Uncertainty Quantification in Risk Assessment

Application of surrogate modelling

Progress Report 2

IBM Research UK
STFC Hartree Centre

IBM Research UK
http://research.ibm.com/labs/uk/
STFC Hartree Centre
https://www.hartree.stfc.ac.uk

Title:
Uncertainty Quantification
in
Risk Assessment

Theme:
Surrogate modelling

Participant(s):

Małgorzata Zimoń*
Alexander Hill
Robert Sawko
Wendi Liu

Date of Completion:
April 3, 2020

Summary:

This work focuses on comparison of two surrogate-based methods for quantifying uncertainty in case of a random input with Pareto distribution.
We analyse polynomial chaos expansion and Gaussian process emulation on a simplified problem and steady-state dispersion simulation.
The report provides estimation of potential computational savings when performing UQ with surrogates. The shortcomings and advantages of each technique are highlighted, with the focus on treatment of random variables with heavy tailed probability distributions. We show that that surrogate modelling can be improved when performing parameter space decomposition.

*Corresponding author: malgorzata.zimon@uk.ibm.com

This work was supported by the STFC Hartree Centre's Innovation Return on Research programme, funded by the Department for Business, Energy & Industrial Strategy.

Contents

1 Introduction

The field of uncertainty quantification (UQ) is concerned with the characterisation and reduction of uncertainties present in real world problems. In computer experiments, in order to account for variability of model inputs, we often rely on many replications of simulations for a large number of probable settings. Such approach is often infeasible due to the high computational cost and time needed to obtain required statistics. More promising strategy to avoid this bottleneck is to use a small number of simulation runs to build an approximation to the model, hereafter referred to as *surrogate*, designed to provide accurate predictions for a given set of random parameters. Constructed using model outputs, the surrogate aims to emulate the behaviour of the simulator but at a fraction of the computational cost.

The report focuses on two of the most popular surrogate methods, polynomial chaos expansion (PCE) and Gaussian process emulation (GPE). Both techniques have been developed independently and are often used in non-intrusive UQ analyses [9]. The PCE approach is more common in the engineering and applied mathematics community, and represents the quantity of interest (QoI) using a series expansion of weighted orthogonal polynomials with respect to the model inputs [5]. An alternative surrogate method which has received much attention in the statistical community is Gaussian process emulation. This statistical method treats the simulator as an unknown function that can be modelled as a realisation of a Gaussian process with mean and covariance functions, which are updated using simulation outputs [14].

The main objective of this study is to compare PCE and GPE, in order to provide useful information for practitioners in UQ. We focus on a particular rare event scenario, which is common in risk analysis, where an input parameter is a random variable with a heavy (or fat) tailed distribution[1]. More specifically, the following questions will be addressed: How accurate are the surrogate methods in approximating the simulator and the stochastic moments (mean and variance)? How do the surrogates compare in terms of the computational costs of their construction, their sensitivity to the input distribution?

[1]The probabilities that decline polynomially or slower. The term heavy tailed can be considered as more general than fat tailed; it is used to refer to distributions which have infinite moments.

The report is organised as follows. In section 2, the theory of PCE and GPE is briefly outlined. In section 3, the application and the uncertainty quantification procedure are described, followed by results and discussion.

2 Uncertainty quantification - theory

In this report, we are focusing on *forward* UQ which aims at estimating the influence of parametric variability on the model outputs. Such analysis is a crucial component of simulation-based reliability study. To increase confidence and the utility of modelling predictions, we need to account for all sources of uncertainty. The PCE and GPE approaches provide surrogate models that can significantly reduce the number of simulations required to perform UQ, while still retaining a low degree of error.

Numerous studies have focused on the development of advanced formulations of both GPE and PCE. Some of these new variants, which offer flexibility and computational efficiency, have been reviewed in [10]. For simplicity of the comparison, we use only the basic formulations of the surrogate methods for UQ analysis.

2.1 Polynomial chaos expansion

The quantity of interest, $Y(\omega)$, can be represented as a sum of weighted polynomials:

$$Y(\omega) = \sum_{k=0}^{\infty} y_k \psi_k \left(\xi(\omega)\right), \tag{2.1}$$

where ω is the random event and $\{y_k\}_{k=0}^{\infty}$ is referred to as a set of polynomial chaos expansion coefficients. The basis functions are orthogonal with respect to measure μ, i.e.

$$\langle \psi_k \psi_j \rangle = \int_{\Omega} \psi_k(\xi)\psi_j(\xi)\mu(\xi)d\xi = \gamma_k \delta_{kj}, \quad \text{for} \quad k, j = 0, \ldots, N_p \tag{2.2}$$

where $\langle . \rangle$ is the inner product, δ_{kj} is the Kronecker delta. Therefore, the choice of the basis is imposed by the probability distribution of uncertain inputs. The normalisation constants γ_k are unique to the chosen basis and are known. The list of polynomials whose weights are associated with a particular random variable can be found in [15].

In practice, the series is truncated at $N_p + 1$ terms which represent the main contribution to the output variance. Construction of orthogonal polynomials for arbitrary probability density functions (PDFs) can be performed by exploiting the three-term recurrence relation [12].

Due to the orthogonality of the basis, the stochastic moments, i.e. the expected value, $\mathbb{E}$, and the variance, $\mathbb{V}$, can be expressed in terms of polynomial coefficients:

$$\mathbb{E}\left[Y\right] \left\langle \Psi_0 Y \right\rangle = \sum_{k=0}^{N_p} y_k \left\langle \Psi_0 \Psi_k \right\rangle = y_0, \tag{2.3}$$

$$\mathbb{V}\left[Y\right] = \mathbb{E}\left[|Y - \mathbb{E}\left[Y\right]|^2\right] = \sum_{k=0}^{N_p} y_k^2 \left\langle \Psi_k^2 \right\rangle, \tag{2.4}$$

where $N_p = \frac{(p+d)!}{p!d!} - 1$ and $\Psi(\boldsymbol{\xi})$ are multivariate polynomials and $\boldsymbol{\xi} = \{\xi_1, \ldots, \xi_d\}$. The zeroth order polynomial is a constant and the convention is that $\Psi_0 = 1$ [5]. As the total number of PCE coefficients grows combinatorially (*curse of dimensionality*) as a function of the number of random inputs, d, and total degree of multivariate polynomials, p, practical applications of the expansion are limited to studies with low stochastic dimension and high regularity [12].

Apart from the stochastic moments, functions, such as the cumulative distribution function and PDF, can be obtained by sampling Eq. 2.1. In addition, the model enables analytical derivation of the Sobol' indices through post-processing of the PCE coefficients [11].

2.1.1 Non-intrusive approach

The PCE coefficients can be obtained non-intrusively (without reformulation of the model) either through regression, using least-squares solution to a linear system [9], or non-intrusive spectral projection (NISP method) which projects the simulator output against each polynomial term:

$$y_k = \frac{\left\langle Y \Psi_k \right\rangle}{\left\langle \Psi_k^2 \right\rangle} = \frac{1}{\gamma_k} \int Y(\boldsymbol{\xi}) \Psi_k(\boldsymbol{\xi}) d\mu(\boldsymbol{\xi}), \quad k = 0, \ldots, N_p. \tag{2.5}$$

To effect approximate evaluations, we can invoke quadrature rules. The realisations of Y are used to determine a surrogate model $\tilde{Y} = \sum_{k=0}^{N_p} \tilde{y}_k \Psi_k \approx Y$ for which

$$\tilde{y}_k = \frac{1}{\gamma_k} \sum_{j=1}^{Q} Y\left(\xi^{(j)}\right) \Psi_k\left(\xi^{(j)}\right) w^{(j)}. \tag{2.6}$$

In this expression, the integration points (nodes) $\xi^{(j)}$ and weights $w^{(j)}$ in each dimension are computed using the theory of orthogonal polynomials.

In the present work, we limit our analysis to Gaussian quadrature grids which are not nested. In polynomial chaos, using Q nodes and weights, this quadrature formula can be made accurate for all polynomials of degree at most $2Q - 1$, and no other formula on Q nodes has higher order of accuracy.

If evaluations of the simulator are computationally expensive, Clenshaw–Curtis quadrature rules might be preferable for UQ study. In many circumstances, they have a comparable accuracy to Gaussian quadrature [13].

2.2 Gaussian process emulation

Let us assume observed values $\Theta_n = Y\left(\xi^{(n)}\right)$ for $n = 1, \ldots, N$ and $\boldsymbol{\Theta} = [\Theta_1, \ldots, \Theta_N]$ without any error. We are going to use the data $\left(\xi^{(n)}, \Theta_n\right)$ to make prediction of $Y\left(\xi^{(n)}\right)$ for the unseen points. A natural approach would be to perform interpolation over observations. Most of the interpolations assume that the values Θ_n are recorded exactly. However, it might be more appropriate to incorporate the observational uncertainties in order to obtain probabilistic predictions as opposed to deterministic estimations. Such stochastic interpolation is referred to as Gaussian process emulation.

The GPE method describes the simulator output as a realisation of a Gaussian process defined over the input space [14]. The simulator Y is treated as an unknown function of uncertain inputs that we are trying to evaluate. Based on the Bayesian paradigm, we can specify our prior beliefs about Y using a Gaussian process (GP). Therefore, it can be fully described by a mean function, $m(.)$, which provides the central estimate for predicting the model outputs $\Theta = Y(\boldsymbol{\xi})$, and a covariance function, $(Cov(.,.))$. The mean function should represent a global behaviour of the simulator for the whole input space; it is common to use a mean of the training data, or simple linear trends when no knowledge of the model is available. The covariance function, referred to as *kernel*, describes our expectation of how the simulator output is correlated as a function of two inputs. Therefore, it controls local changes in the model.

We define the covariance as $Cov(.,.,\boldsymbol{\delta})$, where $\boldsymbol{\delta}$ is a set of parameters, referred to as *correlation lengths*, that adjust the strength of the correlation for each input parameter. Longer correlation lengths define a smoother process, while short lengths result in a more *noisy* system. Unless stated differently, we use a Radial Basis Function (RBF) kernel [14] which leads to a long term, smooth rising trends. The choice of the kernel is still an open problem and can be mitigated using the available information about the model.

Having the mean and covariance functions defined, we can express our prior belief about Y as:

$$\Theta \sim GP\left(m(.), Cov(.,.)\right). \tag{2.7}$$

This initial guess can be updated when introducing a data-set D of simulator outputs for new inputs $\xi^{(1)}, \ldots, \xi^{(N)}$ and producing the posterior distribution $GP_{|D}\left(\boldsymbol{\xi}\right)$ conditional on D [14]. In our experiment, we follow a common approach of optimising the hyperparameters governing $m(.)$ and $Cov(.,.,\boldsymbol{\delta})$ by maximising the log-marginal-likelihood using the L-BFGS-B optimisation algorithm [1].

After the correlation lengths have been optimised, the result is a Gaussian process surrogate model, known as an *emulator*. The GPE surrogate can be use to obtain the statistics and the entire PDF of the outputs by performing Monte

Carlo (MC) sampling. The emulator is the mean value of the posterior distribution $GP_{|D}(\boldsymbol{\xi})$, while the deviation from this mean specifies the uncertainty of its predictions. This uncertainty results from a limited number of model evaluations used to construct the estimate. As expected from the interpolatory nature of the method, the variance of the emulator predictions increases rapidly away from design points. The mean, m, and variance, σ^2, at any point ξ can be estimated as

$$m_{|D}(\xi) = (\xi) + C_{\xi D} C_{DD}^{-1} \left(Y(D) - m(D) \right), \tag{2.8}$$

$$\sigma_{|D}^2(\xi) = Cov(\xi, \xi) - C_{\xi D} C_{DD}^{-1} C_{D\xi}, \tag{2.9}$$

where $C_{\xi D}$ is the vector of $Cov\left(\xi, \xi^{(n)}\right)$ for $n = 1, \ldots, N$, $C_{\xi D}$ is the N by N matrix of $Cov\left(\xi^{(i)}, \xi^{(j)}\right)$, and $C_{D\xi}$ is its transpose.

3 Probabilistic analysis

In this section, the methodology and results of comparison of surrogates in rare event modelling are described. The first part of the study, analyses PCE and GPE using a test model introduced in [2]. To isolate the influence of different types of probabilities on the risk assessment, we limit the analysis to a one-dimensional problem. The aim is to understand the limitations and benefits of two surrogate approaches, particularly in case of random variables with heavy tailed Pareto distribution. The second study applies non-intrusive methods to dispersion modelling performed with FLACS [3], where we consider a flange leak event with probabilities taken from the Hydrocarbon Release Database (HCRD) [7]. Both experiments use one-dimensional Gaussian quadrature rules to define inputs for surrogate modelling.

3.1 Validation of surrogates

To evaluate the accuracy of the surrogates, we introduce the following metrics. We assess how close the emulator is to the original simulation by defining a large set of M validation points $\boldsymbol{x} = [x_1, x_2, \ldots, x_M]$, independent to the experimental designs used to build the surrogates. We use relative approximation error defined as

$$\text{Error}_{\text{approx}} = \frac{|Y(\boldsymbol{x}) - \tilde{Y}(\boldsymbol{x})|}{|Y(\boldsymbol{x})|}, \tag{3.1}$$

where $|.|$ denotes L_2 norm, to assess the accuracy across the input space.

In similar manner, we also perform the convergence analysis of resulting statistical moments, mean and STD. If the real values are unknown, as in the dispersion modelling, the error of approximation can be estimated using results from consecutive surrogates constructed with increasing number of simulation outputs. For example, in case of NISP, the convergence error of expected value $\mathbb{E}$ obtained with Q quadrature nodes can be defined as

$$\text{Error}_{\mathbb{E}}^{\text{Q}} = \frac{|\mathbb{E}_Q - \mathbb{E}_{Q-1}|}{|\mathbb{E}_{Q-1}|}. \tag{3.2}$$

3.2 The influence of a heavy-tail on surrogate modelling

Firstly, we consider a toy problem defined in [2] which aims at approximating pressure coefficient values for varying inputs; exact pressures are not computed as the details of the model were not specified in the article. We calculate the following

$$P \propto \sqrt{\frac{\dot{m}}{U}}, \tag{3.3}$$

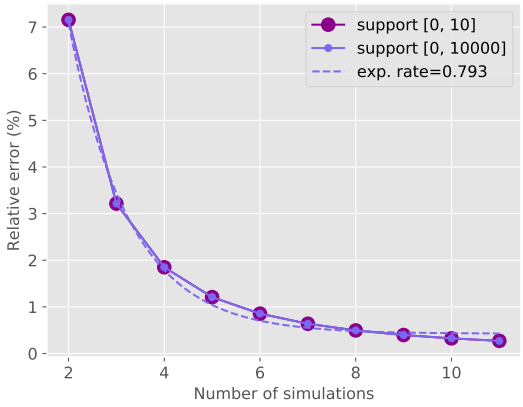
where $\dot{m}$ denotes the leak rate and U the wind speed, which ratio is proportional to pressure P. We treat the leak rate, $\dot{m}$, as a random variable, while the wind speed stays fixed, $U = 10$ m/s.

3.2.1 Modelling with PCE

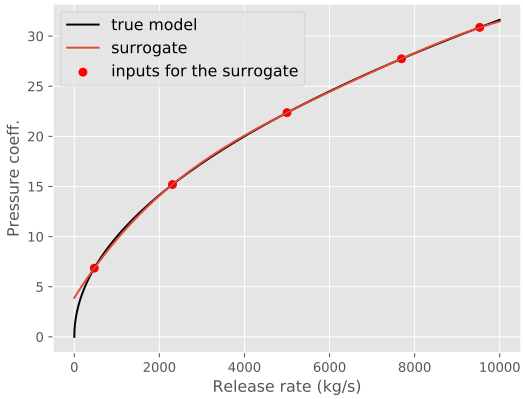
Polynomial chaos expansion aims at obtaining a basis of orthogonal polynomial functionals, which exhibit exponential convergence rates for square-integrable random variables with a specified distribution, for instance Uniform or Normal. This can be shown by defining the leak rate in Eq. 3.3 as a random variable with Uniform distribution. The global relative error between the surrogate and the original model, as defined in Eq. 3.1, decays rapidly, independently of the support of the PDF (see Fig. 3.1a - Fig. 3.1c) verified with $M = 10^4$ equispaced test samples. For smooth response functions without any significant change across the inputs space, the PCE surrogates emulate the true model well for $\dot{m} \sim \text{Uniform}\,(0, 10)$ and $\dot{m} \sim \text{Uniform}\left(0, 10^4\right)$. However, the latter poorly approximates the simulator close to the lower bound of the support. Therefore, we cannot use this model when sampling from Pareto distribution, as the bulk of the values would carry large error.

For Uniform distribution, the low-order statistics, the mean and the standard deviation (STD) of QoI, converge very fast showing the benefit of using surrogate modelling over Monte Carlo (MC) with random sampling. The comparison of the convergence of estimations using these two approaches is depicted in the Fig. 3.2.

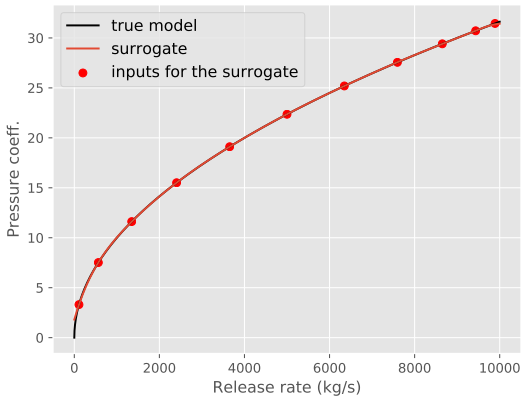
The list of parametric distributions, for which an optimal polynomial expansion exists, includes, among others, the Gaussian, Uniform, Beta and Gamma distribution based on the Askey scheme [15]. For arbitrary distributions, Gram-Schmidt orthogonalisation can be used or, given that the distribution is univariate, it is possible to create orthogonal polynomials stabilised with the three-term recurrence relation [4]. The application of PCE methods to propagate uncertainty caused by heavy-tailed distributions is challenging due to the presence of infinite moments. This can however be circumvented by imposing a cut-off at the boundaries of physical systems to ensure the finiteness of the moments. However, a cut-off extremely far away from the mean can create a huge magnitude difference between the moments. To illustrate the drawback of using PCE in rare event modelling, let us



(a) Approximation errors for different upper bounds of Uniform distribution.

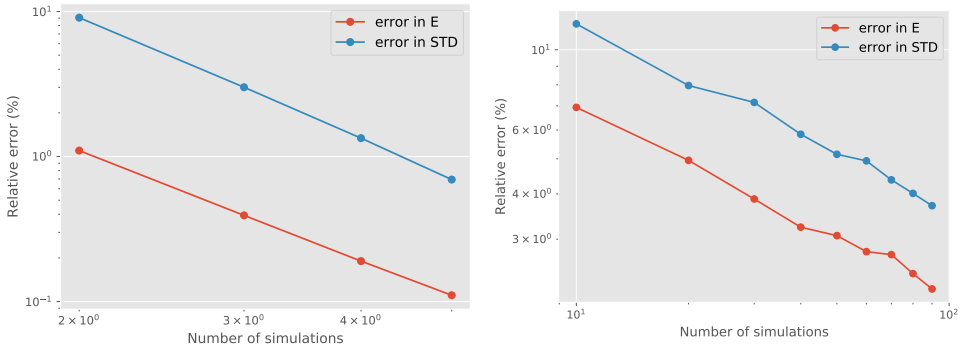


(b) Surrogate for $\dot{m} \sim \text{Uniform}\left(0, 10^4\right)$ obtained using 5 outputs.



(c) Surrogate constructed with 11 simulation points.

Figure 3.1: Analysis of PCE surrogate modelling for random variables with Uniform distribution.



(a) Convergence of PCE surrogate.

(b) Monte Carlo sampling.

Figure 3.2: Comparison of convergence of low-order statistics using PCE and MC for random variable $\dot{m} \sim \text{Uniform}\left(0, 10^4\right)$.

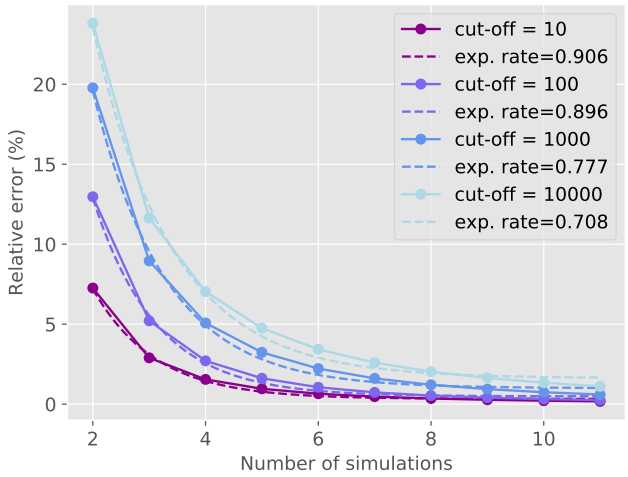
consider the leak rate to have Pareto distribution defined as follows

$$\text{PDF}_{\text{Pareto}} = \frac{\alpha x_m^{\alpha}}{x^{\alpha+1}}, \tag{3.4}$$

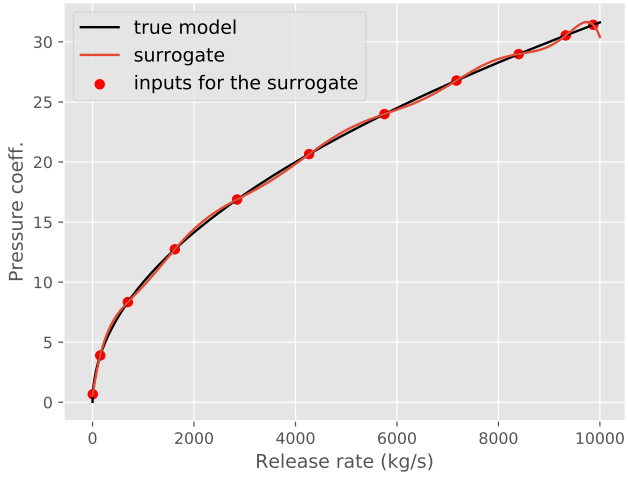
where α is a shape parameter and x_m denotes scale. For the sake of example, we set $\alpha = 2$ and x_m=1.

If we introduce a cut-off close to the mean, a small number of outputs is required to construct an accurate PCE-based surrogate. The approximation error decays exponentially as expected for smooth functions of the random input. However, when the tail of the distribution gets longer, although still exhibiting spectral convergence, the initial error of approximation increases as depicted in Fig. 3.3a, resulting in poorer approximation than in the case of shorter tails (see Fig. 3.3b). Longer tails also affect the estimation of low-order statistics. In case of the Pareto distribution truncated at $\dot{m} = 10$, as expected, the mean converges faster than the STD. The Fig. 3.4 shows that the opposite becomes apparent when the tail increases. With cut-off at $\dot{m} = 10^4$, statistics based on polynomial coefficients obtained from Pareto distribution converge extremely slowly – even after 20 simulations the error in the mean is above 30%.

As mentioned in Sec. 2.1, the PCE of a quantity of interest enables us to estimate the stochastic moments directly from the polynomial coefficients (see Eq. 2.3 and Eq. 2.4). However, we can also simply perform MC sampling using the PCE surrogate. If we want to improve the efficiency of building a surrogate for a random variable with heavy-tailed distribution, we could construct two emulators – one to approximate the variable with Pareto distribution having an upper cut-off close to the mean (short tail), and second which would consider the remaining PDF as plotted in Fig. 3.5a. We could then use the combination of these expansions to perform MC sampling. Experimentation regarding the distribution in simulation runs between the two sub-spaces demonstrated that the primary driver in accurately

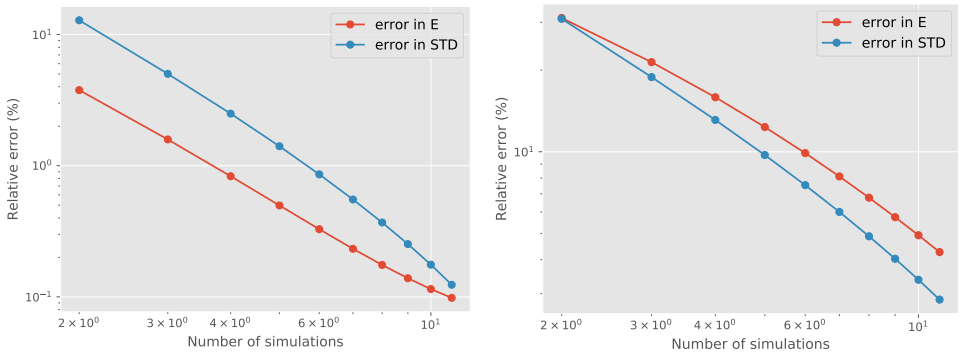


(a) Approximation errors for different cut-offs of Pareto distribution.



(b) Surrogate obtained with 11 simulation points for cut-off at 10^4.

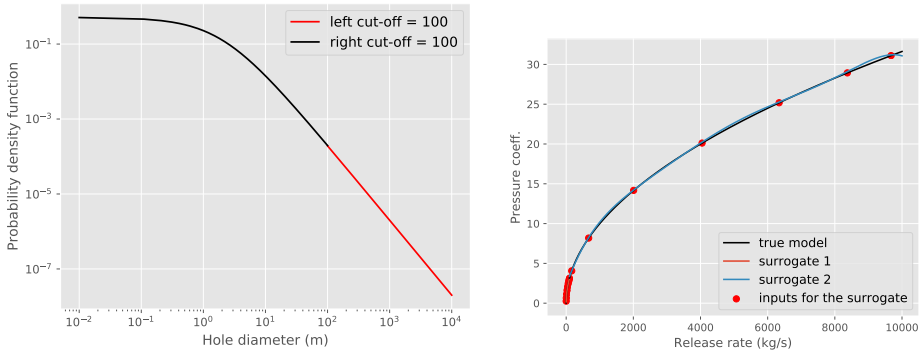
Figure 3.3: The effect of Pareto tail on surrogate modelling with PCE.



(a) Truncated at 10 kg/s.

(b) 1000 kg/s.

Figure 3.4: Comparison of statistical convergence of PCE for increasing tail of Pareto distribution which represents the input leak rate $\dot{m} \geq 10$ kg/s.



(a) Division of Pareto PDF.

(b) Multi-element surrogate.

Figure 3.5: By dividing the Pareto PDF with $\alpha = 2$ and x_m=1 into two parametric sub-spaces, we can significantly reduce the approximation error of the model (below 1% for each sub-model) and its statistics at a fraction of the computational cost.

estimating the mean is resolution in the first sub-space, and in STD is resolution in the second. The *multi-element* surrogate shown in Fig. 3.5b is obtained with 16 simulation runs (nine for the first sub-space and seven for the tail) and provides very small error of low-order statistics – below 1% (with respect to estimations obtained with $M = 10^6$ samples); for the same number of design points, the non-decomposed surrogate estimates the mean with around 35% error.

3.2.2 Modelling with GPE

Gaussian process emulation requires a set of initial simulation runs to update the mean and covariance functions, as outlined in Section 2.2. The input variable values that are chosen for these training runs are known as the *design points*. The selection of these design points is adaptable to the user's main considerations,

though in general it is advisable to ensure that they span the support of the input distribution. Furthermore, if a key interest is reproducing a reasonable realisation of the model's mean and variance, a user should ensure that the region of the support that corresponds to peak of the input random variable's distribution is well sampled by design points. As mentioned before, for the sake of the comparison, we choose our inputs to be Gaussian quadrature nodes estimated based on a given input PDF.

As the model that we are approximating here is smooth, we elect to use an RBF kernel. The mean function is chosen to be a uniform zero mean function. As the initial guess for δ of the RBF kernel, we set the correlation length to be 5% of the input PDF support. The motivation behind this is to make sure that our methodology is as generalised as possible, and easily applicable to supports of various sizes. We use an optimisation algorithm to find values of the kernel's parameters which maximise the log-marginal likelihood, given the training inputs and output. We perform five restarts of this optimiser (with 20 iterations each), where each restart enacts a randomised perturbation onto our initial guess for δ. This methodology ensures that we robustly obtain the optimised value for δ.

Firstly, we emulate the model in Eq. 3.3 assuming the quadrature nodes to be based on Pareto and Uniform distributions (hereafter *Pareto design points* and *Uniform design points*). As shown in Fig. 3.6, we find that the GPE constructed via Pareto design points has a visibly poorer fit to the model than that constructed via Uniform Gaussian quadrature across the support, however its estimation of the mean and STD is notably better (Figs 3.7 and 3.8). This is likely due to the fact that this configuration of design points better samples the parameter space around the mean of the leak rates. Evidence of the poorer general fit in the case of cut-off = 10^4, can be seen in the right panel of these Figures. This suggests that when dealing with Pareto input distributions, in construction of GPE there is a trade-off to be made between sampling the tail or the peak, and thereby the statistical moments or the overall approximation of the surrogate to the model.

We find that the lower the truncation value for the random input variable's support, the quicker the convergence in the mean and STD error. Furthermore, as we have seen in the case of PCE, the error in the estimated mean converges quicker than the STD when dealing with cut-off = 10, while the opposite is seen for cut-off = 10^4. This is likely caused by the fact that with a Pareto distribution, the higher the truncation value, the further the bulk of your surrogate is from the mean. This being the case, the surrogate's predictive power is not focused on the more *important* regions of the support. As seen in the case of PCE, we find that the GPE built with Uniform design points predicts well the global (and smooth) model's behaviour at any cut-off value (displayed in the right panels and Figs. 3.7 and 3.8).

As with PCE in the previous section, we investigate the performance of a multi-

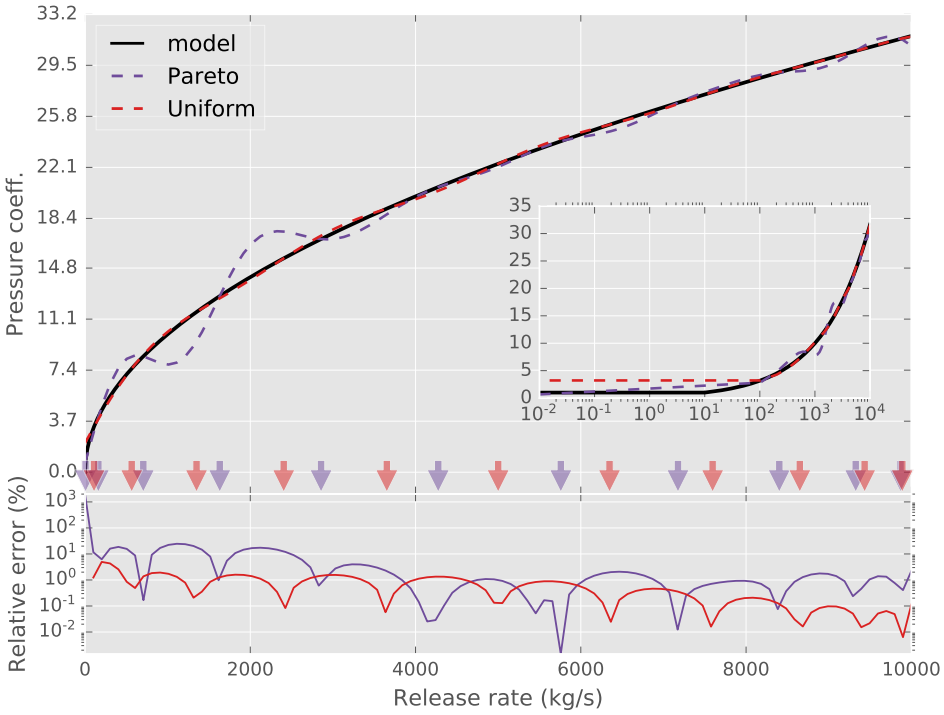


Figure 3.6: GPE surrogate constructed with eleven design points, selected as Uniform and Gaussian quadrature nodes based on a Pareto distribution respectively. Inset plot is identical to the main, except with a logged x-axis. The discrepancy between the two surrogates highlights the sensitivity of GPE to its initial design points. Surrogate is fit to Eq. 3.3.

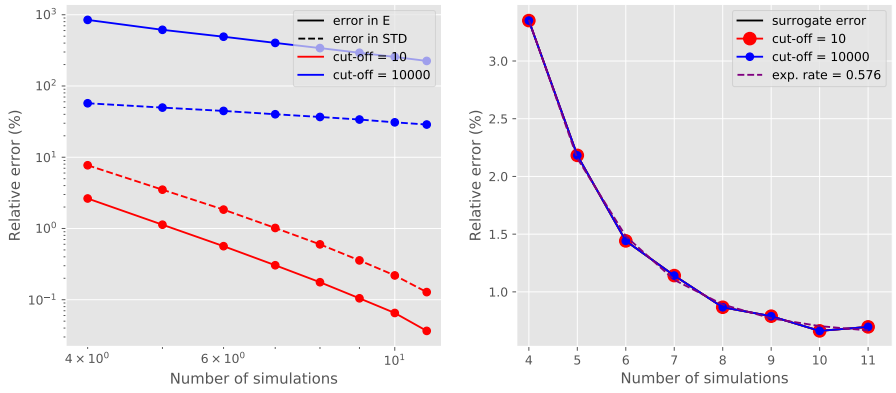


Figure 3.7: Convergence in the mean, standard deviation and approximation errors for a GPE built with Gaussian quadrature nodes based on a Uniform distribution, assuming Pareto random input variables.

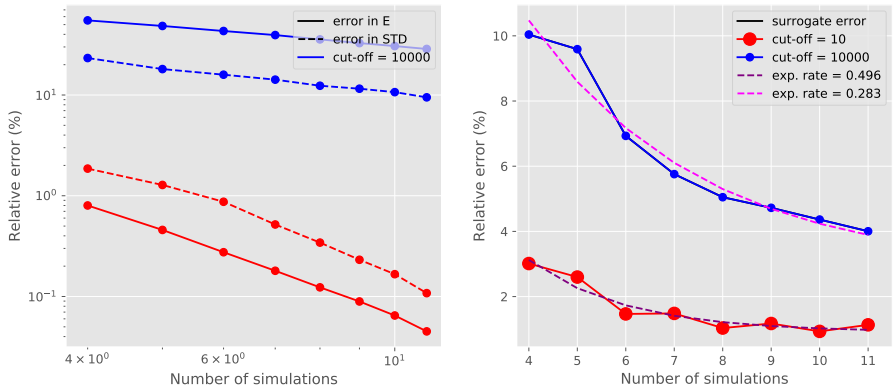


Figure 3.8: Convergence in the mean, standard deviation and approximation errors for a GPE built with Gaussian quadrature nodes based on a Pareto distribution, assuming Pareto random input variables.

element GPE (me-GPE), where we split the support of the input Pareto distribution into *peak* and *tail* components and construct two corresponding GPEs to approximate the model in these regions. The separation point and upper cut-off are set to be 10^2 and 10^4. We obtain 16 design points in total to construct the me-GPE, nine and seven Pareto design points for the two sub-domains respectively. Compared with a standard-GPE (s-GPE) constructed with 16 design points spanning the whole support, we find a significant improvement. The model approximation error is comparable between the me-GPE and s-GPE, 2.9% and 2.7% respectively. However me-GPE is far superior in estimating the statistical moments, with sub-percent errors in the mean and STD, compared with 22% and 7% for s-GPE.

3.3 Surrogate modelling of steady-state dispersion

In this section, we are considering an example of dispersion modelling performed with FLACS (Flame Acceleration Simulator) [3]. The number of accidents in oil and gas terminals can have disastrous impact to both human beings and infrastructure. It is therefore necessary to account for all possible scenarios of leakage associated with a process and/or plant in order to be able to take correct measures and guidelines to mitigate such events. Computational Fluid Dynamics (CFD) models are appropriate to simulate major events with all necessary details.

The present work reports modeling of a steady-state flammable cloud dispersion in a simplified domain shown in Fig. 3.9, which contains a representation of the congestion region with uniformly distributed cylinders. It is assumed that leak occurs in y-direction in the middle of the congested area and its position is fixed, as well as the surrounding conditions. We only vary the hole diameter size of a

leaking flange, resulting in different flow rates of methane, to study its influence on maximum flammable gas volume defined as FLAM (measured in m^3). Here, the entire flammable gas volume is treated as an equivalent stoichiometric cloud [6]. Based on the data from the Hydrocarbon Release Database (HCRD) [7], we consider the diameter size as a random variable with Pareto type-I distribution as shown in Fig. 3.10a The settings of the dispersion scenario are the following. The wind speed is 6 m/s (likely value based on data from *Gexcon-04-F40226-1-Rev01*, plotted in Fig. 3.10b) and direction is at 225° angle. Reservoir pressure and temperature are chosen to be 15 bar and 50°C, respectively. High pressure is chosen to increase the gas size for smaller hole diameters. Otherwise, no cloud is being formed. The atmospheric and wall temperatures are 15°C and 50°C, respectively, while atmospheric pressure is set to 1 bar. The simulations run for 200s. The maximum value of FLAM is computed based on the last 100s.

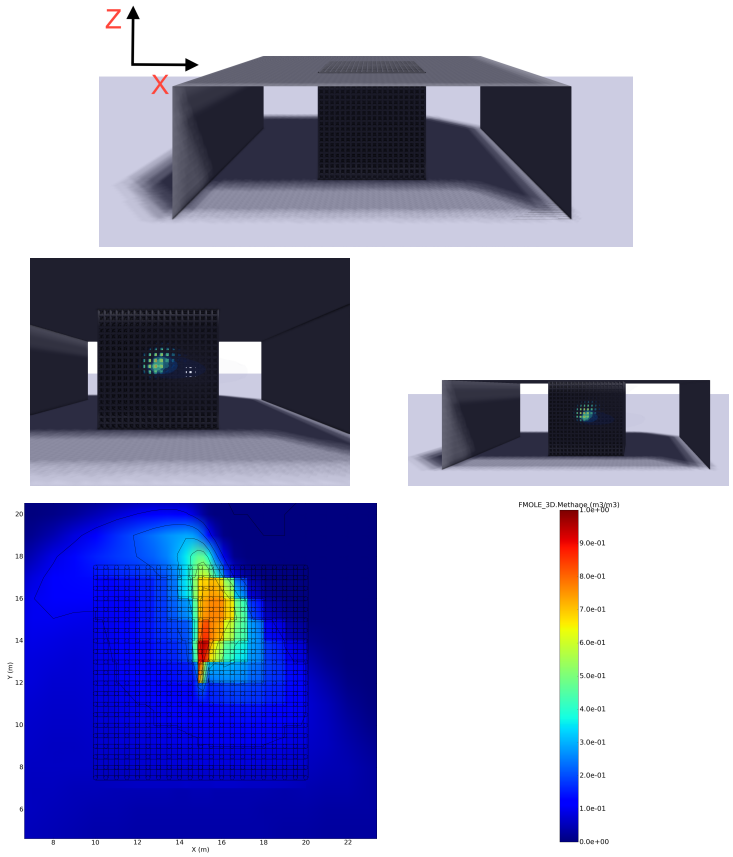
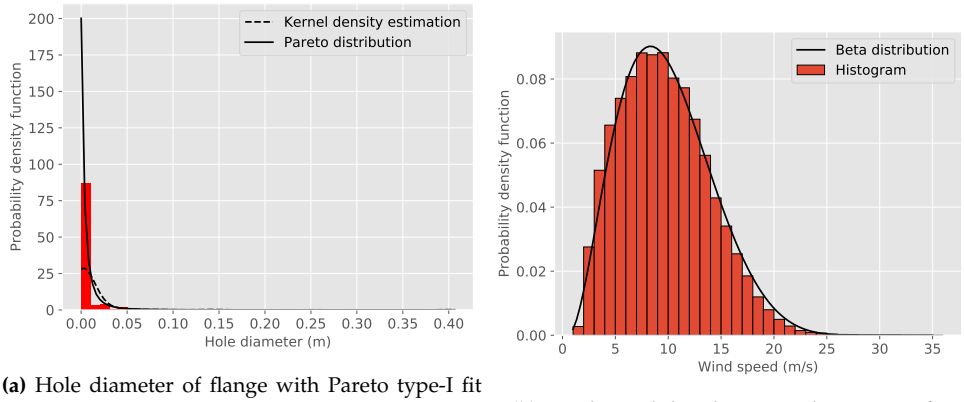


Figure 3.9: Confinement configuration used for dispersion modelling with methane leak in the centre of the domain. The 2D plot shows the distribution of the flammable gas cloud.

Although the Pareto distribution have a long tail and historical data suggests the possibility of hole diameter sizes ranging from 2 mm to 400 mm, the range of



(a) Hole diameter of flange with Pareto type-I fit (parameters: shape $\alpha = 2.0$, scale $x_m = 0.01$, location = -0.00999)

(b) Wind speed distribution with Beta PDF fit.

Figure 3.10: Probability density functions obtained based on the wind speed data and leak information from Hydrocarbon Release Database [7].

values considered is limited to [41 mm, 170 mm]. This is due to lack of stability of the model. For hole sizes with diameter higher than 170 mm, the FLAM values do not converge as shown in Fig. 3.11. For sizes smaller than 41 mm, the flammable cloud is not being detected even though the grid is refined around the leak. Such limitations do not provide enough flexibility to analyse the influence of the tail on surrogate modelling. However, the simulation results pose another challenge – for diameters greater than 80 mm, the maximum volumes of the gas cloud are changing significantly, resulting in a noisy model; in Fig. 3.12, we plot the interpolation over $N = 949$ simulation data points.

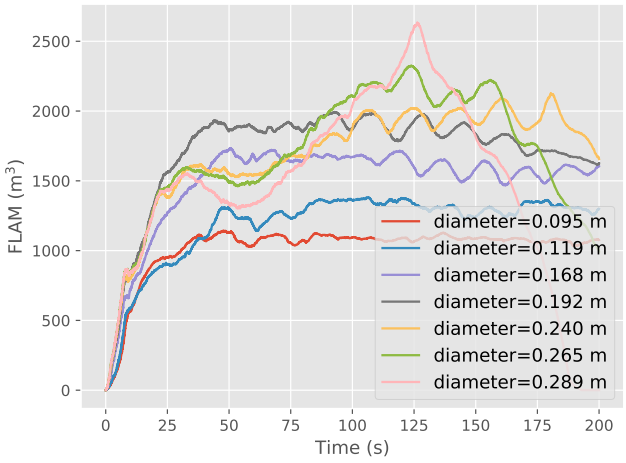


Figure 3.11: Time evolution of FLAM values for different hole diameters.

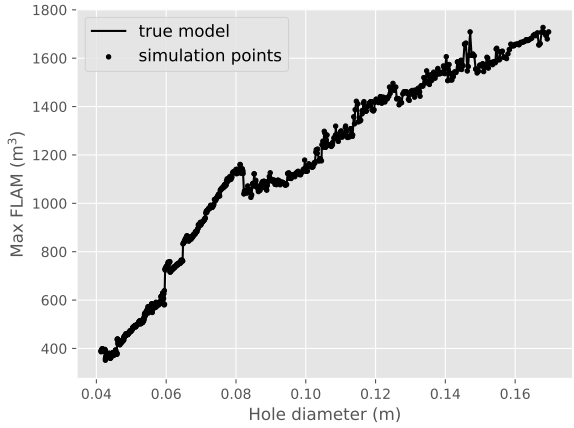


Figure 3.12: Linear interpolation over $N = 949$ maxima of FLAM values from dispersion modelling.

3.3.1 PCE analysis

Firstly, we perform PCE modelling using Pareto distribution truncated at diameter size = 0.17 m (and at 0.041 from the left). We follow the NISP procedure, where the polynomial coefficients are calculated using quadrature rule. We generate Gaussian quadrature nodes based on the Pareto distribution for which the max FLAM values are estimated. Looking at the Fig. 3.13 we can observe that the approximation error is decreasing slowly. In this case, it is not caused by the Pareto tail but rather the noise present in the results for larger diameter values. In order to fit the surrogate to such function, very high polynomial orders are required. Similar results are obtained if the random variable is assumed to have a Uniform distribution with the range of diameter lengths spanning [0.041 m, 0.17 m]. However, parameterising the problem using standard distributions (such as Uniform or Normal) allows us to perform straightforward mapping onto proxy mean-centred PDFs, e.g. Uniform(-1, 1); this leads to more stable surrogate modelling for high polynomial orders. It is more challenging to perform such transform with Pareto distribution and translate any results back to the original space as needed by an appropriate scaling and shifting. An example of PCE-based metamodels constructed with 18 simulation outputs is shown in Fig. 3.14.

As the simulation becomes less stable with increasing diameter size, it is likely that the noise in the model is numerical rather than physical. In such case, calculating the maximum value of the FLAM might be less meaningful. The PCE approach assumes that the model can be approximated with polynomials. Therefore, the surrogate results in a smooth trend which contains mostly low frequencies. If we filter high frequencies from the simulation model, we can see that the PCE approximates the model more effectively as depicted in Fig. 3.15.

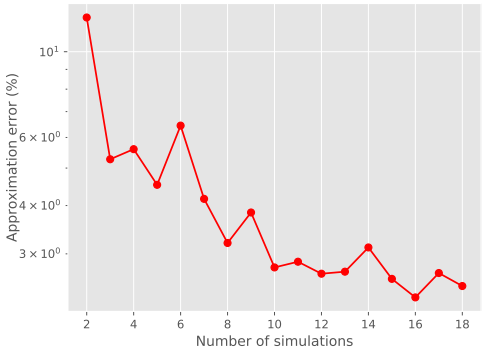
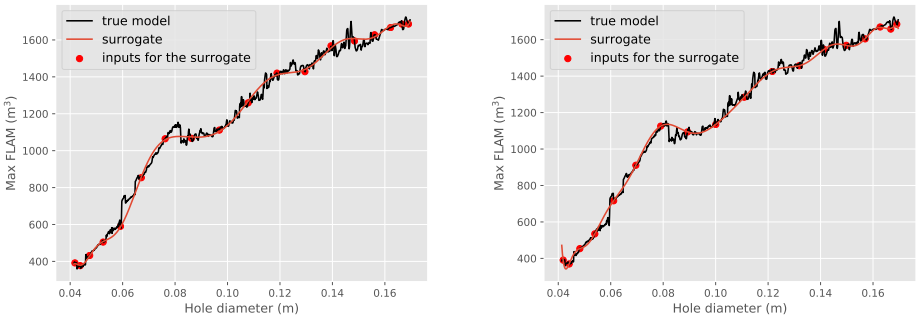


Figure 3.13: Approximation errors for PCE-based surrogate for the hole diameter with Pareto distribution truncated at 0.17 m. The error is computed based on 10^4 equispaced samples.



(a) Pareto with support [0.041 m, 0.17 m].

(b) Uniform(0.041 m, 0.17 m).

Figure 3.14: PCE-based surrogates constructed with 18 data points assuming the random variable with (a) Pareto and (b) Uniform distributions.

3.3.2 GPE study

In attempting to optimise GPE for the FLACS dispersion modelling, we first use an RBF kernel with the initial guess for δ set to be the correlation length to be 5% of the input PDF support range. This is done as we assume a minimal amount of prior information about our model, namely the support range. In Fig. 3.16, we test Uniform and Pareto design points as inputs for the GPE. The design points are again highlighted via the arrow locations on the x-axis. For this model, we find that the two choices of design points are comparable in the performance of the resulting surrogate. This is due to the fact that the Pareto distribution does not have a long tail. The relative error in the output expectation and STD gradually decreases as more simulation runs are used to update the mean and covariance functions (Fig. 3.17). However, this decrease is less rapid than was seen for the smoother model investigated in section 3.2. Increasing the number of design points

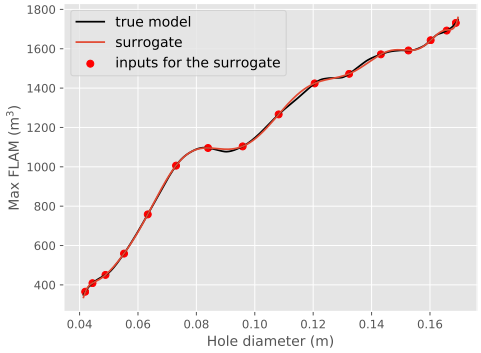


Figure 3.15: PCE surrogate of a filtered dispersion model. Here we use the wavelet thresholding approach [8] to remove high frequencies from simulation data.

beyond the eleven shown here does not significantly improve the performance of the surrogate, as the RBF kernel is *expecting* to have smooth model to fit to, and struggles to approximate the sharp increases in the QoI. Indeed, the main source of error in the GPE surrogates appears to result from poor fitting either side of the point where the model output suddenly decreases at hole diameter 0.08 m. It is clear that the assumption that the model is smoothly varying and infinitely differentiable is not valid in this region. While adding many more design points to the algorithm in this region would improve the fit, it would not satisfactorily incorporate the observed sharp increases in the quantity of interest. Furthermore, it can be an inefficient use of computational resources. Therefore, we investigate the usage of different kernels.

We continue the study with the Matérn kernels, specific formulations of which are identical to the RBF and absolute exponential kernels. Matérn function is parameterised by correlation length, δ, as above, and an additional parameter ν. Generally speaking, the smaller ν is, the rougher the approximated function is assumed to be. It may take several discrete values, 0.5, 1.5, 2.5 and ∞, corresponding to an absolute exponential kernel, once differentiable functions, twice differentiable functions and an RBF kernel respectively.

In the investigation of the performance of various kernels, we fix our design inputs to be Gaussian quadrature nodes based on a Pareto distribution. We test the suitability of RBF and Matérn kernels in Fig 3.18. In this figure, the initial guess for δ is fixed to be the usual 5% of the support range for both types of kernel. The value of ν is fixed to be 0.5 for the Matérn kernel. Other values were tested, but were found to be a poorer approximation than $\nu = 0.5$ and so are not displayed here. By visual inspection, it appears that each kernel reasonably reproduces the output of the model, particularly below hole diameters above 0.08 m. Analysing the performance of the GPEs with respect to the estimation of the mean and STD,

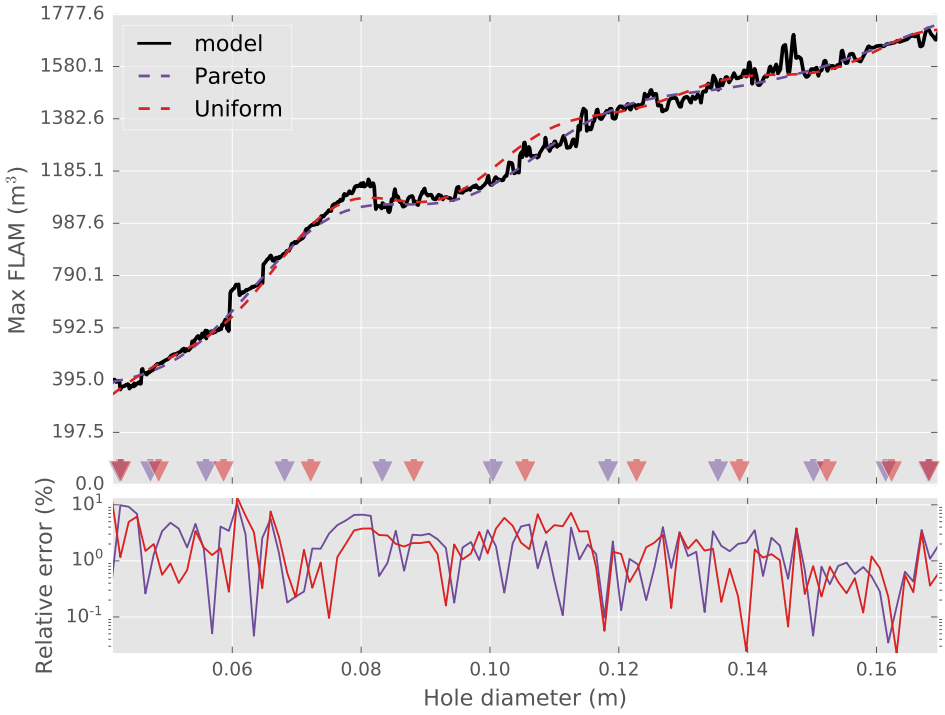


Figure 3.16: GPE surrogate constructed with an RBF kernel and eleven design points, selected as Gaussian quadrature nodes based on a Uniform and Pareto distribution. Surrogate is fit to the interpolation model outlined in Sec. 3.3

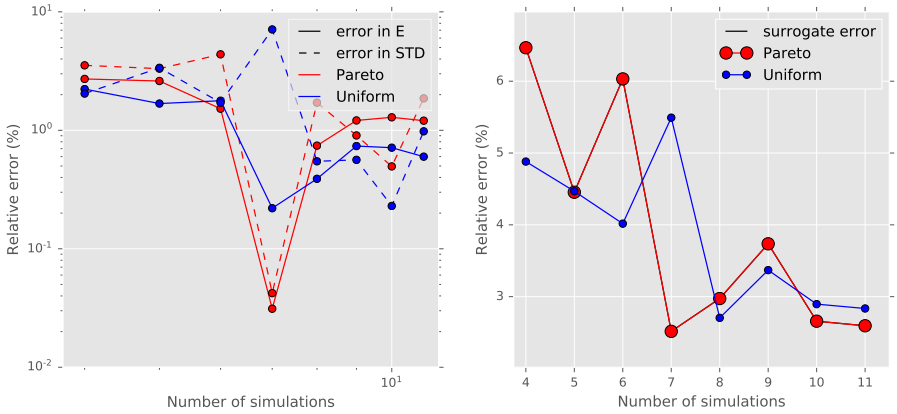


Figure 3.17: Convergence in the relative error between the interpolated simulator and the GPE.

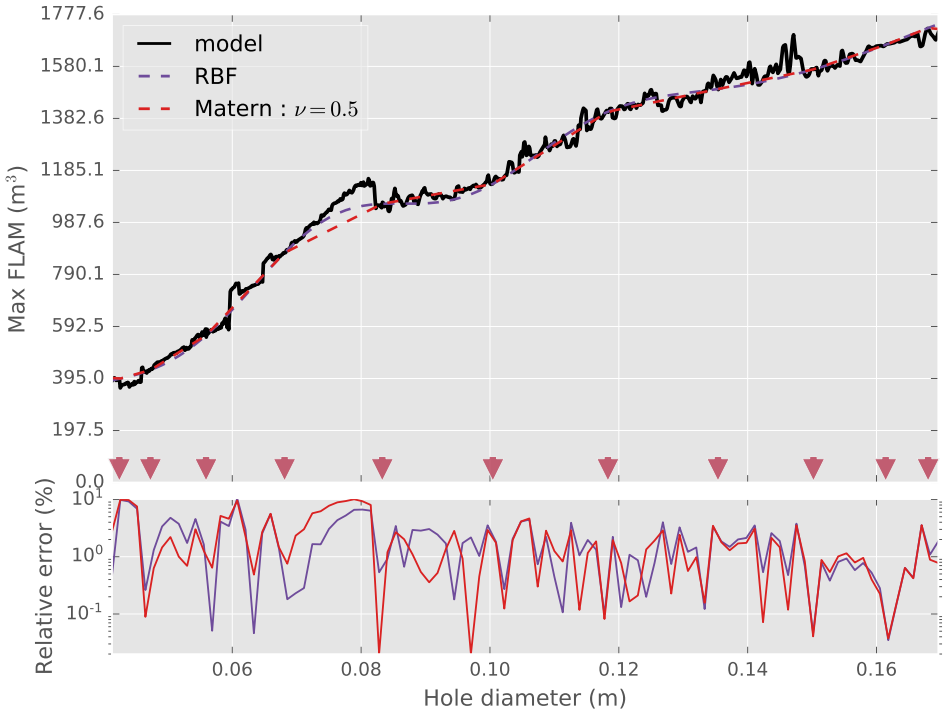


Figure 3.18: GPE surrogates constructed with eleven design points, selected as Gaussian quadrature nodes based on a Pareto distribution. Different kernels are tested, RBF and Matérn for varying values of ν.

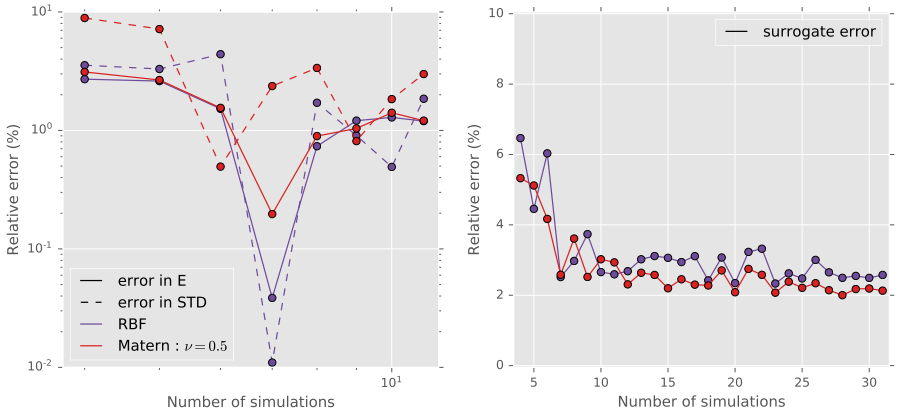


Figure 3.19: Convergence in the relative error between the interpolated simulator and the GPEs.

we find that both kernels are comparable in terms of performance, though the decrease in error with an increasing number of simulations to train the GPE takes place at a slow pace (Fig. 3.19). The statistical moment errors are percent level $N_{\text{sim}} = 11$. As for the model approximation error, we find that the Matérn kernel performs consistently better than the RBF at $N_{\text{sim}} > 10$. We extend the number of simulations shown here to highlight this. At $N_{\text{sim}} \leq 10$, the RBF kernel is comparable to, if not better than, the Matérn kernel in approximating the model output. This is caused by the fact a GPE trained with an RBF kernel fits to the general trend in the model output, whereas the Matérn expects deviations from this trend. With few design points, the only information the GPE has is relating to the trend, which the RBF kernel matches effectively. However increasing the number of design points increases the GPE's knowledge of higher order information, allowing us to benefit from the Matérn kernel. Theredore, a Matérn kernel assuming a fair degree of roughness in the model output appears to be an appropriate choice for the construction of a GPE. Future study could consider the application of random noise vector in the kriging process.

4 Conclusions and recommendations

We have applied two non-intrusive surrogate-based models to quantify uncertainty associated with heavy tailed input distribution in the framework of rare event modelling.

The GPE method generates a surrogate based on Bayes' theorem and the Gaussian hypothesis; however, it is controlled by the kernel function and the experimental design and usually does not utilise prior distribution information. The PCE strategy substitutes the model with orthonormal polynomials, which is more computationally efficient, but performs poorly when facing noisy data. In case of GPE, we can control the *roughness* of the model by more careful choice of the kernel, and, potentially, by adding noise vector to the kriging process. The latter could be the focus of a future study.

When analysing the simplified problem with random input with Pareto distribution, we observe that the long tail of the PDF makes surrogate modelling more challenging. Trying to emulate the model for a large support, fails at estimating the simulator's response and its statistics. We propose decomposing heavy tailed distributions, such as Pareto, into two parameter sub-spaces – one encompassing the bulk of the data, while the other the rare events. By constructing multi-element surrogates based on partial PDFs, we can improve the accuracy and efficiency of the emulation. In addition, we have more flexibility in controlling the resolution of the meta-models regarding their statistical content.

Bibliography

[1] Richard H Byrd et al. "A limited memory algorithm for bound constrained optimization". In: *SIAM Journal on Scientific Computing* 16.5 (1995), pp. 1190–1208.

[2] Chris Coffey, Richard Gibson, and Marios Christou. "Uncertainty in Explosion Risk Assessment". In: *IChemE, HAZARDS29* (2019).

[3] Ankit Dasgotra et al. "CFD modeling of large-scale flammable cloud dispersion using FLACS". In: *Journal of Loss Prevention in the Process Industries* 56 (2018), pp. 531–536.

[4] Jonathan Feinberg and Hans Petter Langtangen. "Chaospy: An open source tool for designing methods of uncertainty quantification". In: *Journal of Computational Science* 11 (2015), pp. 46–57.

[5] Roger G Ghanem and Pol D Spanos. "Spectral stochastic finite-element formulation for reliability analysis". In: *Journal of Engineering Mechanics* 117.10 (1991), pp. 2351–2372.

[6] Olav Roald Hansen et al. "Equivalent cloud methods used for explosion risk and consequence studies". In: *Journal of Loss Prevention in the Process Industries* ().

[7] HSE. *Hydrocarbon Releases System*. 2019. URL: `http://www.hse.gov.uk/offshore/hydrocarbon.htm`.

[8] Maarten Jansen. *Noise reduction by wavelet thresholding*. Vol. 161. Springer Science & Business Media, 2012.

[9] Nathan E Owen et al. "Comparison of surrogate-based uncertainty quantification methods for computationally expensive simulators". In: *SIAM/ASA Journal on Uncertainty Quantification* 5.1 (2017), pp. 403–435.

[10] Mohammad Mahdi Rajabi. "Review and comparison of two meta-model-based uncertainty propagation analysis methods in groundwater applications: polynomial chaos expansion and Gaussian process emulation". In: *Stochastic Environmental Research and Risk Assessment* ().

[11] Bruno Sudret. "Global sensitivity analysis using polynomial chaos expansions". In: *Reliability engineering & system safety* 93.7 (2008), pp. 964–979.

[12] Timothy J Sullivan. *Introduction to Uncertainty Quantification*. Vol. 63. Springer, 2015.

[13] Lloyd N Trefethen. "Is Gauss quadrature better than Clenshaw–Curtis?" In: *SIAM review* 50.1 (2008), pp. 67–87.

[14] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA, 2006.

[15] Dongbin Xiu and George E Karniadakis. "The Wiener–Askey polynomial chaos for stochastic differential equations". In: *Journal on Scientific Computing* 24.2 (2002), pp. 619–644.