# Novel mathematical and statistical approaches to uncertainty evaluation:

# Best practice guide to uncertainty evaluation for computationally expensive models

**Deliverable 2.5.1**
**of Work Package**
**WP2 (Uncertainty evaluation for computationally expensive models)**

**Authors:**

Kurt Rasmussen, Johan Bunde Kondrup
FORCE Technology, DK

Alexandre Allard, Séverine Demeyer, Nicolas Fischer
Laboratoire National de Métrologie et d'Essais, F

Elena Barton, Dale Partridge, Louise Wright
National Physical Laboratory, UK

Markus Bär, André Fiebach, Hermann Gross, Sebastian Heidenreich, Mark-A. Henn, Regine Model, Sonja Schmelter,
Physikalisch Technische Bundesanstalt, DE

Gertjan Kok, Nikola Pelevic
VSL, NL

A report of the EMRP joint research project
**NEW04 "Novel mathematical and statistical approaches to uncertainty evaluation"**

| 1. Work package<br>WP 2 | 2. Deliverable number<br>D2.5.1 | 3. Reporting date<br>May 2015 |
|---|---|---|
| **4. Title**<br>Best practice guide to uncertainty evaluation for computationally expensive models | | |
| **5. Author(s)**<br>Alexandre Allard, Markus Bär, Elena Barton, Johan Bunde Kondrup, Séverine Demeyer, André Fiebach, Nicolas Fischer, Hermann Gross, Sebastian Heidenreich, Mark-A. Henn, Gertjan Kok, Regine Model, Dale Partridge, Nikola Pelevic, Kurt Rasmussen, Sonja Schmelter, Louise Wright. | | **6. Lead author (e-mail)**<br>louise.wright@npl.co.uk |
| **7. Contributing researches (institutes)**<br>FORCE (DK), LNE (F), NPL (UK), PTB (DE), VSL (NL) | | |
| **8. Other contributing work packages**<br>None | | **9. Researchers in other WPs**<br>None |
| **10. Supplementary notes**<br>None | | |

**11. Abstract**
This guide provides a summary of current best practice in uncertainty evaluation for computationally expensive models. A computationally expensive model can, in this case, be considered as a model that takes a sufficiently long time to produce results that the user has rejected Monte Carlo sampling as an uncertainty evaluation method because it requires too many model evaluations to reach the level of accuracy required. This guide assumes that the user has a (possibly approximate) known upper limit on the number of model evaluations that it is feasible to make, and wishes to learn about methods for uncertainty evaluation that could give better results than Monte Carlo sampling for that limited number of model evaluations.The guide is aimed at metrologists, scientists, and engineers who are familiar with the basic ideas of uncertainty evaluation. A reader who is familiar with the Guide to the Expression of Uncertainty in Measurement [1] (GUM) and its supplements [2, 3] or who is familiar with the Monte Carlo sampling method for uncertainty evaluation will be able to understand, and hopefully benefit from, this guide.

**12. Key words**
Uncertainty, computationally expensive, finite element, scatterometry, flow, thermal diffusivity.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

## 1.1 Background

This guide provides a summary of current best practice in uncertainty evaluation for computationally expensive models. Here, a computationally expensive model is such that the time to evaluate the model is too long to permit a sufficient number of model evaluations to provide results to a required level of numerical accuracy using a simple Monte Carlo approach to uncertainty evaluation. This guide assumes that the user has a (possibly approximate) known upper limit on the number of model evaluations that it is feasible to make, and wishes to learn about methods for uncertainty evaluation that could give better results than Monte Carlo sampling for that number of model evaluations.

The guide is aimed at metrologists, scientists, and engineers who are familiar with the basic ideas of uncertainty evaluation. It will be assumed that the reader understands:

- that all measurements are subject to uncertainty,
- that measurement uncertainties can be described using a probability density function (PDF) or a cumulative distribution function (CDF),
- that the uncertainties associated with the input quantities of a model lead to uncertainties associated with the model output quantities,
- that the uncertainties associated with the model output quantities can be evaluated by propagating the uncertainties associated with the input quantities through the model.

A reader who is familiar with the Guide to the Expression of Uncertainty in Measurement [1] (GUM) and its supplements [2, 3] or who is familiar with the Monte Carlo sampling method for uncertainty evaluation will be able to understand, and hopefully benefit from, most of the material in this guide.

Some of the material in this guide involves quite high level mathematics. It is not expected that every reader will understand everything in the guide, but it is hoped that the more challenging material will encourage collaboration with mathematicians or statisticans to develop best practice solutions to problems of uncertainty evaluation with computationally expensive models.

The guide is presented as a set of steps to be considered and decisions to be made when evaluating the uncertainties associated with the outputs of a computationally expensive model. These steps are illustrated by a set of case studies, presented as stand-alone chapters, and further information can be obtained from the references given at the end of the guide. Not all the steps are necessary for all problems, but the steps listed form a useful guide to points that should be considered, if not necessarily implemented. A flow chart presenting the steps and indicating the section number that addresses each step is shown in figure 1.

It is assumed that the reader already has a well-defined problem. A well-defined problem in this case consists of a set of input quantities with well-characterised associated uncertainties, a set of output quantities for which the associated uncertainties are to be determined, and a model and software implementation that links the input quantities and the output quantities. Advice on assigning distributions to input quantities, given various knowledge about the inputs, can be found in the GUM and its supplements [1, 2, 3]. The errors and uncertainties (e.g. rounding errors, discretisation errors, bugs, etc.) associated with the software implementation of the model will not be considered as a source of uncertainty in this guide, but readers should be aware that they exist and should be minimised through software verification and testing. Similarly the uncertainties associated with the choice of model are not explored here, but should be considered and minimised through model validation.

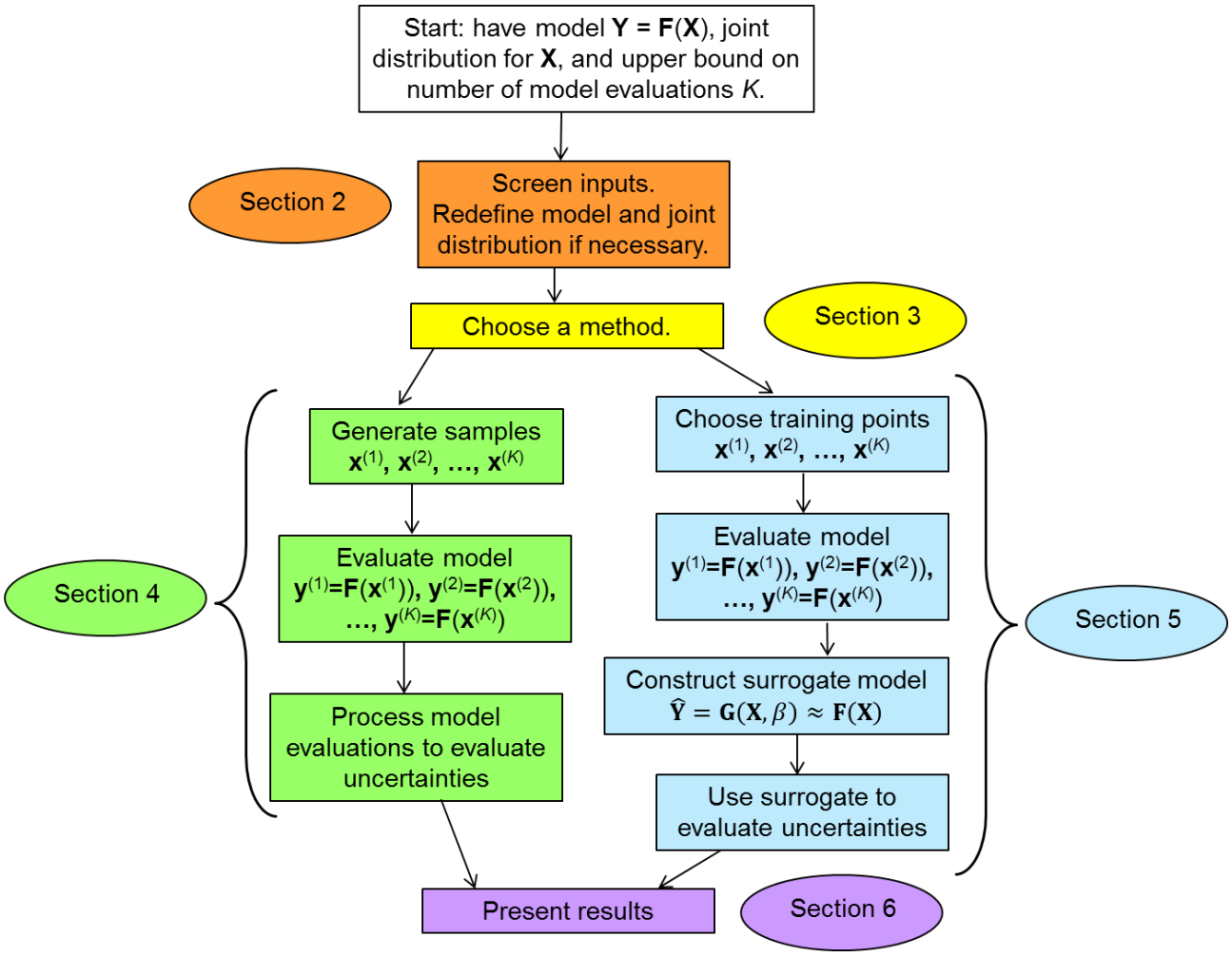



Figure 1: Flow chart illustrating the steps when evaluating the uncertainties associated with the results of a computationally expensive model, addressed in this best practice guide.

## 1.2 Case studies

The chosen case studies are all real metrological problems involving computationally expensive models that will benefit from efficient uncertainty evaluation methods.

### 1.2.1 Case study 1: Scatterometry

Scatterometry is a surface measurement technique that evaluates the geometric parameters that define periodic nanosurfaces. The surface is illuminated by monochromatic light and measurements are made of the intensity of diffracted light at different scattering angles. A finite element model can be used to predict the scattered intensities, and an inverse modelling approach used to evaluate the unknown geometric parameters. The uncertain input quantities represent the noise terms associated with the measured light intensities: material properties, geometric parameters not determined by the model, and optical parameters are assumed to have a negligible effect on the uncertainty.

The model is computationally expensive both because the finite element model takes several minutes to evaluate results, and because the use of an inverse modelling approach requires multiple evaluations of the finite element model to determine a single set of geometric parameters.

### 1.2.2 Case Study 2: Fluid flow

Calibration of flow meters and characterisation of flow measurement devices usually requires the flow conditions to be in a steady "plug flow" state to give good results. The state of a flow within a pipe is difficult to obtain directly, so it is important to have an understanding of how upstream bends and constrictions can affect the flow conditions to ensure that measurements are taken sufficiently far from such obstacles. Computational fluid dynamics (CFD) models can be used to obtain this understanding, even for turbulent flow in rough pipes, but such models are very computationally expensive.

The work reported in this case study has used a family of skew inflow profiles, where the position of the maximum is assumed to be uncertain. The output quantities describe the flow profile at a series of points downstream from the point at which the inflow profile is defined.

In a second part of the case study on fluid flow, a CFD model of a sonic nozzle is studied. This CFD model may be used to improve the understanding of the physical conditions (i.e. temperature, pressure and flow speed distribution) on the flow within sonic nozzles.

### 1.2.3 Case Study 3: Thermophysical properties

Thermal barrier coatings are multi-layer coatings that are used in turbines and engines to protect blades from high temperature environments. The layered nature of the coatings and the method of their construction makes creation of isolated samples of the individual layers challenging, and so some of the common techniques for characterisation of thermal properties of materials cannot be applied to the component parts of such coatings. Hence there has been interest in extending existing characterisation techniques to handle layered materials.

The work reported in this case study uses a finite volume model within an optimisation algorithm to obtain estimates of the thermal conductivity of the components of a layered material by matching model predictions to measured values, and evaluates the uncertainties associated with the estimates. The initial model uses nine uncertain input quantities associated with various aspects of the experiment.

## 1.3 Notation and vocabulary

Suppose we have a model

$$\mathbf{Y} = \mathbf{F}(\mathbf{X}) \tag{1}$$

where $\mathbf{X} = (X_1, X_2, \ldots, X_N)^\top$ is an input vector of quantities, described as random variables of known joint probability distribution, and $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_M)^\top$ is a output vector of quantities for which it is required to determine the joint distribution. It is assumed that the function $\mathbf{F}$ is a black box, so that evaluation of $\mathbf{F}$ is possible but algebraic evaluation of its derivatives with respect to the inputs is not.

Values of an input or an output quantity are denoted in lower case, so that $x_n$ is a value of $X_n$ the variable. The use of this notation becomes important when distinguishing between a quantity that is a function of the input quantities $X_i, i = 1, 2, \ldots, N$, and a value that has been calculated using a set of sample values $x_i, i = 1, 2, \ldots, N$. The mean of the $i$th output quantity will be denoted by $m_i$ and the variance of the $i$th output quantity will be denoted by $s_i^2$. If only one output quantity is present, these symbols will be used without subscripts. The correlation between the $i$th and $j$th output quantities will be denoted $c_{ij}$.

The expectation of a quantity $X$ is written as $E(X)$ ad the variance as $V(X)$. In some figures these quantities are ddenoted as $\mathbb{E}(X)$ and $\mathbb{V}(X)$. Several different types of probability distribution are commonly used in this report. The symbol $\mathrm{T}(a, b)$ is used to denote a symmetric triangular distribution on the interval $[a, b]$. The symbol $\mathrm{R}(a, b)$ is used to denote a rectangular distribution on the interval $[a, b]$. The symbol $\mathrm{N}(\mu,\sigma^2)$ is used to denote a normal distribution with mean $\mu$ and variance $\sigma^2$.

## 1.4 Toy problem

A toy problem will be used to demonstrate the methods described in the guide. The toy problem has three inputs:

- $X_1$ is uniformly distributed on the interval $[0, 1]$,
- $X_2$ has a triangular distribution on the interval $[0, 1]$, with a mode of 0.25,
- $X_3$ is normally distributed with a mean of 0.5 and a standard deviation of 0.01.

The problem has a single output $Y$ for which an uncertainty evaluation is required. The model linking $\mathbf{X} = (X_1, X_2, X_3)^\top$ and $Y$ is treated as a black box, so it is assumed that nothing is known about the model, but the output is given by

$$Y = X_1X_2 + X_2X_3 + X_3X_1 + \sin(2\pi X_1). \tag{2}$$

Large-scale random sampling shows that $Y$ has a mean of 0.67 and a standard deviation of 0.57. It is assumed that the user can afford 18 evaluations of this model.

# 2 Sensitivity analysis and input screening

## 2.1 Background

The idea of sensitivity coefficients will be familiar to many from uncertainty budgets and the uncertainty evaluation method described in the GUM [1]. In that context, sensitivity coefficients are the partial derivatives of the model with respect to the inputs, evaluated at the estimates (means) of the inputs. More generally, sensitivity analysis provides a rigorous framework to support exploration of the relationship between the model input quantities and the model output quantities. Input screening, discussed in the next section, is a simplified form of sensitivity analysis that allows the user to identify the most important input quantities and potentially to reformulate the model so that only the most important input quantities are treated as uncertain.

The input screening process has several benefits as a preparatory step for uncertainty evaluation. The main benefit is a reduction in complexity of the problem, but the results of the process can also lead to a clearer understanding of the links between the model input quantities and output quantities, particularly for complicated models that use "black box" software.

Some uncertainty evaluation methods become difficult to use for a large number of input quantities, either due to instability or high computational cost, so input screening can lead to a broader range of possible methods being available.

Input screening involves evaluation of the model and thus has an associated computational cost. In some cases this cost will be recouped by the ability to use a simpler model after the input quantities have been screened. If the screening suggests that all the model input quantities are necessary, it may still be possible to reuse the model evaluations made for input screening as part of the subsequent uncertainty evaluation process, for instance as parts of a random sample or as training points for a surrogate model.

Input screening is illustrated in the thermophysical properties case study (see section 9).

It should be noted that the other sensitivity analysis methods discussed in this section require repeated model evaluations and are therefore not always suitable for computationally expensive models. However, if a simplified version of the model is available (for instance a linearised model or a model of reduced dimensionality), these methods may be of help in identifying which input quantities are significant and which can be neglected.

## 2.2 Full and fractional factorial designs

One of the most conceptually simple methods for input screening is the full factorial design. An $n$-level full factorial design selects $n$ values for each of the model inputs and evaluates the model using every possible combination of these values of the inputs. For a model with $N$ inputs, a full factorial design requires $n^N$ model evaluations. It is common to set $n = 2$ and to refer to the higher value of each input quantity as the positive value and the lower as the negative value. A two-level design assumes the output quantity behaves approximately linearly in response to variations in the input quantities. It is common to test this assumption by carrying out an extra model evaluation at a centre point, for instance using the mean value of each of the input quantities. All of the subsequent material in this section, unless otherwise stated, discusses two-level designs, but similar analysis techniques can be applied to the results of designs with more levels.

The full factorial design can be used to generate a model of the form

$$Y = \beta + \sum_{i=1}^{N} \beta_i X_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \beta_{ij} X_i X_j + \ldots + \beta_{123\ldots N} X_1 X_2 \ldots X_N \tag{3}$$

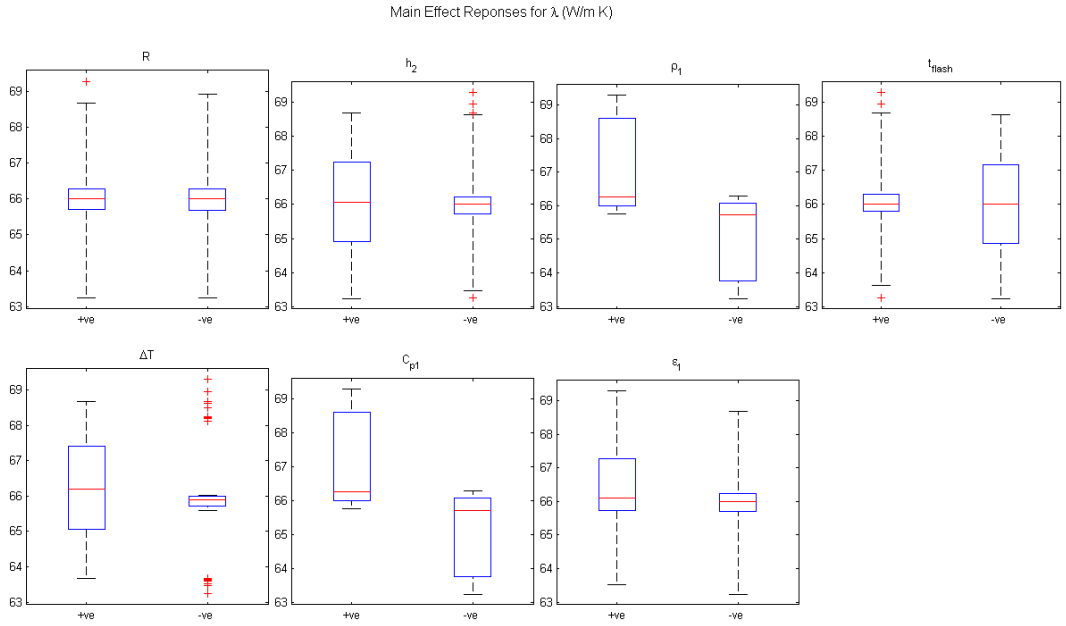



Figure 2: Box plots showing positive and negative response of main effects for a model with seven input quantities. Within each plot, the positive response is shown on the left and the negative is shown on the right.

for each output quantity $Y$, where the $\beta_i, i = 1, 2, \ldots N$ are called the main effects associated with the input quantities (i.e. the extent to which an input quantity affects the output quantity under the assumption of approximate linearity), and the $\beta_{ij}, i = 1, 2, \ldots, N, j = i + 1, i + 2, \ldots, N$ are the effects of the two-input interactions.

After performing the model evaluations, each model input quantity is assessed by comparing the output quantities from a positive input value to those from a negative input value. The main effects can be compared visually using box plots of the output quantities, as in figure 2, which helps to identify which input quantities may be important.

A box plot for a given input quantity shows the distribution of all output values obtained using the positive value of a given input quantity separately from all those obtained using the negative value of that input quantity. The box of the plot represents the interquartile range for each subset of the output values, with the median of the full set of output values marked by a solid line. The length of the stems is a multiple of the interquartile range (in the figures used in this guide the multiplier is 1.5). The points that lie beyond the length of the stems are treated as outliers and are denoted with plus symbols $(+)$. The importance of an input quantity can be assessed qualitatively by looking at the differences between the levels of the centres of the two boxes. Important quantities have a significant difference between the centres of the two boxes, such as the third box on the top row and the second box on the bottom row in figure 2.

To get a quantitative assessment of the main effects associated with each input quantity or of an interaction, the mean response to an input quantity or interaction at its positive level minus the mean response at the negative level is calculated. For a main effect, if we order the output quantity values $y_i$ such that $y_1, y_2, \ldots, y_p$, where $p = 2^{N-1}$, were obtained from the full model using the high (positive) value of $X_j$, and $y_{p+1}, y_{p+2}, \ldots, y_{2p}$

were obtained using the low value of $X_j$, then the main effect associated with $X_j$ is calculated from

$$E_j = \frac{1}{p}\left(\sum_{i=1}^{p} y_i - \sum_{i=p+1}^{2p} y_i\right). \tag{4}$$

The interaction effect associated with the interaction between $X_j$ and $X_i$ is calculated by ordering the model outputs by the value of the interaction, so that $y_1, y_2, \ldots, y_p$ were obtained from the full model when $X_j$ and $X_i$ are either both high or both low and $y_{p+1}, y_{p+2}, \ldots, y_{2p}$ were obtained when one of $X_i$ and $X_j$ is high and the other is low, and calculating the quantity $E_{ij}$ using the expression for $E_j$ above. Similar interaction effects can be obtained for higher-order interactions in the same way.

In order to evaluate if these quantities are significant an estimate of the standard error is required, determined by

$$s_E = \frac{s}{\sqrt{2^N}}, \tag{5}$$

where $s$ is the standard deviation of the $2^N$ responses. Any effect greater in absolute value than the standard error is deemed to be significant. It should be noted that it is possible for an input quantity to be significant in its interactions but not as a main variable. As an example, consider the function $X_1X_2 - X_1X_3$ and the choices $x_1^p = x_2^p = x_3^p = 1$, $x_1^m = x_2^m = x_3^m = -1$. The calculated main effect for all of the quantities $X_i$ is zero, but it is obvious from the function that all of the input quantities are significant.

To carry out a full factorial analysis

- For each input quantity of interest assign a high value, $x_j^p$, and a low value, $x_j^m$. These values should be the end points of a coverage interval of probability close to 1. For a quantity with a rectangular distribution these values could be the limits of the interval over which the quantity is non-zero. For quantities with a Gaussian distribution with mean $\mu$ and standard deviation $\sigma$, setting $x_j^p = \mu + 2\sigma$ and $x_j^m = \mu - 2\sigma$ would mean the points were close to the end points of a 95 % coverage interval.

- Run the model with every possible combination of the chosen input quantities. There will be $2^N$ such combinations. It is often useful to generate these (and keep track of them) by writing the numbers between 0 and $2^N - 1$ in binary form, replacing the 0 values with $-1$ values, and taking a $-1$ in the $j$th column of the $i$th line to mean that $x_j^m$ should be used in the $i$th model evaluation, and a 1 in the $j$th position to mean that $x_j^p$ should be used. An example for $N = 4$ is shown in table 1.

- Calculate the main effects and any interactions of interest using formulae of the form (4). The correct ordering for an interaction of mutiple variables can be found by multiplying the $+1$s and $-1$s for the quantities of interest in each row of a form such as table 1 together, and ordering so that all the products that are equal to $-1$ are the first $p$ variables, and all the products that are equal to $+1$ are the second $p$ variables. So for instance if the interaction between variables 1, 2 and 4 was of interest and the line ordering used in table 1 was used, then one possible ordering of the outputs would be $0, 2, 5, 7, 9, 11, 12, 14, 1, 3, 4, 6, 8, 10, 13, 15$.

- Calculate the standard deviation of the $y_i, i = 1, 2, \ldots, 2^N$, and hence the standard error $s_E$ as in equation (5) to decide whether an input quantity or an interaction is significant. As noted above, quantities that are significant via interaction may have an insignificant main effect so it is worth calculating all interaction effects if the results of a full factorial analysis are available.

| Trial number | $X_1$ | $X_2$ | $X_3$ | $X_4$ |
|---|---|---|---|---|
| 0 | $-1$ | $-1$ | $-1$ | $-1$ |
| 1 | $-1$ | $-1$ | $-1$ | $+1$ |
| 2 | $-1$ | $-1$ | $+1$ | $-1$ |
| 3 | $-1$ | $-1$ | $+1$ | $+1$ |
| 4 | $-1$ | $+1$ | $-1$ | $-1$ |
| 5 | $-1$ | $+1$ | $-1$ | $+1$ |
| 6 | $-1$ | $+1$ | $+1$ | $-1$ |
| 7 | $-1$ | $+1$ | $+1$ | $+1$ |
| 8 | $+1$ | $-1$ | $-1$ | $-1$ |
| 9 | $+1$ | $-1$ | $-1$ | $+1$ |
| 10 | $+1$ | $-1$ | $+1$ | $-1$ |
| 11 | $+1$ | $-1$ | $+1$ | $+1$ |
| 12 | $+1$ | $+1$ | $-1$ | $-1$ |
| 13 | $+1$ | $+1$ | $-1$ | $+1$ |
| 14 | $+1$ | $+1$ | $+1$ | $-1$ |
| 15 | $+1$ | $+1$ | $+1$ | $+1$ |

Table 1: Table showing one possible ordering of the input quantity values for a full fractorial input screening design. Within the table, "$+1$" denotes that the higher value of the input value is to be used, and "$-1$" indicates that the lower value is to be used.

| Trial number | $X_1$ | $X_2$ | $X_4$ | $X_1X_2X_4$ and hence $X_3$ |
|---|---|---|---|---|
| 0 | $-1$ | $-1$ | $-1$ | $-1$ |
| 1 | $-1$ | $-1$ | $+1$ | $+1$ |
| 2 | $-1$ | $+1$ | $-1$ | $+1$ |
| 3 | $-1$ | $+1$ | $+1$ | $-1$ |
| 4 | $+1$ | $-1$ | $-1$ | $+1$ |
| 5 | $+1$ | $-1$ | $+1$ | $-1$ |
| 6 | $+1$ | $+1$ | $-1$ | $-1$ |
| 7 | $+1$ | $+1$ | $+1$ | $+1$ |

Table 2: Table showing one possible ordering of the input quantity values for a fractional fractorial input screening design that confounds $X_3$ with $X_1X_2X_4$. Within the table, "$+1$" denotes that the higher value of the input value is to be used, and ""$-1$" indicates that the lower value is to be used.

For a computationally expensive problem, a full factorial design may be too time-consuming to perform. In such cases, the number of model evaluations can be reduced by using a fractional factorial design, which requires $2^{N-m}$ model evaluations where $m$ is the level of reduction. A good general introduction to fractional factorial designs is given on the NIST website [4], or in "Sensitivity Analysis" by Saltelli et al [5].

Fractional factorial designs are created by choosing a subset of the input quantities, defining a set of model evaluations based on every combination of high and low values of those quantities, associating each of the remaining input quantities with an interaction of some combination of the input quantities in the chosen subset, and assigning high or low values for the remaining input quantities according to the value of the interaction. So for instance, if it was decided to associate the interaction between variables 1, 2 and 4 with variable 3, then table 1 would become table 2, with half the number of model evaluations required.

The drawback of a fractional design is the confounding effect: it becomes impossible to separate the contributions of some combinations of effects and interactions, and in particular the input quantities not in the chosen subset are confounded with interactions of the input quantities that are in the subset. Careful design can help reduce the impact of confounding by ensuring as much as possible that main effects and low level interactions do not confound with each other. Generally, it is likely that physical insight can help to identify the interactions that are most important and hence should not be confounded.

## 2.3 Input screening for the toy problem

The high and low values assigned to the variables were:

- $X_1^m = 0$, $X_1^p = 1$,
- $X_2^m = 0$, $X_2^p = 1$,
- $X_3^m = 0.48$, $X_3^p = 0.52$.

| Trial number | $X_1$ | $X_2$ | $X_3$ | $X_1X_2$ | $X_1X_3$ | $X_2X_3$ | $X_1X_2X_3$ | Value of $Y$ |
|---|---|---|---|---|---|---|---|---|
| 0 | $-1$ | $-1$ | $-1$ | $+1$ | $+1$ | $+1$ | $-1$ | 0.0 |
| 1 | $-1$ | $-1$ | $+1$ | $+1$ | $-1$ | $-1$ | $+1$ | 0.0 |
| 2 | $-1$ | $+1$ | $-1$ | $-1$ | $+1$ | $-1$ | $+1$ | 0.48 |
| 3 | $-1$ | $+1$ | $+1$ | $-1$ | $-1$ | $+1$ | $-1$ | 0.52 |
| 4 | $+1$ | $-1$ | $-1$ | $-1$ | $-1$ | $+1$ | $+1$ | 0.48 |
| 5 | $+1$ | $-1$ | $+1$ | $-1$ | $+1$ | $-1$ | $-1$ | 0.52 |
| 6 | $+1$ | $+1$ | $-1$ | $+1$ | $-1$ | $-1$ | $-1$ | 1.96 |
| 7 | $+1$ | $+1$ | $+1$ | $+1$ | $+1$ | $+1$ | $+1$ | 2.04 |
| Effect | 1.0 | 1.0 | 0.04 | 0.5 | 0.02 | 0.02 | 0 | |

Table 3: Table showing the combinations of input values and the corresponding values of $Y$. The bottom row shows the main effects and interactions calculated from the responses. The standard error is 0.28. Within the central seven columns of the table, "$+1$" denotes that the higher value of the input value is to be used, and "$-1$" indicates that the lower value is to be used.

Table 3 shows the full set of output values obtained, using the same format as table 1 for the inputs. The standard error determined from the output quantity values is approximately equal to 0.28. The values of the output quantities can be used to calculate the main effects and the interactions by adding and substracting various combinations of the model outputs, as described in the section above.

It is worth noting that the results for $X_1$ and $X_2$ are identical, which illustrates that the nonlinearity in $X_1$ has not been identified. The value of $\sin 2\pi X_1$ is zero at both of the values of $X_1$ used in the factorial test, meaning that if the only contribution of $X_1$ was through the sinusoidal term, the variable would be considered to be negligible. This example illustrates the limitations of input screening using a full factorial analysis without an additional internal point to test the assumption of linearity.

These results suggest that the main effect of $X_3$ and all of the interactions involving $X_3$ are insignificant, so the model can be simplified by fixing $X_3$ at its mean value of 0.5. The model function then changes from equation (2) to

$$Y = X_1X_2 + 0.5X_2 + 0.5X_1 + \sin(2\pi X_1). \tag{6}$$

A plot of this function is shown in figure 3. The input screening has used eight model evaluations of the 18 the user can afford, so all subsequent methods will be demonstrated using a sample size of 10.

## 2.4 Other sensitivity methods

During the evaluation of measurement uncertainty, it is common practice to provide an uncertainty budget which is a summary of the contribution of each input quantity to the variance of an output quantity. More generally, sensitivity methods can be used to provide a similar subdivision of the overall variance [5].

In the GUM, this subdivision is usually obtained using the sensitivity coefficients. These coefficients can be estimated using a finite difference approach (perturbing one variable at a time by a small relative amount and dividing the difference in output values by the perturbation), requiring only $N + 1$ model evaluations. This

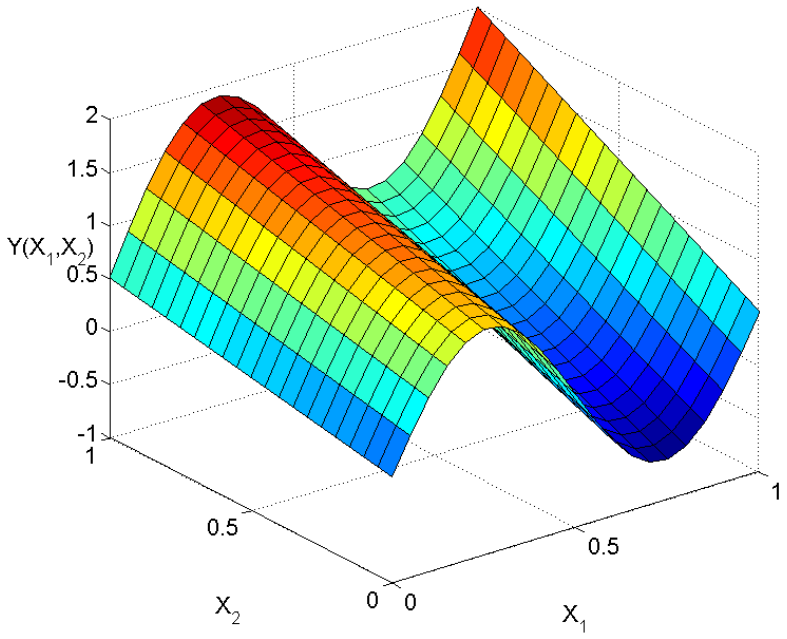



Figure 3: Plot of the toy problem model after input screening.

approach is adequate in many cases, but runs the risk of discarding variables that have no important first order effect but are important nevertheless due to interactions or higher order effects.

However, a propagation of distributions according to the supplement 1 to the GUM does not involve partial derivatives and makes it less straightforward to provide an uncertainty budget [6], although it is possible to evaluate input quantity sensitivities using the Monte Carlo method [7] at a computational cost. Another popular method to estimate conditional variances is the evaluation of Sobol' indices [8]. The conditional variance of a quantity is the variance obtained if only a subset of the input quantities are allowed to vary and the others are held fixed at some value, and is a good measure for how sensitivie teh quantity is to the subset of teh inputs that are allowed to vary. Such an estimation requires additional Monte Carlo simulations, and the number of extra model evaluations increases rapidly as the number of considered input quantities increases. Good practice may be to evaluate the Sobol' indices only for a set of the most significant input quantities, and the Morris design [9] is a quite efficient method to identify input quantities that have a significant effect (alone or through an interaction with another input quantity).

These sensitivity methods have not been used in any of the case studies due to computational cost.

### 2.4.1 Morris designs

Morris designs are input screening designs. The principle is to define a base point and to successively perturb each of the input quantities by a known amount in order to estimate the effect of the varied input quantity. In order to evaluate potential interactions, this process is repeated $R$ times, for example see figure 4 for two input quantities with three repetitions. If an input quantity is involved in an interaction, then its effect will vary according to the value taken by other input quantities so the effects at the three points shown in figure 4 would be different.

For each run of the design, the effect for the $i^{th}$ input quantity evaluated at the $j^{th}$ repetition is denoted as

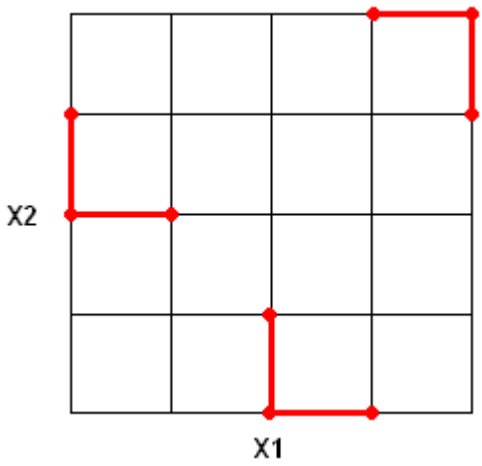



Figure 4: Variation of two input quantities according to a Morris design, for two input quantities and $R = 3$ repetitions

$E_{ij}$ where

$$E_{ij} = \frac{F(x_1^{(j)}, x_2^{(j)}, \ldots, x_i^{(j)} + \Delta x_i^{(j)}, \ldots, x_N^{(j)}) - F(\mathbf{x}^{(j)})}{\Delta x_i^{(j)}}, \tag{7}$$

where $\mathbf{x}^{(j)} = (x_1^{(j)}, x_2^{(j)}, \ldots, x_N^{(j)})^\top$ is the $j$th base point and $\Delta x_i^{(j)}$ is the size of the perturbation of the $i$th variable. The mean effect for the $i^{th}$ input quantity and the standard deviation are then computed according to

$$m_{Ei} = \frac{1}{R} \sum_{j=1}^{R} E_{ij}, \tag{8}$$

$$s_{Ei} = \sqrt{\frac{1}{R-1} \sum_{j=1}^{R} (E_{ij} - \mu_i)^2} \tag{9}$$

Input quantities with a high mean effect $m_{Ei}$ in comparison with the mean error as defined in equation (5) have a significant influence on the output quantity and those with a high effect standard deviation are likely to be involved in a significant interaction and should not be neglected for the estimation of Sobol' indices.

### 2.4.2 Sobol' indices

The Sobol' method is based on the estimation of conditional variances. The total variance decomposition theorem yields that the variance of the output quantity may be written as

$$V(Y) = V[E(Y|X_i)] + E[V(Y|X_i)] \tag{10}$$

where $V(Y)$ denotes the variance of the quantity $Y$ and $E(Y)$ is the expectation of $Y$, the first term $V_i = V[E(Y|X_i)]$ denotes the part of the variance of $Y$ that is due to the variations of the input quantity $X_i$ while the second term denotes the part of the variance of $Y$ that is due to all the input quantities apart from $X_i$. The first and second order sensitivity indices are then defined, respectively, as

$$S_i = \frac{V_i}{V(Y)}, \tag{11}$$

$$S_{i,j} = \frac{V\left[E\left(Y|X_i, X_j\right)\right] - V_i - V_j}{V(Y)}. \tag{12}$$

The effect of a given input quantity should be accounted for whether it comes from a first, second, or even higher order effect. To include higher order effects, some methods use the total sensitivity index $T_i$ for the input quantity $X_i$, given by

$$T_i = S_i + \sum_{j \neq i} S_{i,j} + ... + S_{1,2,...,N}. \tag{13}$$

Sobol' estimation of the sensitivity indices relies on two samples of size $M$; $x_{i,j}, i = 1, ..., N; j = 1, ..., K$, and $x'_{i,j}, i = 1, ..., N; j = 1, ..., K$, that are mixed together in a deterministic way in order to obtain an estimate of the sensitivity index. For the first order index associated with $X_i$, the model is evaluated for all of the $\mathbf{x}_j$ and for an altered version of the $\mathbf{x}'_j$ where the value of $x_{i,j}$ replaces the value of $x'_{i,j}$ for all $j$. The first order sensitivity index is estimated by the quantity

$$S_i^{Sobol} = \frac{\hat{D}_i}{\hat{D}} \tag{14}$$

where

$$\hat{D}_i = \frac{1}{M}\sum_{k=1}^{M} F\left(x_{1,k}, \ldots, x_{N,k}\right) F\left(x'_{1,k}, \ldots, x'_{(i-1),k}, x_{i,k}, x'_{(i+1),k}, \ldots, x'_{N,k}\right) - \hat{F}_0^2 \tag{15}$$

is the estimator of the variance of the conditional expectation of $Y$ given $X_i$,

$$\hat{D} = \frac{1}{M}\sum_{k=1}^{M} F^2\left(x_{1,k}, ..., x_{N,k}\right) - \hat{F}_0^2 \tag{16}$$

is the estimator of the total variance of $Y$, and

$$\hat{F}_0 = \frac{1}{M}\sum_{k=1}^{M} F\left(x_{1,k}, ..., x_{N,k}\right) \tag{17}$$

is the empirical mean of $y$.

$\hat{D}_{ij}$ may be defined in a similar manner with both $x_{i,k}$ and $x_{j,k}$ replacing $x'_{i,k}$ and $x'_{j,k}$ in $\mathbf{x}'_k$, leading to estimates of the second order indices (or effect of the interaction between two input quantities) as

$$S_{i,j}^{Sobol} = \frac{\hat{D}_{ij} - \hat{D}_i - \hat{D}_j}{\hat{D}}. \tag{18}$$

Higher order indices can be obtained in the same manner. As these indices are often null, it may be more efficient to compute the total order sensitivity indices to establish whether any significant contributions exist among higher order terms. The estimation of the total sensitivity indices relies on the estimation of the variance of the output due to the variations of all input quantities other than $X_i$. Sobol' proposes then the corresponding estimate

$$T_i^{Sobol} = 1 - \frac{\hat{D}_{-i}}{\hat{D}}, \tag{19}$$

where

$$\hat{D}_{-i} = \frac{1}{M}\sum_{k=1}^{M} F\left(x_{1,k}, \ldots, x_{N,k}\right) F\left(x_{1,k}, \ldots, x_{i-1,k}, x'_{i,k}, x_{i+1,k}, \ldots, x_{N,k}\right) - \hat{F}_0^2. \tag{20}$$

The estimation of Sobol' indices is computationally expensive. For this reason, best practice is to evaluate only the most significant effects, holding the input quantities with negligible influence at their best estimate, in particular for computationally expensive systems. In order to improve the convergence of these sensitivity estimates, smart sampling methods (such as Latin hypercube sampling) can also by used to generate the samples $\mathbf{x}_k$ and $\mathbf{x}'_k$.

# 3 Method choice

Universal rules on choosing a method for uncertainty evaluation are impossible to develop. Each case must be judged separately. This chapter provides a list of factors to be considered when making a choice, and discusses the aspects of a problem that can limit the choice of method, and some ways of removing these limitations. The main factors discussed here are:

- Number of input and output quantities;
- Computational and mathematical complexity and software availability;
- User knowledge of the model and the input quantities;
- Correlation between the input quantities;
- Existence of historical model evaluations and the need for sample size adaptivity.

It is assumed throughout this section that, as mentioned in the introduction, the user has a known upper bound $K$ on the number of model evaluations that can feasibly be made.

The following section mentions methods that have yet to be described. The reader is referred to the relevant subsection of the report for further details.

## 3.1 Factors to consider

It should be noted that different methods are not necessarily mutually exclusive. In particular, the points that are generated if a sampling method is chosen can often be used as training points for constructing a surrogate model (see section 5.1). In some cases the fitting of a surrogate model to sampled results can identify regions where more samples would improve the estimated uncertainties. However, it is not necessarily the case that the surrogate model will give a more accurate estimate of the uncertainty than the direct analysis of the sample results. This aspect is discussed in the case study on thermophysical properties (see section 9).

One of the most important factors affecting method choice is the number of input quantities. The average distance between a fixed number of randomly chosen points increases as the dimensionality of the space increases, so the sampling points will be distributed more sparsely as the number of input quantities increases. This sparsity can mean that important areas of the input space might not be sampled.

The number of input quantities also strongly affects the performance of surrogate models. Construction of many surrogate models requires the evaluation of parameters for the surrogate model and the number of parameters usually increases at least proportionally to the number of input quantities (see, for instance, sections 5.3 and 5.4). The unique determination of a set of $P$ parameters requires at least $P$ model evaluations, so some methods may be ruled out if $P > K$. Parameter determination also adds computational cost to the method, and some parameter determination methods suffer from instability as the number of input quantities increases. These factors may also rule out some methods for models with a large number of input quantities. In some cases, the number of input quantities can be reduced by using input screening (see section 2) to identify any insignificant inputs.

Some methods may not be suitable for models with multiple output quantities. As an example, consider a model with one input $X_1$ with a rectangular distribution on $[0, 1]$ and two outputs $Y_1 = 1/(0.001 + X_1)$ and $Y_2 = (10X_1)^3$, so that $Y_1$ is sensitive to small values of $X_1$ and $Y_2$ is sensitive to large values of $X_1$. The different areas of sensitivity may make it difficult to use importance sampling (see section 4.2) or stratified sampling (see section 4.3) for both output quantities at once.

Computational complexity and software availability can limit the choice of method. The use of parameters by surrogate models can add computational complexity. For instance, the parameter evaluation process for Gaussian process models (see section 5.4) involves the solution of a complicated nonlinear inverse model requiring the use of optimisation routines. This computational complexity may make a method too difficult to use, and in particular if the user does not have access to suitable optimisation routines then the method may not be usable. Similarly, the complexity of the mathematics that underpins the polynomial chaos method can deter potential users. In some cases the complexity can be avoided by collaborating with an appropriate expert, or by using a trusted software package (various automated uncertainty evaluation packages are available) to generate samples and process the results.

User knowledge about the input quantities and the nature of the model can strongly affect the choice of method. A simple example is that if the user knows that the model exhibits strongly nonlinear or discontinous behaviour, then a quadratic response surface, which assumes a degree of smoothness of the model, is unlikely to give reliable results. Most methods assume that all parts of the input space are equally important. If the user has knowledge that the model output quantities are particularly sensitive to a specific region of the input space then a method such as importance sampling or stratified sampling can be used to ensure that this knowledge is used to direct the sampling.

A lack of knowledge can make use of a particular method challenging. Some methods require the user to make decisions that will strongly affect the results but that involve choices where the best choice is not obvious. Examples include the best choice of subdivision of the input space for stratified sampling (see section 4.3) and the best choice of proposal density for importance sampling (see section 4.2). In such cases it is useful to test the sensitivity of the method to the choice being made, but this approach may not be possible due to the constraints of computational expense.

The nature of the joint distribution associated with the input quantities affects the choice of methods. Several methods, in particular polynomial chaos (see section 4.5) and Latin hypercube sampling (see section 4.4), are difficult to use for correlated input quantities. Polynomial chaos also works best when the independent input quantities have distributions that can be related to a family of orthogonal polynomials (see table 4 for a list of distributions and related polynomials), so if the distributions do not have a known associated polynomial family then the method will exhibit poor convergence.

Sometimes historical model evaluations may be available that could be used in addition to new model evaluations, but some methods cannot easily include extra data. Similarly if the estimate of the number of model evaluations that is feasible increases (for instance if a more powerful computer becomes available), some methods struggle to alter the sample size adaptively. In particular, Latin hypercube sampling cannot easily adjust its sample size without imposing correlation between the input quantities, and the use of sparse grid sampling to evaluate the integrals required for the polynomial chaos method means that the model is evaluated at specific values of the input quantities which may not be consistent with historical values.

## 3.2 Comparison of methods

The performance of methods can vary depending on the distributions assigned to the input quantities, the nature of the model (linear, nonlinear, discontinuous, etc.), the sample size or training point set size, and the individual values making up the sample or training point set. This variability makes choosing a method even more difficult, since it is unlikely that an example that exactly matches a problem of interest can be found in the literature to guide the choice.

The main criteria that are desirable in an uncertainty evaluation method are high accuracy, high repeatability, low sensitivity, and high efficiency. Accuracy can be proven for some statistics and some methods [10] and is often a result of the formulation used to estimate the value of the quantity of interest. If accuracy cannot be proved, it can only be demonstrated by comparison to reference results.

Ideally, the user would run tests and compare possible methods applied to the problem of interest to assess the methods on these criteria, but clearly this is almost always not possible for a computationally expensive model. However, it may be possible to identify a less computationally expensive version of a model (for instance a one-dimensional model instead of a three-dimensional one, or a model that reduces the number of unknowns by neglecting some aspects of the physics) but that still includes the aspects most likely to affect the performance of the uncertainty evaluation method. If such a model is available, this model can be used to compare methods and perhaps to generate reference results.

### 3.2.1 Reference results

Reference results should be generated using a large random sample because random sampling is unbiased and its results will eventually converge to the true cumulative distribution function. An adaptive approach provides a means to decide how many trials are needed to produce results with a required precision or to know the precision for results generated using a fixed number of trials, which can be a more efficient approach. An adaptive Monte Carlo method is described in supplement 1 of the GUM [2] for a single ouput quantity, the main steps being:

1. Fix a sample size $M = \max(10^4, J)$ where $J$ is an integer obtained by rounding $100/(1-p)$ upwards and $p$ is a probability for which a coverage interval is required.
2. Fix a number of significant digits $n_{\text{dig}}$ that are required to be accurate.
3. Generate $M$ random samples of the input quantities and evaluate the model to generate $y_1, y_2, \ldots, y_M$.
4. Repeat step 3 to generate $y_{M+1}, y_{M+2}, \ldots, y_{2M}$.
5. For each sample $y_{(h-1)M+1}, y_{(h-1)M+2}, \ldots, y_{hM}$ of size $M$, calculate the mean $m^{(h)}$, the standard deivation $s^{(h)}$, and the coverage interval endpoints $y^{(h)}_{\text{low}}$ and $y^{(h)}_{\text{high}}$.
6. Calculate the average across the $h$ samples of these quantities, e.g. $y_{\text{high}} = \sum_{r=1}^{h} y^{(r)}_{\text{high}}$.
7. Calculate the sample to sample standard deviation for the mean, standard deviation, and coverage interval endpoints, e.g. $s^2_m = \frac{1}{h}\sum_{r=1}^{h}(m^{(r)} - m)^2$. Here, "sample to sample standard deviation" of a quantity is the standard deviation of the values of the quantity calculated for each sample.
8. Calculate the standard deviation $\hat{s}$ of the whole sample, express the value in the form $\hat{s} = c \times 10^{\ell}$ where $c$ is an integer with $n_{\text{dig}}$ digits, and define a tolerance $\delta = 0.5 \times 10^{\ell}$.
9. If any of the sample to sample standard deviations are greater than $\delta/2$, go back to step 4 and repeat.

If an adaptive approach is not feasible, the user can still subdivide the full sample into subsamples and use the subsample to subsample standard deviations as an indication of the extent to which the estimators have converged for a given total sample size. It is also useful to plot the estimator against sample size to assess convergence.

### 3.2.2 Repeatability, sensitivity and efficiency

For some sampling methods and some classes of model, statements about repeatability of estimators can be made [10]. If no such statements can be made, the easiest way to test repeatability is to generate multiple samples of a fixed size using the method of interest, to evaluate the estimator of interest for each sample,

and to evaluate the sample-to-sample standard deviation of the estimator. This approach has been used in the thermophysical properties and scatterometry case studies.

Methods that are deterministic may be sensitive to the choices made in their implementation, for instance choice of training points for the response surface methodology or for Gaussian process emulation or the maximum polynomial order used for polynomial chaos. In some cases it may be possible to derive analytical expressions that put bounds on the effects of the decisions, for instance it may be possible to estimate the size of the next term in a truncated series.

The sensitivity of a surrogate model to its training points can be tested at low computational cost by taking one training point out of the set, fitting a new model to the reduced training point set, evaluating the new model at the point taken out of the original training set, and comparing the result of the surrogate model to the (known) value of the full model at the training point. This procedure has a low computational cost because it does not require any further evaluations of the main model. To ensure robustness of the conclusions, this "leave one out" procedure should be repeated with each training point in turn. A surrogate model that gives small differences for all points can be regarded as stable and hence more likely to be reliable.

Efficiency is a comparative measure of the sample size required to achieve a given performance. An efficient estimator gets close to the minimum possible variance acheivable by an unbiased estimator. A more efficient estimator needs fewer samples to achieve the required performance than a less efficient one. The statistical efficiency $e(T)$ of an unbiased estimator $T$ of a parameter $\theta$ is formally defined as

$$e(T) = \frac{1/I(\theta)}{s_T^2} \tag{21}$$

where $s_T^2$ is the variance of $T$ and

$$I(\theta) = \int \left( \frac{\partial}{\partial \theta} [\log g(x;\theta)] \right)^2 g(x;\theta) dx \tag{22}$$

is the Fisher information of the sample from a distribution with PDF $g$ and $1/I(\theta)$ is the minimum possible variance for an unbiassed estimator. Here it is required that the estimator is defined by the method used to generate the sample from which it is calculated as well as the calculation itself.

Evaluation of the efficiency either requires knowledge of the distribution from which the samples are drawn (in this case, the distribution of the output quantities), or multiple samples from which the estimator variance and Fisher information can be estimated. If analytical information is not available, it is easier to compare the sample to sample variance of two methods directly rather than attempting to approximate the Fisher information using discrete values.

# 4 Sampling methods

## 4.1 Background

A sampling method consists of a method for generating samples $\mathbf{x}_k, k = 1, 2, \ldots, K$, from the space of input variables (the input space), and a method for post-processing the resulting output values $\mathbf{y}_k = \mathbf{F}(\mathbf{x}_k)$ to obtain estimates of some properties of the desired distribution.

The simplest possible sampling method is random sampling, also known as Monte Carlo sampling. This method, described in full in supplement 1 to the GUM [2] using slightly different notation, consists of the following steps for a sample size $K$:

- Generate $K$ random samples $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(K)}$ from the joint input distribution for $X_1, X_2, \ldots X_N$. If the input quantities are independent, this can be accomplished by
  - For each quantity $X_j$, generate $K$ random samples $\xi_j^{(1)}, \xi_j^{(2)}, \ldots, \xi_j^{(K)}$ of a variable with a rectangular distribution on the interval [0,1].
  - For each input quantity $X_j$, evaluate the inverse of the cumulative distribution function $f_j(X_j)$ at each value $\xi_j^{(i)}, i = 1, 2, \ldots, K$, so that $x_j^{(i)} = f_j^{-1}(\xi_j^{(i)})$.
  - Form the samples as $\mathbf{x}^{(1)} = (x_1^{(1)}, x_2^{(1)}, \ldots, x_N^{(1)})^\top$, $\mathbf{x}^{(2)} = (x_1^{(2)}, x_2^{(2)}, \ldots, x_N^{(2)})^\top$, and so on up to $\mathbf{x}^{(K)} = (x_1^{(K)}, x_2^{(K)}, \ldots, x_N^{(K)})^\top$.
- Evaluate the model function $\mathbf{F}(\mathbf{X})$ at each of the sampled values to obtain $K$ sets of output quantities, so that $\mathbf{y}^{(1)} = \mathbf{F}(\mathbf{x}^{(1)}), \mathbf{y}^{(2)} = \mathbf{F}(\mathbf{x}^{(2)}), \ldots, \mathbf{y}^{(K)} = \mathbf{F}(\mathbf{x}^{(K)})$.

These results $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(K)}$ where $\mathbf{y}^{(i)} = (y_1^{(i)}, y_2^{(i)}, \ldots, y_M^{(i)})^\top$ can then be processed to obtain estimates of various statistical quantities, including:

- Mean values for each quantity: $m_i = \frac{1}{K} \sum_{k=1}^{K} y_i^{(k)}$.
- Standard deviation for each quantity: $s_i = \sqrt{\frac{1}{K-1} \sum_{k=1}^{K} (y_i^{(k)} - m_i)^2}$.
- Correlation coefficients $c_{ij} = \frac{1}{(K-1)s_i s_j} \sum_{k=1}^{K} (y_i^{(k)} - m_i)(y_j^{(k)} - m_j)$.
- If the samples are ordered based on the calculated values for a single output quantity $Y_i$ such that $y_i^{(1)} \leq y_i^{(2)} \leq \ldots \leq y_i^{(K)}$, then the pairs $(y_i^{(k)}, \frac{k-0.5}{K}), k = 1, 2, \ldots, K$ form an estimate of the marginal cumulative distribution function for $Y_i$. The estimated CDF can be used to obtain coverage intervals for $Y_i$.
- If the sample size is sufficiently large, it may be possible to sort the results into "bins" and thus estimate the marginal probability density function for $Y_i$.

It is stated in clause 7 of the first supplement to the GUM [2] (note that the supplement uses $M$ for the number of samples, but this has been changed to $K$ below for consistency with the rest of this document) that

7.2.1 A value of $K$, the number of Monte Carlo trials, i.e. the number of model evaluations to be made, needs to be selected. It can be chosen *a priori*, in which case there will be no direct control over the quality of the numerical results provided by the Monte Carlo method. The reason is that the number of trials needed to provide these results to a prescribed numerical tolerance will depend on the shape of the PDF for the output quantity and on the coverage probability required. Also, the calculations are stochastic in nature, being based on random sampling. NOTE A value of $K = 10^6$ can often be expected to deliver a 95 % coverage interval for the output quantity such that this length is correct to one or two significant decimal digits.

7.2.2 The choice of a value of $K$ that is large compared with $1/(1-p)$, e.g. $K$ at least $10^4$ times greater than $1/(1-p)$, should be made. It can then be expected that the discrete approximation to the CDF of $Y$ determined from the results of the Monte Carlo method will provide a reasonable discrete representation of $G_Y(\eta)$ (the CDF of $Y$) in the regions near the endpoints of a $100p\%$ coverage interval for $Y$.

There are many applications, within metrology and beyond, that require models that take upwards of several minutes to run. The computational expense of such models, which is the prime motivation for this guide, renders random sampling computationally intractable. However, other sampling methods exist that produce better repeatability for small sample sizes.

Random sampling does not guarantee that a given region of the input space will include a sample point, and it does not exclude the possibility of one or more points being very close together. These features mean that it is not an efficient sampling method because some areas of the input space that may be related to critical values of the output quantities may not be sampled, whilst other less critical areas are sampled repeatedly. For small sample sizes, the estimates of the joint distribution of the output variables are strongly affected by every sampled point, and so since the samples are not guaranteed to cover evenly the whole of the input space, estimates of the joint distribution of the output variables obtained using different random samples of the same small size $K$ can exhibit a large sample-to-sample variance.

## 4.2 Importance sampling

Importance sampling was developed as a variance reduction technique to improve on random sampling. Suppose we wish to evaluate an integral $I$ of the form

$$I = \int_D G(\mathbf{X}) q(\mathbf{X}) dV, \tag{23}$$

where $G(\mathbf{X})$ is some function of the input variables $\mathbf{X}$, $D$ is the domain spanned by the input variables, $\mathrm{d}V = \mathrm{d}X_1\ \mathrm{d}X_2 \ldots\ \mathrm{d}X_N$ and $q(\mathbf{X})$ is their joint PDF, using a numerical quadrature method which requires function evaluation at a set of (possibly randomly chosen) values of $\mathbf{X}$.

Integrals of this form are of interest because most quantities of interest related to the model results as described in section 1.1 can be expressed in this form, for example the mean of a model result $Y_i = F_i(\mathbf{X})$ is given by

$$m_i = \int_D F_i(\mathbf{X}) q(\mathbf{X}) dV, \tag{24}$$

the variance is

$$s_i^2 = \int_D (F_i(\mathbf{X}) - m_i)^2 q(\mathbf{X}) dV, \tag{25}$$

and the cumulative probability is given by

$$P(Y \leq Y^*) = \int_D H(Y^* - F(\mathbf{X}))q(\mathbf{X})dV, \tag{26}$$

where $H$ is the Heaviside step function. If an estimate $\bar{I}$ of $I$ is determined in terms of $K$ random samples drawn from $q$, the variance of $\bar{I}$ can be written as

$$\frac{1}{K}\left[\int_D G^2(\mathbf{X})q(\mathbf{X})dV - I^2\right]. \tag{27}$$

If the function $G$ is only non-zero for a small subset of the input space $D$, then the variance will be particularly large because the estimate of $I$ obtained will be determined by how many samples lie within the subset of $D$ where $G$ is non-zero. The most common example is the case when a particularly low cumulative probability is to be determined so that the Heaviside function in the expression (26) is zero within most of $D$. Importance sampling aims to reduce this variance by rewriting the integral $I$ as

$$I = \int_D \frac{G(\mathbf{X})q(\mathbf{X})}{p(\mathbf{X})}p(\mathbf{X})dV, \tag{28}$$

where $p$ is a function of the input variables such that $\int_D p(\mathbf{X})dV = 1$ and $p(\mathbf{X}) \geq 0\ \forall \mathbf{X} \in D$, generating input values $\tilde{\mathbf{x}}_k, k = 1, 2, \ldots, K$, by sampling from $p$ as the joint distribution of $\mathbf{X}$, and estimating $I$ as

$$\tilde{I} = \frac{1}{K}\sum_{k=1}^{K}\frac{G(\tilde{\mathbf{x}}_k)q(\tilde{\mathbf{x}}_k)}{p(\tilde{\mathbf{x}}_k)}. \tag{29}$$

The estimate $\tilde{I}$ has variance

$$\frac{1}{K}\left[\int_D \frac{G^2(\mathbf{X})q^2(\mathbf{X})}{p(\mathbf{X})}dV - I^2\right]. \tag{30}$$

The aim of importance sampling is to reduce the variance of $\tilde{I}$ by an appropriate choice of $p$. The density $p$ is known as the importance sampling density function, proposal density, or instrumental density in the literature.

The main strength of the importance sampling method is that it can produce a significant reduction in the variance of the estimate of $I$, thus improving the accuracy of the estimate beyond that of random sampling. The reduction of the variance is particularly high if the quantity of interest is a low quantile or a low probability because importance sampling ensures that the regions of the input space that lead to these low probability events are sampled.

The main difficulty in implementing the method is that the choice of importance sampling density function $p$ is not obvious. Various references [11, 12, 13] point out that choosing $p(\mathbf{X}) = \alpha G(\mathbf{X})q(\mathbf{X})$, where $\alpha$ is some constant, minimises the variance in theory, but is impossible in practice because if the knowledge of $G$ was sufficiently complete to define $p$ in that way then the integral $I$ could be evaluated more efficiently without using sampling methods (in our case $G$ is insufficiently known because it is a black box).

Anderson [11] supplies four criteria that a good choice of $p$ should meet, and gives a warning about distribution tails. The criteria are:

- $p(\mathbf{X}) > 0$ whenever $G(\mathbf{X})q(\mathbf{X}) \neq 0$,
- $p(\mathbf{X})$ should be close to proportional to $G(\mathbf{X})q(\mathbf{X})$,
- it should be simple to sample values from $p(\mathbf{X})$, and

- it should be easy to evaluate $p(\mathbf{X})$ for any $\mathbf{X} \in D$.

The determination of a good importance sampling density function is even more difficult in high dimensions [11]. For a model with multiple output quantities a suitable importance sampling density function for one output quantity may be incompatible with the other output quantities. A compromise may be difficult to find in such a case.

To carry out an importance sampling calculation for a single ouput quantity, take the following steps:

- Select an importance sampling density function $p$ that satisfies the criteria listed above.
- Generate a sample of size $K$ using $p$ as the joint distribution of $\mathbf{X}$.
- Evaluate the model at the sampled points to generate output quantities $y_1, y_2, \ldots y_K$.
- Evaluate the importance sampling density function $p$ and the true density function $q$ at the sampled points.
- Calculate the estimate of the mean from

$$m = \frac{1}{K} \sum_{k=1}^{K} \frac{y_k q(\mathbf{x}_k)}{p(\mathbf{x}_k)}, \tag{31}$$

the estimate of the variance from

$$s^2 = \frac{1}{K} \sum_{k=1}^{K} \frac{(y_k - \bar{y})^2 q(\mathbf{x}_k)}{p(\mathbf{x}_k)}, \tag{32}$$

and the cumulative probability that $Y \leq y^*$ from

$$P(\tilde{y} \leq y^*) = \frac{1}{K} \sum_{k=1}^{K} \frac{H(y^* - y_k) q(\mathbf{x}_k)}{p(\mathbf{x}_k)}. \tag{33}$$

Consider the toy example (2), and suppose that the user knows that the model output quantity is sensitive to the parts of the input space where $X_1 \approx 0.25$ and $X_1 \approx 0.75$. Then the user could decide to define $p(X_1)$ as shown in figure 5, leaving the distribution of $X_2$ as it is. Evaluating a single sample of size 10 using this distribution gives a mean value of 0.62 and a standard deviation of 0.19, whereas using the same random numbers as a sample of $X_1$ without the importance sampling distribution gave a mean value of 0.49 and a standard deviation of 0.28. Repeated evaluation using a sample size of 10 suggested that importance sampling gives a better estimate of the mean than random sampling, but does not improve the estimate of the standard deviation. A better choice of importance sampling distribuition may improve results further.

The difficulties associated with importance sampling have meant it has not been used in any of the case studies.

## 4.3 Stratified sampling

Stratified sampling is a technique that has lower sample-to-sample variance than random sampling and was developed in an effort to address some of the shortcomings of random sampling. Some references (e.g., [14]) appear to use the names “stratified sampling” and “importance sampling” interchangeably. They are considered as separate here, as importance sampling uses a distinctly different approach to sample selection from that used in stratified sampling.

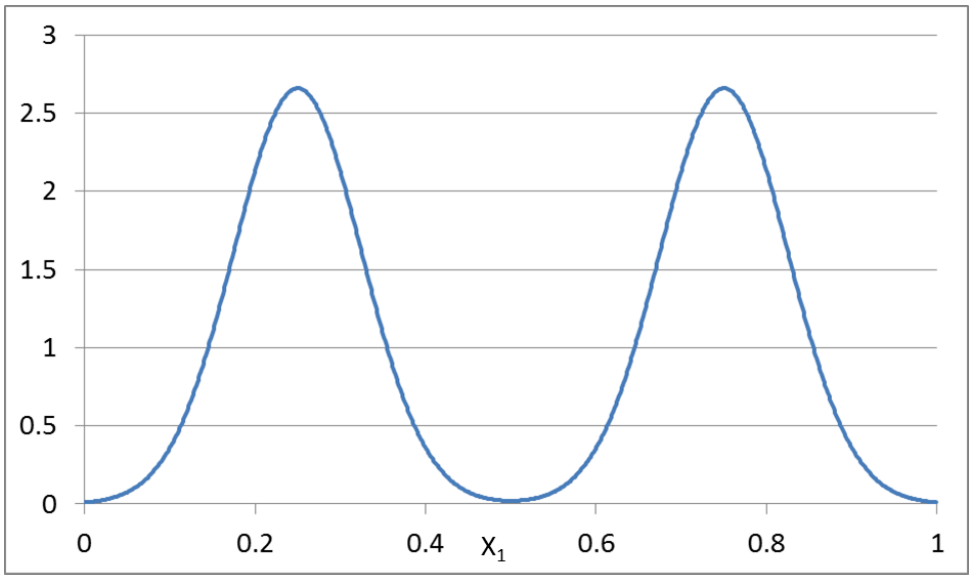



Figure 5: Possible importance sampling distribution for $X_1$ that ensures that regions around 0.25 and 0.75 are sampled.

Stratified sampling divides the complete input space into a number of distinct non-overlapping regions $S_i, i = 1, 2, \ldots, n_S$, and takes $I_i, i = 1, 2, \ldots, n_S$, samples within the $i^{th}$ region, such that $\sum_{i=1}^{n_S} I_i = K$. The regions need not be of equal probability, and it is useful to define the probability of a sample occuring within the $i^{th}$ region as $p_i$.

The estimates of the mean and covariance matrix of the output variables are determined from

$$m_k = \sum_{i=1}^{n_S} \sum_{j=1}^{I_i} w_i F_k(\mathbf{x}_{i,j}) \tag{34}$$

$$c_{kn} = \sum_{i=1}^{n_S} \sum_{j=1}^{I_i} w_i (F_k(\mathbf{x}_{i,j}) - m_k)(F_n(\mathbf{x}_{i,j}) - m_n)^\top \tag{35}$$

where $\mathbf{x}_{i,j}, i = 1, 2, \ldots, n_S, j = 1, 2, \ldots, I_i$, is the $j^{th}$ sample in the $i^{th}$ subregion and $w_i = p_i/I_i$ is a weight dependent on the "size" of the subregion (measured in terms of probability $p_i$) and the number of samples taken within that subregion. The estimator for $c_{kn}$ is biased, but [10] gives bounds on the bias and shows that it tends to zero as the sample size increases.

If a larger sample size is needed, for instance if it is decided that a higher accuracy is required, more points can be generated either by resampling in some regions, so that $I_i$ increases for some values of $i$, or by further subdividing some regions and sampling within any new empty regions. In both cases the values of the corresponding weights $w_i$ will require adjustment. An adaptive method that identifies the most important input variable to subdivide has been developed in the context of sampling as a quadrature method [13].

The main strength of the method is that it allows use of expert knowledge about the underlying model $\mathbf{F}$ to generate samples that will create the most useful information about $\mathbf{Y}$ and to ensure that any regions of particular importance are included in the sampling, whilst maintaining a proper probabilistic consideration of the input distributions. Even when the model is a black box, it is possible that the user will know that some regions of the input space give almost constant values of the output variables and that other regions produce more variation in the output variables and so should be sampled more densely.

The chief drawback of the method is that creating a set of non-overlapping regions of known probability that completely fill the input space efficiently is not simple, particularly for correlated input variables and for problems with a large number of input variables. Another drawback is that if there are multiple output variables of interest then the best choice subdivisions for different output variables could easily be incompatible, meaning that a unique best choice subdivision may not be possible. A lesser disadvantage is the extra care required in interpreting the results for the case where the regions in the subdivisions are not of equal probability.

A detailed step by step guide to applying stratified sampling to a problem is not given because there are only three steps and it is not possible to provide specific detail for any of the steps that is not model-dependent. The three steps are define a stratification of the input space, sample within each stratified region, and process the results using the formulae given above.

Stratified sampling has been used in the flow case study (see section 8) and in the thermophysical properties case study (see section 9).

Figure 6 shows two ways of defining a subdivision of the two-dimensional input space of the toy problem. The plot on the left of figure 6 shows a subdivision with $n_S = 10$ regions of equal probability, so that $p_i = 0.1$ throughout. The plot on the right of figure 6 shows a subdivision with regions of unequal probability, also with $n_S = 10$. The region in the first column has $p_i = 1/3$, regions in the second column have $p_i = 1/9$, and regions in the third column have $p_i = 1/18$. The crosses show the locations of random samples taken within each region with $I_i = 1, i = 1, 2, \ldots, 10$.

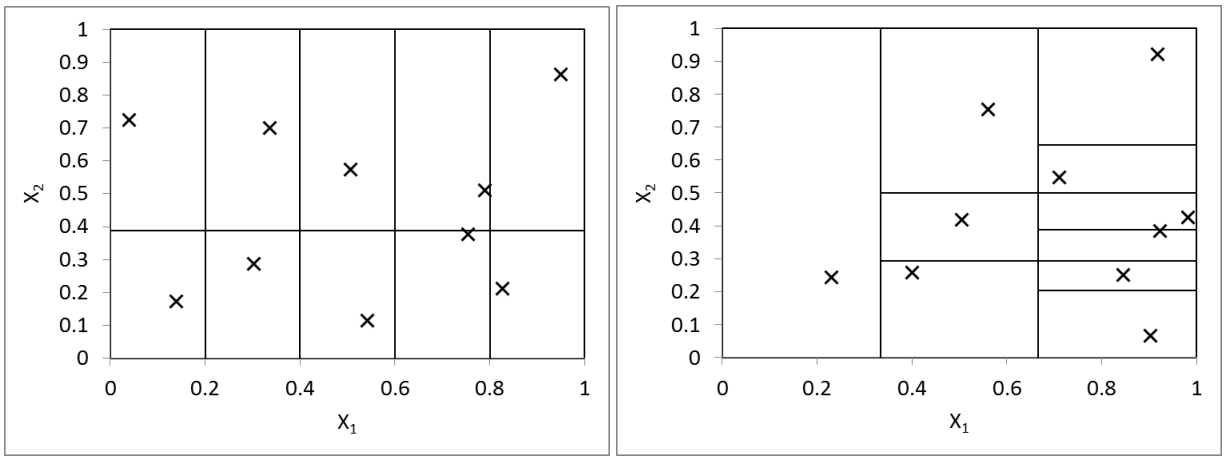



Figure 6: Two ways of subdividing the same sample space into ten regions: the plot on the left has regions of equal probability and the plot on the right does not.

## 4.4 Latin Hypercube sampling

Latin hypercube sampling (LHS) was first described by McKay [10]. Latin hypercube sampling is an extension of stratified sampling to ensure that every subregion of every individual input variable is sampled, and uses a simple method to generate samples spread across the full range of values. Latin hypercube sampling has been used in all three case studies.

The basic form of the sampling method requires that all input variables be independent. The full range of each input variable is divided into $K$ regions of equal probability, and a sample of each input variable is taken from each region. If an input quantity is not defined on a finite region (e.g., if the quantity has a Gaussian distribution), then one or two of the subregions may be infinite. The $K$ samples of each of the $N$ input variables

are then randomly arranged into $K$ vectors of length $N$, with each vector containing exactly one sample of each input variable and each sample being used in exactly one vector. The $K$ vectors are used to generate $K$ vectors of output variable values, and these values are post-processed using the same techniques as are used to post-process random samples.

To generate a Latin hypercube sample of size $K$ for $N$ independent input quantities $X_j$ with cumulative distribution functions $g_j(X_j)$, take the following steps:

- Generate $N \times K$ random samples of a random variable that has a rectangular distribution on the interval $[0, 1]$ and group them as $N$ vectors of length $K$, $\xi_j^{(i)}, i = 1, 2, \ldots, K, j = 1, 2, \ldots, N$.

- Generate $N$ different permutations of the numbers $\{1, 2, \ldots, K\}$, where $\pi_j^{(i)}, i = 1, 2, \ldots, K, j = 1, 2, \ldots, N$, is the value in the $i$th position of the $j$th permutation. In the absence of a random permutation generator, the easiest way of generating a single such permutation is

  - Generate $K$ random samples $v_1, v_2, \ldots, v_K$ of a random variable $V$ that has a rectangular distribution on the interval $[0, 1]$.
  - Sort these numbers in increasing order, and use the resulting rearrangement of the sample indices as the random permutation.
  - For instance, if $K$ = 4 and the random samples are $v_1 = 0.384, v_2 = 0.027, v_3 = 0.725$, and $v_4 = 0.168$ then the rearranged order is $v_2 < v_4 < v_1 < v_3$, so the permutation is $2, 4, 1, 3$.

- Calculate the quantities $u_j^{(i)} = (\pi_j^{(i)} - \xi_j^{(i)})/K, i = 1, 2, \ldots K, j = 1, 2, \ldots, N$. These quantities are samples of $K$ random variables, each uniformly distributed on one of the $K$ sub-regions of [0,1], arranged in an order determined by the random permutation.

- Construct the vectors of input quantities by setting $x_j^{(i)} = g_j^{-1}(u_j^{(i)}), i = 1, 2, \ldots, K, j = 1, 2, \ldots, N$, and setting the $i$th vector to $\mathbf{x}^{(i)}$ consist of the values $(x_1^{(i)}, x_2^{(i)}, \ldots, x_N^{(i)})^\top$. Then each vector contains exactly one value of each input quantity, each subregion of the range of each input quantity is sampled exactly once, and the vectors are assembled randomly.

- Evaluate the model function $\mathbf{F}(\mathbf{X})$ at each of the sampled values to obtain $K$ sets of output quantities, so that $\mathbf{y}^{(1)} = \mathbf{F}(\mathbf{x}^{(1)}), \mathbf{y}^{(2)} = \mathbf{F}(\mathbf{x}^{(2)}), \ldots, \mathbf{y}^{(K)} = \mathbf{F}(\mathbf{x}^{(K)})$.

Latin hypercube sampling combines the space-filling properties and control of stratified sampling with the ease of sample generation of random sampling. In the original paper describing the method [10], it is shown that Latin hypercube sampling leads to unbiased estimates for the moments of the output distribution, and that if the function $\mathbf{F}$ is monotonic in each of its inputs, then the variance of those estimates is lower for Latin hypercube sampling than it is for random sampling. Some papers [10, 14, 15] suggest that the variance is better than that of random sampling for a wider range of models. Extensions to Latin hypercube sampling that optimize the sampling of points [16], handle correlation [17], inequality constraints on the inputs [18], and that enable sensitivity analysis [19] have been proposed.

The main disadvantages of Latin hypercube sampling are the difficulty of correlated inputs, and sample size inflexibility. As with stratified sampling, correlated input variables provide a challenge for Latin hypercube sampling. Iman and Conover [17] developed a method to induce a user-defined rank correlation matrix in a set of uncorrelated multivariate samples. The approach assumes that imposing a rank correlation matrix is equivalent to imposing a sample correlation matrix, which is meaningful in most modelling situations but may not be perfect in some cases.

The dependence of the hypercubes used in Latin hypercube sampling on the sample size $K$ makes changing the sample size difficult, so if there are doubts about the adequacy of the sample size, the trial will have to be rerun from the start. Theoretically it is possible to subdivide each of the $K$ intervals for each input value and sample in the new regions, but this approach would not lead to truly random vectors of input values (since new values would only be input with other new values) and would have a potentially unwanted correlation structure.

For example, consider generating a Latin hypercube sample of size $K = 10$ for the toy problem (2). First generate two random permutations of the integers $\{1, ..., 10\}$: $\boldsymbol{\pi}_1 = (10, 8, 4, 3, 2, 7, 9, 6, 5, 1)$ and $\boldsymbol{\pi}_2 = (1, 6, 2, 7, 3, 10, 8, 5, 4, 9)$.

The point $\left(\pi_1^{(i)}, \pi_2^{(i)}\right)$ lies at the top right-hand corner of the square $\left[\frac{\pi_1^{(i)}-1}{10}, \frac{\pi_1^{(i)}}{10}\right] \times \left[\frac{\pi_2^{(i)}-1}{10}, \frac{\pi_2^{(i)}}{10}\right]$ of the grid represented in figure 7, and the $i$-th two-dimensional point will be sampled within that square. If $\xi_1$ and $\xi_2$ are independent random variables both uniformly distributed on $[0, 1]$ then the uniform random variables $-\frac{\xi_1^{(i)}}{10}$ and $-\frac{\xi_2^{(i)}}{10}$ give the coordinates of the $i$-th LHS point in the sampled square relative to the top right-hand corner of the square.

The $i$-th LHS sample of $U_1$ and $U_2$ has coordinates $\left(u_1^{(i)}, u_2^{(i)}\right) = \left(\frac{\pi_1^{(i)}-\xi_1^{(i)}}{10}, \frac{\pi_2^{(i)}-\xi_2^{(i)}}{10}\right)$. Inserting the values of $u_1^{(i)}$ and $u_2^{(i)}$ into the inverse of the CDF gives the desired sampled values of $X_1$ and $X_2$. The LHS for $X_1$ and $X_2$ is plotted in figure 8 and compared to a sample obtained with simple random sampling. Note that the random sample includes several examples of points in close proximity and that much of the top left-hand corner of the plot is empty.

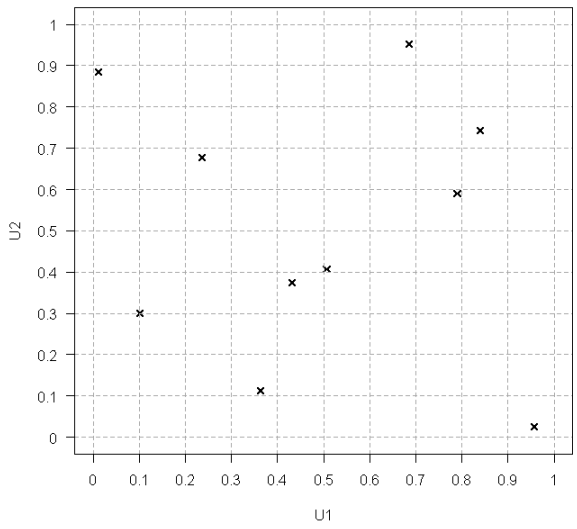
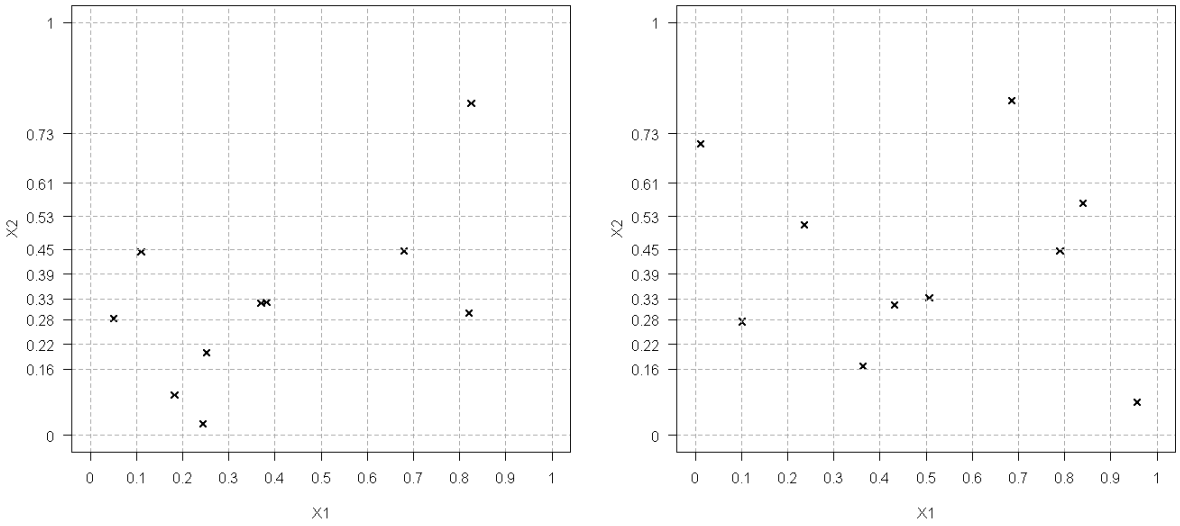



Figure 7: Example of a Latin hypercube sample of size $K = 10$ for two uniformly distributed variables.

## 4.5 Generalized Polynomial Chaos

Generalized polynomial chaos (gPC) is becoming more widely used for uncertainty evaluation [20]. The approach was introduced by Wiener [21] and was generalized in [22]. The approach approximates the output quantity of interest as a weighted sum of orthogonal polynomials of random variables. The weights are de-




(a) Simple Random Sampling

(b) Latin Hypercube Sampling

Figure 8: Examples of two ways to generate a sample of size $K = 10$.

fined as integrals over the space of input quantities, with the integrands being a product of the model function and one of the polynomials. The integrals are evaluated using numerical quadrature, requiring model evaluations. This approach has very rapid convergence properties (in many cases, low order polynomials give a good approximation of the model output), meaning that the methods can outperform most sampling methods.

The method is restricted to models with independent input quantities. In some cases involving correlation, the correlation can be eliminated because it is caused by a well-understood physical effect, and the model should treat this effect as an independent random variable. For instance, many material properties are correlated due to their dependence on temperature. If the temperature dependence is characterised and temperature is treated as a random input quantity, the correlation between input quantities can be removed.

Another drawback of the method is the curse of dimensionality, meaning that the number of model evaluations required increases rapidly as the number of input quantities increases. Additionally, convergence becomes less good for cases when there is not a direct link between the distribution of interest and a family of polynomials. These cases are typically addressed by identifying the most similar distribution that does have an associated family of polynomials, and using that family in the expansion.

The following section briefly reviews the full mathematical definition of the generalized polynomial chaos method, and section 4.5.2 gives guidelines for applying the theory to a practical problem. Good overviews can also be found in [20, 23, 24, 25, 26, 27]. The theory is quite mathematically involved, but it can be reduced to a series of steps to be followed fairly easily.

Polynomial chaos has been used in the scatterometry case study (section 7) and the flow case study (section 8).

### 4.5.1 Theory

Let $(\Omega, \mathcal{F}, P)$ be a probability space consisting of a $\sigma$-algebra $\mathcal{F}$ over the event space $\Omega$ and an associated probability measure $P$. In the following we consider independent random variables $\mathbf{X} = (X_i)_{i=1}^N$, $N \in \mathbb{N}$,

| Distribution | Family of the ansatz polynomials |
|---|---|
| Gaussian | Hermite |
| Gamma | Laguerre |
| Uniform | Legendre |
| Beta | Jacobi |

Table 4: Correspondence of the family of orthogonal polynomials to the distribution of a random variable. Note that most of the random variables must be scaled in a suitable way such that the probability density function is equal to the weight function of the polynomial family.

mapping from $(\Omega, \mathcal{F}, P)$ to $(\mathbb{R}^N, \mathcal{B}^N, P_{\mathbf{X}})$, where $\mathcal{B}$ denotes the Borel $\sigma-$algebra and $P_{\mathbf{X}}$ denotes the range of measure of $\mathbf{X}$. Assuming that $\mathcal{F}$ is generated by $\mathbf{X}$, every square integrable function $y : (\Omega, \mathcal{F}, P) \to (\mathbb{R}^N, \mathcal{B}^N, P_y)$ can be expanded in a series of orthogonal polynomials with modes $(\hat{y}_j)_{j=0}^{\infty}$, i. e.,

$$y(\mathbf{X}(\omega)) = \sum_{j=0}^{\infty} \hat{y}_j \Psi_j(\mathbf{X}(\omega)), \tag{36}$$

where $\omega$ is used to indicate that the $X_i$ are random variables. Here $(\Psi_j)_{j=0}^{\infty}$ is a family of orthogonal polynomials with respect to the weighted inner product

$$\langle \Psi_k, \Psi_l \rangle_w = \int_{\mathbb{R}^N} \Psi_k(\mathbf{s}) \Psi_l(\mathbf{s}) w(\mathbf{s})\, d\mathbf{s}, \quad k, l = 0, \ldots, \infty \quad \text{with} \quad \int_{\mathbb{R}^N} w(\mathbf{s})\, d\mathbf{s} = 1. \tag{37}$$

If the weighting function $w$ for a given family of orthogonal polynomials has all of the properties defining a probability density function, then the dependence of $y$ on any input quantities with PDF $w$ can be expanded as a sum of that family of polynomials.

Using the Kronecker delta $\delta_{kl}$ that returns 1 if $k = l$ and 0 otherwise it follows that

$$\langle \Psi_k, \Psi_l \rangle_w = S_k \delta_{kl} \quad \text{with} \quad S_k = \langle \Psi_k, \Psi_k \rangle_w.$$

According to the Wiener-Askey scheme there is a connection between the polynomial family and the distributions of the random variables $X_i$, $i = 1, \ldots, N$. In Table 4 some continuous distributions and the associated families of orthogonal polynomials are collected. In most cases the random variables must be scaled in a suitable way (e.g. rescaling to zero mean and unit variance) such that the probability density function is equal to the weight function of the polynomial.

Using $\Psi_0 = 1$, the equivalence of the probability density function and the weight function of the polynomial ansatz, as well as the orthogonality of the ansatz polynomials, the expectation $E(y)$ and the variance $V(y)$ can be directly calculated from the series expansion (36), i. e,

$$\begin{aligned} E(y) &= \int_{\mathbb{R}^N} y(\mathbf{X}(\omega))\, dP(\omega) = \sum_{j=0}^{\infty} \hat{y}_j \langle \Psi_j, 1 \rangle_w = \hat{y}_0, \\ V(y) &= E(y^2) - E(y)^2 = \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} \hat{y}_j \hat{y}_k \langle \Psi_j, \Psi_k \rangle_w - \hat{y}_0^2 = \sum_{j=1}^{\infty} \hat{y}_j^2 S_j. \end{aligned} \tag{38}$$

In many practical computations, the degree $p$ of the polynomials has to be restricted, which leads to an approximation

$$\tilde{y}(\mathbf{X}) = \sum_{j=0}^{q} \hat{y}_j \Psi_j(\mathbf{X}), \quad q+1 = \frac{(N+p)!}{N!p!}, \tag{39}$$

of the series expansion (36). In expression (38) we have only to sum up $q+1$ modes $\hat{y}_j$. The length of the series increases very fast with the polynomial degree $p$ and the dimension of the random space $N$.

Many applications for which polynomial chaos is useful are governed by partial differential equations that have dependence on the spatial coordinates $\mathbf{r}$. Such problems can be written as an operator $L$ acting on the solution $y$

$$L(y(\mathbf{r}, \mathbf{X}), \mathbf{r}, \mathbf{X}) = 0, \quad \mathbf{r} \in D \subset \mathbb{R}^d,\, d \in \mathbb{N}, \tag{40}$$

with appropriate initial and boundary conditions. The solution $y(\mathbf{r}, \mathbf{X})$ is a random field. Exploiting the orthogonality of the ansatz polynomial, the modes of the series expansion are given by

$$\hat{y}_j(\mathbf{r}) = \frac{1}{S_j} \langle y(\mathbf{r}, \cdot), \Psi_j \rangle_w, \quad j = 0, \ldots, \infty. \tag{41}$$

In practical applications there exists a robust and efficient numerical solver for the operator $L$, so equation (41) cannot be determined analytically. Therefore the $j$th projection $\hat{y}_j(\mathbf{r})$ of the solution $y(\mathbf{r}, \mathbf{X})$ onto the basis polynomial $\Psi_j$ in equation (38) has to be approximated. This can be done by a cubature rule given by $K \in \mathbb{N}$ cubature nodes $(\mathbf{x}^{(k)})_{k=1}^K$ and weights $(\lambda^{(k)})_{k=1}^K$. The integral of the weighted inner product in equation (41) becomes a sum and equation (41) can be written as

$$\hat{y}_j(\mathbf{r}) \approx \frac{1}{S_j} \sum_{k=1}^{K} y(\mathbf{r}, \mathbf{x}^{(k)}) \Psi_j(\mathbf{x}^{(k)}) \lambda^{(k)}, \quad j = 1 \ldots, q. \tag{42}$$

Thus, the underlying deterministic equation has to be solved for $K$ samples $\mathbf{x}^{(k)}, k = 1, \ldots, K$ given by the cubature rule. For a small number of random variables $\mathbf{X}$, tensor Gauss cubature is the natural choice, and [20] gives tables of nodes $\mathbf{x}^{(k)}$ and weights $\lambda^{(k)}$ for certain quadrature formulas. For moderate or large $N$ this approach becomes infeasible and sparse grid approaches are to be preferred [28, 29].

Once the modes $\hat{y}_j$ are known, an approximation to the solution can easily be calculated. This is discussed in Section 5.5, where the polynomial chaos approximation is used as a surrogate model.

### 4.5.2 Applying generalised polynomial chaos to a problem

The application of the theory is straightforward and is broadly similar to the steps in the evaluation of a Fourier series. For the application of polynomial chaos to a specific problem, the following steps have to be performed:

1. Identify the polynomial family associated with the distribution assigned to each of the input quantities. Some examples are listed in Table 4, but other pairs of distribution and polynomial family are known.

2. Select a polynomial degree $p$. The choice of polynomial degree will affect the accuracy. The less smooth the output quantity is as a function of the input quantities, the higher the degree of polynomial required. The rate of convergence is controlled by how quickly the polynomials of degrees higher than $p$ decay.

3. Select the quadrature points $\mathbf{x}^{(k)}, k = 1, 2, \ldots, K$, using either Gaussian quadrature or sparse grid techniques for high dimensional spaces.

4. Evaluate the model function, all of the polynomials, and the associated weight function $w$ at each of the quadrature points. These values are used in equation (42) to determine the coefficients in the sum of polynomials (36).

5. Mean and variance (as well as other moments) can directly be calculated from equation (38). Note that the sum in equation (38) only runs from 1 to $q$ due to the truncation of the polynomial degree $p$.

### 4.5.3 Toy model

The uniform distribution of $X_1$ means that Legendre polynomials should be used to expand the function in terms of $X_1$. The triangular distribution of $X_2$ requires a transformation of variables.

If $W_2 \sim \mathcal{U}(0,1)$ then $X_2$ given by

$$\begin{aligned} X_2 &= \sqrt{W_2}/2 \quad \text{for } 0 < W_2 < 1/4, \\ X_2 &= 1 - \sqrt{3(1-W_2)}/2 \quad \text{for } 1/4 \leq W_2 < 1, \end{aligned} \tag{43}$$

is triangular on $(0,1)$ with mode $1/4$.

The toy model has been evaluated using ten quadrature nodes for $X_1$ and two for $X_2$, i.e. 20 model evaluations are required in total. The presence of the nonlinear term $\sin(2\pi X_1)$ means that the degree of the polynomial in the expansion terms using $X_1$ has to be as high as possible. The evaluation gave a mean of 0.67 and a standard deviation of 0.56, which are in excellent agreement with the reference values of 0.67 and 0.57 respectively.

# 5 Surrogate models

Surrogate models are meta models used when the output quantity of interest is difficult to calculate. The surrogate model is a simplified model that captures the main features of the full model for a certain range of input quantities. In many cases the surrogate model is constructed from simple high dimensional functions. Surrogate models are approximations of more rigorous calculations. Surrogate models can improve computational efficiency by replacing time demanding finite element calculations used to solve partial differential equations. Therefore, surrogate models enable statistical methods derived elsewhere [30] to be applied to computationally expensive problems. First of all we will introduce different types of surrogate models.

Suppose we have a model problem

$$\mathbf{Y} = \mathbf{F}(\mathbf{X}), \tag{44}$$

where $\mathbf{X} = (X_1, X_2, \ldots, X_N)^\top$ is an input vector of random variables of known joint distribution and $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_M)^\top$ is a vector of model outputs for which it is required to determine the joint distribution. It is assumed that the function $\mathbf{F}$ is a black box, so that the evaluation of $\mathbf{F}$ is possible, and that the evaluation is computationally costly. Typically, the function $\mathbf{Y}$ is a computationally costly solution of a partial differential equation. The main idea is to replace the model equation (44) by an approximation, either using a stochastic process or a deterministic function,

$$\tilde{\mathbf{Y}} = \mathbf{G}(X_1, ..., X_N, \beta_1, ..., \beta_n) \approx \mathbf{F}(\mathbf{X}), \tag{45}$$

where for a deterministic function, $\mathbf{G}$ is commonly taken as a linear combination of simple functions. In some cases the input random variables are not all used in the surrogate model in order to reduce the computational expense further. The subset of variables that are used are generally selected by using a screening method to identify the most important variables, as explained in section 3. It will be assumed throughout the following that any screening of input variables has already taken place, so that $(X_1, X_2, \ldots, X_N)^\top$ are all to be used in the surrogate model.

For both types of approximation, the hyper-parameters $\boldsymbol{\beta} = (\beta_1, ..., \beta_n)^\top$ are model specific and typically have to be determined by using a set of training point outputs $\mathbf{y}_i, i = 1, 2, \ldots \ell$, associated with a set of training point input values $\mathbf{x}_i, i = 1, 2, \ldots, \ell$. Training point outputs are experimentally measured values or rigorously calculated output values of the expensive model which are provided by equation (44).

In the following we introduce four different surrogate models with increasing complexity: nearest neighbor interpolation, response surface methodology, Gaussian process modelling, and the polynomial chaos expansion. Throughout the section, a single output quantity is considered. Multiple output quantities are generally treated by creating separate surrogate models for each output and evaluating these surrogate models using the same input quantity values so that correlations between output quantities can be estimated.

## 5.1 Training point selection

Training points are the points at which the full model is evaluated, and upon which the surrogate model is based. Most surrogate modelling methods permit an almost free choice of training points.

The choice of training input values can strongly affect the quality of the approximation. The aim is to span the space of likely input values. Various schemes are available for choosing these values, amongst the most commonly used being full factorial and fractional factorial designs and various optimized forms of Latin hypercube design. Values need not be chosen randomly (Hammersley sampling [31] is a popular deterministic method), and (as with sampling methods, see section 4) the training points should be most dense in regions where the output quantities are sensitive to the values of the input quantities.

In cases where the training points and input values are taken from experiment or from previous evaluations of an expensive model, there may be no control over that choice and there may be a resultant loss of quality of the approximation or instability in the parameter estimation calculations.

The thermophysical properties case study (see section 9) presents a comparison of several different training point selection methods.

## 5.2 Nearest neighbor interpolation

Nearest-neighbor (NN) interpolation is one of the simplest interpolation techniques. Here, the nonlinear function $F_k(\mathbf{X})$ is calculated for chosen training points $\mathbf{x}_i$ and the interpolated function value at some arbitrary point $\mathbf{x}_k$ is given by the function value at a training point close to $\mathbf{x}_k$. In particular, the interpolated function value is $\tilde{F}_k = \mathbf{F}(\mathbf{x}_i)$ if and only if $\mathbf{x}_k \in U(\mathbf{x}_i)$, where $U$ denotes the neighborhood of $\mathbf{x}_i$. This approximation leads to a piecewise constant discontinuous approximation of the function $F_k(\mathbf{X})$. The method can be implemented in various forms depending on how the neighbourhood $U(\mathbf{X})$ is defined. The example given below defines a set of identically sized hypercubes, defines one training point within each hypercube, and assigns any sampled point lying within a given hypercube the value of the function at the training point within the hypercube.

An alternative approach that allows for more generally defined training points (and hence potential reuse of historical data) is to define the neighbourhood of a given training point to be the region that encloses all space that is closer to that training point than to any other training point (i.e. the cells in a Voronoi tesselation). If this approach is taken, it is sensible to normalise the input quantities by introducing length scales into the definition of "closer" so that large valued quantities do not dominate the definition of distance.

The NN interpolation method is simple to understand and to implement, but many training points are needed to be certain of a good approximation, particularly for models known to be nonlinear.

To implement a surrogate model using NN interpolation and a simple neighbourhood definition for a single output quantity, carry out the following steps:

- Define the range $I_i$ associated with each input quantity $X_i, i = 1, 2, \ldots, N$.
- Choose the number of training point neighbourhoods in each dimension $m$. The total number of volumes, and hence of training points, is given by $K = m^N$.
- Divide the ranges $I_i$ into $m$ non-overlapping intervals of equal size, $I_i^1, I_i^2, ..., I_i^M, i = 1, 2, \ldots, N$.
- Define the neighbourhood volumes as cross products of these one-dimensional intervals and label them consecutively. For instance, for a case with three input quantities ($N = 3$) and a known value of $M$, then a general expression for a volume would be $I(j = j_1 + (j_2 - 1)M + (j_3 - 1)M^2) = I_1^{j_1} \times I_2^{j_2} \times I_3^{j_3}$. This approach can be extended to systems with more input quantities by extending the powers of $M$ used in the expression for $j$ up to $(j_N - 1)M^{N-1}$.
- Choose a training point $\mathbf{x}^{(j)} \in I(j)$.
- Calculate $y^{(j)} = F(\mathbf{x}^{(j)})$ for $j = 1...K$ using the full model.
- To approximate the function value at an arbitrary point $\hat{\mathbf{x}}$, identify the volume $I(j)$ that has $\hat{\mathbf{x}} \in^{(j)}$ and set $G(\hat{\mathbf{x}}) = y^{(j)}$.

Nearest neighbour interpolation is used in the scatterometry (see section 7) and thermophysical properties (see section 9) case studies.

Figure 9 shows a possible definition of the neighbourhoods of a Latin hypercube sample of the toy problem. The dots show the sampled points, and the lines split the domain into regions such that the nearest sampled point to any point in a given region is the sampled point in that region.

These regions were created using a Voronoi tesselation rather than by using the procedure defined above, because the aim was to construct a surrogate model based on existing evaluations of the full model, so the training points had already been selected and could not be redefined.

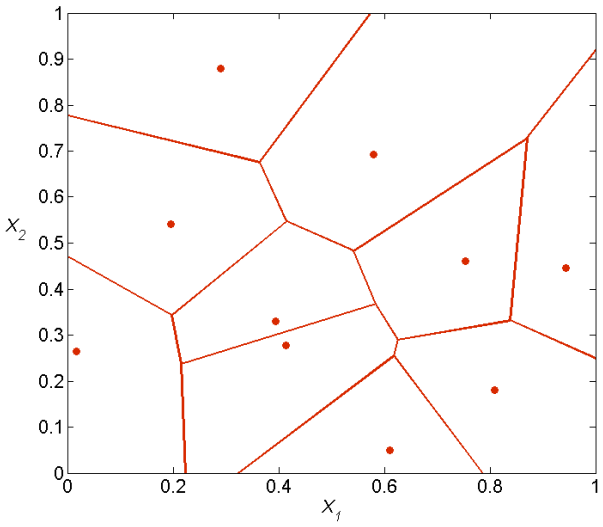



Figure 9: Voronoi tesselation of the toy problem domain into neighbourhoods, based on a Latin hypercube sample.

## 5.3 Response surface methodology

The response surface methodology (RSM), also referred to as polynomial regression, is a parametric regression model. The model uses training points to determine the parameters $\boldsymbol{\beta}$ by regression. Once the parameters are known, the training set is no longer used and only the parameters of the model characterize the response at new points. In the RSM, the model outputs are represented as a polynomial plus an error [32, 33, 34], so for a single model output,

$$\tilde{Y}(\mathbf{X}) = G(\mathbf{X}, \boldsymbol{\beta}) + \epsilon. \tag{46}$$

Here, $\epsilon$ represents random error which is in general assumed to be normally distributed with zero mean and standard deviation $\sigma$. The approximation polynomial can have any order, but most models only use terms up to the second order. Linearization of the model around a best estimate as is done in the GUM corresponds to a first order approximation. For second order polynomials the model is expressed as

$$G(\mathbf{X}, \boldsymbol{\beta}) = \beta_0 + \sum_{i=1}^{N} \beta_i X_i + \sum_{i=1}^{N} \sum_{j \geq i}^{N} \beta_{ij} X_i X_j, \tag{47}$$

where $\beta_0$ is referred to as the intercept, $\beta_i, i = 1, 2, \ldots, N$ are the slopes, and $\beta_{ij}, 1 \leq i \leq j \leq N$, are the interaction coefficients, which together represent the unknown hyperparameters of the surrogate model. To find a best approximation for the desired model the experimentalist makes a series of experimental runs or data sets

are calculated using the expensive model defined by equation (44). The best fit parameters are obtained by least squares estimation, that is by minimizing the error [33]

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{\ell} \epsilon_i^2 = \sum_{i=1}^{\ell} \Big( y_i - G(\mathbf{x}^{(i)}, \boldsymbol{\beta}) \Big)^2 . \tag{48}$$

The discussion here has been restricted to scalar model outputs for simplicity, but the scheme can easily be extended to vector valued outputs. For more details on least squares regression modelling see [33].

The key choices that uncertainty evaluation with RSM requires are a choice of training input values, a choice of polynomial order, and a method for uncertainty evaluation using the surrogate model.

RSM is a well-established method and its implementation has been used for many applications. The evaluation of parameters is generally quick once the training values have been obtained. RSM is best suited for fewer than 10 input random variables and is able to tackle weakly nonlinear problems.

RSM is less efficient for strongly nonlinear or discontinuous problems. In principle, higher order polynomials can be adopted, but it needs a large number of training points to accurately capture the function behavior, reducing the benefits of using RSM. Alternatively the input quantity space could be divided into subsets and a local response surface could be fitted over each subset.

To construct a quadratic response surface given a set of training points $\mathbf{x}^{(k)}, k = 1, 2, \ldots, K$, and corresponding function values $y^{(k)}$:

- Identify suitable normalisation constants $\ell_i, i = 1, 2, \ldots, N$ and $M_i$ for each of the input quantities $X_i, i = 1, 2, \ldots, N$. The normalisation constants should be chosen such that the likely range of values taken by each input quantity (including all training point values) maps onto the interval $[0, 1]$ via $\hat{X}_i = (X_i - M_i)/\ell_i$. This normalisation will improve the stability of the surface fitting process. Define normalised values of the training points as $\hat{x}_i^{(k)} = (x_i^{(k)} - M_i)/\ell_i, i = 1, 2, \ldots, N, k = 1, 2, \ldots, K$.

- Calculate the mean value of the output quantity $m = \frac{1}{K} \sum_{k=1}^{K} y^{(k)}$ and subtract it from each function value to give $\hat{y}^{(k)} = y^{(k)} - m$. If values of $\hat{y}^{(k)}$ are not of the order of one, scale $\hat{y}^{(k)}$ by calculating the standard deviation $s$ and dividing by that value.

- Either use a linear least squares fitting routine to determine the parameters, or calculate all of the following quantities

  - $a_0 = \sum_{k=1}^{K} \hat{y}^{(k)}$,
  - $A_i = \sum_{k=1}^{K} \hat{x}_i^{(k)},\ i = 1, 2, \ldots, N$,
  - $B_{ij} = \sum_{k=1}^{K} \hat{x}_i^{(k)} \hat{x}_j^{(k)},\ i = 1, 2, \ldots, N, j = i, i+1, \ldots, N$,
  - $a_i = \sum_{k=1}^{K} \hat{y}^{(k)} \hat{x}_i^{(k)},\ i = 1, 2, \ldots, N$,
  - $\hat{B}_{ij} = \sum_{k=1}^{K} \hat{x}_i^{(k)} \hat{x}_j^{(k)},\ i, j = 1, 2, \ldots, N$,
  - $C_{mij} = \sum_{k=1}^{K} \hat{x}_i^{(k)} \hat{x}_j^{(k)} \hat{x}_m^{(k)},\ i, m = 1, 2, \ldots, N, j = i, i+1, \ldots, N$,
  - $a_{ij} = \sum_{k=1}^{K} \hat{y}^{(k)} \hat{x}_j^{(k)} \hat{x}_i^{(k)},\ i = 1, 2, \ldots, N, j = i, i+1, \ldots, N$,
  - $D_{ijmn} = \sum_{k=1}^{K} \hat{x}_j^{(k)} \hat{x}_i^{(k)} \hat{x}_n^{(k)} \hat{x}_m^{(k)},\ i, m = 1, 2, \ldots, N, j = i, i+1, \ldots, N, n = 1, 2, \ldots, N$.

- The following system of linear equations can be solved to obtain the best fit values of the parameters $\boldsymbol{\beta}$ for the normalised variables:

$$K\beta_0 \quad + \sum_{i=1}^{N} \beta_i A_i + \sum_{i=1}^{N} \sum_{j \geq i}^{N} \beta_{ij} B_{ij} = a_0 \tag{49}$$

$$A_m \quad + \sum_{i=1}^{N} \beta_i \hat{B}_{mi} + \sum_{i=1}^{N} \sum_{j \geq i}^{N} \beta_{ij} C_{mij} = a_m, \; m = 1, 2, \ldots, N, \tag{50}$$

$$B_{mn} \quad + \sum_{i=1}^{N} \beta_i C_{imn} + \sum_{i=1}^{N} \sum_{j \geq i}^{N} \beta_{ij} D_{ijmn} = a_{mn}, \; m = 1, 2, \ldots, N, n = 1, 2, \ldots, N. \tag{51}$$

  Note that these parameters link $\hat{\mathbf{X}}$ and $\hat{Y}$.

- To evaluate the surface value at a new point $\tilde{\mathbf{x}}$, substitute the calculated best fit parameter values and the normalised values of $(\tilde{x}_1 - M_1)/\ell_1, (\tilde{x}_2 - M_2)/\ell_2, \ldots, (\tilde{x}_N - M_N)/\ell_N$ into equation (47) and calculating

$$\tilde{y} = m + s \left( \beta_0 + \sum_{i=1}^{N} \beta_i \frac{\tilde{x}_i - M_i}{\ell_i} + \sum_{i=1}^{N} \sum_{j \geq i}^{N} \beta_{ij} \frac{\tilde{x}_i - M_i}{\ell_i} \frac{\tilde{x}_j - M_j}{\ell_j} \right). \tag{52}$$

Response surface methodology is used in the scatterometry (see section 7) and thermophysical properties (see section 9) case studies.

The Latin hypercube sample of the toy model plotted in figure 9 was used for construction of a quadratic response surface. The surface and the training points are plotted in figure 10. The surface does not pass through most of the points and (as expected) does not capture the sinusoidal behaviour shown in figure 3. Many models exhibit behaviour that is sufficiently close to quadratic that a response surface is a sufficiently good approximation, particularly when used locally. If it was known that sinusoidal behaviour was likely, a more suitable surrogate model might combine a low order Fourier series in $X_1$ with a low order polynmial series in $X_1$ and $X_2$. The parameters for such a series could be determined by solving a linear least squares problem.

## 5.4 Kriging and Gaussian process modelling

Kriging was developed as a geostatistical estimator for single realizations of random fields that infers the value of the field at unobserved locations from samples. The theory was developed by Georges Matheron and Danie G. Krige [35, 36]. It is a specific application of a broader class of surrogate models known as Gaussian process emulators.

Gaussian process emulation in its simplest form is optimal interpolation based on regression against values of the surrounding training points. The predicted value of the emulator for a new set of input variable values is a weighted sum of the values of the training points. In the absence of noise in the evaluation of $\mathbf{Y}$, as is the case for most models, the emulator has zero variance at the training points. The weights in the sum determining the mean and variance of the function at a new set of input variable values are determined by a covariance function that depends on the distances between the new set of variable values and the training input values.

A Gaussian process is a (possibly infinite) set of random variables, any finite number of which have a joint Gaussian distribution [37]. A Gaussian process emulator maps between such a collection of random variables and the space of input variables of the model, such that each of the random variables represents the value of $\mathbf{Y}(\mathbf{X})$ for a given value of $\mathbf{X}$. A Gaussian process emulator is a description of the model as a random function,

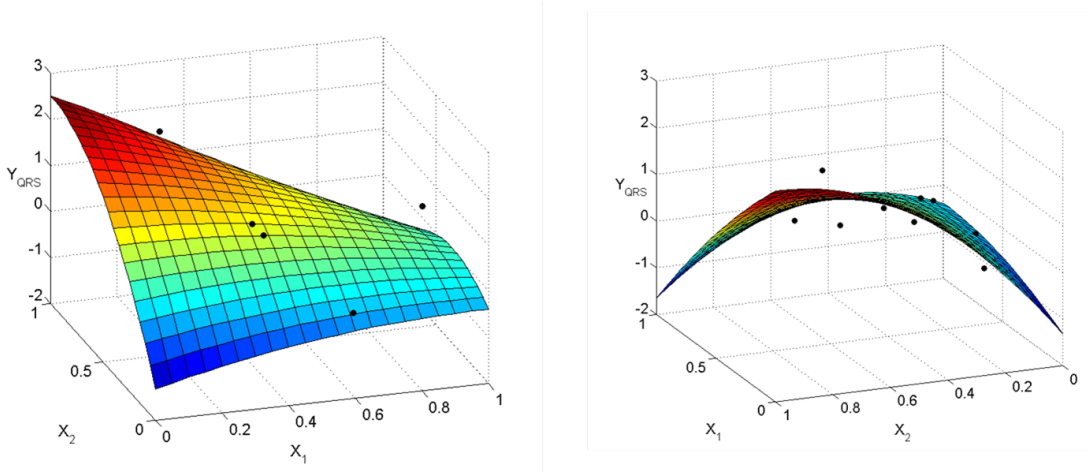



Figure 10: Quadratic response surface fitted to a Latin hypercube sample for the toy problem. The training points are shown as black dots.

in as much as for a given set of training points and hyperparameters, the emulator predicts the distribution of the function value for any new set of input quantity values.

Note that throughout the following, the joint distribution of $\mathbf{X}$ is not taken into account: the emulator treats $\mathbf{X}$ as a control variable, creates a surrogate model that estimates $\mathbf{Y}(\mathbf{X})$ at lower computational cost than the full model, and requires a separate technique for uncertainty evaluation based on this surrogate model.

Throughout the following, the case of a single output $Y$ is considered, but the technique can be extended to multiple outputs. The Gaussian process approximation [38] is given by

$$\tilde{Y}(\mathbf{X}) = \beta(\mathbf{X}) + Z(\mathbf{X}), \tag{53}$$

where $\beta(\mathbf{X})$ is the unknown mean function of $\tilde{Y}(\mathbf{X})$. The second contribution $Z(\mathbf{X})$ is a zero-mean Gaussian process with known covariance function. Alternative forms to equation (53) exist. One common form occurs in regression problems, where a parametric form of the mean function is specified in terms of a set of basis functions $\mathbf{h}(\mathbf{X})$, and parameters $\boldsymbol{\beta}$ are to be determined as part of the model fitting. Then the expression becomes

$$\tilde{Y}(\mathbf{X}) = \mathbf{h}(\mathbf{X})^{\top}\boldsymbol{\beta} + Z(\mathbf{X}). \tag{54}$$

It is not necessary to specify $\boldsymbol{\beta}$ unless there is specific information about its form: the development of the Gaussian process emulator produces an explicit form for $\tilde{Y}(\mathbf{X})$ conditional on the training data and the parameters used to define the covariance function of the Gaussian process. In practical applications it is often found to be sufficient to set $\beta = 0$, $\beta = \beta_0$ for some constant $\beta_0$, or to set $\mathbf{h}(\mathbf{X})$ to span a space of low-order polynomials. The case $\beta = \beta_0$ is known as the ordinary kriging model.

The outputs of the emulator are very strongly influenced by the form chosen for the covariance function of the stochastic process. A common approach [39, 40, 41] gives the covariance function $C$ as

$$C(Z(\mathbf{X}), Z(\mathbf{X}')) = \sigma^2 R(\mathbf{X}, \mathbf{X}'). \tag{55}$$

$R(\mathbf{X}, \mathbf{X}')$ is a parametrized correlation function that can be tuned to the training data, and $\sigma^2$ is the variance of the Gaussian process. In many practical applications exponential functions are typically chosen for the

correlation functions:

$$R(\mathbf{X}, \mathbf{X}') = \prod_{j=1}^{N} \exp\left[ -\left| \frac{X_j - X'_j}{\ell_j} \right|^{p_j} \right], \tag{56}$$

where $\ell_j$ and $p_j$ are hyperparameters. Typical values for the parameters satisfy $\ell_j > 0$ and $0 < p_j \leq 2$. For the special case $p_j = 2$ the function is a Gaussian and the covariance and hence all potential realisations of the random function are infinitely differentiable. $\ell_j$ is a length scale controlling how far the influence of any training point spreads in a given dimension. A smaller value of $\ell_j$ leads to a more rapidly changing function, tending towards white noise as $\ell_j \to 0$.

The hyperparameters for a chosen correlation function can be estimated by minimising the negative marginal log-likelihood function for the hyperparameters given the training points (see section 5 of [37] for more details). This minimisation process is a nonlinear least squares problem and is likely to require multiple runs from randomly generated initial estimates to obtain best fit hyperparameters.

Once the hyperparameters have been obtained, the Gaussian process emulator can be used to estimate the function value $\tilde{y}$ for a new point $\tilde{\mathbf{x}}$. Let $\mathbf{y} = (y^{(1)}, y^{(2)}, \ldots, y^{(K)})^\top$ denote the vector of function values at the training points, and let the correlation function be $R(\mathbf{x}, \tilde{\mathbf{x}})$. Then the joint distribution of $(\tilde{y}, \mathbf{y})$ has the multivariate Gaussian distribution

$$\begin{pmatrix} \tilde{y} \\ \mathbf{y} \end{pmatrix} \sim N_{1+K} \left[ \begin{pmatrix} \beta(\tilde{\mathbf{x}}) \\ \boldsymbol{\beta}(\mathbf{x}) \end{pmatrix}, \sigma^2 \begin{pmatrix} 1 & \mathbf{r}(\tilde{\mathbf{x}})^\top \\ \mathbf{r}(\tilde{\mathbf{x}}) & \mathbf{R} \end{pmatrix} \right]. \tag{57}$$

where $\boldsymbol{\beta}(\mathbf{x})$ is the vector of values of the mean function $\beta$ evaluated at the training points, $\mathbf{r}(\tilde{\mathbf{x}}) = (R(\tilde{\mathbf{x}}, \mathbf{x}_1), R(\tilde{\mathbf{x}}, \mathbf{x}_2), ..., R(\tilde{\mathbf{x}}$ , and $\mathbf{R}$ denotes the correlation matrix whose $i, j$ component is $R(\mathbf{x}_i, \mathbf{x}_j)$.

It follows that the conditional distribution of the model output $\tilde{y}$ at the new point $\tilde{x}$ is the univariate distribution

$$\tilde{y}|\mathbf{y}, \tilde{\mathbf{x}}, \mathbf{x} \sim N\left[ \beta(\tilde{\mathbf{x}}) + \mathbf{r}(\tilde{\mathbf{x}})^\top \mathbf{R}^{-1}(\mathbf{y} - \boldsymbol{\beta}(X)), \sigma^2 \left( 1 - \mathbf{r}(\tilde{\mathbf{x}})^\top \mathbf{R}^{-1} \mathbf{r}(\tilde{\mathbf{x}}) \right) \right]. \tag{58}$$

This distribution depends on the parameters used to specify the mean function $\beta(\mathbf{x})$ and the variance parameter $\sigma^2$ of the Gaussian process. Once these parameters have been estimated (for instance by maximum likelihood estimation; see [37]), [42], their best estimates $\hat{\beta}(\mathbf{x})$, $\hat{\beta}(\tilde{\mathbf{x}})$ and $\hat{\sigma}^2$ can be plugged into equation (58).

The mean of equation (58) is the predictive mean at input variable values $\tilde{\mathbf{x}}$ and is the Best Linear Unbiased Predictor of $Y(\tilde{\mathbf{x}})$. The variance is the mean squared error of the predicted mean.

Note that this is not a distribution for the model output $\tilde{Y}$, it is a conditional distribution for the value of $\tilde{Y}$ for a given set of values of the input variables. The distribution of $Y$ can be estimated using a sampling method and treating $\tilde{Y}$ as a surrogate model.

Figure 11 shows the mean function (plotted in black) and a 95 % confidence interval (plotted in blue) obtained by calculating the Gaussian process emulator of the function $\exp(-1.4x) \times \cos(3.5\pi x)$ (plotted in red) on $[0, 1]$ using correlation functions with different length scales. $R_1$ (left-hand plot) has $\ell = 0.0856$ and $R_2$ (right-hand plot) has $\ell = 0.2$. The calculation used seven evenly-spaced training points. Kriging predictions and 95 % confidence intervals were computed with the R package Dice Kriging [43]. The plots clearly show the effects of the length parameter $\ell$.

Gaussian process emulation is an extremely flexible method that imposes minimal assumptions on behaviour. Further flexibility is given by the wide range of covariance functions available. The functions can be tailored to fit particular features of a training data set, see section 5.4.3 of [37] for an example.

The technique can be applied to problems with many input quantities (examples with up to 50 input quantities appear in the literature). The performance is good and comparable to second order polynomial regression

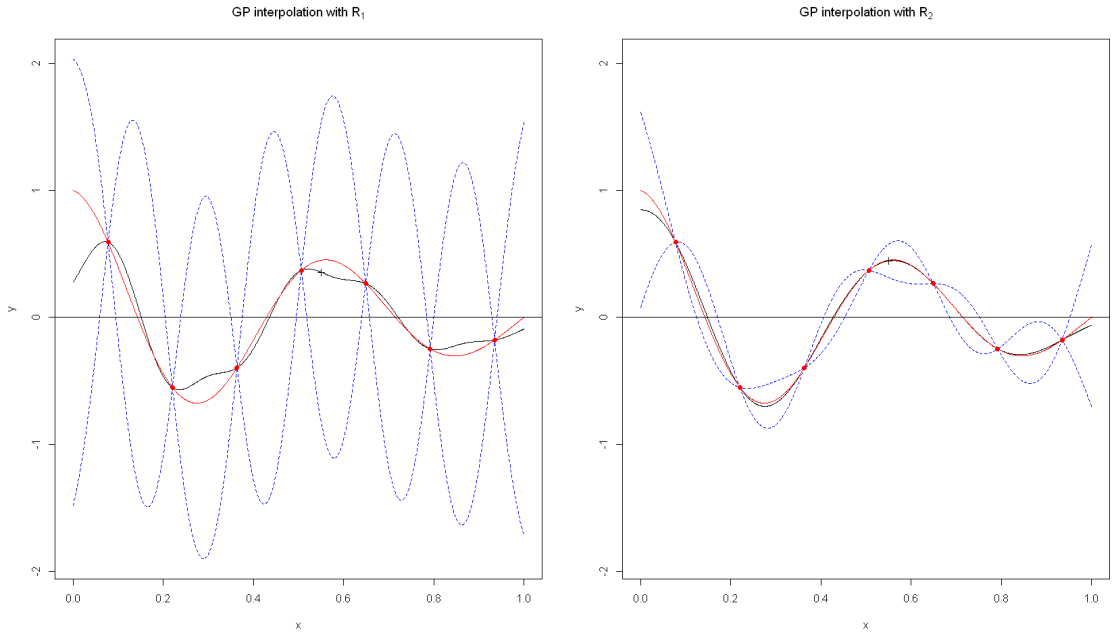



Figure 11: GP emulators with the Gaussian correlation functions $R_1$ and $R_2$. Each plot shows the true curve in red, the mean function curves in black, the bounds of a $95\%$ confidence interval in blue, and the predictions at $\tilde{x} = 0.55$ as a black cross.

models. The method can be used to identify regions of the input space that require extra training points, making the algorithm more efficient.

Drawbacks of the method include the difficulty of choosing an appropriate training data set, particularly since closely-spaced points can lead to ill-conditioning of the correlation matrix $\mathbf{R}$. The minimisation process used to determine the most likely hyperparameter values can be difficult to code and may have multiple local minima. The model construction is rather time consuming and the calculations are sophisticated, although software packages are available to carry out many of the common calculations. Addition of extra training points requires recalculation of all terms in the predictor, and since $\mathbf{R}$ is an $\ell \times \ell$ matrix that must be inverted, this can be time-consuming and potentially unstable.

To construct a Gaussian process emulator given a set of training points $\mathbf{x}^{(k)}, k = 1, 2, \ldots, K$, and corresponding function values $y^{(k)}, k = 1, 2, \ldots, K$, carry out the following steps:

- Select a correlation function. Guidance on selecting the best function for a given set of training points is available [37], but a common choice is to use an exponential function such as (56) with $p_i = 2, i = 1, 2, \ldots, N$ as this has desirable smoothness and differentiability properties.

- Given the correlation function and the training points, identify a suitable set of hyperparameters for the correlation function. Guidance is available [37], but the work reported in the thermophysical properties case study (see section 9 minimised the negative marginal log-likelihood function for the hyperparameters given the training points.

- If a parameterised mean function is required, use an appropriate method (e.g. maximum likelihood estimation) to determine the parameters of the mean function and evaluate the mean function at the training points.

- Evaluate the correlation matrix $R_{ij} = R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}), 1 \leq i, j \leq K$.

- To evaluate the emulator at a new point $\tilde{\mathbf{x}}$, use the calculated best fit parameters and the correlation matrix in equation (58).

Gaussian process emulation is used in the thermophysical properties case study (see section 9).

A Gaussian process model was fitted to the Latin hypercube sample of the toy problem plotted in figure 9. A Gaussian correlation function was used, with hyperparameters $\sigma = 2.74$, $\ell_1 = 0.44$, and $\ell_2 = 3.99$. The large value of the length parameter for $X_2$ in comparison with the typical size of $X_2$ suggests that the model is not very sensitive to $X_2$, which agrees with the functional form. The surface is plotted in figure 12. The emulator interpolates the training points (shown as black dots) and is clearly very similar to the full model as plotted in figure 3.

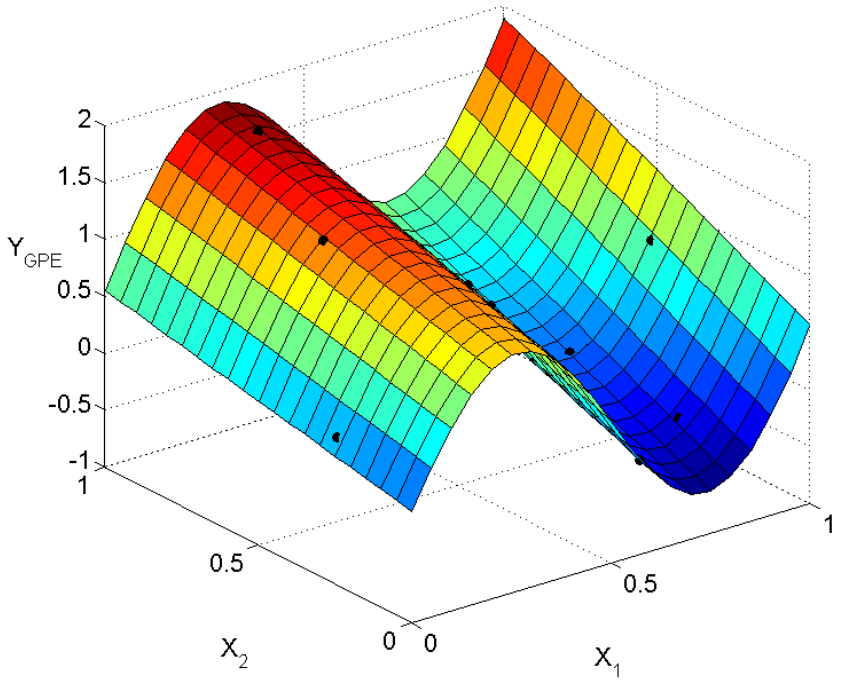



Figure 12: GP emulator for the toy problem using a Latin hypercube sample as its training points (shown as black dots).

## 5.5 Polynomial Chaos

The polynomial chaos method approximates a stochastic process by a series of polynomials. For a detailed introduction and an explanation of how to construct the approximation, the reader is referred to Section 4.5. When the polynomial chaos method is used as a surrogate model the model function $\mathbf{F}$ is approximated by $\tilde{\mathbf{F}}$.

Thus, the approximated output $\tilde{\mathbf{Y}}$ is given by

$$\tilde{\mathbf{Y}} = \tilde{\mathbf{F}}(\mathbf{r}, \mathbf{X}) = \sum_{j=0}^{q} \hat{y}_j(\mathbf{r}) \Psi_j(\mathbf{X}) \quad \text{with} \quad \hat{y}_j(\mathbf{r}) = \frac{1}{S_j} \sum_{k=1}^{K} \mathbf{F}(\mathbf{r}, \mathbf{x}^{(k)}) \Psi_j(\mathbf{x}^{(k)}) \lambda^{(k)}. \tag{59}$$

Here, $\mathbf{r}$ denotes the spatial coordinates and $\mathbf{X}$ the vector of random variables, which model the uncertain parameters in the system. Furthermore, $\{\Psi_j\}_{j=0}^{q}$ are orthogonal polynomials, which should be chosen according to the distribution of the random variable $\mathbf{X}$. This correspondence is given by the Wiener-Askey scheme, see Table 4. The modes $\hat{y}_j$, $j = 0, \ldots, q$, are calculated by a cubature rule with nodes $\mathbf{x}^{(k)}$ and weights $\lambda^{(k)}$, $k = 1, \ldots, K$.

Note that the surrogate model $\tilde{\mathbf{F}}$ can now be used to approximate the solution of the computationally expensive model $\mathbf{F}$ for a realisation $\mathbf{X}^h$. In this case, only the values $\Psi_j(\mathbf{X}^h)$ have to be calculated and plugged into equation (59) yielding $\tilde{\mathbf{F}}(\mathbf{r}, \mathbf{X}^h)$.

# 6 Processing and presenting results

## 6.1 General guidance

Whilst the presentation of results has few aspects that are unique to computationally expensive models, it is still worth presenting some general advice and tips here if only for completeness.

There are two important aspects to presentation of results. The first is the presentation of a clear statement of the process that has been used to generate a set of results so that the reader can potentially follow the same steps to obtain comparable values. The second is the presentation of a clear statement, whether numerical or graphical, of the results that have been obtained using the stated method.

The first aim requires inclusion of the following points:

- Specification of all input quantities and definition of their fixed values or distributions as appropriate.
- Specification of all choices of function or parameter associated with the uncertainty evaluation method, such as order of polynomial or choice of correlation function.
- For a sampling method: statement of sample size used and any associated choices (e.g. definition of stratification for stratified sampling, space-filling approach if a space-filling Latin hypercube sample was used, etc.).
- For a surrogate modelling method: statement of the number of training points used and their method of selection, statement of any hyperparameter values and/or the methods used to evaluate these values (e.g. "the response surface parameters were calculated by minimising the mean square error at the training points").
- Statement defining the model used to generate the results: this statement could be a complete specification of the equations solved and the numerical approach used to do so, or it could be a short statement that "Ansys been used to solve a static stress analysis with 17 482 degrees of freedom", but it should be detailed enough that the reader can ensure that their model can be reasonably expected to be comparable with the model being reported.

There are many ways of presenting results, and the choice of presentation method should be driven by the point the results are intended to highlight. For reporting numerical summaries, Supplement 1 to the GUM [2] states that

> 5.5.1 The following items would typically be reported following the use of the propagation of distributions:
>
> - an estimate $y$ of the output quantity $Y$;
> - the standard uncertainty $u(y)$ associated with $y$;
> - the stipulated coverage probability $100p\%$ (e.g. 95 %);
> - the endpoints of the selected $100p$ % coverage interval (e.g. 95 % coverage interval) for $Y$;
> - any other relevant information, such as whether the coverage interval is a probabilistically symmetric coverage interval or a shortest coverage interval.

5.5.2 $y$, $u(y)$ and the endpoints of a $100p$ % coverage interval for $Y$ should be reported to a number of decimal digits such that the least significant decimal digit is in the same position with respect to the decimal point as that for $u(y)$. One or two significant decimal digits would usually be adequate to represent $u(y)$.

and this excellent advice can be applied directly to results of the methods described in this guide.

Various graphical techniques are available to highlight particular aspects of results. Box plots were discussed in section 2 as a good way of comparing the two "halves" of a full factorial model for each variable. Box plots are also a useful way of comparing results of different samples of the same size. As was stated in section 2, the box of the plot represents the interquartile range for each sample, the median is marked by a solid line, the length of the stems is a multiple (1.5) of the interquartile range, and the points that lie beyond the length of the stems are treated as outliers and are denoted with plus symbols. Figure 13 shows two box plots comparing the results of the thermophysical model evaluations for ten Latin hypercube samples of size 10 (left hand plot) and ten random samples of size 10 (right hand plot). The plots suggest that the medians of the Latin hypercube samples are more consistent than those of the random samples, but that both sets of samples show a lot of sample-to-sample variability.

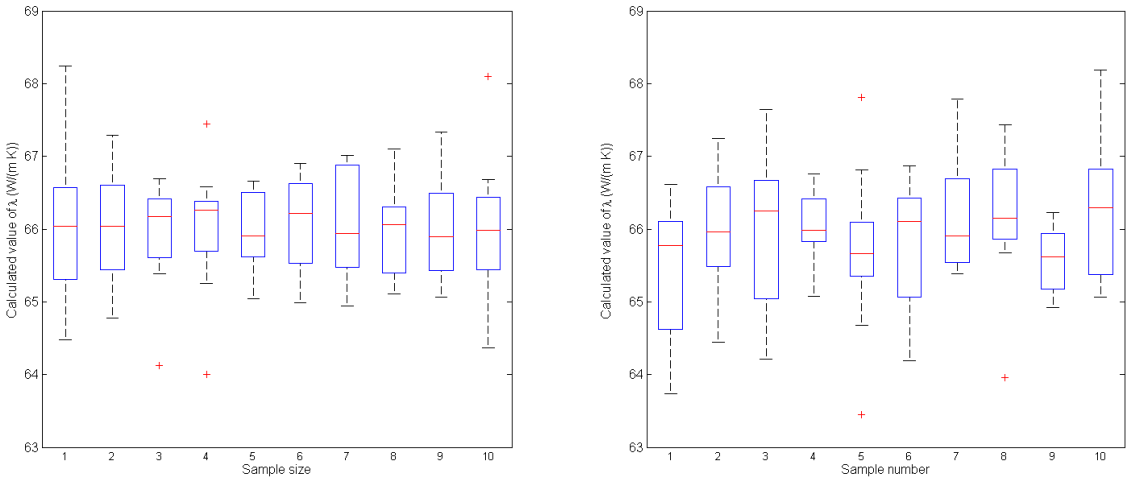



Figure 13: Comparison of the results of a set of ten Latin hypercube samples of size 10 (left hand plot) to the results of a set of ten random samples of size 10.

Another useful way of looking at repeatability across multiple samples of the same size is an envelope CDF plot. This plot shows the highest and lowest values of the output quantity at given percentiles that occur within the samples, and are constructed by taking the maximum and minimum value of the quantity at each percentile from the CDFs of the samples. An example is shown in figure 14. Again, this plot compares the results of ten Latin hypercube samples of size 10 (left hand plot) and ten random samples of size 10 (right hand plot), with the red line showing the reference CDF. These plots suggest that the percentiles evaluated using the Latin hypercube samples are closer to the reference distribution than those evaluated using random sampling.

It should be noted that the envelope width will increase as the number of samples increases. If a large number of samples are being compared, it may be better to take the 10th and 90th percentiles of the range of values at each probability level to ensure comparability of envelopes across different numbers of samples.

A good way of comparing two distributions is a quantile-quantile plot. These are scatter plots of quantiles computed from two samples, with a line drawn between the first and third quartiles. These are a particularly

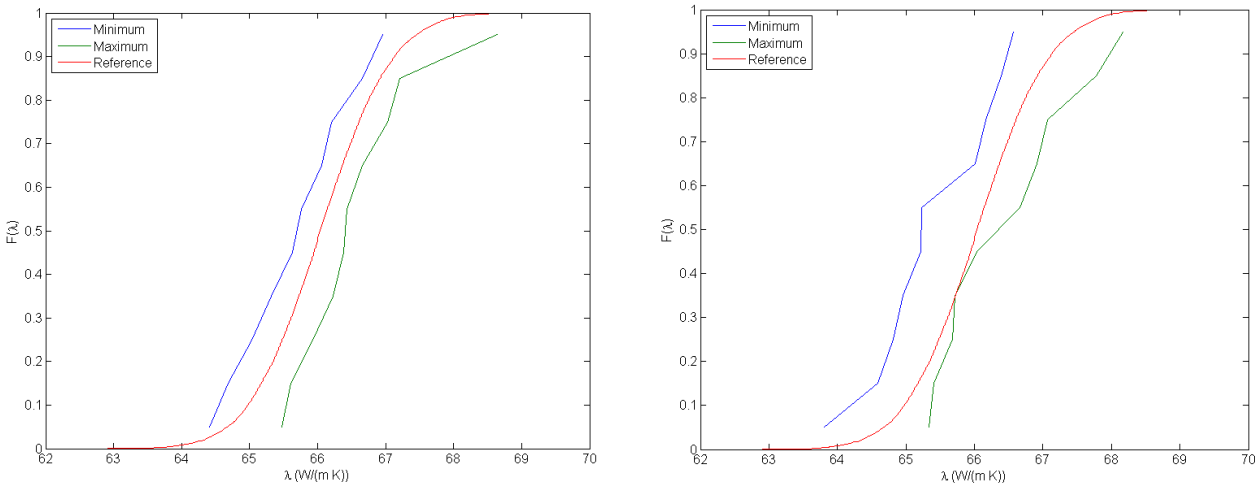



Figure 14: CDF envelop plot comparing the maximum and minimum values at percentiles across ten Latin hypercube samples (left hand plot) to those obtained across ten random samples (right hand plot). In each plot the red curve is a set of reference values calculated separately.

convenient way of comparing distributions to a reference distribution. An example is shown in figure 15, where the results of Gaussian process emulators based on three different training point determination methods are being compared to a reference distribution. The reference distribution is indicated by the line $y = x$. It is clear from the plot that the emulator using the Latin hypercube sampling for the training points shows signifcant deviation from the line $y = x$, suggesting that the CDF calculated with this emulator may not be accurate at the tails of the distribution.

## 6.2 Toy problem results

The toy problem has been used to illustrate the various methods used in this report. It should be noted that some of the models are not suitable for a problem with the sinusoidal nonlinearity that occurs in this problem: this does not mean that the methods are “bad”: they are just not suitable for this problem.

Each of the sampling methods was used to create 10 samples of size 10. The mean and standard deviation of each sample were calculated, and for each method the mean of means and the mean of the standard deviation were calculated, along with the sample-to-sample variation of the sample mean and standard deviation.

The ten Latin hypercube samples were used as training points for each of the surrogate modelling methods. Once the surrogate models had been constructed, random sampling with a sample size of $10^4$ was used to calculate the mean and standard deviation of the surrogate model, and the sample-to-sample variation was analysed.

The polynomial chaos method was used to evaluate the mean and standard deviation using deterministic quadrature, so repeatability is not relevant.

The results of all methods are listed in table 5. These results suggest that for a problem like this which is not well approximated by a single quadratic function, the methods that give best accuracy and repeatability are Latin hypercube sampling, polynomial chaos, nearest neighbour spproximation, and Gaussian process modelling.

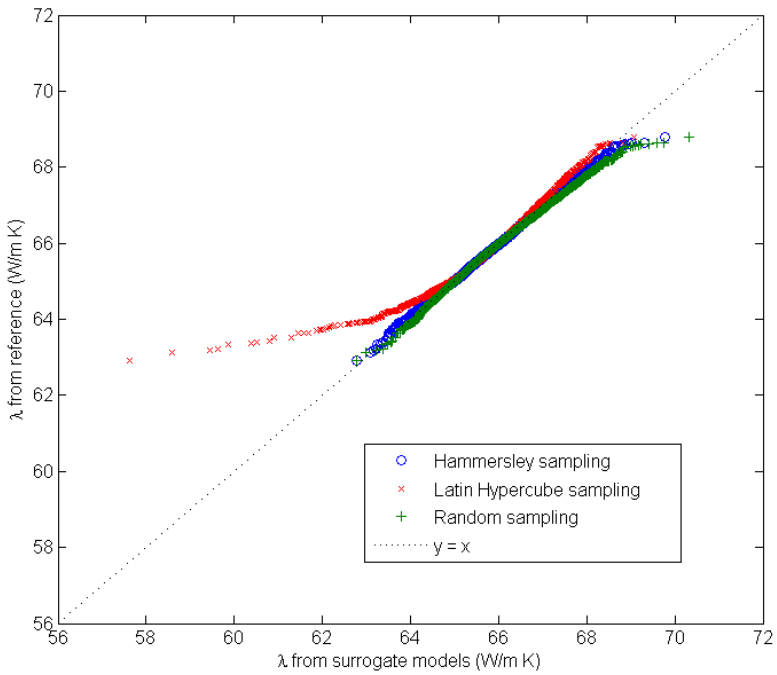



Figure 15: Quantile-quantile plot comparing the results obtained from Gaussian process emulators constructed based on three different training point generation methods.

| Method | Mean of means | Mean of standard deviations | Standard deviation of means | Standard deviation of standard deviations |
|---|---|---|---|---|
| Reference results | 0.67 | 0.57 | | |
| Random sampling | 0.58 | 0.51 | 0.22 | 0.05 |
| Importance sampling | 0.68 | 0.31 | 0.13 | 0.12 |
| Stratified sampling | 0.56 | 0.54 | 0.14 | 0.05 |
| Latin hypercube sampling | 0.65 | 0.58 | 0.03 | 0.05 |
| Polynomial chaos | 0.67 | 0.56 | | |
| Nearest neighbour | 0.64 | 0.57 | 0.04 | 0.05 |
| Response surface | 0.68 | 0.67 | 0.07 | 0.26 |
| Gaussian processes | 0.66 | 0.57 | 0.01 | 0.01 |

Table 5: Collected results of the toy problem illustrating accuracy and repeatability.

# 7 Case Study 1: Scatterometry

## 7.1 Problem description

In scatterometry, periodic nano-structured surfaces are illuminated by a monochromatic light source and diffracted intensities are measured. Relative maxima of diffracted intensities (efficiencies) are used to reconstruct the unknown geometry. In this context, the forward model is defined as a map from parametrized geometries onto the efficiencies of diffracted light.

In our case study we used a model of an EUV photo mask geometry that consists of a multilayer, two capping layers, and periodic straight absorber lines of three different materials. The period of the line structure is 280 nm and the geometry parameters are bottom-CD $= 140$ nm (width of bottom), and height of the second absorber layer $= 80$ nm. The angle of the upper layer is fixed to employ corner rounding; the angle of the second TaN layer is called side wall angle (SWA). Typically, the SWA, bottom-CD and the height are reconstructed. These are the geometric parameters considered in this work. A cross section of the geometry is shown in figure 16.

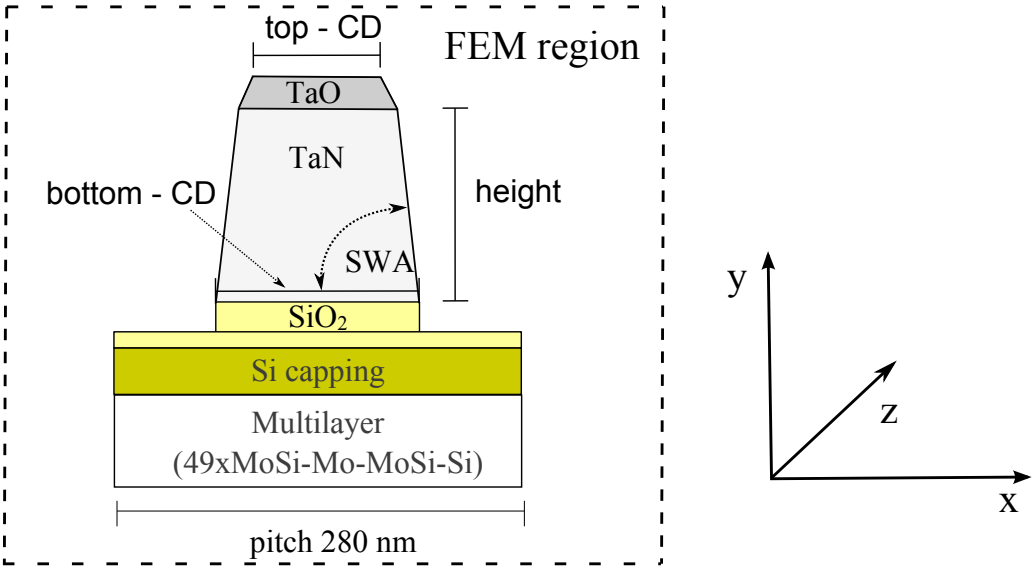



Figure 16: Cross section of the EUV photo mask for one period.

## 7.2 Mathematical model

The mathematical model is characterized by the propagation of electromagnetic waves through optical materials. Wave propagation of electromagnetic waves in the context of scatterometry is well described by Maxwell's equations.

The specific line structure chosen implies invariance of optical properties in one spatial direction, here labeled as the $z$-direction. Therefore, Maxwell's equations can be simplified and reduced to the two dimensional

Helmholtz equation in the $x-y$ plane,

$$\Delta u\left(x,y\right)+k^{2}\left(x,y\right)u\left(x,y\right)=0, \tag{60}$$

where $u$ is the transversal field component that oscillates in the z-direction and $k\left(x,y\right)$ is the wave number

$$k\left(x,y\right)=\omega\left(\mu_{0}\epsilon\left(x,y\right)\right)^{1/2}, \tag{61}$$

that is assumed to be constant for areas filled with the same material. The boundary conditions that are imposed on the PDE are periodic on the lateral boundaries due to the periodic structure and usual outgoing wave conditions in the infinite regions [44]. For rigorous calculations we used the finite element (FEM) solver DIPOG (developed at WIAS; http://www.wias-berlin.de/software/DIPOG). We fixed the wavelength of the incident light $\lambda=13.4\ \mathrm{nm}$, chose an incident angle of $6^{\circ}$ and used values of the optical parameters given in [45]. For every set of geometry parameters $\mathbf{p}=\left(p_{1}....p_{N}\right)^{\top}$, there is a set of efficiencies $f_{1}(\mathbf{p})...f_{M}(\mathbf{p})$ which defines a nonlinear map (forward model)

$$\mathbf{p}\mapsto f_{j}(\mathbf{p}). \tag{62}$$

In the work reported here, $M=22$ efficiencies were used. Efficiencies obtained in experiments include additional noise. For scatterometry, two main sources of noise are known. Variations of the incoming light beam and noise from background fluctuations. To include these effects, we extend our model by adding a Gaussian distributed noise term $\epsilon_{j}$:

$$y_{j}=f_{j}\left(\mathbf{p}\right)+\epsilon_{j}. \tag{63}$$

Here, the standard deviation of Gaussian distributed noise consists of two independent contributions embodied in the parameters $\tilde{a}$ and $\tilde{b}$, that yield

$$\tilde{\sigma}_{j}^{2}=\left(\tilde{a}f_{j}\right)^{2}+\tilde{b}^{2}, \tag{64}$$

where the first term models fluctuations of the light beam and the second represents background noise. When we use numerical approximation of the forward model there is an additional error. A simple Ansatz of a complete error model is given by a multivariate Gaussian distribution with standard deviations

$$\sigma_{j}^{2}=\left(af_{j}\right)^{2}+b^{2}, \tag{65}$$

where the error parameters include the measurement error and the approximation error, i.e., $a^{2}=\tilde{a}^{2}+a_{\mathrm{approx}}^{2}$ and $b^{2}=\tilde{b}^{2}+b_{\mathrm{approx}}^{2}$.

For the estimation of geometry parameters an inverse problem has to be solved, which makes it necessary to evaluate the forward model equation (62) several times which is computationally expensive.

## 7.3 Implementation of surrogate methods

Three different types of surrogate model were created to replace the FE model within the inverse problem. The input quantities were the SWA, bottom-CD and height as shown in figure 16. The parameters $a$ and $b$ are not part of the computationally expensive (i.e. FE) part of the model and so were not included as inputs into the surrogate model.

### 7.3.1 Nearest-neighbor interpolation

For the NN interpolation we divided the input parameter space into 729 cubes and calculated output values from the midpoints of all cubes rigorously with the FEM solver. From this set we constructed a look up table to approximate the full physical model. Every output value is obtained by the identification of the cube containing the input values and reading off the output value which is related to that cube.

### 7.3.2 Response surface methodology

In our case study, we used 609 training point input values. The training point outputs $Y_m(\mathbf{x}_k), m = 1, 2, \ldots, M, , k = 1, 2, \ldots, K$, are rigorously calculated by a FEM solver. The hyper-parameters are calculated by projecting $Y_m(\mathbf{x})$ onto the polynomial basis. With the obtained hyper-parameters we construct the quadratic response surface as an approximation of the output quantities.

### 7.3.3 Polynomial chaos

In this case study all input distributions are chosen as rectangular distributions $\mathrm{R}[p_i^{\min}, p_i^{\max}]$, where $p_i = \mu_i + \omega_i$ where $\mu_i$ is the mean of $p_i$ (the $i$th input quantity) and $\omega_i$ is the half-range. According to the Wiener-Askey scheme, we have chosen Legendre polynomials as basis functions $\Psi_k$ and the expansion coefficients have been calculated by projections onto the basis functions $\Psi_k$, i.e.,

$$\beta_{im}^{\alpha}(\boldsymbol{\mu}) = \frac{\int Y_m(\mathbf{x})\Psi_k(\omega_i)U(\omega_i)d\omega_i}{\int \Psi_k^2(\omega_i)d\omega_i} \approx \frac{1}{\int \Psi_k^2(\omega_i)U(\omega_i)d\omega_i}\sum_k \lambda_k^i Y_j(\omega_i^k)U(\omega_i^k), \tag{66}$$

The integral in the denominator was solved analytically and other integrations are approximated by sparse Legendre quadrature rules. In our case study, we calculated expansion coefficients up to the 10th order. By using sparse grid techniques 608 training points are needed for good approximation results. The location of training points were given by the sparse Legendre grid.

## 7.4 The Bayesian approach

The Bayesian approach provides a statistical method to solve the inverse problem by including prior information. For more details see [30]. We will apply this approach to computationally expensive problems.

We assume normal distributed measurement errors and therefore the likelihood function is given by

$$\mathcal{L}(\mathbf{p}, a, b; \mathbf{y}) = \prod_{j=1}^{M} \frac{1}{\sqrt{2\pi\sigma_j^2(\mathbf{p}, a, b)}} \exp\left[-\frac{(y_j - f_j(\mathbf{p}))^2}{2\sigma_j^2(\mathbf{p}, a, b)}\right], \tag{67}$$

$$\text{with } \sigma_j^2 = (af_j(\mathbf{p}))^2 + b^2. \tag{68}$$

The estimates of parameters $\boldsymbol{\theta} = (\mathbf{p}^\top, a, b)$ are given by the expectation values $\bar{\theta}_i = E[\theta_i]_\pi$ and the associated variances are given by $V(\theta_i) = E[\theta_i^2]_\pi - E[\theta_i]_\pi^2$ with $E[\theta_i]_\pi = \int \theta_i d\pi$, where $d\pi$ is the posterior measure. For our study we have chosen uniform prior distributions: $\mathrm{R}[128.8, 151.2]$ nm for the bottom CD, $\mathrm{R}[74.4, 85.6]$ nm for the height and $\mathrm{R}[82.5, 92.5]\circ$ for the sidewall angle. These distributions are identical to those used in the polynomial chaos expansion described above. The parameters $a$ and $b$ were also assigned uniform prior distributions, $\mathrm{R}[0.01, 0.1]$ for $a$ and $\mathrm{R}[0.001, 0.1]$ % for b.

## 7.5 Results of the case study

### 7.5.1 Comparison of surrogates

All surrogate models are constructed from 608 to 729 training points and validated with rigorous FEM calculations. We have randomly chosen one hundred geometries from the prior distribution, calculated all efficiencies with each surrogate model and compared the results with rigorous FEM calculations.

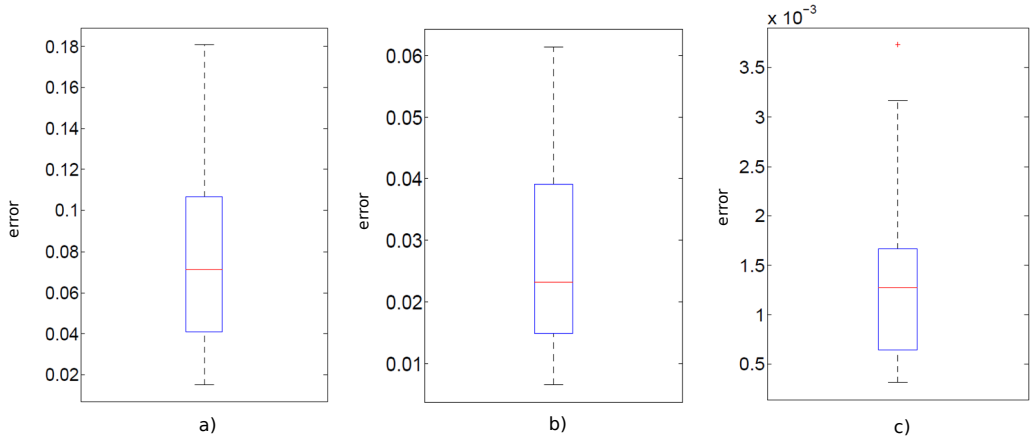



Figure 17: Boxplots of the $L^2$-norm of the difference between rigorous FEM calculations and surrogate approximations of the forward model (approximation error) for hundred geometries chosen randomly from the prior distribution. a) nearest neighbor method, b) quadratic response surface methodology and c) polynomial chaos approach.

| Application | Function evaluations | FEM | NN | RSM | PC |
|---|---|---|---|---|---|
| Sensitivity Analysis | $3 \times 10^4$ | 41 d | 20 h + 5 min | 20h +9.0 min | 20h + 0.1 s |
| Bayesian MCMC | $\sim 10^5$ | 139 d | 20 h + 17 min | 20 h + 30 min | 20 h + 47 min |
| Maximum Likelihood | 30 | 1 h | 20 h + 0.3 s | 20 h + 0.54 s | 20h + 0.84 s |
| Least squares | 15 | 0.6 h | 20 h + 0.15 s | 20 h + 0.27 s | 20 h + 0.42 s |

Table 6: Speed up on various applications for different surrogate models. All surrogate methods need a pre-calculation time of about 20 h to determine the hyper-parameters. Here, d denotes days, h hours, min minutes and s seconds.

For every geometry we calculated the $L^2$-Norm of the difference between rigorous calculations and the approximation, i.e., error = $\sqrt{\sum_{j=1}^{22}(f_j^{\mathrm{FEM}}(\mathbf{p}) - f_j^{\mathrm{surrogate}}(\mathbf{p}))^2}$. Figure 17 shows box-plots of the error of each surrogate model. The approximation error of the surrogate model based on the polynomial chaos expansion is more than one order of magnitude smaller than that of the others.

To estimate the speed up of each model, we measured the time of function evaluations. Since the construction of the surrogate model requires the calculation of about 700 training points, there is a pre-calculation time of about 20 hours for each model. For applications like Bayesian inference or sensitivity analysis, typically more than $10^4$ function evaluations are necessary and the computational cost of the forward model becomes significant. In table 6 we compare the time duration of all models with rigorous FEM calculations for various hypothetical applications of the model. For the estimation of one parameter set within the Bayesian approach using rigorous FEM calculations a time of more than six months is needed. In contrast, the computational time reduces to less than an hour when a surrogate model is applied.

| Surrogate | $a$ | $a_{\text{approx}}$ | $\tilde{a}$ | $b$ | $b_{\text{approx}}$ | $\tilde{b}$ |
|---|---|---|---|---|---|---|
| RSM | 0.03666 | 0.03096 | 0.01932 | 0.002670% | 0.00159% | 0.002146% |
| PC | 0.02467 | 0.01160 | 0.02177 | 0.002334% | 0.00113% | 0.0020434% |

Table 7: Error parameters $a$, $b$ and $a_{\text{approx}}, b_{\text{approx}}$ are calculated from the posterior distribution. $\tilde{a}$ and $\tilde{b}$ are calculated with equation (65). For simulations of the measurements we used an error of $\tilde{a} = 0.02$ and $\tilde{b} = 0.002\%$.

### 7.5.2 Parameter estimation and uncertainty quantification

We simulated a set of measurement data by choosing randomly a geometry from the prior distribution, calculating the corresponding efficiencies rigorously and superimposing noise accordingly to the error model (68). Here, we have chosen $\tilde{a} = 0.02$ and $\tilde{b} = 0.002\%$ corresponding to experimental experiences in EUV-scatterometry [45, 46].

Furthermore, we used a Markov-Chain Monte Carlo (MCMC) sampling method using each surrogate model to estimate geometry parameters and determine its uncertainties from the approximated posterior distribution. MCMC methods are described in detail elsewhere [30]. For diagnostics we analyzed mixing and used the Gelman-Rubin criterion [47].

To estimate the geometry parameters and associated uncertainties from the posterior distribution we construct Markov chains with $2 \times 10^5$ steps. For the burn in phase we used $25000$ steps according to the diagnostics result.

We estimated all parameters by calculating the means of the posterior distributions and determined the uncertainties by calculating $95\%$ credible intervals of the distribution. Figure 18 depicts results for the bottom-CD. The error bars indicate the credible intervals and can be asymmetric if the underlying distribution is asymmetric. For all data sets, the geometry was the same ('true' geometry). Differences appear due to randomly distributed error according to equation (68). The 'true' geometry is marked by the red dashed lines. The green dashed lines give the mean value of the set of estimations.

In the case of the NN method, Markov-Chains are drifting to the boundary of the support of the prior and the posterior becomes asymmetric. The 'true' bottom-CD was not estimated and uncertainties are not consistent with the true value. Estimations using the RSM are consistent but not very precise and with large uncertainties ($\sigma \approx 3.0$ nm). In addition there is a bias of about $2$ nm which is independent of the simulated measurement noise. In the case of PC expansion results are almost all consistent, uncertainties are smaller ($\sigma \approx 1.5$ nm) and there is no bias. It turns out that the PC method provides precise and accurate results in contrast to the other two methods.

### 7.5.3 Approximation error

So far uncertainties include the measurement noise and the approximation error. To separate both errors, we first set $\tilde{a} = \tilde{b} = 0$ and used MCMC sampling to obtain the posterior distribution for the approximation error in the same way as explained in the previous section. Secondly, we calculated the posterior distribution including the full error, where the measurement error was set to $\tilde{a} = 0.02$ and $\tilde{b} = 0.002\%$. In table 7 results for the estimated error parameters are compared. The values of $a_{\text{approx}}$ and $b_{\text{approx}}$ for the RSM are large due to the large approximation error. But estimations for $\tilde{a}$ and $\tilde{b}$ are in a good agreement with the chosen values for both methods.

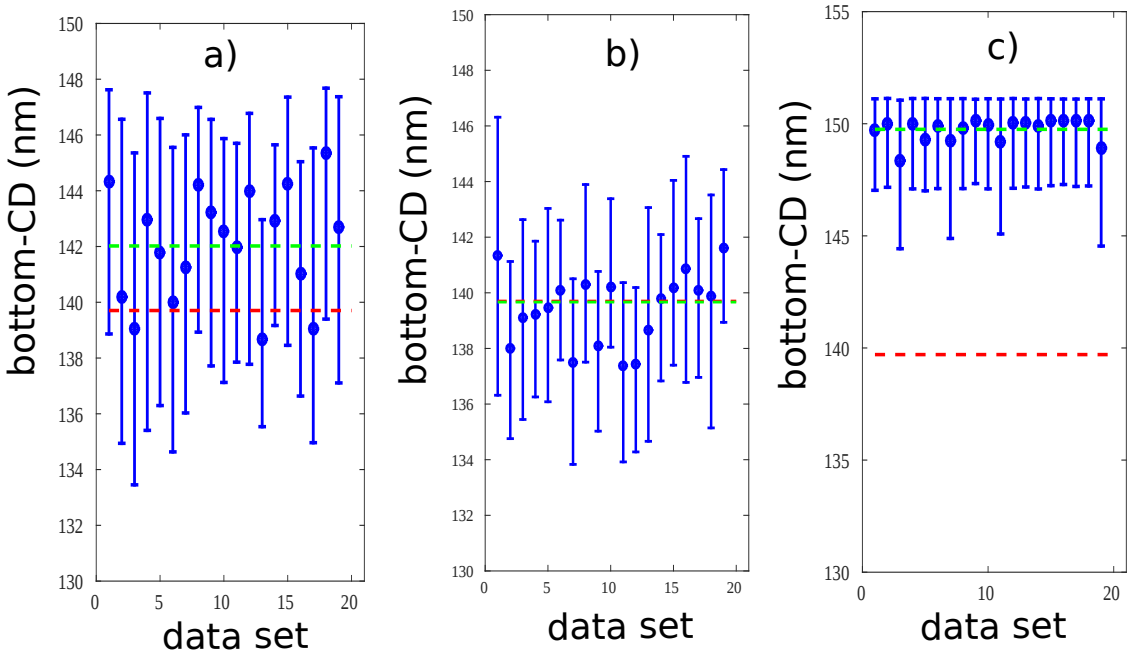



Figure 18: Estimation of the bottom-CD (line bottom width) by using different surrogate models and the Bayesian approach. a) Quadratic response surface methodology, b) Polynomial chaos approach and c) Nearest neighbor interpolations. For each surrogate model 100 000 Markov-Chain Monte Carlo samplings are used, where the burn in phase was to be up to 25 000 sampling steps. Estimations are obtained from a data set of 20 simulated measurements. Red dashed line is the initial bottom-CD and the green dashed line is the average of 20 estimated bottom-CDs. In figure b), the red and green lines overlay one another.

This short consideration shows that the approximation error can be determined in advance and could be used to determine the pure measurement error.

### 7.5.4 Conclusions for the case study scatterometry

In this case study, we applied and discussed different surrogate models (NN, RSM, PC) to approximate the forward model of a scatterometry measurement experiment. We validated each model by rigorous finite element calculations and calculated the associated approximation error. For the same number of training points the approximation error was large for NN interpolations, moderate for the RSM and small for the PC expansion.

Furthermore, surrogate models are used to enable the application of the Bayesian approach to scatterometry. By using a Markov-Chain Monte Carlo method (Metropolis Hastings) uncertainties are determined. We found reasonable agreement and consistency with the geometry used for the RSM and the PC method. For the number of training points chosen the PC method gives the best performance with respect to accuracy and precision.

# 8 Case Study 2: Fluid flow

This case study is presented in two distinct parts: one section analysing the uncertainties in pipe flow measurement caused by the inflow profile, and the other section evaluating the uncertainties associated with a Venturi nozzle.

## 8.1 Pipe flow

### 8.1.1 Numerical model

In the following, the influence of uncertain inflow profiles modeling bends or restrictions further upstream in the pipe on the flow field is investigated. This is an illustrative example how the non-intrusive polynomial chaos approach introduced in section 4.5 can be applied. We consider the flow of water at 20° C in a "DN 50" pipe of diameter $d = 55$ mm and length $L = 5500$ mm. The geometry is shown in Figure 19. Note that cylindrical

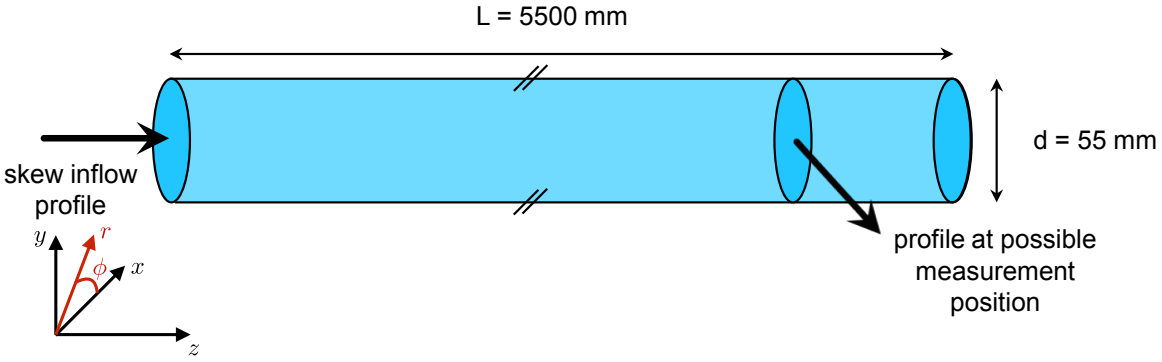



Figure 19: Schematic drawing of the considered geometry.

coordinates $(r, \phi, z)$ are used. The origin $(0, 0, 0)$ is at the inlet in the middle of the pipe. The pipe is assumed to be hydraulically smooth. The stationary flow through a pipe is described by the stationary Navier-Stokes equations, modeling conservation of mass and momentum, given by

$$\nabla \cdot \mathbf{u} = 0, \qquad \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u}. \tag{69}$$

Here, $\mathbf{u} = (u_r, u_\phi, u_z)^\top$ is the velocity field with radial, angular, and axial component, $p$ the pressure, $\rho$ the density of the fluid, and $\nu$ its kinematic viscosity. We prescribe standard no-slip boundary conditions on the walls, a zero-gradient boundary condition on the outlet, and an inflow boundary condition on the inlet:

$$u_r = 0 \text{ for } r = R, \quad \frac{\partial u_z}{\partial z} = 0 \text{ for } z = L, \quad u_z(r, \phi, 0) = U(r, \phi) \text{ for } z = 0.$$

The numerical simulations of the turbulent flow through such a pipe are performed by ANSYS CFX® in three dimensions. The turbulence is modeled by Reynolds-averaging the Navier-Stokes equations and applying the SST-model of Menter, see [48]. The ouptuts of interest are the values of $u_z$ across the cross-section of the pipe for various values of $z$.

### 8.1.2 Validation

In order to check the accuracy of the CFD model, we compare with experimental data[1]. After a half aperture, the flow profile was measured by a laser Doppler velocimetry (LDV) system. From this data, we use the velocity vectors and the turbulence data as input data for the CFD model. Further downstream the velocity profile was again measured by a second LDV system. These data are used for comparison with the CFD simulations. In Figure 20 a vertical cut of the measured profile, one of the numerically obtained profile on different meshes, and a curve of the ideal profile of Gersten and Herwig [49, 50] are shown. The curves of the numerically obtained

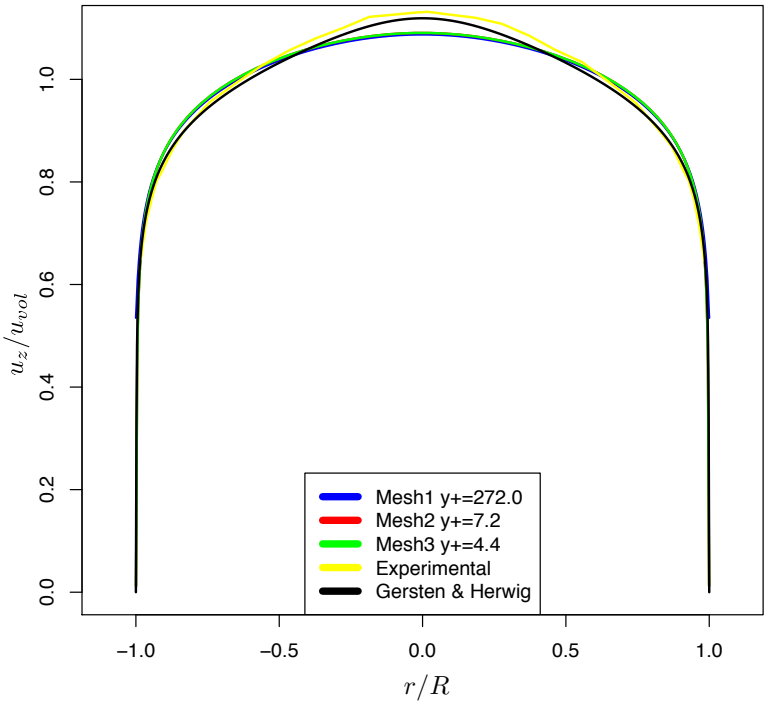



Figure 20: Vertical cut of the measured profile, a vertical cut of the numerical obtained profile on different meshes and the ideal profile of Gersten and Herwig. The difference between the measured profile and the with CFD calculated solution is less then $4\%$.

profiles are close together, which means that the numerical method is converged. The profile of Gersten and Herwig is in good agreement with the experimental data. The numerical profile is flatter than the curve of the ideal profile and the measured profile. However, the maximal difference is under $4\%$.

### 8.1.3 Uncertain inflow profiles

The propagation of uncertain inflow profiles is investigated by a generalized polynomial chaos approach. A family of skew inflow profiles is considered, i.e., $U = (0, 0, U_z)^\top$ with

$$U_z(r, \phi) = \frac{1}{2} \left( W(-r) - W(r) \right) (\cos \phi + 1) + W(r), \tag{70}$$

[1] Obtained at Physikalisch-Technische Bundesanstalt (PTB), Bundesallee 100, 38116 Braunschweig, Germany

and

$$W(r) = \begin{cases} -\frac{W_{\max}(r+R)(r-R-2r_M)}{(R+r_M)^2}, & -R \le r \le r_M, \\ -\frac{W_{\max}(r-R)(r+R-2r_M)}{(R-r_M)^2}, & r_M < r \le R. \end{cases}.$$

Here $r$ is the radial, $\phi$ the angular direction, $r_M$ denotes the position of the maximum, which is assumed to be uniformly distributed in the interval $[-R, R]$ (where $R$ = 27.5 mm), and $W_{\max} = 5.377\mathrm{m/s}$ is the maximal axial velocity in the pipe. The Reynolds number is about Re = $1.82 \times 10^5$.

Since only one random parameter is considered, we have $N = 1$ and $p = d_p$ in equation (39). Since we model that all positions of the maimum velocity have equal probability, the random variable is assumed to be uniformly distributed in the interval $[-R, R]$. The output of interest is the axial component of the velocity further downstream in the pipe. This models the position of a possible flow meter in practice. According to the assumed distribution of the random variable and the Askey scheme, Legendre polynomials were used as ansatz functions in the polynomial chaos approach, see Table 4. Furthermore, we have used a Gauss-Legendre quadrature rule for the calculation of the modes in the polynomial chaos approximation, for details see section 4.5. Note that this quadrature rule is exact for polynomials of degree $2q - 1$ or less.

For the generalized polynomial chaos approximation we have used $q = 50$ collocation points. The Gauss-Legendre quadrature rule is exact at least for polynomials up to order $2q - 1 = 99$. If we assume that the output of interest behaves like a polynomial of degree $d_s$ with respect to the uncertain parameter, then the ansatz polynomials should be of order $2q-1-d_s = 99-d_s$. The polynomial degree $d_p$ was chosen as $d_p = q-1 = 49$. So it is assumed that the solution behaves like a polynomial of degree $d_s = 2q - 1 - d_p = q = 50$.

### 8.1.4 Results

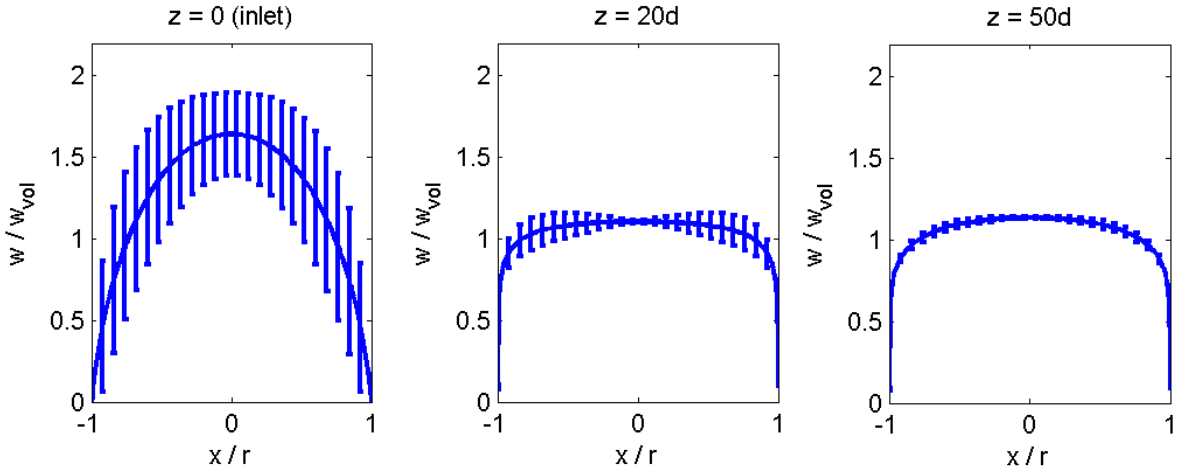



Figure 21: Expectation value $\pm$ standard deviation of velocity profiles at different $z$-positions (cut line $\phi = 0$), calculated by generalized polynomial chaos with $q = 50$ collocation points and polynomial degree $d_p = q-1 = 49$.

Figure 21 shows the radial distribution of the expectation value $E(u)$ $\pm$ standard deviation $\sqrt{V(u)}$ of velocity profiles at different $z$-positions for a cut line at $\phi = 0$. The first picture of Figure 21 gives the expectation value $\pm$ standard deviation for the prescribed inflow profiles. One can see that the family of prescribed profiles covers quite a large range of possible inflow profiles. Further downstream, which can be seen in the next pictures, the variation between the different profiles decreases. At $z = 50D$ the variance has

become very small and all prescribed profiles look almost like a “standard” fully-developed turbulent profile, independent of the position of the maximum prescribed at the inlet.

In order to check the convergence of the method, the expectation and variance of the approximated solution $\tilde{\mathbf{u}}$, see equation (39), at the inlet are compared with the expectation and variance of the prescribed family of parabolic inflow profiles, see equation (70). Figure 22 compares polynomial chaos approximations of different orders with the analytically calculated solution at the inlet. In the left picture, the expectation value is depicted.

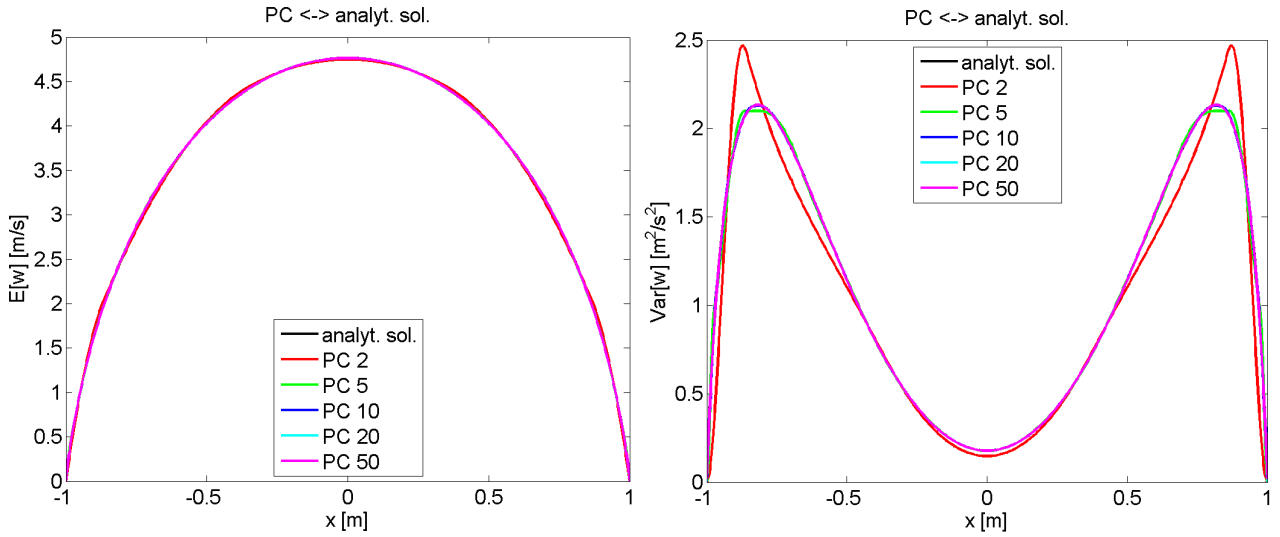



Figure 22: Comparison of the expectation value and the variance between generalized polynomial chaos of different orders and the analytical solution at the inlet.

The right picture shows the variance. One can see that the polynomial chaos ansatz with only $q = 5$ collocation points gives already good approximations of the true expectation value and variance. Table 8 shows the decay of the error of the polynomial chaos approximation with increasing order. Here, $U$ denotes the prescribed inflow profile and $\tilde{u}(r, \phi, 0)$ the polynomial chaos approximation at the inlet $z = 0$.

| $q$ | $\|E(U) - E(\tilde{u}(\cdot,\cdot,0))\|_\infty$ in m/s | $\|V(U) - V(\tilde{u}(\cdot,\cdot,0))\|_\infty$ in $\mathrm{m}^2/\mathrm{s}^2$ |
|---|---|---|
| 2 | 0.1183 | 1.1127 |
| 5 | 0.0291 | 0.2212 |
| 10 | 0.0131 | 0.0694 |
| 20 | 0.0074 | 0.0197 |
| 50 | 0.0073 | 0.0098 |

Table 8: Maximum absolute error of expectation value and variance for different number of collocation points $q$.

### 8.1.5 Summary

We show that gPC is a promising approach to treat flow problems with random input parameters, which are well-known to be computationally expensive. In contrast to classical Monte-Carlo methods, the approach only

needs a few evaluations of the underlying deterministic problem. However, if the output of interest does not depend smoothly on the random parameters, a relatively high polynomial degree and thus many evaluations are necessary even for a method as efficient as gPC. In this case, a stochastic finite-element approach may be preferable [51].

## 8.2 Sonic nozzle

### 8.2.1 Problem Description

The physical problem of interest concerns a Venturi nozzle through which air is flowing, see figure 23. The air speed at the narrowest part of the nozzle, the throat, is sonic under standard operating conditions. The inlet pressure $P_{\text{in}}$, the inlet temperature $T_{\text{in}}$, the outlet pressure $P_{\text{out}}$, the molar mass $M$, dynamic viscosity $\mu$, specific heat at constant volume $C_v$ and the Prandtl number $Pr$ of the air are all known with some uncertainty. In table 9 the corresponding probability distributions are listed. The input quantities are assumed to be independent. The governing mathematical equations are the Navier-Stokes equations. The quantity of interest is the value of the mass flow rate $q$ flowing through the nozzle and its uncertainty due to the uncertainty in the parameters mentioned above. The mathematical model was solved using the OpenFOAM [52] solver sonicFoam. Performing four model evaluations in parallel on a desktop computer with 4 Intel(R) Core (TM) i7 CPU 870@2.93 GHz processors and 7.8 GB of RAM memory took approximately 30 minutes. The memory usage was slightly over 30 %, whereas the 4 CPUs were performing at 100 %.

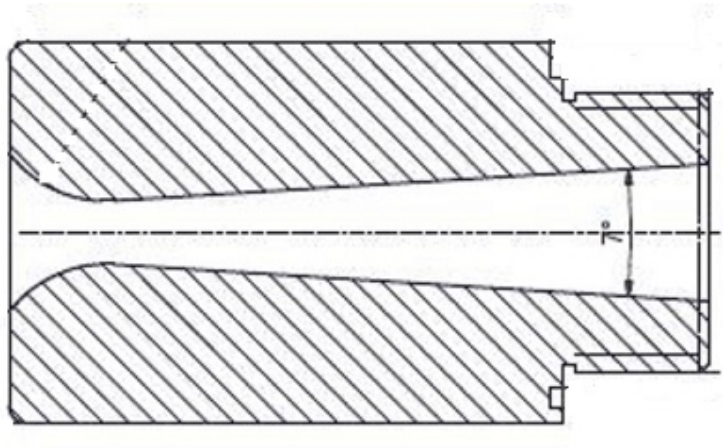


Figure 23: Schematic drawing of a critical flow Venturi nozzle with toroidal throat.

### 8.2.2 Sampling methods

The following methods have been applied to evaluate the uncertainty of the mass flow rate:

- Monte Carlo sampling (MC);
- Law of Propagation of Uncertainty (LPU);
- Stratified Sampling (SS);
- Latin Hypercube Sampling (LHS);

| Variable symbol | Description | Unit | Distribution |
|---|---|---|---|
| $P_{\text{in}}$ | Inlet pressure | Pa | N(100 000, $13^2$) |
| $P_{\text{out}}$ | Outlet pressure | Pa | N(60 000, $13^2$) |
| $T_{\text{in}}$ | Inlet temperature | K | N(295, $0.1^2$) |
| $M$ | Molar mass | kg/ kmol | N(28.85, $0.005^2$) |
| $\mu$ | Dynamic viscosity | $\mu$Pa s | R(18.09, 18.51) |
| $C_v$ | Specific heat at constant volume | kJ/ (kg K) | R(710.5, 725.0) |
| $Pr$ | Prandtl number | - | R(0.687, 0.729) |

Table 9: Definition of distributions of the uncertain input quantities of the model.

- Polynomial Chaos (PC).

The second method is the standard GUM approach [1] using numerically calculated sensitivity coefficients and the Law of Propagation of Uncertainty (LPU). It may yield good results when the uncertainties in the parameters are small relative to the non-linearity of the model. LPU was implemented by evaluating the model once using the best estimates of all of the parameters, and 7 times at the best estimates of the parameters except for one parameter at a time, which was evaluated at its mean plus twice its standard deviation. This amounts to a total of 8 model evaluations. Note that the calculated linear sensitivity coefficients could be used to fix some uncertain input parameters to their mean value, but this has not been done in order to be sure not to miss possibly present higher order interaction effects (which appear to be absent in this problem, as revealed by the reference result).

The Stratified Sampling (SS) method was used with $2^7 = 128$ samples, with each input quantity being split into two regions of equal probability. This method was tested 3 times. Latin Hypercube sampling (LHS) was assessed with 8 samples in order to compare it with LPU-sampling, and with 128 samples in order to compare it with SS-sampling. For both sample sizes the method was repeated 3 times. Polynomial Chaos (PC) was performed on a full 7-dimensional tensor product grid with 2 sample points per dimension, resulting in $2^7 = 128$ input samples to be evaluated by the model. As the PC-method is deterministic for a given sample size, this method was only used once. The Monte Carlo procedure was applied using sample sizes of 8 and 128 in order to compare MC-sampling with the methods LPU, LHS, SS and PC. Again, the method was repeated 3 times for each sample size. Reference results were established by means of the Monte Carlo method using approximately 2000 samples.

### 8.2.3 Results

The model has been evaluated with the samples described in the last section. Each sampling method has been run 3 times in order to assess the variability of each method. For reasons of computation time no additional runs were performed. The final objective is to assess the uncertainty of the output quantity, i.e. the mass flow rate $q$ through the nozzle in this particular case. A measure for the uncertainty of the estimate is the standard deviation of the model evaluation results, called the 'standard uncertainty' of $q$ and denoted by $u(q)$. The mean of the results is of less interest, as one would usually only evaluate the model at the best estimates of the input

quantities and not repeat the model evaluation multiple times to determine the mean of the output distribution. Nevertheless the mean of the calculated flow rates by the model is presented as well. This mean value is the best estimate of the flow rate according to GUM supplement 1 [2].

The reference mean mass flow rate and the associated standard deviation can be found in table 10. The number in parentheses denotes the uncertainty of the last digit of the presented value. This value has been calculated by calculating a pooled standard deviation based on a subdivision in 6 sets of approximately equal size of the 1998 samples, and dividing it by the square root of the total number of samples, 1998. The reference result is denoted by 'REF-MC-1998'.

| Sample ID | $q_{\mathrm{ref}}$ [mg/s] | $u_{\mathrm{ref}}(q_{\mathrm{ref}})$ [mg/s] |
|---|---|---|
| REF-MC-1998 | 43720 (1) | 27 (5) |

Table 10: Reference results for the calculated mass flow rate and its standard uncertainty with the uncertainty of the values in parentheses. The values are based on 1998 Monte Carlo simulations.

The results for the calculated mass flow rate and the standard uncertainty for several sampling methods and several sample sizes can be found in table 11. The results are presented as differences from the reference values of table 10. In parentheses the uncertainty of each value is indicated. This uncertainty only takes into account the variability of the sampling method and not the uncertainty of the reference results. The sampling method in tables 10 and 11 is indicated by a letter combination abbreviating the sampling method, followed by a dash and the number of samples used in each of the 3 individual repetitions. The mean results and their uncertainties in parentheses are shown. The results for the individual repetitions of the sampling methods are not shown.

| Sampling method | $q - q_{\mathrm{ref}}$ [mg/s] | $u(q) - u_{\mathrm{ref}}(q_{\mathrm{ref}})$ [mg/s] |
|---|---|---|
| MC-8 | 2 (5) | 2 (8) |
| LPU-8 | 0 (0) | 0 (0) |
| LHS-8 | 0 (2) | 0 (6) |
| MC-128 | 0 (2) | 1 (1) |
| SS-128 | -1 (2) | 0 (0) |
| LHS-128 | 0 (0) | 0 (1) |
| PC-128 | 0 (0) | -1 (0) |

Table 11: Results for the calculated mass flow rate and the standard uncertainty for several sampling methods and several sampling sizes, presented as differences from the reference values of table 10. In parentheses the uncertainty of each value is indicated. This uncertainty only takes into account the variability of the sampling method and not the uncertainty of the reference results.

### 8.2.4 Discussion of the results

In table 10 it can be seen that practically all results are consistent with the reference values when the variability of the sampling method is taking into account, even when the uncertainty of the reference values is not taken into account. Only the deterministic PC result for the standard uncertainty is slightly different.

For a sample size of 8, the Monte Carlo sampling method shows more variability in both the values for the flow rate amd the standard uncertainty than Latin Hypercube sampling, the other stochastic method. The deterministic method LPU is superior to both methods in this case. The problem with this method, however, is that it is not clear beforehand if it will produce correct results or not because the assumption of model linearity is not known to be valid initially, whereas MC and LHS are guaranteed to yield correct results as long as the sample size is sufficiently large. The results of MC and LHS can be checked for convergence whereas this is not possible for the results of LPU. For a sample size of 128 model evaluations all methods perform well.

Overall, the best method for this particular problem is LPU, as it happens that the model function can be linearized around the input values given the sizes of the uncertainties of the input values. If it is not known beforehand if the model can be linearized, as was the case for this potentially very non-linear flow problem, LPU cannot be used as the only method. In that case LHS-8 seems to be a suitable method, though the variability of the calculated uncertainty may be higher than desirable. In that case LHS with more samples can be used. It does not seem to be necessary to use as many as 128 samples to determine the mean and standard uncertainty of the model output value.

A combination of methods is also possible. First one could investigate with LPU the sensitivity of the flow rate to each of the input parameters, taking into account the size of the uncertainty associated with the input parameters. In a second step one could use a different number of input values for each input parameter, if either the Stratified Sampling or the Polynomial Chaos method is used. For input parameters to which the output is sensitive, a greater variety of input values would be allocated than for less sensitive input parameters.

The statistical significance of the presented results was not assessed. As all methods have been repeated only 3 times, the difference in the results may not be significant to the 5 % level. However the presented results and conclusions may nevertheless provide at least qualitative information.

### 8.2.5 Complete distribution functions

Often one is not only interested in the standard uncertainty of the result, but also in a coverage interval with a specified coverage probability or in the mode or median of the result distribution. All this information is encoded in the cumulative distribution function (CDF). In figure 24 the reference CDF based on the REF-MC-1998 samples, the CDF obtained using LPU-8 in case of variation of the input parameters with 1 standard deviation, the CDF using PC-128 and the maximum and minimum observed values for the CDFs at a given cumulative relative frequency in each of the 3 repetitions the simulations for the MC-8, LHS-8, MC-128, LHS-128 and SS-128 sampling methods are shown. For the CDF obtained by using the Law of Propagation of Uncertainty the rectangular distribution was used, as the dominating input parameter $C_v$ had this probability distribution. Polynomial chaos yields a polynomial expansion for the probability distribution of the output quantity in terms of the input distributions. Instead of explicitly evaluating the sum of polynomials of the definition, the (same) polynomial expansion can easily be evaluated by calculating a sum of 128 terms each consisting of a product of 7 linear Lagrange polynomials and a specific calculated flow rate as evaluated by the CFD model. This approximate CDF has been numerically evaluated by means of a Monte Carlo method using $10^5$ samples. This large number of samples poses no problem here, as the approximate CDF is not computationally expensive. Furthermore, note that the CDFs for MC-8 and LHS-8 start at an ordinate value of 0.125, corresponding to a cumulative relative frequency of 1/8. For the purpose of better visualization of the values on the x-axis all flow rates have been reduced by 0.0437 kg/s (an approximation to the reference mean value in table 10).

It can be seen that the reference CDF closely resembles the rectangular distribution found by means of LPU-8. The CDFs only start to deviate from the rectangular distribution at probabilities smaller than approximately 0.05 or bigger than 0.95. The limiting CDFs found by applying the sampling methods MC-128, LHS-128 and SS-128 are much closer to the reference CDF than the CDFs for MC-8 and LHS-8. In table 12 the maximum

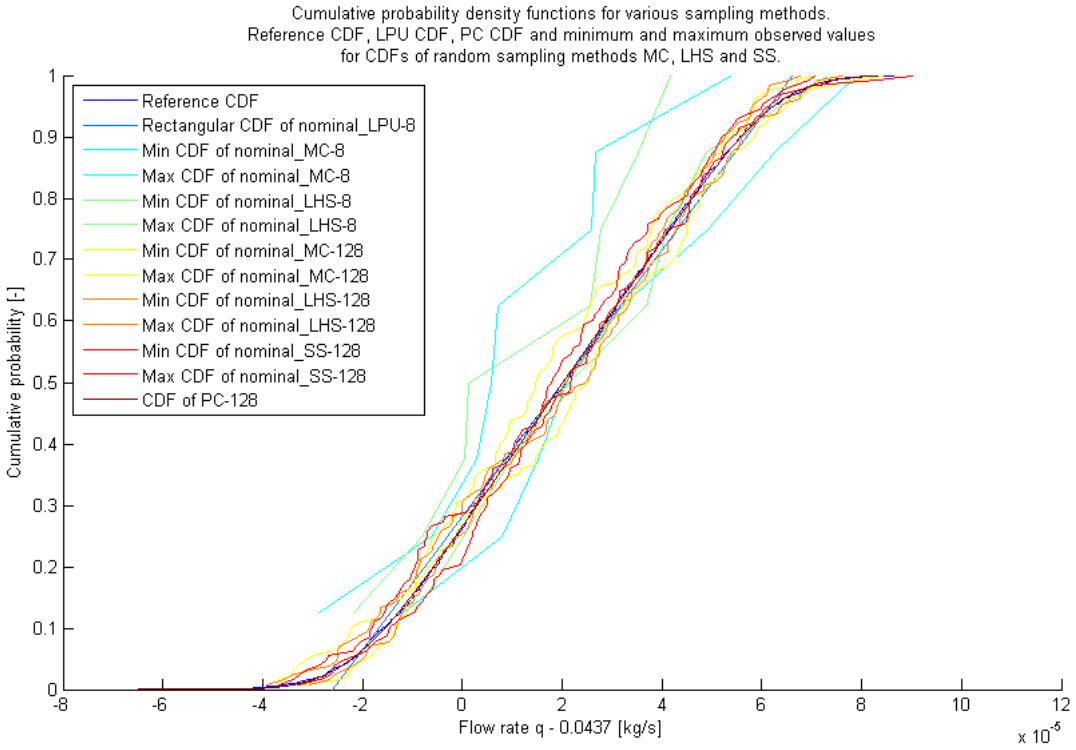



Figure 24: Cumulative probability distributions for the flow rate for various sampling methods. The reference CDF is shown, the CDF for LPU and the minimum and maximum observed values for MC, LHS and SS random sampling methods. The CDFs for MC-8 and LHS-8 start at an ordinate value of 1/8 = 0.125. All flow rates have been reduced by 0.0437 kg/s for better visualization of the values on the x-axis.

of the absolute differences of the constructed CDFs based on the sampling results with the reference CDF is shown. The difference between the CDFs was evaluated at all values for the flow rate found in the reference sample set REF-MC-1998. Only LPU-8, LHS-128 and PC-128 have a maximum deviation smaller than 5 % in the entire domain. The maximum deviations for MC-8 and LHS-8 are very large. In this case study PC performs better than LHS, LHS performs better than SS and SS performs better than MC. Note that as the reference distribution is based on 1998 samples, the agreement is not expected to be better than 0.005 at best. It is even possible that the PC-128 CDF is more accurate than the REF-MC-1998 CDF.

| Sampling method | maximum absolute difference with CDF of the minimum values | maximum absolute difference with CDF of the maximum values |
|---|---|---|
| MC-8 | 0.308 | 0.131 |
| LPU-8 | 0.033 | 0.033 |
| LHS-8 | 0.250 | 0.156 |
| MC-128 | 0.088 | 0.076 |
| SS-128 | 0.077 | 0.056 |
| LHS-128 | 0.047 | 0.048 |
| PC-128 | 0.013 | 0.013 |

Table 12: Maximal absolute difference of the CDF constructed from the samples with the reference CDF for various sampling strategies. As LPU-8 and PC-128 are deterministic methods the values in both columns are equal.

### 8.2.6 Summary

In this case study a computationally expensive CFD model for calculating the flow rate through a sonic Venturi nozzle has been studied. Different sampling methods have been tested in order to evaluate the uncertainty of the output quantity due to uncertainties in the input parameters. It appears that the problem can be linearized around the best estimates as the uncertainties are sufficiently small. Physically speaking this means that the nozzle is operated in a stable regime with stable sonic flow. For this reason the "Law of Propagation of Uncertainty" approach (LPU-8) performed very well for calculating the mean value of the flow rate, its standard uncertainty and its associated cumulative probability density function. For calculating the mean value and the standard uncertainty all sampling methods performed reasonably well and only small sample sizes were needed. In the case when the full probability distribution is required, MC-8 and LHS-8 do not perform well. The best methods are first PC-128, then LPU-8, then LHS-128, then SS-128 and then MC-128.

It is not generally known beforehand if a model can be linearized, given that input quantities are independent, and thus if LPU can be applied. The results of the LPU method give no information about this criterion. The method of choice is therefore not LPU-8. In the case when only the mean value and the standard uncertainty is needed, one could repeat LHS-8 at least three times and assess from the variability in the results if they have converged. If not, one can apply additionally a higher order LHS-approach, e.g. LHS-50 or LHS-100. An alternative approach is to assess the sensitivity of the output quantity to the various input quantities first, and then apply PC with different polynomial orders for each dimension. If a coverage interval or the full CDF is required, then PC is clearly the method of choice. A disadvantage of PC is that it is more complicated to

implement than LHS.

This example also shows that if a model can be linearized, then LPU-8 is a very powerful tool needing only a few model evaluations and outperforming MC-128, SS-128 and LHS-128.

# 9 Case Study 3: Thermophysical properties

The problem discussed in this case study concerns the estimation of the thermophysical properties, typically the thermal conductivity, of a material by matching the temperatures predicted by a model of the laser flash thermal diffusivity experiment to voltages generated by a temperature sensor. The laser flash thermal diffusivity experiment [55] heats one side of a small cylindrical material sample with a short burst of energy supplied by a laser, and measures the resulting temperature rise of the opposite face over time. In the simplest cases and for a uniform material, an analytical solution to the governing equation and boundary conditions can be used to estimate the thermal diffusivity of a material [54, 55, 53]. Thermal diffusivity is defined as

$$\alpha = \frac{\lambda}{\rho c_p} \tag{71}$$

where $\lambda$ is the thermal conductivity of the material, $\rho$ is the density of the material, and $c_p$ is the specific heat capacity of the material.

For layered materials such as thermal barrier coatings, this analytical approach cannot be used and a numerical approach is needed. The work reported here uses a finite volume model to solve the heat equation

$$\rho c_p \frac{\partial T}{\partial t} = \nabla.(\lambda \nabla T) + Q\delta(z)I(t < t_{\text{flash}}), \tag{72}$$

where $Q$ is the energy flux into the sample due to the laser and $I(t < t_{\text{flash}})$ is a function that takes the value 1 if the condition in the brackets is true and 0 otherwise, with radiation-like boundary conditions of the form

$$\mathbf{n}.\nabla T = \sigma\epsilon(T_a^4 - T^4), \tag{73}$$

where $T_a$ is the ambient temperature and $\epsilon$ is an effective emissivity parameter not constrained to lie between 0 and 1. The inverse model uses an optimisation algorithm to estimate the unknown model parameters $\lambda$ and $Q$ by matching the model predictions of temeprature to the measured voltages. This matching process is an inverse problem and so provides a link to related work [30].

An initial assessment of potential methods of interest was carried out using a simulation of a uniform material (ArmCo Iron). The best performing methods were then applied to a layered material, with the thermal conductivity of one of the layers being regarded as unknown. The results of both sets of tests are reported here.

For the initial model, the model output quantities are the thermal conductivity of the sample, $\lambda$, and the laser power $Q$. The inputs include the measured values used as the target data in the optimisation routine, the sample radius $R$ and thickness $h_2$, the sample material density $\rho$ and specific heat capacity $c_p$, the duration of the laser pulse $t_{\text{flash}}$, the equivalent emissivity $\epsilon_1$ and the temperature conversion coefficient (used to convert from voltage to temperature) $\Delta T$. The assumed distributions for the input quantities are listed in table 13.

The layered sample consisted of a layer of copper (denoted by a subscript 2) and a layer of braze (denoted by a subscript 1). The thermal conductivity of the copper was regarded as unknown. The other parameters of the model and their associated distributions or fixed values, are listed in table 14. The various differences between tables 13 and 14 will be explained as the case study progresses. The effective emissivity $\epsilon$ was treated as unknown because it became clear that too little was known about the parameter for it to be assigned a meaningful distribution.

## 9.1 Measured values and reference results

The two sets of measured values used to match to model results are shown in figure 25. It is clear from these plots that the measured values are noisy. If the data points, of which there are more than 1 000, are treated

| Symbol | Quantity, unit | Distribution |
|---|---|---|
| Density, $\mathrm{kg\,m^{-3}}$ | $\rho$ | $\mathrm{N}(7870, 78.7^2)$ |
| Specific heat capacity, $\mathrm{J\,kg^{-1}\,K^{-1}}$ | $c_p$ | $\mathrm{N}(465, 4.65^2)$ |
| Effective emissivity, dimensionless | $\epsilon$ | $\mathrm{N}(226.6, 0.37^2)$ |
| Laser pulse duration, ms | $t_{\text{flash}}$ | $\mathrm{R}(0.7, 0.9)$ |
| Sample layer thickness, mm | $h_2$ | $\mathrm{R}(1.91253392, 1.91346608)$ |
| Sample radius, mm | $R$ | $\mathrm{R}(4.9992, 5.0008)$ |
| Temperature normalisation, K | $\Delta T$ | $\mathrm{T}(1.5, 6.5)$ |

Table 13: Definition of distributions of the uncertain input quantities of the model.

| Quantity, unit | Symbol | Distribution |
|---|---|---|
| Braze density, $\mathrm{kg\,m^{-3}}$ | $\rho_1$ | N(8 600, $(86)^2$) |
| Braze specific heat capacity, $\mathrm{J\,kg^{-1}\,K^{-1}}$ | $c_{p1}$ | N(377, $(3.77)^2$) |
| Braze thermal conductivity, $\mathrm{W\,m^{-1}\,K^{-1}}$ | $\lambda_1$ | N(112, $(2)^2$) |
| Copper density, $\mathrm{kg\,m^{-3}}$ | $\rho_2$ | N(8 930, $(89.3)^2$) |
| Copper specific heat capacity $\mathrm{J\,kg^{-1}\,K^{-1}}$ | $c_{p2}$ | N(397, $(3.97)^2$) |
| Furnace temperature, K | $T_a$ | R[373, 383] |
| Temperature scaling factor, K | $\Delta T$ | R[1.5, 6.5] |
| Laser flash duration, ms | $t_{\text{flash}}$ | R[0.5, 0.7] |
| Sample radius, mm | $R$ | 6.2 mm |
| Braze layer thickness, mm | $h_1$ | 1.018 5 mm |
| Copper layer thickness, mm | $h_2$ | 1.018 5 mm |
| Copper thermal conductivity, $\mathrm{W\,m^{-1}\,K^{-1}}$ | $\lambda_2$ | Unknown |
| Laser heat flux, $\mathrm{MW\,m^{-2}}$ | $Q$ | Unknown |
| Effective emissivity, dimensionless | $\epsilon$ | Unknown |

Table 14: Probability distributions associated with the input quantities for the copper and braze layered sample model runs.

as a set of uncertain inputs then most surrogate methods cannot be used. A curve was fitted to the data in figure 25(a), the noise on the data was characterised as being $\mathrm{N}(0, (4\mathrm{mV})^2)$, and two Monte Carlo runs were carried out. One run treated the fitted curve as the target data and evaluated $\lambda$ and $Q$ given values of the uncertain inputs listed in table 13, and one run treated the noise on each of the measured points as independent identically distributed random input quantities and evaluated $\lambda$ and $Q$. The sample sizes of these runs were dictated by the time available. The first run used 5 048 samples and the second used 2 021 samples. The CDFs for $Q$ and $\lambda$ using idealised and noisy data differed very little, so it was assumed that the noise associated with the data could be neglected.

The results of the first Monte Carlo run were extended to include almost 9 000 samples, and were used as reference data to which the results of the other methods could then be compared. A similar set of reference results were generated for the layered material using 2 632 samples (again, the sample size was determined by the time available). The results of these reference runs, which should not be compared directly because they are for different materials, are summarised in table 15. The correlation coefficients associated with the various pairs of output quantities were sufficiently small that the output quantities can be regarded as uncorrelated.

| | Uniform material | | Layered material | | |
|---|---|---|---|---|---|
| | $\lambda$ (W m$^{-1}$ K$^{-1}$) | $Q$ (MW m$^{-2}$) | $\lambda_2$ (W m$^{-1}$ K$^{-1}$) | $Q$ (MW m$^{-2}$) | $\epsilon$ |
| Mean | 66.0 | 60.5 | 277.7 | 45.1 | 90 |
| Std. dev. | 0.84 | 15.9 | 13.4 | 4.2 | 18 |
| Median | 66.0 | 60.1 | 276.9 | 44.6 | 190 |
| 2.5% | 64.4 | 31.4 | 253.8 | 38.7 | 144 |
| 97.5% | 67.7 | 92.2 | 306.1 | 53.2 | 235 |

Table 15: Summary of the reference results for the two models.

## 9.2 Input screening

Input screening was carried out for the problem with the uniform material. Full factorial and fractional factorial designs were considered, and the results were in agreement. The full set of direct effects and the top seven influence factors for each output quantity are shown in table 16. Note that although the interactions are denoted by a $\times$ sign, their interaction may not be multiplicative.

The values in table 16 suggests that variations in $R$ and $h_2$ do not significantly affect either output quantity, so only their mean values and not their distributions are required by the model. These results have motivated the use of fixed values for the geometric parameters $R$, $h_1$, and $h_2$ in table 14. It is reassuring to note that the effects of $\rho_1$ and $c_p$ are equal, as would be expected since they only ever appear as the product $\rho c_p$ in equation (72) and they both have the same relative uncertainty of 1 %.

## 9.3 Sampling methods

Three sampling methods were compared using the model of the uniform material: random sampling, stratified sampling, and Latin hypercube sampling. Different sample sizes were used for each method, the sample sizes used being:

- random sampling: 10, 25, 50, 75, 100, 150, 200, 250.

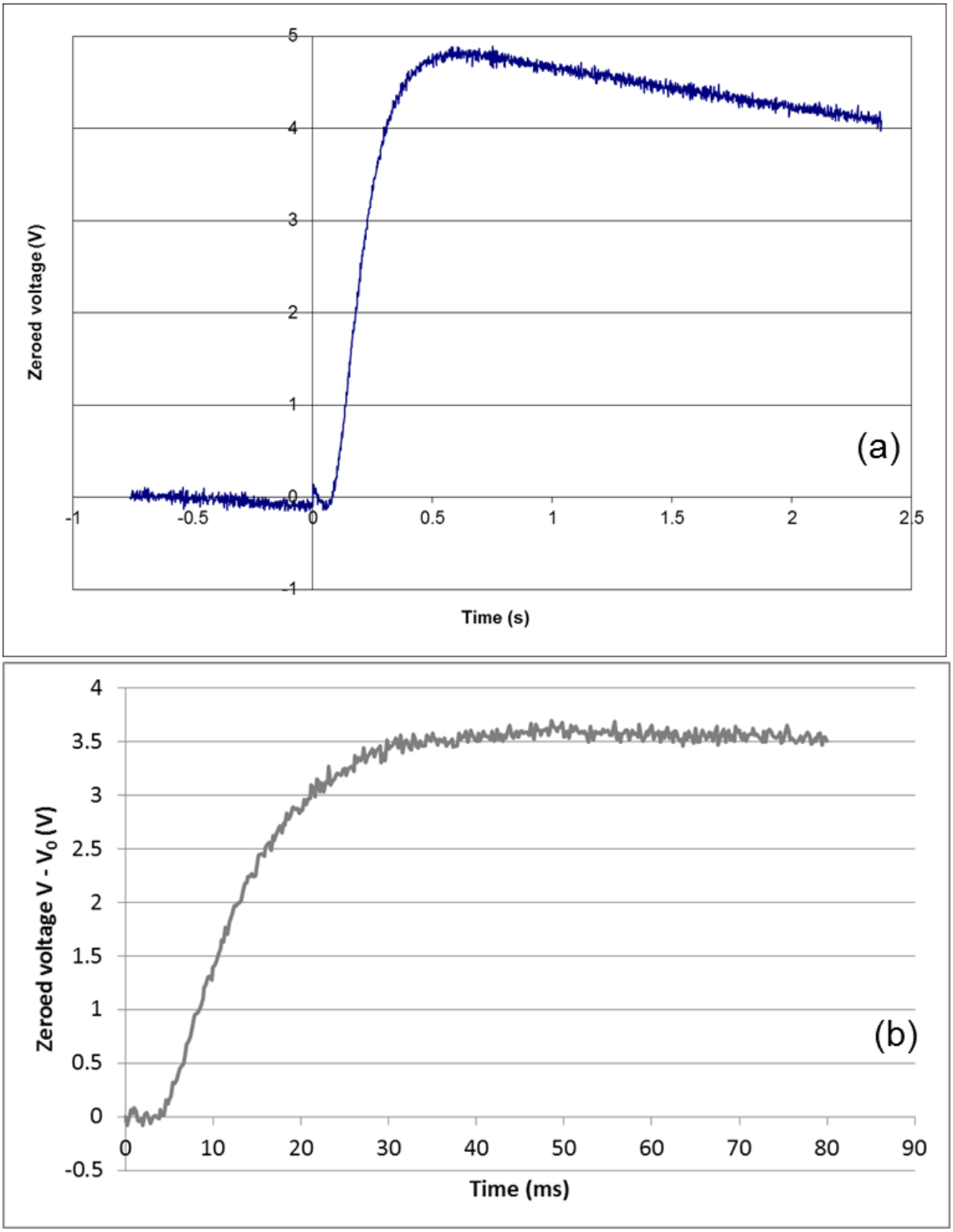



Figure 25: Measured values used in the work. Plot (a) was measured using a uniform Armco iron sample, and plot (b) was measured using a layered sample consisting of a layer of copper and a layer of braze.

| Input | Effect on $\lambda$ (W/(m K)) | Interaction | Effect on $\lambda$ (W/(m K)) |
|---|---|---|---|
| $\rho$ | **2.26** | $\Delta T \times \epsilon$ | **0.15** |
| $c_p$ | **2.26** | $h_2 \times c_{p1}$ | 0.14 |
| $\Delta T$ | **0.25** | $h_2 \times \rho$ | 0.14 |
| $\epsilon$ | **0.19** | $h_2 \times \epsilon$ | 0.13 |
| $t_{\text{flash}}$ | **0.16** | $\rho \times t_{\text{flash}}$ | 0.13 |
| $R$ | 0.032 | $t_{\text{flash}} \times c_p$ | 0.13 |
| $h_2$ | 0.0025 | $\rho \times \Delta T$ | 0.12 |

| Input | Effect on $Q$ (MW/m$^2$) | Interaction | Effect on $Q$ (MW/m$^2$) |
|---|---|---|---|
| $\Delta T$ | **76.2** | $t_{\text{flash}} \times \Delta T$ | **9.52** |
| $t_{\text{flash}}$ | **15.2** | $\Delta T \times c_p$ | 1.47 |
| $\rho$ | 2.34 | $\rho \times \Delta T$ | 1.47 |
| $c_p$ | 2.34 | $\rho \times t_{\text{flash}}$ | 0.31 |
| $h_2$ | 0.0077 | $t_{\text{flash}} \times c_p$ | 0.31 |
| $\epsilon$ | 0.0038 | $\rho \times c_p$ | 0.059 |
| $R$ | 0.0004 | $h_2 \times \rho$ | 0.014 |

Table 16: Absolute change in output quantity for each contributing input quantity, and the top 7 interaction effects. Effects are significant if greater than the standard error, 0.15 W/(m K) for $\lambda$ and 3.48 MW/m$^2$ for $Q$. The significant effects are marked in bold.

- stratified sampling: 32, 48, 72, 108, 243.
- Latin hypercube sampling: 10, 25, 50, 75, 100, 150, 200.

The stratified sampling sample sizes were generated by splitting the range of each of the input quantities into either 2 or 3 regions of equal probability and using every combination of these regions to define stratification of the complete input space into regions of equal probability. The choice of subdivisions was partly motivated by the results of the input screening, which suggested $R$ and $h_2$ were not important. The full list of intervals for each sample size is shown in table 17. Whilst two separate stratifications were used for sample size 48, no significant differences were found between the results for the two sets of samples.

| | Number of intervals for each input quantity | | | | | | |
|---|---|---|---|---|---|---|---|
| Sample size | $R$ | $h_2$ | $\rho$ | $t_{\text{flash}}$ | $\Delta T$ | $c_p$ | $\epsilon$ |
| 32 | 1 | 1 | 2 | 2 | 2 | 2 | 2 |
| 48 (1) | 1 | 1 | 3 | 2 | 2 | 2 | 2 |
| 48 (2) | 1 | 1 | 2 | 2 | 3 | 2 | 2 |
| 72 | 1 | 1 | 3 | 2 | 2 | 3 | 2 |
| 108 | 1 | 1 | 3 | 2 | 3 | 3 | 2 |
| 243 | 1 | 1 | 3 | 3 | 3 | 3 | 3 |

Table 17: Number of intervals associated with each input quantity for a given sample size for the stratified sampling method.

Ten samples of each sample size were generated using each method so that the repeatability of the methods could be assessed. The results of each sample were used to calculate an estimate of the mean and the standard deviation of each output quantity. The means and standard deviations of the ten samples generated using the same method and sample size were then used to estimate an inter-sample mean and standard deviation of the sample means and sample standard deviations. The inter-sample standard deviation of the sample means gives a good insight into the likely sample-to-sample variation of estimates of the mean, and hence a good indication of the repeatability of the method.

## 9.4 Surrogate models

Three surrogate modelling methods were compared: nearest neighbour interpolation, response surface methodology, and Gaussian process emulation. Nearest neighbour interpolation was implemented in both scaled and unscaled form (the scaled form transformed all variables to be zero mean and unit standard deviation). The response surface methodology was used to fit a quadratic response surface to the full set of training points. A Gaussian process emulator was fitted to the training points using a covariance function of the form, as specified in equations (55) and (56)

$$\text{Cov}(X, X') = \sigma^2 \prod_{j=1}^{N} \exp\left[ -\left| \frac{x_j - x'_j}{\ell_j} \right|^2 \right]. \tag{74}$$

The hyperparameters $\sigma^2$ and $\ell_j$ required for the Gaussian process model were determined using an optimisation routine to minimise the negative marginal log-likelihood function for the hyperparameters given the training

points. The optimisation process was run repeatedly to ensure that the best set of hyperparameters was identified.

Three different methods were used for the training point selection: random sampling, Latin hypercube sampling, and Hammersley sampling. Hammersley sampling [31] is a deterministic method that ensures the training points are spread throughout the input space. The method will not be described in full here, but is based on the use of Halton sequences with increasing prime bases. Each method was used to generate a single set of 37 training points. The size of the training point set was chosen as being one more than the number of points required to uniquely define a quadratic response surface passing through all the data points.

A surrogate model of each type was fitted to each set of training points. Models for each of the two output quantities were fitted independently. Once the surrogate models had been fitted, a random sample of the input quantities of size 10 000 was generated, the surrogate model was evaluated at those 10 000 points, and the results were used to evaluate the uncertainties associated with the output quantities.

## 9.5 Initial results

### 9.5.1 Sampling methods

A visual summary of the results of the sampling methods is shown in figures 26 and 27. The plots in figure 26 show the variation of the estimates of the mean and standard deviation of $\lambda$ with sample size, and figure 27 shows the equivalent information for $Q$. The upper plot shows the mean of the sample means (left hand plot) or sample deviations (right hand plot) as a line joining different sample sizes and the inter-sample standard deviation of the quantity as error bars, and the lower plot shows the inter-sample standard deviation of the sample mean or sample standard deviation separately for clarity.

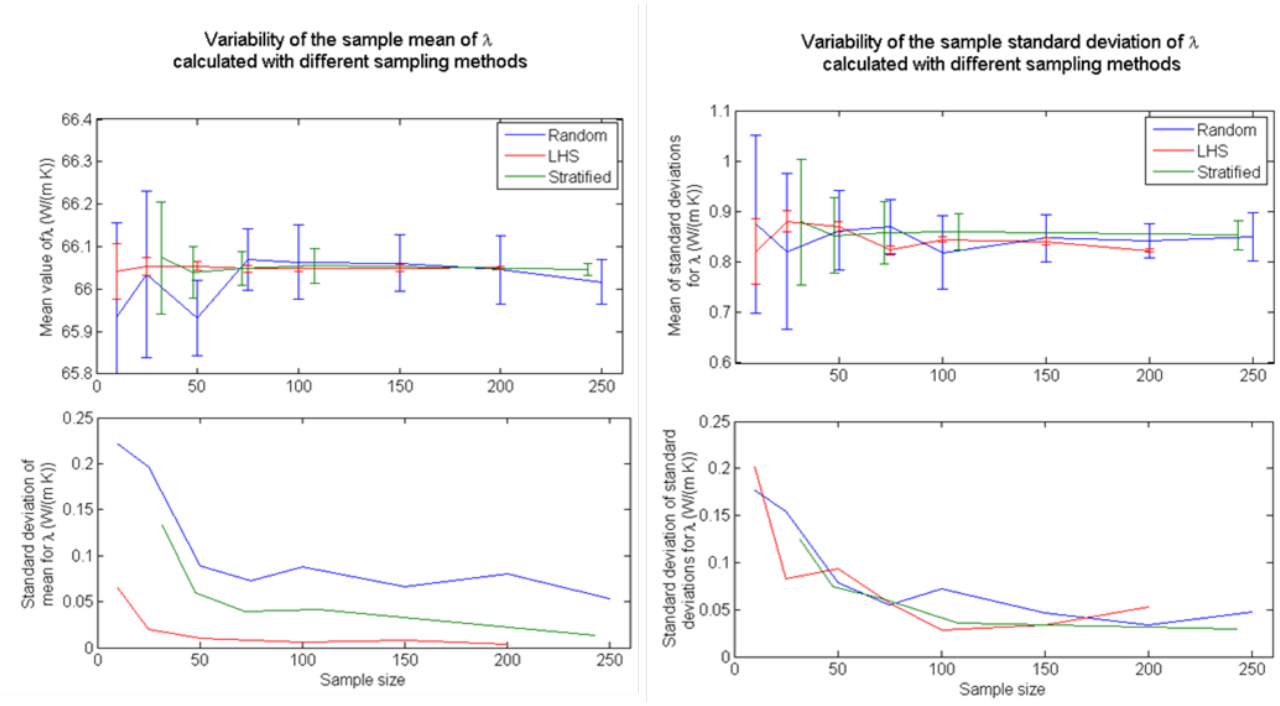



Figure 26: Variation of sample mean and standard deviation and sample-to-sample variability with sample size for $\lambda$.

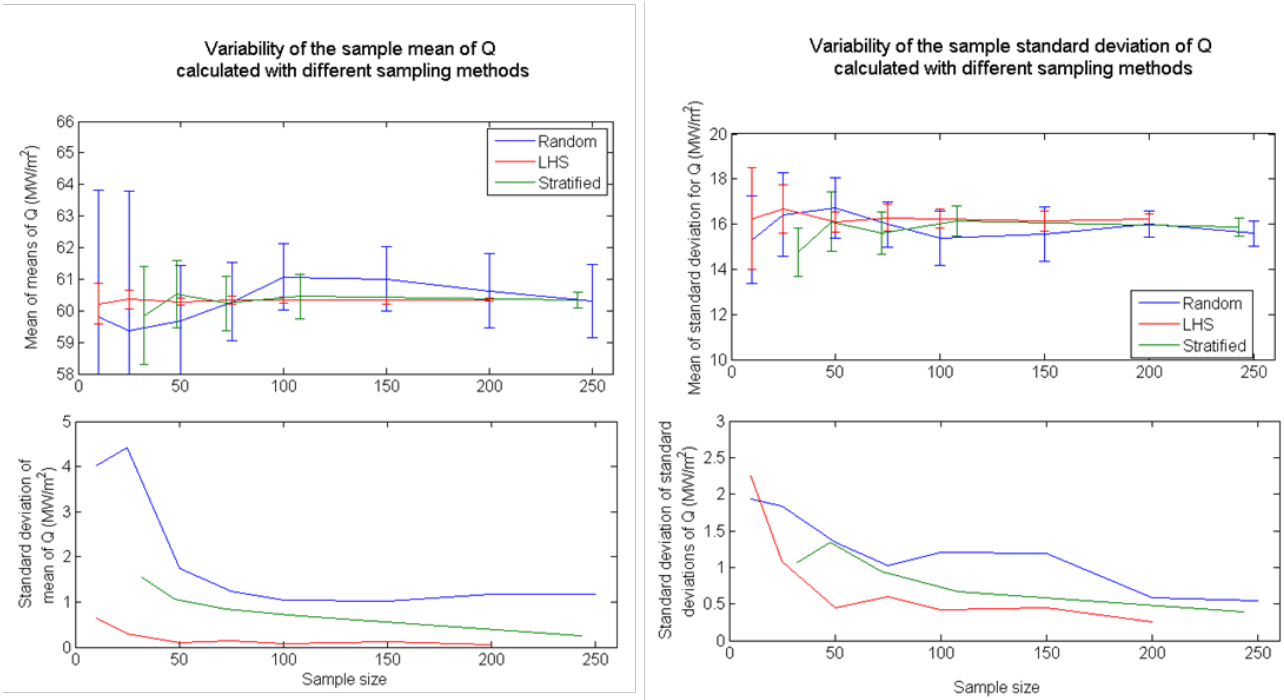



Figure 27: Variation of sample mean and standard deviation and sample-to-sample variability with sample size for $Q$.

These results suggest that the sample to sample variability of the estimate of the mean and standard deviation generated by Latin hypercube sampling is significantly smaller than that generated by random sampling and stratified sampling for all sample sizes. For the problem studied here, Latin hypercube sampling gives good repeatability and accuracy for surprisingly small sample sizes. This statement applies to estimates associated with $\lambda$ and with $Q$. Estimates obtained using stratified sampling have a lower sample-to-sample standard deviation than those obtained using random sampling.

### 9.5.2 Surrogate models

A comparison of the results of the different training point sets for each method suggested that for the problem studied here, the choice of training points made no difference to the results. One possible reason for this lack of sensitivity was highlighted by the input screening analysis. The input screening suggested that $\lambda$ is strongly affected by the values of $\rho$ and $c_p$ and is less sensitive to the other input quantities, and that $Q$ is strongly affected by the values of $\Delta T$ and $t_{\text{flash}}$ and is almost independent of the other input quantities. Hence the 37 points used to train the models are effectively spanning a two-dimensional space rather than a seven-dimensional space (the initial model has seven input quantities), and all training point selection methods can fill the space reasonably well.

A comparison between the results of the different surrogate models using the Latin hypercube training points is shown in table 18. The results in this table are expressed as absolute differences between the values calculated using the surrogate models and the reference values presented in table 15, with a negative value meaning that the reference result is larger.

The results suggest that all four models give good estimates of the mean, standard deviation and CDF of $\lambda$, and that the estimates of the mean, standard deviation and CDF of $Q$ are generally quite good, although all the

| | Nearest Neighbour | | Scaled Nearest Neighbour | | Quadratic Response | | GP emulator | |
|---|---|---|---|---|---|---|---|---|
| | $\lambda$ (W/m K) | $Q$ (MW/m$^2$) | $\lambda$ (W/m K) | $Q$ (MW/m$^2$) | $\lambda$ (W/m K) | $Q$ (MW/m$^2$)) | $\lambda$ (W/m K) | $Q$ (MW/m$^2$) |
| Mean | 0.0 | −2.0 | 0.1 | −0.3 | 0.2 | −2.5 | 0.0 | 0.2 |
| Std. dev. | 0.06 | −4.0 | 0.02 | −5.3 | 0.08 | −2.3 | 0.07 | 0.1 |
| Correlation | 0.18 | | −0.06 | | 0.03 | | 0.03 | |
| Median | 0.0 | −0.4 | 0.1 | −0.4 | 0.2 | −0.5 | 0.1 | 0.0 |
| 2.5% | −0.6 | 3.0 | 0.2 | 9.5 | 0.0 | −1.0 | −0.5 | −0.3 |
| 97.5% | 0.1 | −13.9 | 0.3 | −13.9 | 0.4 | −11.4 | −0.2 | −0.5 |

Table 18: Comparison of the results obtained using the different surrogate modelling methods on the Latin Hypercube sampling training points, expressed as differences between the reference results in table 15 and values calculated using each method.

methods apart from the GP emulator significantly underestimate the standard deviation of $Q$.

The good performance of the nearest neighbour methods is surprising. As with the choice of training points, this result may be affected by the dominance of two of the input quantities for each output quantity and the resulting space-filling nature of the training points.

Checks on goodness of fit using the "leave one out" method described in section 3 were carried out. The results will not be reported in full here, but they suggested that nearest neighbour interpolation was very sensitive to the removal of a training point (as would be expected), that Gaussian process emulation was not sensitive to the removal of a point, and that the quadratic response surface was sensitive for some training points sets but not others. It is likely that the removal of isolated points has a strong effect on the surrogate models and that this has a disproportionate effect for the quadratic response surface because the number of training points in the full set was only one more than the minimum number required for unique definition of a quadratic response surface.

## 9.6 Final results

The results of the initial model on the uniform material suggested that Latin hypercube sampling was the best choice of sampling method and that Gaussian process modelling was the best choice of surrogate model for this problem. These two methods were applied to a model with a layered material.

Three sample sizes (10, 50, and 100) were chosen, and ten Latin hypercube samples of each size were generated. These samples were processed to obtain sample means and standard deviations. Random samples of sizes 10 and 100 were also generated and evaluated for comparison.

A Gaussian process emulator was constructed using each Latin hypercube sample as the training points. The correlation function given in equation (74) was used, and the best fit hyperparameters were determined using nonlinear optimisation as before. The Gaussian process emulator was then evaluated at 10 000 randomly chosen points to obtain estimates of the uncertainties associated with the output quantities.

A summary of the results associated with $\lambda_2$, the thermal conductivity of the copper layer, is shown in tables 19 and 20. Results for other output quantities showed similar trends.

| Method | Sample size | Mean of sample means ($\mathrm{W\,m^{-1}\,K^{-1}}$) | Standard deviation of sample means ($\mathrm{W\,m^{-1}\,K^{-1}}$) | Minimum value of sample mean ($\mathrm{W\,m^{-1}\,K^{-1}}$) | Maximum value of sample mean ($\mathrm{W\,m^{-1}\,K^{-1}}$) |
|---|---|---|---|---|---|
| LHS | 10 | 277.27 | 1.49 | 275.29 | 278.84 |
| LHS | 50 | 277.26 | 0.33 | 276.55 | 277.63 |
| LHS | 100 | 277.47 | 0.29 | 276.80 | 277.83 |
| GP | 10 | 277.39 | 1.29 | 275.39 | 279.65 |
| GP | 50 | 276.84 | 0.54 | 276.02 | 277.56 |
| GP | 100 | 277.13 | 1.36 | 274.55 | 278.65 |
| RS | 10 | 277.62 | 3.87 | 271.80 | 282.21 |
| RS | 100 | 277.77 | 1.29 | 276.06 | 279.39 |

Table 19: Statistics associated with the means of the ten samples for each size. Reference mean is 277.7 $\mathrm{W\,m^{-1}\,K^{-1}}$.

| Method | Sample size | Mean of standard deviations ($\mathrm{W\,m^{-1}\,K^{-1}}$) | Standard deviation of standard deviations ($\mathrm{W\,m^{-1}\,K^{-1}}$) | Minimum value of sample standard deviation ($\mathrm{W\,m^{-1}\,K^{-1}}$) | Maximum value of sample standard deviation ($\mathrm{W\,m^{-1}\,K^{-1}}$) |
|---|---|---|---|---|---|
| LHS | 10 | 14.60 | 2.35 | 12.33 | 19.24 |
| LHS | 50 | 13.43 | 0.51 | 12.56 | 14.38 |
| LHS | 100 | 13.21 | 0.54 | 12.21 | 13.85 |
| GP | 10 | 16.33 | 4.38 | 11.47 | 24.64 |
| GP | 50 | 14.37 | 1.05 | 12.86 | 16.75 |
| GP | 100 | 14.51 | 1.32 | 13.11 | 16.87 |
| RS | 10 | 13.42 | 3.13 | 11.32 | 18.48 |
| RS | 100 | 12.78 | 0.75 | 11.35 | 14.08 |

Table 20: Statistics associated with the standard deviations of the ten samples for each size. Reference standard deviation is 13.4 $\mathrm{W\,m^{-1}\,K^{-1}}$.

The first point of interest in these results is that all sample sizes give very good estimates of the mean with both methods. The maximum deviation of the sample mean from the reference mean is about 2 % even for a sample size of 10 using random sampling. The sample standard deviations are less good estimates of the reference standard deviation, as would be expected, but are still reasonably accurate, particularly for the Latin hypercube samples.

The Latin hypercube sample estimates become more accurate and more repeatable (lower sample-to-sample standard deviation) as the sample size increases, but the same is not true of the Gaussian process model results even though the Gaussian process models are based on the Latin hypercube sample points. For the large sample sizes, the Gaussian process model estimates of mean and standard deviation are less accurate and more variable than the corresponding Latin hypercube estimates. This phenomenon is shown for a sample size of 100 in figures 28 for sample means and 29 for sample standard deviations. The dotted line in each graph denotes the reference value, the solid squares are the Gaussian process estimates plotted against sample number, and the empty circles are the Latin hypercube estimates for the same samples. In most cases, the Latin hypercube estimate is closer than the Gaussian process estimate to the reference value.

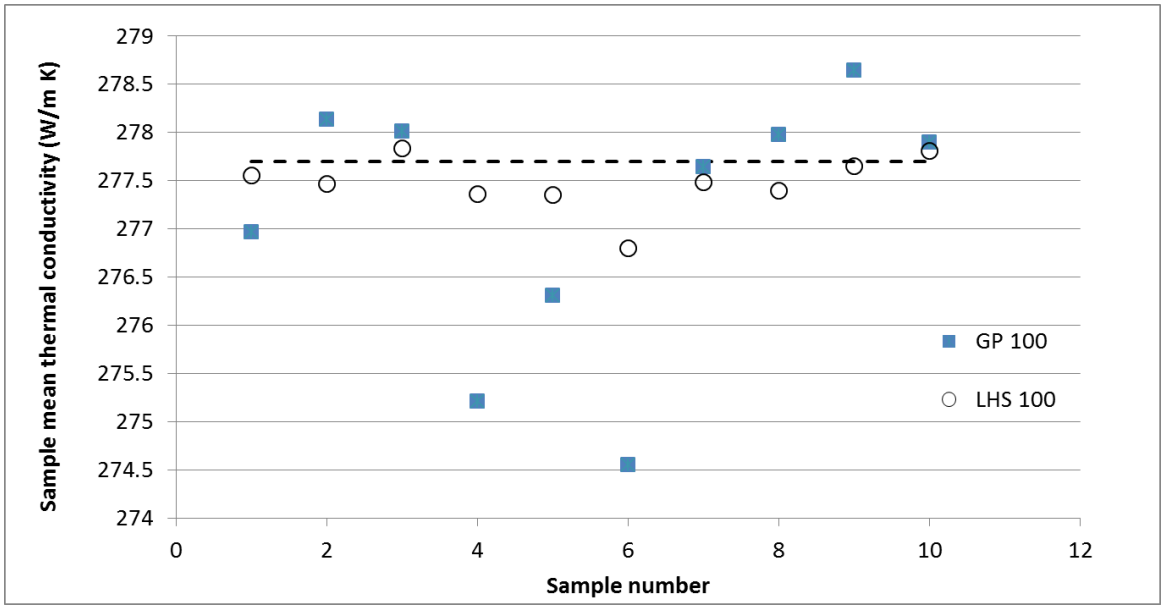



Figure 28: Sample means for a sample size of 100.

These figures suggest that the Gaussian process model adds no value to the Latin hypercube sample on which it is based. It is difficult to explain the behaviour of the Gaussian process models. The models have eight input quantities, which makes useful visualisation of the model surfaces challenging. It was noted above that sometimes the hyperparameter fitting process can become numerically unstable for large numbers of training points, and it may be that the hyperparameter values determined are not suitable.

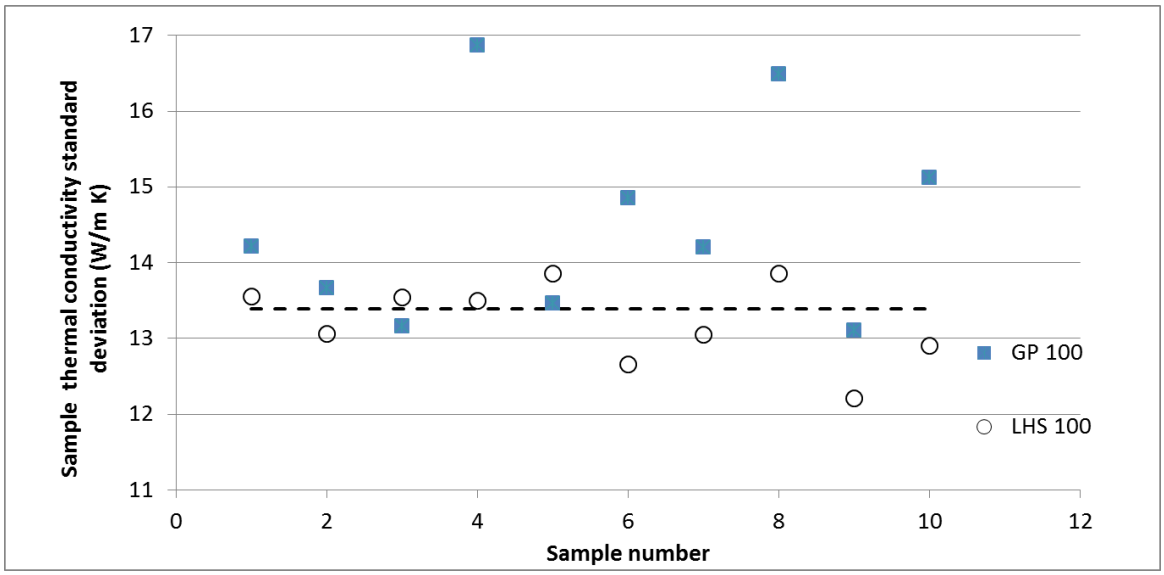



Figure 29: Sample standard deviations for a sample size of 100.

## Acknowledgements

The EMRP is jointly funded by the EMRP participating countries within EURAMET and the European Union.

## References

[1] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP and OIML. *Evaluation of measurement data - Guide to the expression of uncertainty in measurement,* Joint Committee for Guides in Metrology, JCGM 100:2008, 2008.

[2] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP and OIML. *Evaluation of measurement data - Supplement 1 to the Guide to the expression of uncertainty in measurement - Propagation of distributions using a Monte Carlo method,* Joint Committee for Guides in Metrology, JCGM 101:2008, 2008.

[3] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP and OIML. *Evaluation of measurement data Supplement 2 to the Guide to the expression of uncertainty in measurement Extension to any number of output quantities,* Joint Committee for Guides in Metrology, JCGM 102:2011, 2011.

[4] http://www.itl.nist.gov/div898/handbook/pri/section3/pri3341.htm. Retrieved March 2015.

[5] Saltelli, A., Chan, K., and Scott, E. M. (eds). *Sensitivity Analysis, first edition* John Wiley & Sons, ISBN 0-471-99892-3, 2001.

[6] Allard, A., and Fischer, N., *Recommended tools for sensitivity analysis associated to the evaluation of measurement uncertainty* Advanced Mathematical and Computational Tools in Metrology and Testing IX, World Sc. **84**, 1-12, 2012.

[7] Cox, M. G., and Harris, P. M. Software Support for Metrology Best Practice Guide No 6 - uncertainty evaluation. NPL Report DEM-ES 011, 2006.

[8] Sobol', I. M., *Sensitivity estimates for nonlinear mathematical models* Mathematical Modelling and Computational Experiments, 1:407-414, 1993.

[9] Morris, M. D., *Factorial sampling plans for preliminary computational experiments* Technometrics **33** 161-174, 1991.

[10] McKay, M D, Beckman, R J, and Conover, W J, *A comparison of three methods for selecting values of input variables in the analysis of output from a computer code,* Technometrics, 21(2), 1979.

[11] Anderson, E C, *Monte Carlo methods and Importance Sampling,* Lecture notes for Stat 578C, Statistical Genetics, 1999. Available from http://ib.berkeley.edu/labs/slatkin/eriq/classes/guest_lect/mc_lecture_notes.pdf

[12] Owen, A, and Zhou, Y *Safe and effective importance sampling,* Journal of the American Statistical Association, 95, 449, 135-143, 2000.

[13] Press, W H, Teukolsky, S A, Vetterling, W T, and Flannery, B P, *Numerical Recipes: The Art of Scientific Computing, Third Edition,* Cambridge University Press ISBN-10: 0521880688, section 7.9, 2007.

[14] Helton, J C, and Davis, F J, *Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems,* Reliability Engineering and System Safety 81, 23-69, 2003.

[15] Esward, T J, Matthews, C E, Wright, L, and Yang, X-S, *Sensitivity analysis, optimisation, and sampling methods applied to continuous models: three test problems,* NPL Report MS 3, November 2010.

[16] Stocki, R. *A method to improve design reliability using optimal Latin hypercube sampling,* Computer Assisted Mechanics and Engineering Sciences 12, 87105.

[17] Iman, R L, and Conover W J *A distribution-free approach to inducing rank correlation among input variables,* Communications in Statistics - Simulation and Computation, 11(3), 311-334, 1982.

[18] Petelet, M, Iooss, B, Asserin, O, Loredo, A, *Latin hypercube sampling with inequality constraints,* Advances in Statistical Analysis, 94, 325-339, 2010.

[19] McKay, M D *Latin hypercube sampling as a tool in uncertainty analysis of computer models,* Proceedings of the 1992 Winter Simulation Conference, eds. Swain, J J, Goldsman, D, Crain, R C, et al. 557-564, 1992.

[20] Le Maître, O. P., and Knio, O. M. *Spectral Methods for Uncertainty Quantification*, Springer, Dordrecht, 2010.

[21] Wiener, N., *The homogeneous chaos,* American Journal of Mathematics, **60** (4), 897-936, 1938.

[22] Xiu, D., and Karniadakis G. E. *The Wiener-Askey polynomial chaos for stochastic differential equations*, SIAM, 24, 1118–1139, 2002.

[23] Ghanem, R. G., and Spanos, P. D., *Stochastic finite elements: a spectral approach,* Dover Publications, revised edition 2003.

[24] Najm, H. N., *Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics* Annual Review of Fluid Mechanics, **41**, 35, 2009.

[25] Xiu, D., *Fast numerical methods for stochastic computations: a review* Communications in Computational Physics, **5** (2-4), 242–272, 2009.

[26] Xiu, D. *Numerical Methods for Stochastic Computations: A Spectral Method Approach*, Princeton University Press, 2010.

[27] Augustin, F., Rentrop, P., and Wever, U. *Wiener Calculus for Differential Equations with Uncertainties*, Progress in Industrial Mathematics at ECMI 2010, 271–282, Mathematics in Industry 17, Günther, M., Bartel, A., Brunk, M., Schöps, S., and Striebel, M. (eds.), Springer, 2012.

[28] Smolyak, S. A. *Quadrature and interpolation formulas for tensor products of certain classes of functions,* Dokl. Akad. Nauk SSSR, 148, 1042–1045, 1963.

[29] Gerstner, T. and Griebel, M. *Numerical integration using sparse grids,* Numer. Algorithms, 18, 209–232, 1998.

[30] Elster, C., Klauenberg, K., Walzel, M., Wübbeler, G., Harris, P., Cox, M., Matthews, M., Smith, I., Wright, L., Allard, A., Fischer, N., Cowen, S., Ellison, S., Wilson, P., Pennecchi, F., Kok, G., van der Veen, A., Pendrill, L. *A Guide to Bayesian Inference for Regression Problems* Report, available for download from https://www.ptb.de/emrp/new04-publications.html

[31] Diwekar, U. M., and Kalagnanam, J. R. Efficient sampling technique for optimization under uncertainty. *AIChE Journal*, 43(2):440–447, 1997.

[32] Box, G. E. P., and Draper, N., *Response surfaces, mixtures, and ridge analysis,* Wiley, New Jersey, 2007.

[33] Myers, R. H., Montgomery, D. C., and Anderson-Cook, C. M., *Response surface methodology* Wiley, New Jersey, 2009.

[34] Myers, R. H., Khuri, A. I., and Carter, W. H., *Response surface methodology:1966-1988* Technometrics, 31:2, 137-157 1981.

[35] Krige, D. G. *A statistical approach to some mine valuations and allied problems at the Witwatersrand* Master's thesis of the University of Witwatersrand, 1951.

[36] Matheron, G., *Principles of Geostatistics* Economic Geology, 58, 1246-1266, 1963.

[37] Rasmussen, C. E., and Williams, C. K. I., *Gaussian process for machine learning,* MIT press, Cambridge, Massachussetts, 2006.

[38] Cressie, N., *The origins of kriging,* Mathematical Geology, 22:3,239-252, 1990.

[39] Ahmed, M. Y. M., and Quin, N., *Comparison of response surface and kriging surrogates in aerodynamic design optimization of hypersonic spiked blunt bodies* ASAT-13-AE-15, 2009.

[40] Keane, A. J., and Prasanth, B. N., *Computational approaches for aerospace design: The pursuit of excellence* John Wiley, West Sussex, 2005.

[41] Martin, J. D., and Simpson, T. W. *On the use of kriging models to approximate deterministic computer models,* Proc. of DETC2004, DAC-57300, 2004.

[42] Fang, K. T., Li, R., and Sudjianto, A., *Design and modelling for computer experiments,* Chapman & Hall/CRC, 2006.

[43] Roustant, O., Ginsbourger, D., and Deville, Y., *DiceKriging, DiceOptim: Two R Packages for the Analysis of Computer Experiments by Kriging-Based Metamodeling and Optimization,* Journal of Statistical Software, 51(1), 1-55, 2012.

[44] Gross, H., Model, R., Bär, M., Wurm, M., Bodermann, B., and Rathsfeld, A., *A scatterometry inverse problem in optical mask technology* Measurement, **39**, 782 (2006).

[45] M.-A. Henn, H. Gross, F. Scholze, M. Wurm, C. Elster, and M. Bär, *A maximum likelihood approach to the inverse problem of scatterometry* Optics Express, **20**, 12771– 12786, 2012.

[46] Henn, M-A., Gross, H., Heidenreich, S., Scholze, F., Elster, C., and Bär, M., *Improved reconstruction of critical dimensions in extreme ultraviolet scatterometry modeling systematic errors* Meas. Sci. Tech. **25** (4), 044003 (2014).

[47] A. Gelman and D. B. Rubin, *Inference from interative simulations using multiple sequences* Statistical Science, **7**, 457, 1992.

[48] Menter, F. R. *Two-equation eddy-viscosity turbulence models for engineering applications*, AIAA J., 32(8), 1598–1605, 1993.

[49] Gersten, K. *Fully developed turbulent pipe flow* in Fluid Mechanics of Flow Metering, 1–22, Merzkirch, w. (ed.), Springer, Berlin, 2005.

[50] Gersten, K., and Herwig, G. author = K. Gersten and G. Herwig, *Strömungsmechanik: Grundlagen der Impuls-, Wärme- und Stoffübertragung aus assymptotischer Sicht*, Vieweg, Braunschweig, 1992.

[51] Matthies, H. G. *Stochastic finite elements: Computational approaches to stochastic partial differential equations*, ZAMM, Z. Angew. Math. Mech., 88(11), 849–873, 2008.

[52] OpenFOAM, free, open source CFD software package developed by OpenCFD Ltd at ESI Group and distributed by the OpenFOAM Foundation, `www.openfoam.com`.

[53] Taylor, R.E., and Clark III, L.M. *Finite pulse time effects in flash diffusivity method* High Temperatures - High Pressures, 6(1), 65-72, 1974.

[54] Clark III, L.M., and Taylor, R.E. *Radiation loss in the flash method for thermal diffusivity* Journal of Applied Physics, 46(2),714-19, 1975.

[55] Parker, W. J., Jenkins, R. J., Butler, C. P., and Abbott, G. L. *Flash method of determining thermal diffusivity, heat capacity, and thermal conductivity*. Journal of Applied Physics, 32(9), 1679-1684, 1961.