{y}|x) \propto \exp\left(-\frac{1}{2}\sum_{i=1}^{N}(y_i - f_p(x))^T\mathbf{C}_m^{-1}(y_i - f_p(x))\right) \tag{3}$$

The problem with this formulation is that the observations are assumed to follow the point model. Only the uncertainties on the observations are taken into account in the inverse problem. This is especially problematic for such an ill-posed inverse problem where a small error in the direct model can lead to dramatic errors in the inverse problem resolution. Thus our objective is dual in this work. We would like to build a surrogate model which should provide a better description of the physical processes, and reliable covariance predictions, such that the model error can be included in the likelihood during the resolution of the inverse problem.

3.2 Surrogate models and Bayesian inference

Let us assume from now on that we are provided with a surrogate model of the direct model, which is able to provide a predictive mean $\overline{f_s(x)} \in \mathbb{R}^D$ of the $D = 3$ outputs and quantify the associated error $\eta(x)$ at any given point x. Then the observations can be written as :

$$\mathbf{y} = \overline{\mathbf{f_s}(x)} + \boldsymbol{\eta}(x) + \boldsymbol{\epsilon} \tag{4}$$

where $\overline{\mathbf{f_s}(x)} = \left(\overline{f_s(x)}, ..., \overline{f_s(x)}\right)^T \in \mathbb{R}^{DN}$ is the flattened vector of the surrogate model mean predictions, $\boldsymbol{\eta}(x) = (\eta(x), ..., \eta(x))^T \in \mathbb{R}^{DN}$ is the flattened vector of the model errors and the measurement error is $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_{\text{obs}})$ where $\mathbf{C}_{\text{obs}}$ is a block diagonal matrix.

$$\mathbf{C}_{\text{obs}} = \begin{pmatrix} \mathbf{C}_m & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{C}_m \end{pmatrix} \in \mathbb{R}^{DN \times DN} \tag{5}$$

In this work, we consider Gaussian processes surrogate models. By definition, the multi-output Gaussian process provides at any given point x a predictive mean $\overline{f_s(x)}$ and a predictive covariance $\mathbf{C}_s(x)$ such that the predictive distribution $f_s(x)$ is Gaussian.

$$f_s(x) \sim \mathcal{N}\left(\overline{f_s(x)}, \mathbf{C}_s(x)\right) \tag{6}$$

With this specific framework, the model error $\boldsymbol{\eta}(x)$ in equation 4 follows a zero-mean Gaussian distribution $\boldsymbol{\eta}(x) \sim \mathcal{N}\left(0, \mathbf{C}_{\text{mod}}(x)\right)$.

$$\mathbf{C}_{\text{mod}}(x) = \begin{pmatrix} \mathbf{C}_s(x) & \dots & \mathbf{C}_s(x) \\ \vdots & \ddots & \vdots \\ \mathbf{C}_s(x) & \dots & \mathbf{C}_s(x) \end{pmatrix} \in \mathbb{R}^{DN \times DN} \tag{7}$$

This model leads to a modified likelihood which accounts for both the model error and the noise error.

$$L(\mathbf{y}|x) = (2\pi)^{-DN/2} |\mathbf{C}_{\text{tot}}(x)|^{-1/2} \exp\left(-\frac{1}{2} \left(\mathbf{y} - \overline{\mathbf{f_s}(x)} \right)^T \mathbf{C}_{\text{tot}}(x)^{-1} \left(\mathbf{y} - \overline{\mathbf{f_s}(x)} \right) \right) \tag{8}$$

where $\mathbf{C}_{\text{tot}}(x) = \mathbf{C}_{\text{mod}}(x) + \mathbf{C}_{\text{obs}}$.
The posterior distribution can be sampled with the help of Monte-Carlo Markov Chain (MCMC) methods. In section 5.2, a detailed discussion on these MCMC methods is presented.

4 GAUSSIAN PROCESSES SURROGATE MODELS

4.1 Introduction to Gaussian process regression

In this section we describe briefly the principle of Gaussian process regression for a single scalar output[12]. A Gaussian process is a collection of random variables, such that any finite subset follows a multivariate normal distribution. The distribution of a Gaussian process is completely determined by its mean function $m(\cdot)$ and covariance function $k(\cdot, \cdot)$. If f is a Gaussian process with mean function m and covariance function k, it is denoted :

$$f \sim \mathcal{GP}\left(m(x), k(x, x')\right) \tag{9}$$

The covariance function k defines the regularity of the functions sampled from the Gaussian process. Different families of covariance functions exist and are used depending on the expected shape of the function to be learnt. In this work, Matérn covariance kernels are used.

Let us consider a known set of input-output pairs $(\mathbf{X}, \mathbf{f})$. A nugget noise σ^2 is added to account for the noise in the training data. The predictive distribution for the new outputs $\mathbf{f}_*$ associated with the inputs $\mathbf{X}_*$ and given $\mathbf{X}$ and $\mathbf{f}$ is Gaussian and given by :

$$\begin{aligned} \mathbf{f}_* | \mathbf{f}, \mathbf{X}, \mathbf{X}_* &\sim \mathcal{N}\left(\mu_C, K_C\right) \\ \mu_C &= K(\mathbf{X}, \mathbf{X}_*)^T \left(K(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathcal{I}_N\right)^{-1} \mathbf{f} \\ K_C &= K(\mathbf{X}_*, \mathbf{X}_*) - K(\mathbf{X}, \mathbf{X}_*)^T \left(K(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathcal{I}_N\right)^{-1} K(\mathbf{X}, \mathbf{X}_*) \end{aligned} \tag{10}$$

This conditional distribution provides a way to predict the mean output from any given input points $\mathbf{X}_*$ as well as the covariance. The main interest of Gaussian process regression for our application is its ability to quantify the uncertainty on the predictions.

In order to provide reasonable predictions for regression or classification problems, a Gaussian process has to be trained. The training phase consists of optimizing the choice of the hyperparameters in the covariance kernels based on the training data. The common practice for optimizing the hyperparameters is to find the values that maximize the marginal likelihood $p(\mathbf{f}|\mathbf{X})$. The log-marginal likelihood is optimized using common optimization algorithms. In our case, the limited memory Broyden–Fletcher–Goldfarb–Shanno algorithm for bound constraint (also known as L-BFGS-B) is used [13]. The optimization algorithm is restarted 10

times with different initial values for the hyperparameters. The optimal set of hyperparameters chosen is the one that provide the highest log-marginal likelihood of the 10 iterations. With this approach, the risk of being stuck in a local optimum is reduced. Once the optimal set of hyperparameters is found, predictions can be made using equations 10.

In this work however, we would like to build a surrogate model with $D = 3$ different outputs. The first naive approach would be to build three independent surrogate models, one for each output. However, this approach neglects the correlations between the outputs. For our particular task, the outputs are strongly correlated especially the second and third Feynman moments Y_∞ and X_∞. More complex approaches are described in the next paragraphs.

4.2 Linear Model of Coregionalization

In order to build a multi-output covariance kernel, one possible method is to create independent scalar Gaussian processes and mix them afterwards with a transition matrix chosen to be definite positive [3]. With this approach, it is possible to correlate the output channels while maintaining a positive definite covariance structure. A brief description of this method is presented hereafter. Let us consider Q independent zero-mean scalar Gaussian processes, which will be referred to as the latent Gaussian processes:

$$u_q \sim \mathcal{GP}\left(0, k_q(x, x')\right) \tag{11}$$

Now let us consider a real mixing matrix $W \in \mathcal{M}_{D,Q}(\mathbb{R})$. For $x \in \mathbb{R}^I$, let f_d be the output for channel d built by the following relation :

$$f_d(x) = \sum_{q=1}^{Q} w_{d,q} u_q(x) \tag{12}$$

Each output channel is a linear combination of independent Gaussian process. With this definition, for a collection of n inputs $\mathbf{X} = (X^{(k)})_{k \leq n}$, the flattened output vector follows a multivariate normal distribution.

$$\begin{pmatrix} f_1(X^{(1)}) \\ \vdots \\ f_1(X^{(n)}) \\ \vdots \\ f_D(X^{(1)}) \\ \vdots \\ f_D(X^{(n)}) \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \sum_{q=1}^{Q} \begin{pmatrix} w_{1,q}w_{1,q}\mathbf{K}_q(\mathbf{X},\mathbf{X}) & \dots & w_{1,q}w_{D,q}\mathbf{K}_q(\mathbf{X},\mathbf{X}) \\ \vdots & \ddots & \vdots \\ w_{D,q}w_{1,q}\mathbf{K}_q(\mathbf{X},\mathbf{X}) & \dots & w_{D,q}w_{D,q}\mathbf{K}_q(\mathbf{X},\mathbf{X}) \end{pmatrix}\right)$$

where $w_{d,q}$ is the element of the matrix W on the d-th row and q-th column, and $\mathbf{K}_q(\mathbf{X}, \mathbf{X})$ is the covariance matrix of the inputs $\mathbf{X} = (X^{(k)})_{k \leq n}$ obtained with the covariance kernel k_q. It is also possible to write the full covariance matrix using the Kronecker product $\otimes$:

$$\mathbf{K}_{\text{LMC}}(\mathbf{X}, \mathbf{X}) = \sum_{q=1}^{Q} WW^T \otimes \mathbf{K}_q(\mathbf{X}, \mathbf{X}) \tag{13}$$

Similarly, the covariance matrix between two collections of inputs $\mathbf{X}$ and $\mathbf{X}_*$ is given by :

$$\mathbf{K}_{\text{LMC}}(\mathbf{X}, \mathbf{X}_*) = \sum_{q=1}^{Q} WW^T \otimes \mathbf{K}_q(\mathbf{X}, \mathbf{X}_*) \tag{14}$$

With this method, it is possible to build a multi-output Gaussian process able to account for correlations between the outputs. The training phase is similar to scalar GP, except that the inversion of the covariance matrix $\mathbf{K}_{\text{LMC}}(\mathbf{X}, \mathbf{X})$ now has a complexity of $O(D^3n^3)$ instead of $O(n^3)$ for a training set $(\mathbf{X}, \mathbf{f})$ containing n samples. For low output dimension, and not too large datasets, the exact GP training can be carried out but otherwise approximate methods such as sparse variational Gaussian processes may be required [14].

4.3 Bayesian Polynomial Chaos Expansion

In this section, we will derive a new surrogate model based on a Bayesian framework of the polynomial chaos expansion. Let us first introduce briefly the standard polynomial chaos expansion.

4.3.1 Polynomial Chaos Expansion

Let $X = (X_i)_{1 \leq i \leq I}$ be a random variable representing the uncertain input, and such that the X_i are independent and have marginal probability distributions μ_i. Let $(p_{\alpha_i}^{(i)})_{\alpha_i \in \mathbb{N}}$ be a family of polynomials orthogonal for the measure μ_i. Let us consider the output $Y = f(X)$ where $f : \mathbb{R}^I \mapsto \mathbb{R}$ and $\mathbb{E}\left[f(X)^2\right] < +\infty$.

Then the tensorized polynomials defined by $p_\alpha(x) = \prod_{i=1}^{I} p_{\alpha_i}^{(i)}(x_i)$ for $x \in \mathbb{R}^I$ form a strongly orthogonal family of multivariate polynomials [15]. The family or orthogonal polynomials is a basis of the functional Hilbert space $L^2(\mu)$ for the product measure μ defined by $d\mu(x) = d\mu_1(x_1)...d\mu_I(x_I)$, and $f \in L^2(\mu)$. It is thus possible to expand the output Y on the polynomial basis :

$$Y = \sum_{\alpha \in \mathbb{N}^I} y_\alpha p_\alpha(X) \tag{15}$$

The strong orthogonal property simplifies the expression of the coefficients, which can be obtained by orthogonal projection :

$$y_\alpha = \frac{\mathbb{E}\left[f(X)p_\alpha(X)\right]}{\mathbb{E}\left[p_\alpha(X)^2\right]} \tag{16}$$

For scalar inputs, the polynomial expansion is truncated to a given degree to build a surrogate model.

In the case of multi-dimensional inputs, we are dealing with multi-indices α and the truncation is the selection of a subset of all the possible multi-indices. The first intuition that comes to mind is to limit the development up to a certain degree of the multivariate polynomial α_m such that $\alpha \in \mathcal{D}_{\alpha_m} = \{\alpha \in \mathbb{N}^I, |\alpha| \leq \alpha_m\}$, where $|\alpha|$ is the order of the multi-index defined by $|\alpha| = \sum_{i=1}^{I} \alpha_i$.

However, when the number of dimension increases, the number of retained multi-indices dramatically explodes. Indeed $\text{card}(\mathcal{D}_{\alpha_m}) = \frac{(I+\alpha_m)!}{I!\alpha_{m]!}}$. As an illustration, for $I = 7$ and $\alpha_m = 10$,

card$(\mathcal{D}_{\alpha_m}) = 19448$. Hence, the number of multi-indices must be reduced. One of the most common techniques is to consider that the most impactful polynomials are the univariate polynomials of high degrees, and the multivariate of low degrees. This corresponds to the hyperbolic truncation scheme [16]. This assumption is reflected by considering the subset $\mathcal{A}_{\alpha_m,r}$ defined for $r \in (0,1]$ by :

$$\mathcal{A}_{\alpha_m,r} = \left\{ \alpha \in \mathbb{N}^I, \left(\sum_{i=1}^{I} \alpha_i^r \right)^{1/r} \leq \alpha_m \right\} \tag{17}$$

When $r = 1$, $\mathcal{A}_{\alpha_m,1} = \mathcal{D}_{\alpha_m}$ and for $r < 1$ the subset $\mathcal{A}_{\alpha_m,r}$ filters out the multivariate polynomials of high degree, which reduces the number of coefficients to estimate. Based on this subset, it is possible to build a surrogate model by truncating the expansion in equation 15 to the subset $\mathcal{A}_{\alpha_m,r}$:

$$Y \simeq Y_{\alpha_m,r} = \sum_{\alpha \in \mathcal{A}_{\alpha_m,r}} y_\alpha p_\alpha(X) \tag{18}$$

4.3.2 Estimation of the coefficients

The coefficients can be estimated by regression or by Monte-Carlo. The first approach is used in this work. Consider a given truncation $\mathcal{A}_{\alpha_m,r}$, with $P = \text{Card}(\mathcal{A}_{\alpha_m,r})$. A particular order of the subset of multi-indices is chosen such that $\mathcal{A}_{\alpha_m,r} = (\alpha_p)_{1 \leq p \leq P}$. The objective is to find the coefficients which minimize the error between $Y_{\alpha_m,r}$ and the true output $Y = f(X)$. Let $\mathbf{p} = (p_\alpha)_{\alpha \in \mathcal{A}_{\alpha_m,r}}$. The coefficients $\widehat{\mathbf{c}} = (\widehat{c_\alpha})_{\alpha \in \mathcal{A}_{\alpha_m,r}}$ verifying this are given by :

$$\widehat{\mathbf{c}} = \text{argmin}_{\mathbf{c} \in \mathbb{R}^P} \, \mathbb{E}\left[\left(\mathbf{c}^T \mathbf{p}(X) - f(X) \right)^2 \right] \tag{19}$$

The true mean is unknown but can be approximated assuming we have n independent samples $\mathbf{X} = (X^{(k)})_{1 \leq k \leq n}$.

$$\widehat{\mathbf{c}} \simeq \text{argmin}_{\mathbf{c} \in \mathbb{R}^P} \frac{1}{n} \sum_{k=1}^{n} \left(\mathbf{c}^T \mathbf{p}(X^{(k)}) - f(X^{(k)}) \right)^2 \tag{20}$$

This minimization problem can be solved directly in matrix form. Let us introduce the matrix $\mathbf{A} \in \mathcal{M}_{P,n}(\mathbb{R})$ defined by $A_{k,j} = p_{\alpha_j}(X^{(k)})$ for $1 \leq j \leq P$ and $1 \leq k \leq n$, and the vector $\mathbf{f} \in \mathbb{R}^n$ defined by $f_k = f(X^{(k)})$ for $1 \leq k \leq n$. Similarly to GP regression, the input-output pairs $(\mathbf{X}, \mathbf{f})$ introduced in this section are the training dataset from which the coefficients are inferred. The solution of the minimization problem yields the vector of coefficients $\widehat{\mathbf{c}}$ which is given by the product of the pseudo-inverse of $\mathbf{A}$ with the training outputs $\mathbf{f}$:

$$\widehat{\mathbf{c}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{f} \tag{21}$$

4.3.3 Bayesian extension

The polynomial chaos expansion method described so far allows to build a surrogate model but does not provide a quantification of the error of the model, for any given input point. For this reason, the Bayesian paradigm is applied to the PCE method. The coefficients of the polynomial expansion are considered stochastic now. Our goal is to find the posterior distribution of the coefficients $\mathbf{c}$ given the training dataset $\mathcal{D} = (\mathbf{X}, \mathbf{f})$. In order to account for the noise in the

data, a noise parameter σ^2 is also added. For a given sample of coefficients $c^{(0)} = (c_\alpha)_{\alpha \in \mathcal{A}_{\alpha_m,r}}$, the prediction for input point x is a Gaussian distribution with mean $\sum_{\alpha \in \mathcal{A}_{\alpha_m,r}} c_\alpha p_\alpha(x)$ and variance σ^2. The model is thus parametrized by $\xi = (\mathbf{c}, \sigma^2)$, where $\mathbf{c}$ is the vector of coefficients $\mathbf{c} = (c_\alpha)_{\alpha \in \mathcal{A}_{\alpha_m,r}}$. Let $p(\xi|\mathcal{D})$ be the posterior distribution of the parameters given the dataset. Once the posterior distribution is known, predictions can be obtained by marginalizing over the parameters :

$$p(y|\mathcal{D}, x) = \int p(y|\xi, x) p(\xi|\mathcal{D}) d\xi \tag{22}$$

In practice, this integral is obtained by Monte-Carlo estimation. Assuming we have M independent samples of the parameters $(\xi^{(m)})_{1 \leq m \leq M}$ such that $\xi^{(m)} \sim p(\xi|\mathcal{D})$, then :

$$p(y|\mathcal{D}, x) \simeq \frac{1}{M} \sum_{m=1}^{M} p(y|\xi^{(m)}, x) \tag{23}$$

However, the posterior distribution is difficult to sample in practice. The two main approaches to sample the posterior distribution are variational inference and MCMC methods.

4.3.4 Connection to Gaussian Process Regression

The Bayesian Polynomial Chaos Expansion (BPCE) method described in the previous paragraph is another way of building a surrogate model, while being able to quantify the uncertainties on the predictions. When independent Gaussian priors are set on the coefficients, one can view this method as an example of GP regression with an specific choice of covariance kernel. This approach is similar to what was done in [17].

The BPCE model is described by $f(x) = \sum_{\alpha \in \mathcal{A}_{\alpha_m,r}} c_\alpha p_\alpha(x) + \varepsilon$. Let us consider independent Gaussian priors for the coefficients and the noise term. With the notations of section 4.3.1, for $1 \leq p \leq P$ and $\alpha_p \in \mathcal{A}_{\alpha_m,r}$, the prior on the coefficients and the noise are $c_{\alpha_p} \sim \mathcal{N}(0, \sigma_0^2)$ and $\varepsilon \sim \mathcal{N}(0, \sigma_1^2)$. This model defines a prior in the functional space. For any collection of inputs $\mathbf{X} = (X^{(k)})_{1 \leq k \leq n}$:

$$f(\mathbf{X}) \sim \mathcal{N}\left(\mathbf{0}, K(\mathbf{X}, \mathbf{X})\right) \tag{24}$$

where the covariance matrix is defined by :

$$(K(\mathbf{X}, \mathbf{X}))_{i,j} = \operatorname{cov}\left(f(X^{(i)}), f(X^{(j)})\right) = \sigma_1^2 \delta_{i,j} + \sigma_0^2 \sum_{\alpha \in \mathcal{A}_{\alpha_m,r}} p_\alpha(X^{(i)}) p_\alpha(X^{(j)}) \tag{25}$$

for $1 \leq i, j \leq n$. The covariance matrix can be rewritten $K(\mathbf{X}, \mathbf{X}) = \sigma_1^2 \mathcal{I}_n + \sigma_0^2 H H^T$ where $H_{k,p} = p_{\alpha_p}(X^{(k)})$ for $1 \leq p \leq P$ and $1 \leq k \leq n$. Hence, the BPCE is analog to a Gaussian process with zero mean and a non-stationary covariance function given by :

$$k(x, x') = \sigma_1^2 \delta_{x,x'} + \sigma_0^2 \sum_{\alpha \in \mathcal{A}_{\alpha_m,r}} p_\alpha(x) p_\alpha(x') \tag{26}$$

So far, only the scalar output case was considered. In order to extend this method to multiple correlated outputs, one must define one PCE for each output channel and change the scalar noise term ε to a multivariate noise random variable with zero-mean and covariance

$\mathbf{C} = (c_{d,d'})_{1 \leq d,d' \leq D} \in \mathcal{M}_{D,D}(\mathbb{R})$.
The multi-output BPCE model is described by :

$$f(x) = \begin{pmatrix} g_1(x) \\ \vdots \\ g_D(x) \end{pmatrix} + \varepsilon \tag{27}$$

with $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$ and for the d-th output, $g_d(x) = \sum_{\alpha \in \mathcal{A}^{(d)}_{\alpha_m,r}} c^{(d)}_\alpha p^{(d)}_\alpha(x)$ with $c^{(d)}_\alpha \sim \mathcal{N}(0, \sigma^2_{0,d})$ where $p^{(d)}_\alpha$ is the polynomial of multi-index α, $\mathcal{A}^{(d)}_{\alpha_m,r}$ is the truncation for the d-th output, and $\sigma^2_{0,d}$ is the variance of the prior on the coefficients. The coefficients of the covariance matrix are hyperparameters of the covariance kernel associated to this multi-output BPCE model. The covariance is now given by :

$$\text{cov}(f(X^{(i)})_d, f(X^{(j)})_{d'}) = c_{d,d'}\ \delta_{i,j} + \delta_{d,d'}\ \sigma^2_{0,d} \sum_{\alpha \in \mathcal{A}^{(d)}_{\alpha_m,r}} p^{(d)}_\alpha(X^{(i)}) p^{(d)}_\alpha(X^{(j)}) \tag{28}$$

The choice of the truncation $\mathcal{A}^{(d)}_{\alpha_m,r}$ for each channel d was obtained with a cross validation scheme using standard PCE. The hyperbolic truncation scheme described in equation 17 is used for each channel, only the values of α_m and r differ from one output to the other. The truncation yielding the lowest Leave-One-Out (LOO) error $\mathcal{E}_{\text{LOO}}$ is kept. The LOO error[16] is given by :

$$\mathcal{E}_{\text{LOO}} = \frac{1}{n} \sum_{k=1}^{n} \left(\frac{\widehat{\mathbf{c}}^T \mathbf{p}(X^{(k)}) - f(X^{(k)})}{1 - q^{(k)}} \right)^2 \tag{29}$$

where $q^{(k)}$ is the k-th diagonal element of the matrix $\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$.
With this specific Gaussian process surrogate model, the hyperparameters of the kernel are selected by maximizing the marginal likelihood as described in section 4.1.

4.4 Bias learning

Instead of learning directly the Feynman moments and the average count rate, it is possible to learn the disparities between the point model and the simulated data. This idea is adapted from [18] and [19] where the low-fidelity code is the deterministic point model f_p and the high-fidelity code is MCNP6 (or a real experiment):

$$y = f_p(x) + f_s(x) + \epsilon \tag{30}$$

This time, the Gaussian process f_s models the bias of the point model. Since the training data are not necessarily anymore in this case, the Box-Cox transform cannot be used. Instead, the data is preprocessed using the Yeo-Johnson transform [20].

To account for systematic biases caused by the point model approximations, one could introduce an hyperparameter $\rho \in \mathbb{R}^D$ and define a Gaussian Process f_s by weighting the contribution of the point model predictions with ρ.

$$y = \rho \odot f_p(x) + f_s(x) + \epsilon \tag{31}$$

where $\odot$ is the element-wise Hadamard product. The training of the GP works similarly except ρ is identified as an hyperparameter to be tuned by maximizing the marginal log-likelihood.

However, the addition of ρ has not yielded any significant improvement in the performance of the surrogate models and is thus not further investigated in the remainder of the paper and ρ is fixed to $\rho = (1, ..., 1)^T$.

4.5 Performance of the surrogate models

The different surrogate models compared here are the independent scalar GPs (IGP), the Linear Model of coregionalization with 2 latent GPs (LMC2) and 3 latent GPs (LMC3), the Bayesian Polynomial Chaos Expansion (BPCE), and a LMC model where the 3 latent processes are BPCE. (LMC-BPCE). Each model is trained on the dataset, which is split equally into a training set and a validation set. The Normalized Mean Absolute Error on the validation set is displayed in Table 1 for each surrogate model. The best predictions are highlighted in bold. One can see that the predictions of each surrogate model outperform the predictions of the

	IGP	LMC2	LMC3	BPCE	LMC-BPCE
R	0.011	**0.011**	0.011	0.027	0.011
Y_∞	**0.047**	0.055	0.059	0.051	0.056
X_∞	**0.168**	0.245	0.302	0.330	0.342

Table 1: Normalized mean absolute error on the validation set

point model. Besides, the independent GPs seem to yield the best prediction, when looking at each output individually. This would be expected as the same amount of information is used to predict only one output instead of three. However the covariance predictions of this model are expected to be biased.

The same table is shown in Table 2, except that bias learning is used this time. For almost all

	IGP	LMC2	LMC3	BPCE	LMC-BPCE
R	0.0083	**0.0081**	0.0082	0.0150	0.0097
Y_∞	0.029	0.028	**0.027**	0.033	0.045
X_∞	0.155	**0.073**	0.084	0.140	0.242

Table 2: Normalized mean absolute error on the validation set - Bias learning

the outputs and models, including the information of the point model improves the prediction performance. The best model here seems to be the LMC2 model, and the independent GPs model.

Now let us look at the coverage probabilities of the surrogate models. The output of a GP surrogate model at input point x follows a normal distribution with predictive mean $\overline{f_s(x)}$ and predictive covariance $\mathbf{C}_s(x)$. Let us consider the validation set $\mathcal{D}_{\text{val}} = (\mathbf{X}_*, \mathbf{f}_*)$ where $\mathbf{f}_* = (f_*^{(k)})_{1 \leq k \leq n_*}$ and $\mathbf{X}_* = (X_*^{(k)})_{1 \leq k \leq n_*}$ are the validation outputs and inputs.
Let $D_M\left(f_*^{(k)}, f_s(X_*^{(k)})\right)$ be the Mahalanobis distance between predictions and validation outputs defined by :

$$D_M\left(f_*^{(k)}, f_s(X_*^{(k)})\right) = \sqrt{\left(f_*^{(k)} - \overline{f_s(X_*^{(k)})}\right)^T \mathbf{C}_s(X_*^{(k)})^{-1} \left(f_*^{(k)} - \overline{f_s(X_*^{(k)})}\right)} \tag{32}$$

Then the squared Mahalanobis distance between the predictions and validation outputs should follow a χ^2 distribution with $D = 3$ degrees of freedom.

$$D_M\left(f_*^{(k)}, f_s(X_*^{(k)})\right)^2 \sim \chi^2_D \tag{33}$$

Thus, one can evaluate the reliability of the covariance predictions by estimating the Mahalanobis distance on the validation set and comparing the empirical quantiles of the Mahalanobis distance samples, to the actual quantiles of the χ^2_D distribution. This defines a multi-dimensional coverage probability. Our goal is to find the fraction of points that are in the credible region of level α defined by :

$$I_\alpha = \left\{ D_M\left(f_*^{(k)}, f_s(X_*^{(k)})\right)^2 \leq q_\alpha \mid 1 \leq i \leq N_* \right\} \tag{34}$$

where q_α is the quantile of level α of the chi-squared distribution with D degrees of freedom. For a reliable surrogate model, the fraction of points in I_α is expected to be close to α.

$$\frac{\text{card}(I_\alpha)}{N_*} \simeq \alpha \tag{35}$$

The coverage probabilities are estimated with this method for all the surrogate models and for different values of the credibility levels α. The results are displayed in Figure 1. As expected,

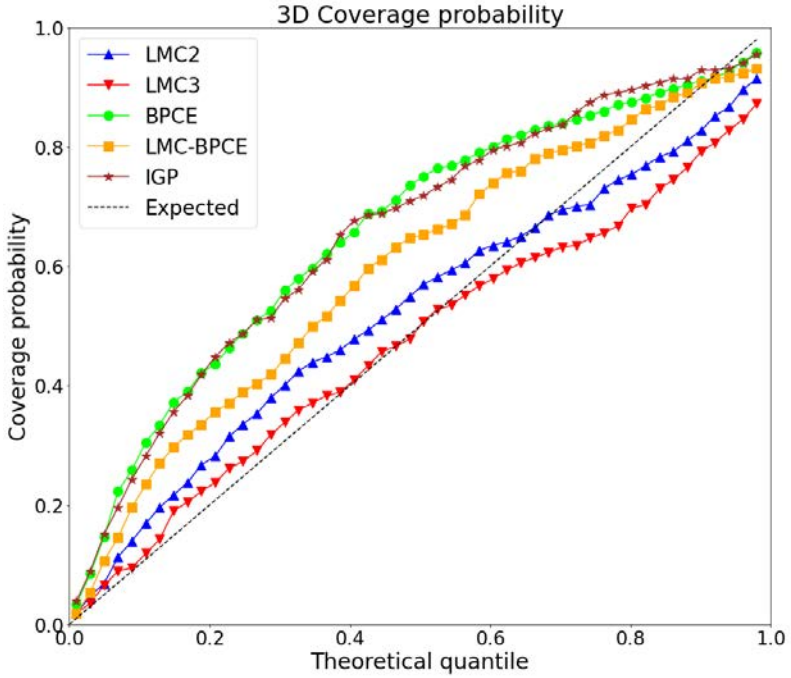

Figure 1: Coverage probabilities of the surrogate models for different credibility levels

the covariance predictions for the independent GPs model are not reliable since the correlations between the outputs are not considered. The predicted covariances for the BPCE model are also not reliable. This may be caused by the $\delta_{i,j}$ in the expression of the covariance kernel in equation 28, which may be too restrictive. However, the LMC models all yield better covariance

predictions, especially with Matérn latent GPs, even though some slight discrepancies with the theoretical quantiles are still observed.

Overall, of all the surrogate models explored, the LCM2 model seems to be the best choice since it provides satisfying mean predictions and reliable covariance predictions. This model is thus chosen to sample the posterior distribution in the inverse problem resolution.

5 APPLICATION

5.1 Description of the problem

This first example is taken from the set of experiments FUND-NCERC-PU-HE3-MULT-003 of the ICSBEP Handbook, experiment n°1 [21]. The experiment is a measurement of neutron correlations on metallic plutonium sphere surrounded by a stainless steel cladding. The neutron detector is a cylinder of ^{3}He with a polyethylene moderator layer. Some simplifications were made on the numerical model compared to the experiment. All nuclear data, including effective cross sections and neutron multiplicities, are considered known. In some works, the uncertainties in the nuclear data are propagated to the neutron transport simulations. This is technically feasible with the methodology described in this paper but that would further complicate the statistical models involved. For simplification purposes, the nuclear data are thus considered known. Besides, in this work, the main source of uncertainty comes from the limited number of neutron detections, and the model error. The impact of the nuclear data uncertainties is expected to be negligible in comparison.

Overall, the simplified MCNP6 model built is close to the benchmark model and the experiment. Some disparities arise due to the simplifications made. They can be linked to the removal of the aluminum support plate for example. Our goal here is not necessarily to exactly retrieve the benchmark results but rather to show the improvements brought by the method presented in this paper with an application on a well-documented experiment.

5.2 MCMC sampling

The posterior distribution $p(x|\mathbf{y})$ is sampled with Monte-Carlo Markov Chain methods based on the likelihood in equation 8. For this particular problem, due to the non-linearity of the direct model, the posterior distribution is very degenerate. In this work, a degenerate distribution is a distribution whose support lies mainly in a manifold of dimension one or two, within the parameter space of dimension $I = 7$. This definition of degeneracy is not rigorous. More precisely, a degenerate distribution is rigorously defined as a distribution whose support has zero Lebesgue measure. However the notion of degeneracy is considered in this work as a practical limitation to MCMC methods and not as a formal mathematical definition. For that reason, the method used is the Adaptive Metropolis algorithm described in [22]. The principle is similar to Metropolis-Hastings, but the covariance of the proposal law is adapted during the run to match the empirical covariance of the sampled Markov Chain. Let $(x^{(m)})_{m\leq M}$ be the sampled Markov chain. At iteration m, the proposal distribution for the new candidate point $\widehat{x}$ is of the form $\widehat{x} \sim \mathcal{N}\left(x^{(m)}, \mathcal{C}^{(m)}\right)$, where the covariance is modified at each step to match the empirical covariance of the points of the chain:

$$\mathcal{C}^{(m)} = s \,\mathrm{Cov}\left(x^{(0)}, ..., x^{(m-1)}\right) \tag{36}$$

$$\mathrm{Cov}\left(x^{(0)}, ..., x^{(m-1)}\right) = \frac{1}{m-1}\sum_{l=0}^{m-1}\left(x^{(l)} - \overline{x}\right)\left(x^{(l)} - \overline{x}\right)^T \tag{37}$$

with $\overline{x} = \frac{1}{m}\sum_{l=0}^{m-1} x^{(l)}$, where $x^{(l)}$ is the l-th element of the Markov chain created by the MCMC algorithm. The scaling factor s is a fixed real value, which has to be tuned to reach the desired acceptance rate.
The direct calculation of the empirical covariance is cumbersome when the chain becomes long. A recursive formula is preferred to evaluate $\mathcal{C}_k$.

$$\begin{aligned}\mathcal{C}^{(m+1)} &= \frac{s}{m}\left(m\overline{x^{(m)}}\,\overline{x^{(m)}}^T - (m+1)\overline{x^{(m+1)}}\,\overline{x^{(m+1)}}^T + (x^{(m+1)})\,(x^{(m+1)})^T\right) \\ &+ \frac{m-1}{m}\,\mathcal{C}^{(m)} \end{aligned} \tag{38}$$

This recursive formula helps speed up the calculation of the covariance. In practice, it is advised to add a small term in the form $\epsilon\mathcal{I}_I$ with $\epsilon > 0$ and I the dimension of the input, in order to guarantee the matrix stays definite positive. Indeed, numerical approximations can lead to a singular covariance matrix, which can be problematic for the sampling of the candidate points.

The adaptation of the covariance matrix is not started directly from the beginning, but rather after a certain number of accepted points n_0 is reached in order to make sure the empirical covariance is calculated on enough points. n_0 is set at 5000 in our case. It is also possible to dynamically change the scaling factor to reach a target acceptance rate for example one can multiply the covariance by a factor r_k defined as :

$$r_k = \exp\left(\alpha_{\text{current}} - \alpha_{\text{target}}\right) \tag{39}$$

where α_{target} and α_{current} are respectively the target and current acceptance rates. This method is implemented in the AM algorithm in our work.

To guarantee that the Markov chain distribution approaches the target distribution, the autocorrelation of the chain is studied. For a given integer h, the autocorrelation of the Markov chain $(x^{(m)})_{m\leq M}$ of length M is defined by the average correlation between $x^{(m)}$ and $x^{(m+h)}$.

$$\text{AC}(h) = \overline{\text{Corr}(x^{(m)}, x^{(m+h)})} \tag{40}$$

where $\text{Corr}(x^{(m)}, x^{(m+h)})$ is the correlation between $x^{(m)}$ and $x^{(m+h)}$, and the average is done over the whole chain, for $m \leq M - h$. The decorrelation time τ is the number of iterations required to completely decorrelate two points of the chain:

$$\tau = \min\left\{h \leq M | \text{AC}(h) = 0\right\} \tag{41}$$

It can be seen as the time taken by the chain to explore the whole target distribution. To guarantee enough statistics when sampling the posterior distribution, the length K of the chain should be much larger than the decorrelation time. This is why we need to evaluate the decorrelation time. An example of autocorrelation curve is shown in Figure 2. The estimated decorrelation time is $\widehat{\tau} \simeq 1000$ for the Adaptive Metropolis algorithm, compared to $\widehat{\tau} \simeq 3000$ for standard Metropolis-Hastings, for a similar computing time. Adaptive Metropolis allows to divide the number of iterations in MCMC by a factor 3. To guarantee enough statistics, the total number of points sampled by MCMC is fixed to 5×10^6 which is much larger than the estimated decorrelation time $\widehat{\tau}$. To obtain the density of the posterior distribution, the Markov chain is thinned by a factor $\widehat{\tau}$, such that the samples of the chain are independent. Finally, the density of the posterior distribution can be estimated with a kernel density estimation.

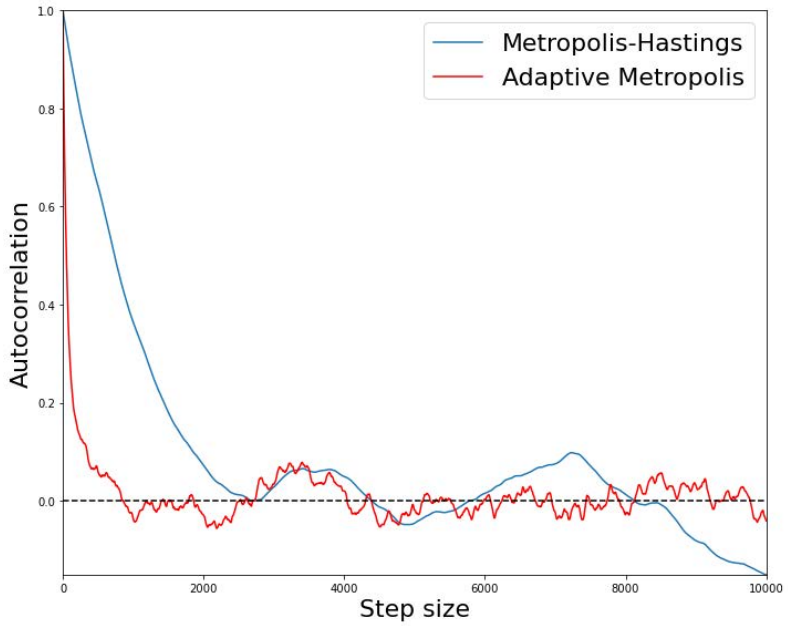

Figure 2: Autocorrelation of the Markov chain, with standard Metropolis-Hastings and with Adaptive Metropolis

5.3 Posterior distribution

First of all, to highlight the improvements brought up by the methodology presented in this work, the posterior distribution is sampled using the likelihood in equation 3, which only uses the point model as a direct model. This posterior distribution thus only includes the uncertainty linked to the observation noise. It is displayed in Figure 3. To simplify the visualization, only

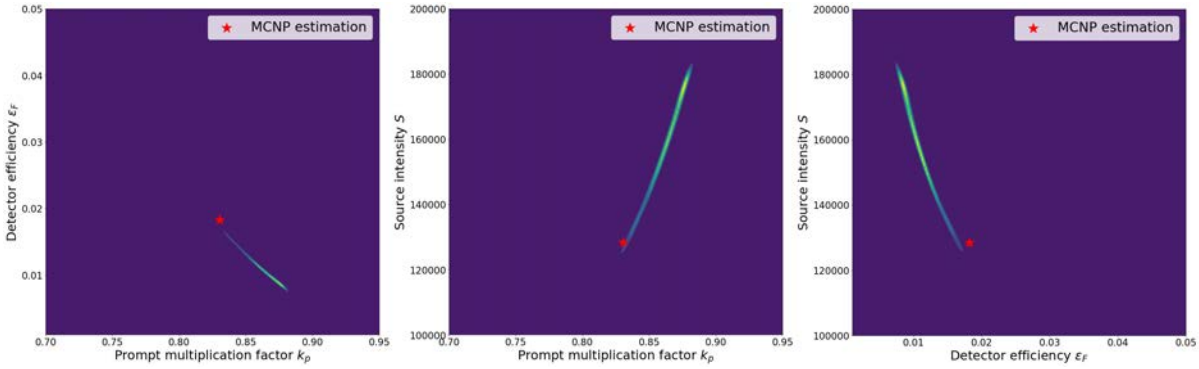

Figure 3: Posterior distribution sampling with the point model, based on equation 3

the 2D marginal distributions are shown here, for the three main parameters of interest. The theoretical values of the parameters (k_p, ε_F, S) as obtained by the numerical simulation with MCNP6 are shown in red. One can see that the posterior distribution obtained here is thin, yet it is biased as the true parameters are not lying on the distribution support. The posterior distribution does not account for the inherent bias of the point model.

Now, let us sample the posterior distribution using the modified likelihood in equation 8, which includes the model error and is based on the surrogate model LMC2, whose predictions are more precise than the point model. The new posterior distribution obtained is shown in

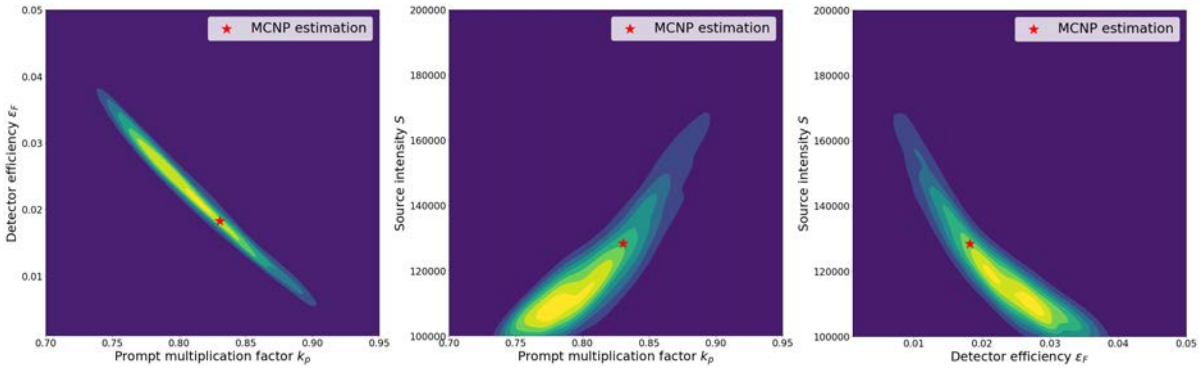

Figure 4: Posterior distribution sampling with the surrogate model, based on equation 8

Figure 4. As expected, this distribution is broader since it accounts for the model error. Besides, the actual values of the parameter lie within the distribution support. The surrogate models presented in this work were not designed to provide the best possible prediction performance, but rather to provide reliable uncertainty predictions. The surrogate models could be improved by increasing the size of the dataset and its dimension. Due to the non-scalability of exact GP inference for large datasets, sparse approximations of GP regression could become necessary in that case [23].

6 CONCLUSION

In this work, a general methodology for the uncertainty quantification in ill-posed inverse problem is presented. Different surrogate models specifically suited for strongly correlated outputs were investigated. The posterior distribution sampled with MCMC methods, encompasses both the model error and the noise of the observations. The study of the multi-dimensional coverage probabilities of the surrogate models showed the reliability of the uncertainty quantification scheme.

This work focuses on Gaussian process surrogate models, however other surrogate models could be investigated. Bayesian neural networks have been considered as they provide more flexibility in the regression task, but their main drawback resides in the reliability of the uncertainty predictions. Heteroscedastic GP models are also investigated since the training data is not actually homoscedastic in this work, and the GP surrogates presented here cannot account for the input-dependent noise as of now. Finally, the models could be improved by increasing the size and dimension of the dataset, to reduce the model error which is still significant in the posterior distribution, and to improve the generalization of the models to other types of nuclear materials.

REFERENCES

[1] R. P. Feynman, F. De Hoffmann, and R. Serber, "Dispersion of the neutron emission in U-235 fission," *Journal of Nuclear Energy (1954)*, vol. 3, no. 1-2, pp. 64–IN10, 1956.

[2] P. Lartaud, P. Humbert, and J. Garnier, "Multi-output gaussian processes for inverse uncertainty quantification in neutron noise analysis," *Nuclear Science and Engineering*, pp. 1–24, 2023.

[3] E. V. Bonilla, K. Chai, and C. Williams, "Multi-task gaussian process prediction," *Advances in neural information processing systems*, vol. 20, 2007.

[4] K. Böhnel, "The effect of multiplication on the quantitative determination of spontaneously fissioning isotopes by neutron correlation analysis," *Nuclear Science and Engineering*, vol. 90, no. 1, pp. 75–82, 1985.

[5] D. M. Cifarelli and W. Hage, "Models for a three-parameter analysis of neutron signal correlation measurements for fissile material assay," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 251, no. 3, pp. 550–563, 1986.

[6] D. Langner, J. Stewart, M. Pickrell, M. Krick, N. Ensslin, and W. Harker, "Application guide to neutron multiplicity counting," Los Alamos National Laboratory, Los Alamos, NM, Tech. Rep., 1998.

[7] I. Pázsit and L. Pál, *Neutron fluctuations: A treatise on the physics of branching processes*. Elsevier, 2007.

[8] T. Goorley, M. James, T. Booth, F. Brown, J. Bull, L. Cox, J. Durkee, J. Elson, M. Fensin, R. Forster *et al.*, "Initial MCNP6 release overview," *Nuclear technology*, vol. 180, no. 3, pp. 298–315, 2012.

[9] R. M. Sakia, "The box-cox transformation technique: a review," *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 41, no. 2, pp. 169–178, 1992.

[10] J. M. Bernardo and A. F. Smith, *Bayesian theory*. John Wiley & Sons, 2009, vol. 405.

[11] A. C. Davison and D. V. Hinkley, *Bootstrap methods and their application*. Cambridge university press, 1997, no. 1.

[12] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer school on machine learning*. Springer, 2003, pp. 63–71.

[13] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on scientific computing*, vol. 16, no. 5, pp. 1190–1208, 1995.

[14] M. Titsias, "Variational learning of inducing variables in sparse gaussian processes," in *Artificial intelligence and statistics*. PMLR, 2009, pp. 567–574.

[15] O. G. Ernst, A. Mugler, H.-J. Starkloff, and E. Ullmann, "On the convergence of generalized polynomial chaos expansions," *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 46, no. 2, pp. 317–339, 2012.

[16] G. Blatman and B. Sudret, "An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis," *Probabilistic Engineering Mechanics*, vol. 25, no. 2, pp. 183–197, 2010.

[17] R. Schobi, B. Sudret, and J. Wiart, "Polynomial-chaos-based kriging," *International Journal for Uncertainty Quantification*, vol. 5, no. 2, 2015.

[18] M. C. Kennedy and A. O'Hagan, "Predicting the output from a complex computer code when fast approximations are available," *Biometrika*, vol. 87, no. 1, pp. 1–13, 2000.

[19] L. Le Gratiet and J. Garnier, "Recursive co-kriging model for design of computer experiments with multiple levels of fidelity," *International Journal for Uncertainty Quantification*, vol. 4, no. 5, 2014.

[20] I.-K. Yeo and R. A. Johnson, "A new family of power transformations to improve normality or symmetry," *Biometrika*, vol. 87, no. 4, pp. 954–959, 2000.

[21] J. B. Briggs, L. Scott, and A. Nouri, "The international criticality safety benchmark evaluation project," *Nuclear science and engineering*, vol. 145, no. 1, pp. 1–10, 2003.

[22] H. Haario, E. Saksman, and J. Tamminen, "An adaptive metropolis algorithm," *Bernoulli*, vol. 7, no. 2, pp. 223–242, 2001.

[23] J. Quinonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate gaussian process regression," *The Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.