Report on suitability and potential of Reduced Order Modelling (ROM) to fusion models

Gaussian Process ROM for Solvers with High-dimensional Outputs

Deyu Ming and Serge Guillas
University College London

UKAEA Report: 2047352_2-TN-02 D1.1
Final report
October 28, 2021

1 Disclaimer

We are very grateful to Dr Patrick Farrell for the provision of the proxyApp modelling the anisotropic heat transport problem. It is the only fusion model we could access over the short period of the funded project (4 January 2021 - 31 July 2021). We discussed with the NEPTUNE team (Benjamin Dudson and Patrick Farrell) the possibility of using another model to couple two models in a one-way coupling for UQ using ROM: the anisotropic heat transport model and the isotropic heat conduction to the solid wall. But the wall boundary proxyApp is not yet available. As a result, we could not examine in practice the possibility of implementing ROM for UQ in the context of nuclear fusion modelling where models are coupled. We nevertheless provided some examples of UQ coupling at the end of this report from the paper [16] and discussed future directions below.

2 Introduction

Many modern physical computer models involve solving PDEs with numerical solvers, such as finite element methods (FEM), which can be computationally expensive due to

- ever more complex and larger-scale models;
- high-dimensional input and output;
- large demands on computational resources.

These create challenges to efficient uncertainty quantification of computer models, such as the fusion models, as we often need to run the models many times for tasks such as sensitivity analysis, uncertainty propagation and model calibration. To tackle these challenges, reduced order models (ROM) are needed to

- serve as low-dimensional replacements with comparable accuracy;
- reduce evaluation time of original solvers;
- save storage, e.g., for high-dimensional output.

Traditional reduced order models, also known as intrusive reduced order models, often are constructed using reduced basis methods [19], among which the Proper Orthogonal Decomposition (POD) is perhaps the most popular technique. The intrusive reduced order models for original high-fidelity models with high-dimensional output are typically built using a two-phase procedure called offline-online decomposition:

- *offline phase*: high-fidelity solutions/outputs are obtained and reduced basis is calculated;
- *online phase*: the original problems are projected onto the reduced space for efficient computation of solutions at new inputs.

However, the online phase of the intrusive reduced order modelling is challenging in practice because:

- expertise and domain knowledge are required to project the equations and physics of the original high-fidelity problems to constructed reduced space;
- dimensionality reduction techniques are largely constrained by the problem formulation;
- uncertainty is not incorporated.

For these reasons, in this report we focus on non-intrusive reduced order models for problems with high dimensional outputs, utilising the family of Gaussian process (GP) surrogates (also known as emulators). GP emulators have been successfully implemented for dimension reduction of either outputs or inputs. For instance:

- [10] used Functional Principal Components Analysis (FPCA) as an equivalent approach to POD for time series outputs of tsunami waves, and [1] used Spherical Harmonics and Gaussian Markov Random Fields for optimal reduction of surfaces outputs.
- For inputs, [14] employed a kernel-based approach to extract the few input field directions of most influence for the outputs in order to build GPs with few input dimensions (orders of magnitude gain in dimension).

The report is organised as follows. In Section 3, a non-intrusive ROM with GP surrogates and POD is described and applied in a anisotropic heat transport problem. We then propose and discuss an active learning procedure to construct the introduced non-intrusive ROM with an illustrative example in Section 4. Future directions are discussed in Section 5.

3 Non-intrusive ROM with Gaussian Process Surrogates

The non-intrusive reduced order modelling is a data-driven approach that uses a statistical surrogate model to mimic the functional relations between the model input and constructed reduced output space in the online phase of the offline-online decomposition. The utilisation of statistical surrogates alleviates the difficulties involved in reformulating the original high-fidelity problems under the intrusive reduced order modelling. In particular, with GP surrogates we

are able to quantify uncertainty of the high-dimensional outputs predicted at unobserved input positions.

Let $\mathbf{X} \in \mathbb{R}^{N\times D}$ contain N sets of D dimensional input of a computer model, which produces N corresponding sets of K dimensional output $\mathbf{Y} \in \mathbb{R}^{N\times K}$ accordingly. Then, one can mimic the functional relationships between the input $\mathbf{X}$ and each output dimension $\mathbf{Y}_k \in \mathbb{R}^{N\times 1}$ by a GP surrogate $\mathcal{GP}_k$ independently for $k = 1, \ldots, K$ without considering the dependence between output dimensions [7]. Ignoring the potential cross-dependence does not pose a serious issue unless we are interested in the joint distribution of the output, and it can be shown [12] that the independently constructed GP surrogates correspond to the marginal GPs of a joint GP surrogate under certain dependence structures. The GP surrogate $\mathcal{GP}_k$ is formally defined as a multivariate normal distribution with respect to $\mathbf{Y}_k$:

$$\mathbf{Y}_k \sim \mathcal{N}(\boldsymbol{\mu}_k(\mathbf{X}), \sigma_k^2 \mathbf{R}_k(\mathbf{X})),$$

in which the i-th element of $\boldsymbol{\mu}_k(\mathbf{X}) \in \mathbb{R}^{N\times 1}$ is often specified by a trend function $f_k(\mathbf{X}_i)$ with $\mathbf{X}_i \in \mathbb{R}^{1\times D}$ being the i-th row of $\mathbf{X}$, and the ij-th element of $\mathbf{R}_k(\mathbf{X}) \in \mathbb{R}^{N\times N}$ is given by $c_k(\mathbf{X}_i, \mathbf{X}_j)$, where c_k is a given kernel function. The trend function f_k can be formulated as a linear combination of a set of basis functions of $\mathbf{X}_i$ and we assume a constant trend function $f_k(\mathbf{X}_i) = b_k$ in this report.

There are various choices for c_k (see [20]). In this report, we use the separable kernel function:

$$c_k(\mathbf{X}_i,\, \mathbf{X}_j) = \prod_{d=1}^{D} c_{k,d}(X_{id},\, X_{jd}),$$

where $c_{k,d}$ is a one-dimensional kernel function. A typical choice for $c_{k,d}$ in computer model emulation is the squared exponential (SExp) kernel:

$$c_{k,d}(X_{id},\, X_{jd}) = \exp\left\{-\frac{(X_{id} - X_{jd})^2}{\gamma_{k,d}^2}\right\},$$

where $\gamma_{k,d} > 0$ is the range parameter. However, the SExp kernel has been criticised for its over-smoothness [25] for physical problems as well as its associated ill-conditioned problems [3, 9]. Another popular kernel choice is the Matérn kernel [20]:

$$c_{k,d}(X_{id},\, X_{jd}) = \exp\left(-\frac{\sqrt{2p+1}\, r_{ij,d}}{\gamma_{k,d}}\right) \frac{p!}{(2p)!} \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} \left(\frac{2r_{ij,d}\sqrt{2p+1}}{\gamma_{k,d}}\right)^{p-i},$$

where $r_{ij,d} = X_{id} - X_{jd}$. The Matérn kernel is known to be less prone to ill-conditioning issues and provides a reasonably adequate smoothness to the GP surrogates. In particular, the Matérn-2.5 kernel, which is defined as the Matérn kernel with $p = 2$:

$$c_{k,d}(X_{id},\, X_{jd}) = \left(1 + \frac{\sqrt{5}|X_{id} - X_{jd}|}{\gamma_{k,d}} + \frac{5(X_{id} - X_{jd})^2}{3\gamma_{k,d}^2}\right) \exp\left\{-\frac{\sqrt{5}|X_{id} - X_{jd}|}{\gamma_{k,d}}\right\},$$

is the default kernel choice for many computer model emulation packages, such as `DiceKriging` [22]

and `RobustGaSP` [8]. Therefore, we employ the Matérn-2.5 kernel in this report.

The posterior predictive distribution $\mathcal{N}(\widehat{\mu}_k(\mathbf{x}^*), \widehat{\sigma}_k^2(\mathbf{x}^*))$ of $\mathcal{GP}_k$ with respect to the output $Y_k^*(\mathbf{x}^*)$ at an unobserved input position $\mathbf{x}^*$ is given in different analytical forms depending on how the model parameters b_k, σ_k^2 and $\{\gamma_{k,d}\}_{d=1,\ldots,D}$ are estimated. Different maximum-likelihood-based estimation approaches and the corresponding expressions for $\widehat{\mu}_k(\mathbf{x}^*)$ and $\widehat{\sigma}_k^2(\mathbf{x}^*)$ are discussed in [22, 9].

The main computational bottlenecks of the GP surrogate construction are the number of data points N and the dimension K of the output of a computer model. Since the inference of GP surrogates involve inversions of the correlation matrix $\mathbf{R}_k \in \mathbb{R}^{N\times N}$ with computational complexity of $\mathcal{O}(N^3)$, it soon becomes computationally prohibitive to build GP surrogates in practice when N is more than several thousands. In such a case, one may need sparse approximations [13] to the GP to reduce the computational complexity induced by the big data.

In computer model experiments, one often does not have big data (i.e., realisations from the underlying computer model) due to the limited computational budget. However, if the input dimension D is large, then small data are insufficient to explore adequately the whole input domain and thus the resulting GP surrogates can be inaccurate. High input dimension also causes challenges to the model estimation because a large number of range parameters $\{\gamma_{k,d}\}_{d=1,\ldots,D}$ need to be estimated for each output dimension. To alleviate this issue, one can reduce the input dimension D to P such that $P \ll D$ by dimension reduction techniques such as POD, kernel dimension reduction [14], and active subspace [26].

A high output dimension K creates the issue that it can be computational burdensome to build K independent GP surrogates: without parallel implementation the training and validation of a huge amount of GP surrogates are practically infeasible. This report tackles the latter issue on high-dimensional outputs (e.g., a snapshot where each point on the snapshot represents a FE solution and contributes to the output dimensionality) produced by computer models. Perhaps the most straightforward approach to address the issue is to reduce the output dimension K to L such that $L \ll K$ by POD.

The POD of $\mathbf{Y} \in \mathbb{R}^{N\times K}$ can be done with following steps:

1. Compute the sample mean $\boldsymbol{\mu}_{\mathbf{Y}} \in \mathbb{R}^{1\times K}$ of $\mathbf{Y}$ and obtain the centred output matrix $\mathbf{Y}_c = \mathbf{Y} - \boldsymbol{\mu}_{\mathbf{Y}}$;

2. Implement the eigendecomposition of $\mathbf{G} = \frac{1}{N}\mathbf{Y}_c\mathbf{Y}_c^\top$ such that $\mathbf{G} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\top$, where the columns of $\mathbf{V} \in \mathbb{R}^{N\times N}$ contains the eigenvectors of $\mathbf{G}$ and the diagonal of $\boldsymbol{\Lambda} \in \mathbb{R}^{N\times N}$ contains the corresponding eigenvalues $(\lambda_1, \ldots, \lambda_N)$ in descending order;

3. Compute $\tilde{\mathbf{V}} = \mathbf{Y}_c^\top\mathbf{V} \in \mathbb{R}^{K\times N}$, which contains the eigenvectors of sample covariance matrix $\mathbf{C} = \frac{1}{N}\mathbf{Y}_c^\top\mathbf{Y}_c$;

4. Choose $L \leq N$ and obtain the low dimensional output $\widehat{\mathbf{Y}} = \mathbf{Y}_c\tilde{\mathbf{V}}_L \in \mathbb{R}^{N\times L}$, where $\tilde{\mathbf{V}}_L \in \mathbb{R}^{K\times L}$ contains the first L eigenvectors included in $\tilde{\mathbf{V}}$.

One can also obtain $\tilde{\mathbf{V}}$ by performing the singular value decomposition (SVD) of $\mathbf{Y}_c$ that is implemented, e.g., in the PCA function of Python package `scikit-learn` [18]. After obtaining

the low dimensional data $\widehat{\mathbf{Y}}$, we then construct L independent GP surrogates of each of L dimensions of $\widehat{\mathbf{Y}}$. Let $\mathcal{N}(\widehat{\mu}_l(\mathbf{x}^*), \widehat{\sigma}_l^2(\mathbf{x}^*))$ be the posterior predictive distribution of $\widehat{Y}_l^*(\mathbf{x}^*)$, the l-th dimension of the low dimensional output, predicted at an unobserved input position $\mathbf{x}^*$. Then the posterior predictive distribution of the corresponding high dimensional output $\mathbf{Y}^*(\mathbf{x}^*) \in \mathbb{R}^{1\times K}$ is given by

$$\mathcal{N}\left(\widehat{\boldsymbol{\mu}}(\mathbf{x}^*)\tilde{\mathbf{V}}_L^\top + \boldsymbol{\mu}_{\mathbf{Y}}, \tilde{\mathbf{V}}_L\widehat{\boldsymbol{\Sigma}}(\mathbf{x}^*)\tilde{\mathbf{V}}_L^\top\right),$$

where $\widehat{\boldsymbol{\mu}}(\mathbf{x}^*) = (\widehat{\mu}_1(\mathbf{x}^*), \ldots, \widehat{\mu}_L(\mathbf{x}^*))$ and $\widehat{\boldsymbol{\Sigma}}(\mathbf{x}^*) = \mathrm{diag}(\widehat{\sigma}_1^2(\mathbf{x}^*), \ldots, \widehat{\sigma}_L^2(\mathbf{x}^*))$.

Figure 1 demonstrates the procedure to build non-intrusive reduced order model with GP surrogates. In the offline phase, dimension-reduction techniques, e.g., POD, are applied to reduce the high-dimensional output to a low-dimensional space. Then in the online phase, GP surrogates are constructed independently on each reduced dimension. Using the constructed GP surrogate and reduced basis, one can obtain the predicted low-dimensional and in turn the high-dimensional output at new input positions with little computational efforts.

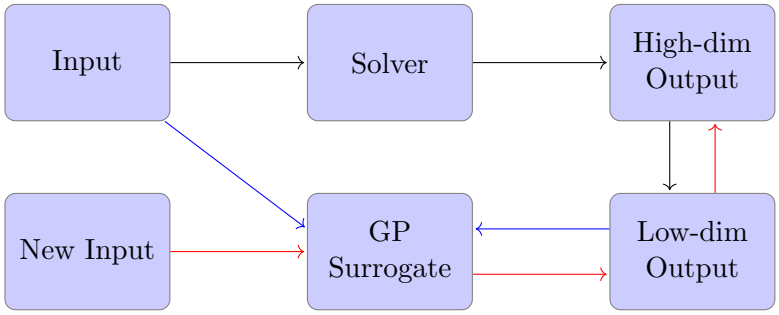


Figure 1: The workflow to construct non-intrusive ROM with GP. The black arrows represent the offline phase; the blue arrows represent the online phase; the red arrows represent the prediction procedure using the constructed non-intrusive ROM with GP.

3.1 Example: 2-D model of anisotropic heat transport

In this section, we explore the non-intrusive ROM with GP to mimic the FE solver to the 2-D problem "Open field lines with oscillating anisotropy directions" in [5]. The problem has two key inputs m and α that control the anisotropy of the solution field, i.e., the anisotropy direction is defined by

$$\mathbf{b} = \frac{\mathbf{B}}{|\mathbf{B}|}, \quad \mathbf{B} = \begin{pmatrix} \alpha(2y-1)\cos(m\pi x) + \pi \\ \pi\alpha m(y^2 - y)\sin(m\pi x) \end{pmatrix},$$

where $m/2$ is the number of oscillation periods in the computational domain and α is the amplitude. The output is a high-dimensional 2-D field defined on the square computational domain $[0,1]\times[0,1]$ and allows a closed form solution.

3.1.1 Experimental Setup

To construct the reduced basis via the POD and the GP surrogate, N=40 samples are arranged in a Latin hypercube over $m \in [0, 12]$ and $\alpha \in [0, 3]$ (see the left plot in Figure 2). We then run

the FE solver (implemented in `Firedrake` [21]) of the toy problem to obtain the corresponding 2-D outputs, each of which contains FE solutions on $K = 78961$ nodes. These 40×78961 high-dimensional outputs are then reduced to 40 low-dimensional outputs (40×25) using POD by retaining the first 25 principal components out of the total 40 components, see the right plot in Figure 2, where the cumulative explained variance is defined as $\frac{\sum_{i=1}^{L} \lambda_i}{\sum_{i=1}^{N} \lambda_i}$ with L be the number of components.

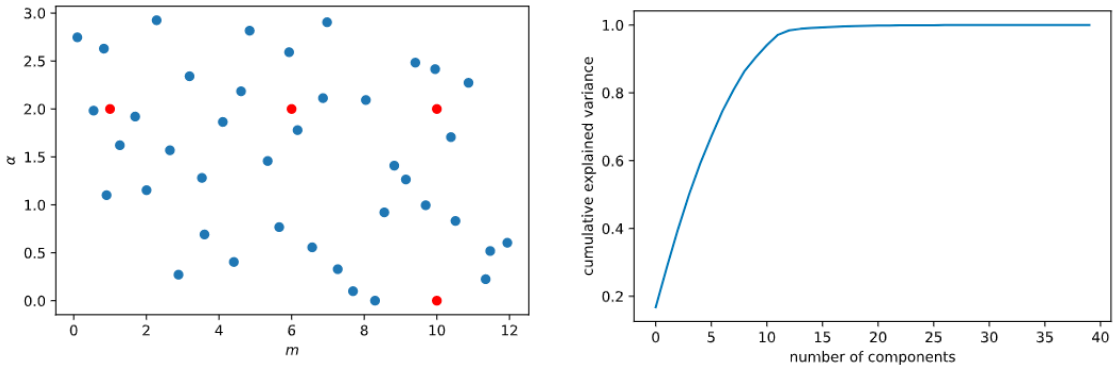


Figure 2: *(Left)*: Training and designing points generated for the inputs m and α. The blue points are design input locations generated from the Latin hypercube design and the red points are testing input locations; *(Right)*: cumulative explained variance given by the POD.

GP surrogates are then constructed independently for each of the 25 dimensions of the reduced order data. GP surrogates are trained with the Matérn-2.5 kernel using the `RobustGaSP` package in R.

3.1.2 Experimental Results

We test the constructed non-intrusive ROM at four testing input positions $(m, \alpha) = (6, 2)$, $(10, 2)$, $(1, 2)$ and $(10, 0)$ (see the left plot of Figure 2). The FE solutions (from the `Firedrake`) and the predicted solutions from the built ROM are compared in Figure 3. The normalised (to the range of FE solutions) errors between the FE solutions and the predicted solutions from the built ROM are shown in Figure 4. The coverage of the ROM (i.e., the instances that the FE solutions fall within the predictive bounds of GP-based ROM) are also given in Figure 5.

It can been seen from these results that the constructed ROM with GP could predict well the FE solutions of the anisotropic problem at input locations that are not realised. Among the four testing positions, the final case with $m = 10$ and $\alpha = 0$ presents the largest normalised errors up to 13%. This is not a surprising result because m has no effect on the FE solution of the problem when $\alpha = 0$. However, this information is not fully captured in the training data and thus not gained by the non-intrusive ROM with GP, which is pure data-driven method that only understands the functional relation between m, α and the solution field from the training set. As a result, we could observe 5 blurred oscillation periods in the predicted solutions from ROM in Figure 3. However, the predictive interval (whose upper and lower bounds are given at two standard deviations $2\widehat{\sigma}$ above and below the predictive mean $\widehat{\mu}$) of the GP-based ROM covers the FE solutions sufficiently in this case, demonstrating that one can benefit from the predictive uncertainty embedded in the non-intrusive ROM coupled with GP emulation.

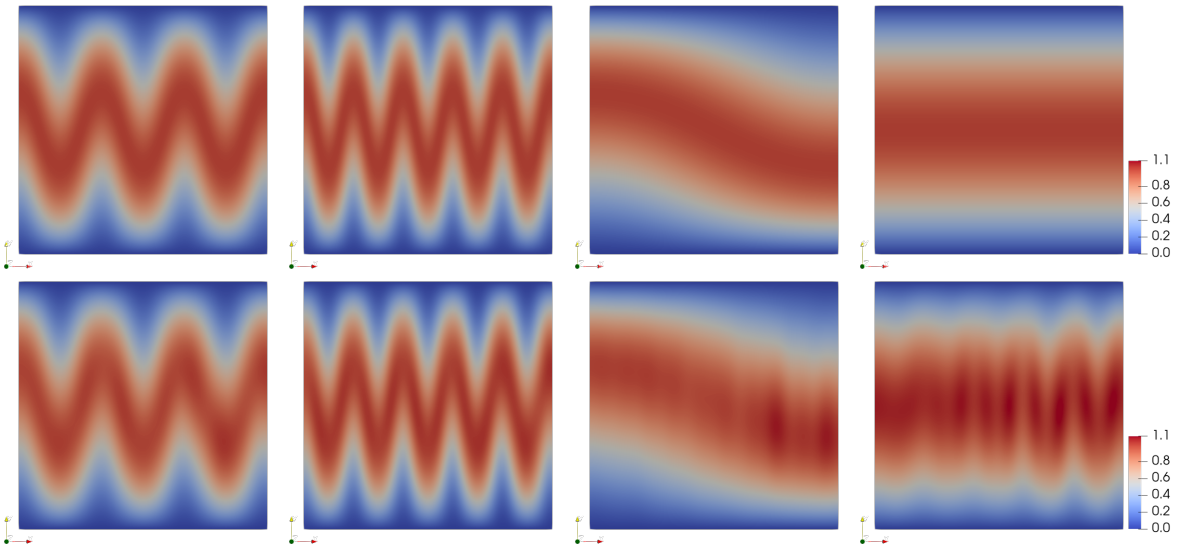


Figure 3: Comparisons of FE solutions to the predicted solutions given by the constructed GP-based ROM. The first row gives the FE solutions. The second row gives the predicted solutions from the GP-based ROM. The columns from left to right correspond to testing input positions $(m, \alpha) = (6, 2)$, $(10, 2)$, $(1, 2)$ and $(10, 0)$ respectively.

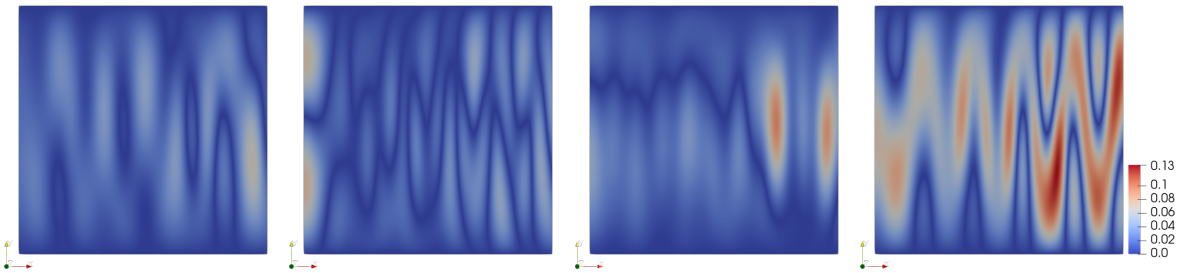


Figure 4: The normalised errors between FE solutions and the predicted solutions from the ROM with GP surrogate. The plots from left to right correspond to testing input positions $(m, \alpha) = (6, 2)$, $(10, 2)$, $(1, 2)$ and $(10, 0)$ respectively.

4 Active learning for Non-intrusive ROM with Gaussian Process Surrogates

4.1 Why Active Learning?

Active learning, also known as sequential design, is a collection of approaches that adaptively enrich the training points for surrogate modelling of computer solvers. In comparison to one-shot designs, such as Latin hypercube designs (LHD), the active learning is preferred in many cases:

- One wants a proper utilisation of computational resources. Active learning allows one to choose computer model input locations adaptively, and therefore can monitor the quality of the resulting surrogate model while the active learning is in progress and determine

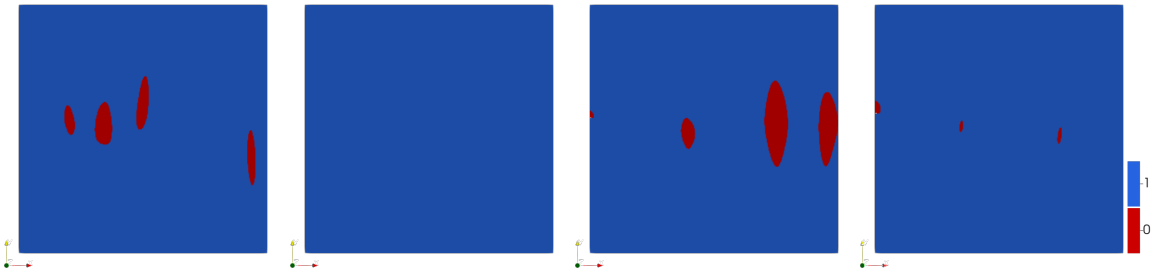


Figure 5: The coverage of constructed ROM with GP, giving the instances that FE solutions fall within the predictive bounds provided by the ROM with GP. 1 indicates that the FE solution is covered by the predictive interval (whose upper and lower bounds are given at two standard deviations $2\widehat{\sigma}$ above and below the predictive mean $\widehat{\mu}$) and 0 indicates otherwise. The plots from left to right correspond to testing input positions $(m, \alpha) = (6, 2)$, $(10, 2)$, $(1, 2)$ and $(10, 0)$ respectively.

whether to pause or continue the model evaluations;

- More computer model evaluations are needed in the input region of interest. Unlike static space-filling designs, such as LHD, active learning, depending on the quality of the underlying surrogate model (as we will discuss in Section 4.4), could direct the computer models to evaluate at input locations where the model response exhibits more variations and thus are more of interest;

- There are existing computer model evaluations, but are potentially large in size and/or not produced with a careful design. It can be computationally inefficient to generate a new design, e.g., a static space-filling design, if one has an existing set of model evaluations because one could utilise the data available. However, it can be both numerically inefficient (e.g., the design formed by the existing data is poor) and computationally burdensome (e.g., the existing data is of large size) to use the whole existing model realisations for surrogate modelling. Thus, one can use active learning to choose training data adaptively from the existing model evaluations from a small design size while at the same time prevent from the numerical instabilities induced by poor designs;

- There is a system of coupled computer models. It has been shown in [16] that active learning is essential to construct Gaussian process (GP) based surrogate models in a computationally efficient and effective manner. Static designs of global inputs can produce poor designs, and thus numerical issues, to sub-models of a computer system, and can also waste computational resources over input regions of sub-models that are not contributing to the global outputs (that correspond to the global input region of interest).

4.2 Implementation

Assume that we have data $\mathcal{D}_n = \{\mathbf{X}_n, \mathbf{Y}_n\}$ that consists of input $\mathbf{X}_n \in \mathbb{R}^{n \times D}$ and the responding high-dimensional computer model output $\mathbf{Y}_n \in \mathbb{R}^{n \times K}$. Then, a generic active learning procedure

that selects the next input position $\mathbf{x}_{n+1}$ to be evaluated by the computer model for refinement of GP based non-intrusive ROM (abbreviated as GP-ROM in the remainder of the report) introduced in Section 3 is given in Algorithm 1. Once $\mathbf{x}_{n+1}$ is determined, one can then obtain the augmented data $\mathcal{D}_{n+1} = \{\mathbf{X}_{n+1}, \mathbf{Y}_{n+1}\}$ by concatenating $\mathbf{x}_{n+1}$ and its corresponding high-dimensional output $\mathbf{y}_{n+1}$ to $\mathcal{D}_n$ and update GP-ROM $\{\mathcal{GP}_l\}$ by re-invoking Algorithm 1.

Algorithm 1 Active learning for GP-ROM

Input: (i) $\mathcal{D}_n = \{\mathbf{X}_n, \mathbf{Y}_n\}$; (ii) a candidate set $\mathcal{C}$ of input locations $\{\mathbf{x}_i\}_{i=1,\ldots,M}$.

Output: The next input position $\mathbf{x}_{n+1}$ to be evaluated by the computer model.

1: Compute the low-dimensional output $\widehat{\mathbf{Y}}_n \in \mathbb{R}^{n\times L}$ of $\mathbf{Y}_n$ and the corresponding eigenvalues $\lambda_{l=1,\ldots,L}$ using POD;
2: Construct GP-ROM $\{\mathcal{GP}_l\}$ using $\{\mathbf{X}_n, \widehat{\mathbf{Y}}_n\}$;
3: Calculate the criterion $I_l(\mathbf{x})$ at each input locations in $\mathcal{C}$ using $\mathcal{GP}_l$ for all l;
4: Choose for the next input position $\mathbf{x}_{n+1}$ by solving

$$\mathbf{x}_{n+1} = \underset{\mathbf{x}\in\mathcal{C}}{\operatorname{argmax}} \sum_{l=1}^{L} w_l I_l(\mathbf{x}) \quad \text{with} \quad w_l = \frac{\lambda_l}{\sum_{i=1}^{n}\lambda_i}$$

We present two candidates for the criterion $I_l(\mathbf{x})$ based on the Active Learning MacKay (ALM) [15] and the Active Learning Cohn (ALC) [2] respectively for the selection of $\mathbf{x}_{n+1}$. ALM aims to find the next input location that corresponds to the maximum predictive variance exhibited by the GP-ROM. Thus, $I_l(\mathbf{x})$ is defined by

$$I_l(\mathbf{x}) = \widehat{\sigma}_l^2(\mathbf{x}),$$

where $\widehat{\sigma}_l^2(\mathbf{x})$ is the posterior predictive variance of $\mathcal{GP}_l$ evaluated at $\mathbf{x}$. However, ALM has a well-know issue that it selects excessive input locations around boundaries of the input region because of the lack of data beyond boundaries. To alleviate this issue, ALC aims to select the input position such that the integrated predictive variance of GP-ROM over the input region is minimised after augmenting $\mathbf{x}_{n+1}$ to $\mathbf{X}_n$. Formally, $I_l(\mathbf{x})$ under ALC is defined by

$$I_l(\mathbf{x}) = -\int_{\mathbf{x}^*\in\mathcal{X}} \widehat{\sigma}_l^2\left(\mathbf{x}^*|[\mathbf{X}_n^\top, \mathbf{x}^\top]^\top\right) \mathrm{d}\mathbf{x}^*.$$

where $\widehat{\sigma}_l^2\left(\mathbf{x}^*|[\mathbf{X}_n^\top, \mathbf{x}^\top]^\top\right)$ is interpreted as the posterior predictive variance of $\mathcal{GP}_l$ evaluated at $\mathbf{x}^*$ given the input data $\mathbf{X}_n$ being augmented by $\mathbf{x}$. It is worth noting that the computation of $\widehat{\sigma}_l^2\left(\mathbf{x}^*|[\mathbf{X}_n^\top, \mathbf{x}^\top]^\top\right)$ does not require evaluations of the associated computer model at $\mathbf{x}$ because the predictive variance of GP does not depend on the output data. In practice, the integral involved in ALC can be approximated by the Monte Carlo integration over a reference set $\mathcal{X}$ (that can be the same as the candidate set $\mathcal{C}$) generated by the LHD. To implement a full active learning procedure, one often starts with a small data set that is generated by a static design, such as LHD, and then execute T iterations of Algorithm 1 to enrich the initial data set with T additional realisations from the computer model.

4.3 Active learning for the GP-ROM emulation of the 2-D anisotropic heat transport model

In this section, we demonstrate how efficiency gains can be made using active learning for the GP-ROM of the FE solver to the 2-D problem described in Section 3.1

4.3.1 Experimental Setup

To initiate the active learning to build GP-ROM, N=20 initial training data points, whose input locations are generated via the LHD over $m \in [0, 12]$ and $\alpha \in [0, 3]$ with the corresponding 2-D output (that contains $K = 78961$ solution nodes) determined by running the FE solver (implemented in `Firedrake` [21]). We then iterate Algorithm 1 for both ALM and ALC 80 times to augment additional 80 training data points to the initial data set. At each iteration of the active learning, we choose the number of components L (in Line 1 of Algorithm 1) to be retained from POD based on the following criteria:

$$L = \underset{L^* \in \{1,\dots,n\}}{\operatorname{argmin}} \left| \frac{\sum_{i=1}^{L^*} \lambda_i}{\sum_{i=1}^{n} \lambda_i} - 0.9998 \right|,$$

where $\lambda_1 > \lambda_2 > \cdots > \lambda_n$. To take into account the effects of initial data set on the active learning, we repeat both ALM- and ALC-based active learning 10 times, each with a different initial training data set. For the comparison between ALM and ALC, we generate 2500 testing data points over $m \in [0, 12]$ and $\alpha \in [0, 3]$ and compute the Normalised Root Mean Squared Error (NRMSE) at each active learning iteration by

$$\text{NRMSE} = \frac{1}{2500} \sum_{i=1}^{2500} \frac{\sqrt{\frac{1}{K}(\widetilde{\mathbf{z}}_i - \mathbf{z}_i)(\widetilde{\mathbf{z}}_i - \mathbf{z}_i)^\top}}{\max(\mathbf{z}_i) - \min(\mathbf{z}_i)} \times 100\%,$$

where $\widetilde{\mathbf{z}}_i \in \mathbb{R}^{1\times K}$ and $\mathbf{z}_i \in \mathbb{R}^{1\times K}$ are 2-D FE solution fields generated by the GP-ROM and `Firedrake` at the i-th testing input location, respectively. $max(\mathbf{z}_i)$ and $min(\mathbf{z}_i)$ are maximum and minimum values of $\mathbf{z}_i$ for a given i.

In terms of implementation, we construct GP-ROM and compute corresponding ALM and ALC criteria at each iteration of the active learning using the `laGP` package in R.

4.3.2 Experimental Results

Figure 6 presents the NRMSEs of GP-ROMs built with ALM- and ALC-based active learning over 80 iterations, in comparison to those constructed with the static LHD at various design sizes. It can be observed that for design size less than 50, GP-ROMs trained using the active learning, regardless of ALM or ALC, provide higher accuracy than those trained using the static LHD. However, as the design sizes increases, the accuracy of GP-ROMs built by the active learning and LHD are comparable. This is because with a large design size, the input domain is densely space-filled by the LHD and thus the NRMSE of the corresponding GP-ROM converges to that of the GP-ROM trained with the active learning.

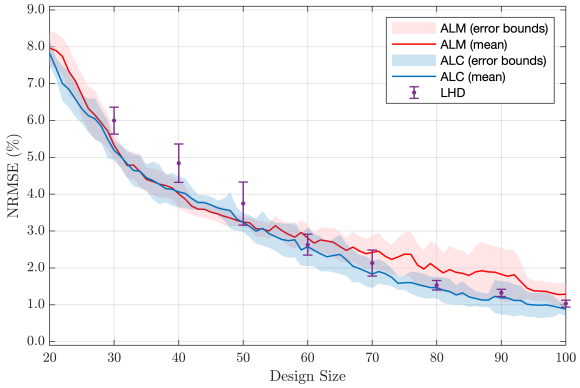
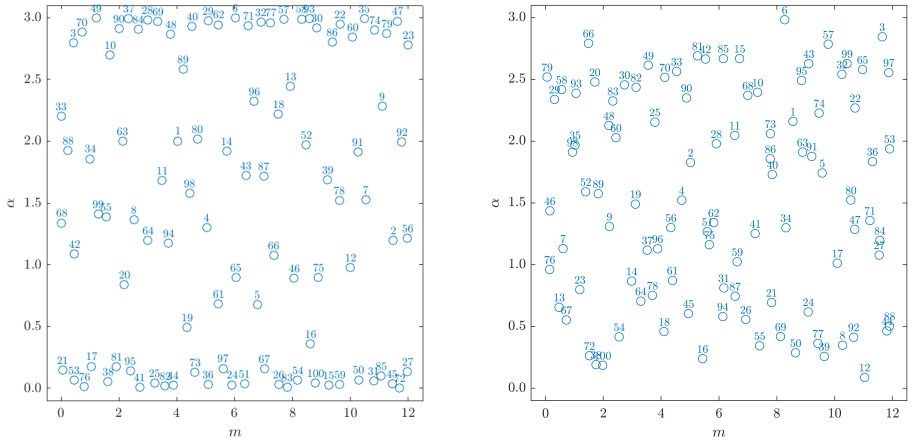


Figure 6: Comparison of NRMSEs of GP-ROM constructed using the ALM-based active learning, the ALC-based active learning, and the static LHD.

We also observe from Figure 6 that for design size larger than 60 GP-ROMs constructed by LHD perform better (in terms of overall lower NRMSE) than those built by ALM-based active learning. This observation can be explained by the fact that ALM-based active learning has the tendency to choose excessive input locations around boundaries of the input domain (see Figure 7(a)) and thus could fail to achieve a satisfactory design, in which input locations are preferred to be scattered within the input domain of interest (see Figure 7(b)).



(a) ALM (b) ALC

Figure 7: Designs produced by a random trial (out of 10 repeated trials) of ALM- and ALC-based active learning.

4.4 Discussion

In this section, we introduce a simple and effective procedure to implement the active learning for GP-ROM construction. Although the active learning may eventually produce a space-filling design, it gives the computer model experimenters more controls over their computational resources. One may criticise that active learning is not computationally efficient in the sense that it directs model runs sequentially and thus can be time-consuming in comparison to static

one-shot designs in which model runs can be done in parallel. This statement is sensible when one possesses sufficient computational power (for parallel computing) and active learning also produces a space-filling design. However, in real-world data these conditions may not be fulfilled. Our computational resources may not permit us to obtain model realisations that cover adequately the input region of interest (for an accurate surrogate model) and a space-filling design may not capture sufficiently (without tremendous computational efforts) the input regions where the model response exhibits abrupt changes, even if we have an advanced surrogate model (that is suitable for both stationary and non-stationary data). On the contrary, active learning has the ability to focus on input regions where the corresponding output surfaces show more variations, given that the underlying surrogate model provides a satisfactory uncertainty quantification (e.g., highlighting the regions with higher predictive standard deviations). A fact often forgotten in computer model experiments is that design and surrogate modelling are not two separate tasks. Good designs produce good surrogates with less numerical issues and more reliable uncertainty quantification, which in turn induces designs that better represent the functional behaviours of computer models under the consideration. These are the reasons why active learning could be preferred to static space-filling designs, which could cause the surrogate modelling difficulty (e.g., a large number of realisations that are needed to capture well the computer model can make the GP-ROM computationally prohibitive) and do not utilise the uncertainties quantified by surrogate models for design improvement.

It is worth noting that active learning does not guarantee the locations of (possibly very small but important) input regions of a computer model that correspond to abrupt changes to the model responses. The design produced by the active learning depends on the quality of the underlying surrogate model, which in turn depends on the information contained in the training data (assuming that the surrogate represents the training data adequately and produces sensible uncertainty quantification). Therefore, whether active learning could find input regions that has very localised and important features depends on whether the information of the regions exists in the training data. For this reason, it is vital to have a good initial design that incorporates such information for the active learning. However, in practice this can be difficult to achieve, particularly for high-dimensional cases, even we have some prior knowledge that such non-stationary features exist in the computer model, and as a consequence we may obtain a surrogate that completely ignores these regions with significant computational costs being wasted. To alleviate this issue, one could simply evaluate the computer model with a high-resolution design using the parallel computing. In this way, the local behaviours of a computer model can be captured within a reasonable amount of time. Nevertheless, it is not advisable to use all model evaluations for surrogate modelling, especially for GP-based surrogates because the large amount of data can cause GP surrogates computationally prohibitive and some evaluations (e.g., that form a flat response surface) are redundant for surrogate improvement. As a result, we propose the following hybrid static-active learning procedure to address the scenario in which we aim to construct efficiently (in terms of computation and time) a surrogate model that could mimic the underlying computer model with localised behaviours:

1. Generate a data set by evaluating the computer model over a dense space-filling design in

parallel;

2. Choose a subset of the produced data set as the initial design for the active learning;

3. Implement the active learning that adaptively refines the design and the surrogate model, e.g., GP-ROM, by selecting data points from the data set produced in Step 1.

There are several benefits provided by the above procedure. Firstly, the high-resolution design provides some guarantees that our data contain information of localised behaviours embedded in the underlying computer model. In addition, unlike typical active learning that evaluates models sequentially, active learning in Step 3 uses the data set already generated with a parallelisable strategy and thus could save a considerate amount of time (especially when computer models are very expensive to run). Furthermore, with active learning one is able to pick (potentially a small amount of) data points (from the generated data set) that contribute most to the surrogate quality, instead of naively pouring the whole data set into the surrogate construction (causing computational difficulties). Perhaps the most decisive and challenging step of the above procedure is Step 2 because, as discussed, one expects to incorporate some information of localised behaviours of a computer model into the initial design such that the resulting surrogate is less likely to overlook these features. How to integrate experts' knowledge about the localised features into the initial design is worth exploring in the future, but the procedure above indicates a potentially brutal but simple implementation for Step 2: choose multiple random subsets of the data set, then proceed to Step 3 for multiple surrogate constructions, and choose the surrogate that gives the best predictive accuracy (e.g., lowest overall predictive error against the generated data set). This implementation is computationally efficient because active learnings in Step 3 initiated by different random designs can be executed in parallel and do not involve computer model evaluations.

5 Future Directions

We demonstrate in this report that a GP-ROM could be used to replace computationally expensive computer solvers for problems with high-dimensional output, in one of the building blocks of nuclear fusion modelling. However, dimension reduction techniques such as POD lose information when the original data are projected onto a lower dimensional space, and thus some extreme but important events could be masked in the low dimensional data, a scenario called the masking effect. As a result, if the surrogate is built on the low dimensional data one may not be able to recover these outlying events using the constructed non-intrusive ROM. Therefore, other dimension reduction methods that may be more resistant to the masking effect could be examined. In addition, although GP-ROM requires no domain knowledge and access to the source code of original problems, it ignores the physics implied by the underlying problem and thus may be inaccurate comparing to the its intrusive counter-party. Therefore, it would be worth exploring the trade-off between the speed and accuracy of intrusive and non-intrusive MOR, especially in context of UQ. It would also be interesting to find a middle ground where one could exploit the benefits (e.g., accuracy, speed and uncertainty) of both intrusive and non-intrusive ROM, producing a physics-informed non-intrusive ROM. Some relevant literature on physics-informed

machine learning (say using a boundary condition or other approaches) include [27, 11, 28]. To give one example, the use of known boundary conditions imposed by the physics can constrain the emulator much further as it constitutes a sort of infinite number of points on a boundary, compared to the small and expensive sampling of the simulator used to build the emulator. For Reduced Order Modelling, use of symmetries, lower limits (e.g. positive values), and boundaries could further help reduce the complexity of the problem and improve the ROM itself. It is definitely future work, not existing research.

Recommendation: Investigate how to apply physics-informed GP-ROM in key nuclear fusion models. Examine how to build new types of GP-ROM for the case of particle-based models (PIC) whose outputs need to be understood as a continuum.

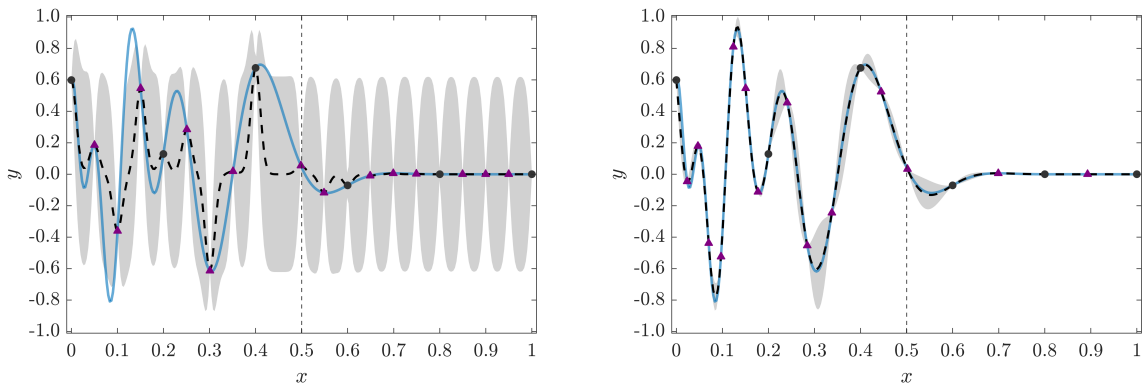
5.1 Deep GP for Non-intrusive ROM

In Section 4 we explored how to construct GP-ROM using active learning. Active learning is particularly useful when the underlying computer model exhibits non-stationary features as it has the ability to produce a non-uniform design that accommodates the non-stationarity. However, the success of the active learning relies on the quality of uncertainty quantified by the surrogate model. Since conventional GP surrogates assume stationarity, more advanced non-stationary GP models, such as deep Gaussian processes [4], would be good candidates for non-intrusive ROM of fusion models that exhibits non-stationarity. Deep Gaussian processes (DGPs) are feed-forward compositions of conventional stationary GPs with flexible model expressiveness, particularly for non-stationary data. However, training and prediction of DGP based emulators are challenging due to the non-linearity induced by the kernel functions involved in GPs. Various inference methods thus are introduced to tackle this issue. Variational inferences, such as Doubly Stochastic Variational Inference (DSVI) [23], is computationally thrifty but is not accurate because simplified assumptions over the latent variables in DGP hierarchy are assumed. On the contrary, the fully-Bayesian approach introduced by [24] gives a comprehensive uncertainty quantification of DGPs, but at the expense of computation. The stochastic imputation approach recently proposed by [17] is a DGP inference method that enjoys both computational speed and the predictive accuracy, and could be a competitive and potential candidate for DGP emulations of non-stationary fusion models. See Section 3 in [17] for comparisons between stochastic imputation, DSVI and fully-Bayesian method in terms of computational efficiency on DGP emulations. The approach is implemented in the `dgpsi` package[1].

Figure 8 showcases the ALM-based active learning using a two-layered DGP surrogate (i.e., composition of two stationary GPs with zero means and squared exponential kernels) trained with the stochastic imputation in comparison to that using a stationary GP (with zero mean and squared exponential kernel) trained with maximum likelihood approach implemented in the `R` package `RobustGaSP`. It can be observed that DGP surrogate outperforms the GP surrogate in both mean predictions and uncertainty quantification. In particular, we note that the predictive uncertainties produced by the stationary GP do not provide useful information to distinguish the rough region of the function from the flat region over $(0.5, 1]$. This is because the correlation

[1] https://github.com/mingdeyu/DGP

between the outputs of two points specified in a stationary GP is determined by the distance between the input positions of the two points. As a result, an input position in the rough region will have a similar predictive variance with that in the flat region if the two input positions have similar distances to the input positions of the training points. In addition, with DGP the active learning could produce a non-uniform design that accommodates the non-stationarity of the underlying data. While the active learning essentially produces a quite space-filling design under GP, it assigns three time more number of design points to the rough (and more interesting) regime over $[0, 0.5]$ than the flat regime over $(0.5, 1]$ under DGP. Although this is a simple 1-D example, it gives motivations why DGP surrogate should be seriously considered if the reduced-order output of a fusion model is non-stationary and the active learning is employed. In this illustrative example, we only considered the ALM-based active learning to construct the DGP surrogate using the stochastic imputation because ALC-based active learning is much more computationally intensive due to complicated calculations of integrated predictive variances under the DGP hierarchy. How to implement ALC-based active learning for DGP surrogates under the stochastic imputation efficiently and effectively is a promising future research direction. Applications of ALC-based active learning for DGP surrogates using the fully-Bayesian approach are recently explored by [24].



(a) GP (b) DGP

Figure 8: ALM-based active learning using GP and DGP emulations. Solid line represents the underlying true function; Dashed line is the mean prediction; Shaded area represents 95% predictive interval; Dots (6 in total) are initial training points and triangles (14 in total) are training points enriched by the active learning procedure using GP and DGP surrogates. The vertical dashed line indicates a visual split of the underlying true function into a rough regime over $[0, 0.5]$ and a flat regime over $(0.5, 1]$.

Recommendation: Investigate how to reduce dimensionality of outputs for key nuclear fusion models whose behaviour may present sharp transitions or various regimes, such as turbulence models. The key question is then how to understand and represent the continuum of outputs features across regimes. Indeed these features shown in Section 3 can vary across regimes and must be made consistent by some form of joint augmentation possibly at a small cost but with large benefits for emulation.

5.2 Active subspace for efficient dimension reduction of inputs

The efficiency of reducing dimensions in the inputs was demonstrated in [14]. Gains of orders of magnitude can be achieved. For instance, the application to a surface of inputs (a mesh of 3200 elements) enabled a reduction from dimension 3200 to 5 with fast and accurate emulation. Only about 100 simulations were needed to come up with 5 key dimensions as a recombination of the original 3200 dimensions. A summary of the method is presented below. It is implemented in the Alan Turing Institute Package *Multi-Output Gaussian Process Emulator* (MOGP)[2]. The context is:

- Simulator input X (high dimension $\mathbb{R}^p$) and output $Y = f(X)$ (one dimension $\mathbb{R}^1$)
- GP emulation: fit a GP and predict $f(x_{new})$ using a sample of simulations $f(X_1), ..., f(X_n)$
- Find a reduced space (known as sufficient dimension reduction SDR) $R(X) \in \mathbb{R}^d$, $d < p$, such that there is (nearly) no loss of information in predicting Y by providing $R(X)$ instead of X
- To achieve SDR, employ the gradient-based Kernel Dimension Reduction (gKDR) approach [6]:

  $$R(X) = B^T X, \quad B^T B = I_d, \quad d < p.$$

  Estimate B from simulation samples $(X_1, Y_1), ..., (X_n, Y_n)$. Note that no strong assumption are made on the variables (type, distribution, dimension).

The specific technical steps in gKDR involve two Reproducing kernel Hilbert spaces (RKHS):

- Prepare kernels $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$, with the associated (RKHS) $\mathcal{H}_{\mathcal{X}}$ and $\mathcal{H}_{\mathcal{Y}}$
- The quantities of interest are the gradients $\frac{\partial E[g(Y)|X]}{\partial X}$ for any $g \in \mathcal{H}_{\mathcal{Y}}$ as their evaluation is the ingredient for the identification of the reduced subspace, by looking at the most influential directions.
- Estimate (see [6] for details)

  $$\hat{M}_n = \frac{1}{n} \sum_{i=1}^{n} \nabla \mathbf{k}_X(X_i)^T (G_X + n\epsilon_n I)^{-1} G_Y (G_X + n\epsilon_n I)^{-1} \nabla \mathbf{k}_X(X_i)$$

  where G_X and G_Y are the Gram matrices $(k_{\mathcal{X}}(X_i, X_j))$ and $(k_{\mathcal{Y}}(Y_i, Y_j))$, and $\nabla \mathbf{k}_X(x) = (\partial k_{\mathcal{X}}(X_1, x)/\partial x, ..., \partial k_{\mathcal{X}}(X_n, x)/\partial x)^T \in \mathbb{R}^{n \times m}$ for any $x \in \mathbb{R}^m$.
- Eigen-decompose $\hat{M}_n$ into $\hat{M}_n = \hat{Q}\hat{\Lambda}\hat{Q}^T$ and partition

  $$\hat{\Lambda} = \begin{bmatrix} \hat{\Lambda}_1 & \\ & \hat{\Lambda}_2 \end{bmatrix}, \quad \hat{Q} = [\hat{B} \;\; \hat{C}],$$

  where $\hat{\Lambda}_1 = diag(\hat{\lambda}_1, ..., \hat{\lambda}_d)$ consisting of the first d largest eigenvalues, to ultimately provide the dimension reduction.

[2] https://github.com/alan-turing-institute/mogp-emulator

The emulation with dimension reduction can be carried out and its loss quantified [14]:

- $f(X) \approx \hat{f}(\hat{B}^T X)$

$$\left\| f - \hat{f} \right\|_{L_2} =$$
$$\mathcal{O}_p \left(\frac{4}{\lambda_d - \lambda_{d+1}} n^{-\min\{\frac{1}{3}, \frac{2\xi+1}{4\xi+4}\}} \left(\sum_{i=1}^{d} b_i \hat{\lambda}_i^2 \right)^{\frac{1}{2}} + \left(\sum_{i=d+1}^{m} b_i \hat{\lambda}_i^2 \right)^{\frac{1}{2}} \right)$$

  where ξ and the b_i's are positive constants.

- Build emulator $\tilde{f} \approx \hat{f}$ on low dimensional space $\hat{B}^T X$

- Approximation procedure:

$$f(X) \approx \hat{f}(\hat{B}^T X) \approx \tilde{f}(\hat{B}^T X)$$

The choice of retained dimension d and hyperparameters is performance based (e.g. in the quality of the predictions in a leave-one-out strategy) and can result in very large gains [14].

Recommendation: Investigate how to reduce dimensionality of inputs of key nuclear fusion models such as the magnetic field modelled input of the anisotropic heat transfer model.

5.3 Linked GP for Non-intrusive ROM

Since fusion models are often multi-disciplinary and multi-physics, the recent advances on linked Gaussian process surrogates [16] must be considered. The linked GP can be seen as a DGP whose latent layers are fully observable (see [17] for details), and is implemented in the `dgpsi` package[3]. As an illustration, consider a toy system that consists of two feed-forward connected computer models shown in Figure 9. By directly applying conventional GP, one fails to capture the extreme local feature (over $[-1, 1]$) of the underlying system with ten system runs, see Figure 10(d). Similarly, the linked GP emulation with the static design, in which GP surrogates of individual sub-models are built using the training points propagated through the system, cannot reproduce the local feature of the system over $[-1, 1]$ either, see Figure 10(c). This is because the space-filling property of the static design for f_2 is lost (see Figure 10(b)) when the well-spaced design for f_1 propagates through f_1, which has a steep transition over $[-1, 1]$. However, if the linked GP is employed with the active learning to the system (see Figure 11(a) and 11(b)), one could recover the extreme local feature of the overall system sufficiently (see Figure 11(c)). Since the extreme local feature of the entire system over $[-1, 1]$ is created by the composition of simpler individual sub-models, constructing a system surrogate on the basis of elementary emulators could achieve better emulation performance, in comparison to a GP surrogate of the whole system. Besides, using the active learning one could optimise the designs for individual sub-models, and thus obtain better corresponding GP surrogates, which in turn produce a system surrogate with higher accuracy with less likelihood of missing extreme local features.

[3] https://github.com/mingdeyu/DGP

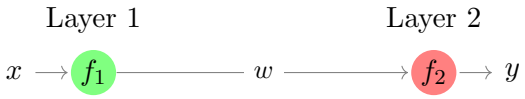
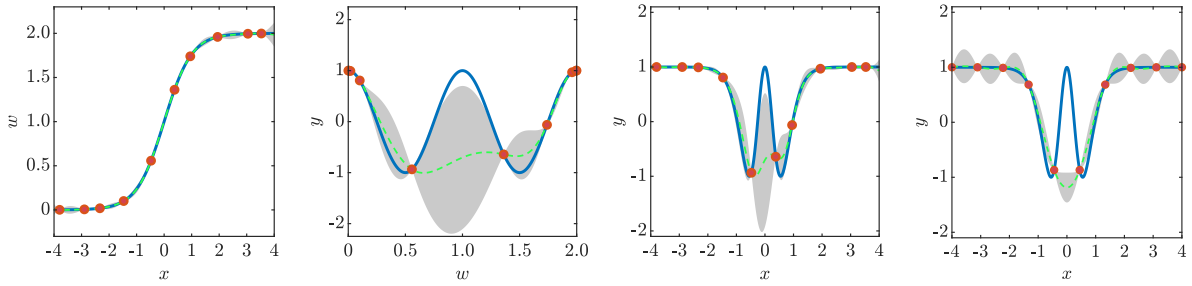
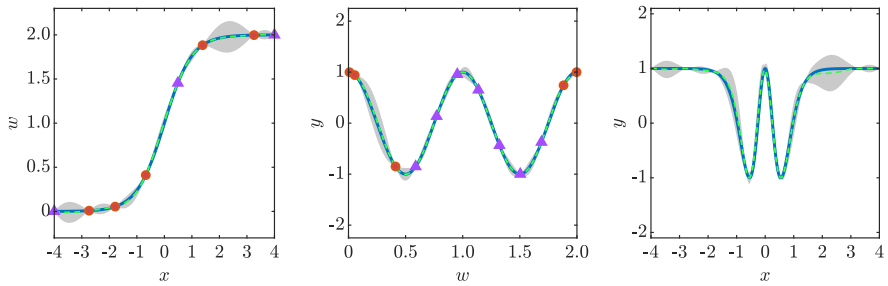


Figure 9: An illustrative example of a system of two computer models f_1 and f_2. Note this is only for illustration. Linked GP in [16] can work on any feed-forward computer systems.



(a) GP of f_1 (b) GP of f_2 (c) Linked GP of $f_2 \circ f_1$ (d) GP of $f_2 \circ f_1$

Figure 10: Linked GP (in (c)) and stationary GP (in (d)) emulators (with the static design) of a feed-forward system ($f_2 \circ f_1$) of two computer models f_1 and f_2 connected as shown in Figure 9. The filled circles are training points; the solid line is the underlying true function; the dashed line is the mean prediction; the shaded area represents 95% prediction interval.



(a) GP of f_1 (b) GP of f_2 (c) Linked GP of $f_2 \circ f_1$

Figure 11: Linked GP (in (c)) emulators (with the active learning) of a feed-forward system ($f_2 \circ f_1$) of two computer models f_1 and f_2 connected as shown in Figure 9. The filled circles in (a) and (b) represent the initial design of the active learning to build GP surrogates of individual computer models f_1 and f_2 for linked GP in (c); the filled triangles in (a) and (b) are training points created by the active learning; the solid line is the underlying true function; the dashed line is the mean prediction; the shaded area represents 95% prediction interval.

The toy example motivates further explorations of linked GP in constructing non-intrusive ROM for fusion systems by linking non-intrusive ROM of individual sub-models. For example, to construct the ROM of the two-layered system in Figure 9, one could first build GP-based non-intrusive ROM (as demonstrated above) for all individual sub-models (f_1 and f_2) and then construct the non-intrusive ROM of the whole system by linking the non-intrusive ROM of f_1 to that of f_2 through the reduced space w analytically. One key benefit of this approach for system-wise reduced order modelling is that one only needs to do dimensionality reduction to the outputs of sub-models. Whereas, to build intrusive ROM, one has to make extra challenging efforts to reformulate the original high-fidelity model f_2 under reduced input w and output y.

Implementing the active learning for linked GP surrogates for systems of computer models with high-dimensional outputs is also challenging. In comparison to the static design (in which the training input data of one sub-model matches the training output data produced by the feeding sub-models), the active learning (e.g., the adaptive design introduced in [16]) could lose the input/output data matching (i.e., training input data of one sub-model may not match the training output data of the feeding sub-models because in each iteration of the active learning only one sub-model rather than the whole system is executed), and thus further explorations are required to examine how to conduct dimension reductions for the internal sub-model input/output so that all information contained in the training data of linked sub-models are utilised.

Recommendation: Investigate how to jointly reduce dimensionality of outputs that are inputs of key nuclear fusion models, such as the heat from the anisotropic heat transfer model propagated to the wall heat transfer model. Emulation with high-dimensional outputs (GP-ROM) of the first simluator and active subspace for dimension reduction of the subsequent inputs of the following simulator should be used in synergy. To establish such a combined strategy will require examining carefully how to weigh variations in outputs of the first model and the influence of inputs for the second. The sampling approach of Section 4 needs to be tailored to this new context as well. It is necessary to carry out such combination of methods and strategies due to the very high dimensions, heavy data transfers, and extremely costly simulations.

References

[1] K.-L. Chang, S. Guillas, et al., *Computer model calibration with large non-stationary spatial outputs: application to the calibration of a climate model*, Journal of the Royal Statistical Society Series C, 68 (2019), pp. 51–78.

[2] D. A. Cohn, *Neural network exploration using optimal experiment design*, Neural networks, 9 (1996), pp. 1071–1083.

[3] K. R. Dalbey, *Efficient and Robust Gradient Enhanced Kriging Emulators*, Tech. Rep. SAND2013–7022, Sandia National Laboratories: Albuquerque, NM, USA, 2013.

[4] A. Damianou and N. D. Lawrence, *Deep Gaussian processes*, in Artificial intelligence and statistics, PMLR, 2013, pp. 207–215.

[5] F. Deluzet and J. Narski, *A two field iterated asymptotic-preserving method for highly anisotropic elliptic equations*, Multiscale Modeling & Simulation, 17 (2019), pp. 434–459.

[6] K. Fukumizu and C. Leng, *Gradient-based kernel dimension reduction for regression*, Journal of the American Statistical Association, 109 (2014), pp. 359–370.

[7] M. Gu and J. O. Berger, *Parallel partial Gaussian process emulation for computer models with massive output*, The Annals of Applied Statistics, 10 (2016), pp. 1317–1347.

[8] M. Gu, J. Palomo, and J. O. Berger, *RobustGaSP: robust Gaussian stochastic process emulation in R*, arXiv:1801.01874, (2018).

[9] M. Gu, X. Wang, and J. O. Berger, *Robust Gaussian stochastic process emulation*, The Annals of Statistics, 46 (2018), pp. 3038–3066.

[10] S. Guillas, A. Sarri, S. J. Day, X. Liu, F. Dias, et al., *Functional emulation of high resolution tsunami modelling over Cascadia*, Annals of Applied Statistics, 12 (2018), pp. 2023–2053.

[11] K. Kashinath, M. Mustafa, A. Albert, J. Wu, C. Jiang, S. Esmaeilzadeh, K. Azizzadenesheli, R. Wang, A. Chattopadhyay, A. Singh, et al., *Physics-informed machine learning: case studies for weather and climate modelling*, Philosophical Transactions of the Royal Society A, 379 (2021), p. 20200093.

[12] K. N. Kyzyurova, *On uncertainty quantification for systems of computer models*, PhD thesis, Duke University, 2017.

[13] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, *When Gaussian process meets big data: a review of scalable GPs*, IEEE transactions on neural networks and learning systems, 31 (2020), pp. 4405–4423.

[14] X. Liu and S. Guillas, *Dimension reduction for Gaussian process emulation: An application to the influence of bathymetry on tsunami heights*, SIAM/ASA Journal on Uncertainty Quantification, 5 (2017), pp. 787–812.

[15] D. J. MacKay, *Information-based objective functions for active data selection*, Neural computation, 4 (1992), pp. 590–604.

[16] D. Ming and S. Guillas, *Linked Gaussian process emulation for systems of computer models using Matérn kernels and adaptive design*, SIAM/ASA Journal on Uncertainty Quantification (in press). ArXiv preprint arXiv:1912.09468, (2021).

[17] D. Ming, D. Williamson, and S. Guillas, *Deep gaussian process emulation using stochastic imputation*, arXiv:2107.01590, (2021).

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, *Scikit-learn: Machine learning in Python*, Journal of Machine Learning Research, 12 (2011), pp. 2825–2830.

[19] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced Basis Methods for Partial Differential Equations: An Introduction*, vol. 92, Springer, 2015.

[20] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*, The MIT Press, Cambridge, MA, 2006.

[21] F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. McRae, G.-T. Bercea, G. R. Markall, and P. H. Kelly, *Firedrake: automating the finite element method by composing abstractions*, ACM Transactions on Mathematical Software (TOMS), 43 (2016), pp. 1–27.

[22] O. Roustant, D. Ginsbourger, and Y. Deville, *DiceKriging, DiceOptim: two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization*, Journal of Statistical Software, 51 (2012).

[23] H. Salimbeni and M. Deisenroth, *Doubly stochastic variational inference for deep Gaussian processes*, in Advances in Neural Information Processing Systems, 2017, pp. 4588–4599.

[24] A. Sauer, R. B. Gramacy, and D. Higdon, *Active learning for deep Gaussian process surrogates*, arXiv:2012.08015, (2020).

[25] M. L. Stein, *Interpolation of Spatial Data: Some Theory for Kriging*, Springer, New York, 1999.

[26] R. Tripathy, I. Bilionis, and M. Gonzalez, *Gaussian processes with built-in dimensionality reduction: applications to high-dimensional uncertainty propagation*, Journal of Computational Physics, 321 (2016), pp. 191–223.

[27] I. Vernon, S. E. Jackson, and J. A. Cumming, *Known boundary emulation of complex computer models*, SIAM/ASA Journal on Uncertainty Quantification, 7 (2019), pp. 838–876.

[28] D. Watson-Parris, *Machine learning for weather and climate are worlds apart*, Philosophical Transactions of the Royal Society A, 379 (2021), p. 20200098.