A multi-element non-intrusive Polynomial Chaos method using agglomerative clustering based on the derivatives to study irregular and discontinuous Quantities of Interest

Nicolas Vauchel, Éric Garnier, Thomas Gomez

A multi-element non-intrusive Polynomial Chaos method using agglomerative clustering based on the derivatives to study irregular and discontinuous Quantities of Interest

Nicolas VAUCHEL[a]*, Éric GARNIER[a], Thomas GOMEZ[a]
a. Univ. Lille, CNRS, ONERA, Arts et Metiers Institute of Technology, Centrale Lille, UMR 9014 - LMFL - Laboratoire de Mécanique des fluides de Lille - Kampé de Fériet, F-59000 Lille, France *
Corresponding author - e-mail adress: nicolas.vauchel@onera.fr

Abstract

A non-intrusive method to get a multi-element Polynomial Chaos model is developed. This method is called ME-ACD, for Multi-Element based on Agglomerative Clustering on Derivatives. It aims at approximating a Quantity of Interest which presents discontinuities or irregularities making it difficult to be accurately approximated by standard Polynomial Chaos models. The method permits to efficiently split the parameter space and to train local polynomial models of lower degrees on every element where the local pieces of the Quantity of Interest are smoother. The algorithm is based on agglomerative clustering of the observations in a well-chosen abstract space taking into account the value of the Quantity of Interest and of its derivatives with respect to the stochastic input parameters. The same observations are used for both partitioning the space and training the local models. Several partitions of the parameter space are tested, and the one leading to local models minimizing a cross-validation error is selected. Once the training observations are labelled with a class number indicating the element they are located in, a neural network classifier is trained to determine which local model to use for further evaluations. The method has proven to efficiently split the parameter space for a set of applications of moderate dimension. The piecewise chaos model is compared with the ones of a standard Polynomial Chaos non-intrusive method and of a Gradient Boosted Trees method in terms of accuracy.

Keywords : Uncertainty Quantification, Multi-element Polynomial Chaos, Machine learning.

1 Introduction

Uncertainty Quantification (UQ) has become paramount in physical simulations in a wide range of fields. Deterministic simulations, with fixed parameters, have been the core of numerical analysis during many years, but they do not take into account the potential uncertainties of the parameters, notably the ones coming from data. UQ focuses on the impact of these uncertainties on an output, which is often denoted the "output" or the "Quantity of Interest" (QOI). The studied uncertainties are called the "input parameters" and are supposed to follow a stochastic distribution law. The determination of the distribution law of an input parameter is one step of UQ which is not discussed in the context of this article. Here is assumed that the input parameters are independent with a known probability distribution function (pdf). Some transforms are known in the literature to handle the case of independent input parameters and map them to independent ones [20, 16].
The classical Monte-Carlo (MC) method is known to be robust and simple to implement, but has a low

convergence rate and only gives information on the statistic moments of the QOI. The use of surrogate models overcomes these drawbacks. Surrogate models approximate the relation between the QOI and the input parameters by an analytical formula. In this article, we are interested in the Polynomial Chaos expansion (PC), first introduced by Wiener in [28]. Originally, this expansion aimed at building an approximation of a stochastic process as a series of Hermite polynomials of gaussian random variables. PC was later extended in [32] for other stochastic distributions using other orhogonal families, introducing the gPC (generalized polynomial chaos) framework. In the article, PC denotes an expansion in the gPC framework.

Constructing a PC model for an Ordinary Differential Equation or a Partial Differential Equation is classically done by Stochastic Galerkin (SG) methods, also called "Intrusive methods" [32] in the field of chaos polynomials. SG methods extend both input parameters and QOI into a chaos series and use orthogonalisation to obtain a system of coupled differential equations with every coefficient of the polynomial approximation series of the QOI. These methods can therefore result in large systems and need to develop new simulation codes, which is extremely difficult for complex systems. In the context of this article, "Non-intrusive" (NI) methods are considered [31, 2, 29, 14, 4, 8]. NI methods consist in fixing a value of the input parameters and then running the deterministic associated system, which is suitable for QOIs with complex computation, seeing it as a "black box". NI methods are then close to Machine learning (ML) problems. A set of fixed values of the input parameters and the corresponding value of the QOI with these parameters is called an "observation" (or in some references a "node"). Observations can be represented in an abstract space. The parameter space is one possible abstract space where every coordinate of a point corresponds to the value taken by the input parameters. The number of input parameters is then representing the dimension of this space. The PC model is obtained with the values of the QOI on a set of observations, called "training observations". There are numerous NI methods. Some of them are based on classical interpolation or linear regression methods but with observations representing stochastic uncertainties. For more details on "Intrusive" or "Non-intrusive" methods, the reader is referred to [30].

PC models (both obtained with SG and NI) have been proven to be very efficient in many cases when the QOI is smooth with respect to the input parameters but its accuracy highly deteriorates in presence of irregularities and discontinuities. In this article, the term of irregularities refers to a continuous behaviour needing a high truncation degree of the chaos expansion to be accurately approximated. As shown below, the number of terms of the expansion grows in factorial with the truncation degree and with the dimension of the parameter space (the number of input parameters), which can become a burden both in term of computational time and memory to approximate such quantities. Moreover, a discontinuous QOI suffers from Gibbs phenomenon, which does not vanish when the truncation degree is increased.

Among other methods, Multi-Element (ME) methods were developed to address these issues [27, 5, 10, 22]. The first NI ME method, ME-PCM [5] (the non-intrusive equivalent of the multi-element intrusive method ME-PC [27]), consists in refining the parameter space into smaller and smaller hypercubes, and applying stochastic collocation methods on these elements to find the chaos coefficients of the local PC models. In that way, a QOI can be approximated locally by models with lower degrees, which can reduce the cost of the algorithm in time and in memory. In the case of a discontinuous QOI, the Gibbs phenomenon is confined in the hypercubes which contain the discontinuity so that its impact on the variance of the QOI is reduced. However, if the discontinuity or the irregularity is not aligned with the Cartesian axis of the parameter space, the number of hypercubes can grow dramatically.

In [10], the minimal multi-element method has been developed to study a discontinuous QOI. Relying on polynomial annihilation edge detection [1], this algorithm aims at splitting the parameter space along the discontinuities, which totally removes the Gibbs phenomenon as the QOI is smooth on each element. The same observations are used to divide the parameter space and to train the surrogate models, using the least orthogonal interpolation [15] on every element. Similar methods split the parameter space along the discontinuity using Bayesian inference to detect the discontinuity and to approximate its curve equation [22]. Nevertheless, these two methods only focus on discontinuous behaviours but do not simplify the study of irregular continuous behaviours as ME-PCM did. Moreover, to the author knowledge, these methods

do not permit to split the parameter space in presence of a discontinuity which runs partially through it. In this article, a new NI ME method is developed. This method aims at partitioning the parameter space according to a potential discontinuity or efficiently partitioning it to use local models of lower degree in presence of irregularities. This method is called "ME-ACD", which stands for Multi Element method based on Agglomerative Clustering on Derivatives. To split the parameter space, the used method is agglomerative clustering on the training observations represented in an well chosen abstract space taking into account the value of the QOI and the values of its derivatives with respect to the input parameters. This method also permits to split the parameter space in presence of a discontinuity which runs partially through it. The same training observations are used for both partitioning the parameter space and for training the local surrogate models with linear regression. Once the piecewise model is obtained, the observations are labelled with an index stating in what element they are in and a supervised classifier is trained to determine which local polynomial model is used for a given evaluation. The chosen classifier is the Multi-Layer Perceptron (MLP) classifier, based on artificial neural network.
The use of clustering to study discontinuous UQ has already been done in [19] and [23] but both are only driven by the QOI values on the observations. To the author knowledge, the development of a method which selects the most suitable partition of the parameter space using clustering on a abstract space containing the derivatives of the QOI had never been done in the field of UQ.

The article is outlined as following : PC framework and its extension to multi-element is developed in Section 2. Section 3 presents in details the different steps of ME-ACD. Section 4 shows its performances on a set of applications. The accuracy of ME-ACD is compared with the one of an algorithm of Gradient Boosted Trees (GBT) [6] and with the accuracy of classical NI PC method on a set of discontinuous and irregular continuous QOIs. GBT is not a PC method, but a ML one which can be used for regression. As NI PC methods are classical regression or interpolation methods used in the context of UQ with a polynomial surrogate model, GBT can be used in this same context, except that it does not give orthogonal polynomial series. GBT are adapted to the study as they are said to be able to approximate even irregular and discontinuous QOIs. In a similar way, the method developed in this article in the context of UQ could be used in other approaches as a new regression method.

2 from PC expansion to ME model

In this section, notations and the standard PC framework are introduced and its limits are underlined. Then, some NI methods are recalled. Finally, the ME approach is introduced.

2.1 QOI and input parameters

The studied QOI is denoted f. f can be a given function, the solution of an Ordinary Differential Equation system at a given instant or the solution of a Partial Differential Equation system at a given location and instant. In this work, the possible dependencies in time and in space of the QOI are ignored to only focus on its dependency in the input parameters, without loss of generality. f depends on d stochastic input parameters $\mathbf{p} = (p_1, ..., p_d)^T$ which are supposed stochastic and independent. With this hypothesis, $\mathbf{p}$ follows the following multivariate pdf

$$\rho(\mathbf{p}) = \prod_{m=1}^{d} \rho_m(p_m) \tag{1}$$

where ρ_m is the pdf of p_m. All the pdfs are supposed known. f is therefore stochastic and is supposed having a finite variance.

2.2 PC expansion

PC methods aims at approximating the dependence of the QOI f with respect to the stochastic input parameters $\mathbf{p}$ as a sum of multivariate orthogonal polynomials. As orthogonal polynomials take input parameters in a standard form, a bijective mapping is needed. In [2], every input parameter is mapped with a standard Gaussian variable and multivariate Hermite polynomials are selected as orthogonal basis for the polynomial expansion. Isoprobabilisitic transformations can also be used to map every input parameter to an input parameter following the same kind of stochastic distribution but which is standardized. In that way, the used multivariate orthogonal polynomials are the ones being products of univariate polynomials corresponding to the standard distribution law of the input parameter through the Wiener-Askey scheme [32]. In this article, another mapping, developed below, is considered. For NI methods, as for ML methods, a sampling of the input parameters is effectuated. In the case of UQ, the observations are representing stochastic parameters, so the sampling is usually done in respect to their distribution. Observations are then mapped. The orthogonal properties are then used, notably to estimate the moments of the QOI and the Sobol sensitivity indices [26].

With the ME decomposition, the orthogonal properties of the polynomial family cannot be used as local models only fits data on its element and is ignored in the rest of the parameter space, whereas the orthogonal properties need information of the model on the whole parameter space. A method permitting to use the orthogonal properties of the polynomials and then estimating Sobol indices with a ME model is the subject of further researches. Here, as orthogonal properties are not required for the learning of the model, the canonical multivariate polynomials basis $(\Psi_{\mathbf{k}})$ is used. For $\mathbf{k} \in \mathbb{N}^d$

$$\Psi_{\mathbf{k}} = \prod_{m=1}^{d} X^{k_m} \tag{2}$$

Moreover, a bijective mapping from $\mathbf{p}$ to $\mathbf{x}$, where every x_m follows an uniform law on $[0, 1]$, is used. To do this, another hypothesis is considered on top of the others. Every ρ_m is strictly positive (instead of being only positive), so that its cumulative distribution function C_m is bijective.

$$x_m = C_m(p_m) \tag{3}$$

From now on, the input parameters refer to $\mathbf{x}$, the parameter space refers to $\mathcal{X} = [0, 1]^d$, the unit $d-$dimensional hypercube which contains all possibles $\mathbf{x}$ and the QOI refers to $\tilde{f}$

$$\tilde{f}(\mathbf{x}) = f(C_1(p_1), ..., C_d(p_d)) = f(\mathbf{p}) \tag{4}$$

Once the model is trained, the same mapping is necessary for evaluating it with new values.

The truncation degree of the chaos series is denoted $n \in \mathbb{N}$. There are several way to truncate a series of multidimensional polynomials, which leads to a finite number of terms M. The classical truncation scheme consists on selecting all the polynomials with $||\mathbf{k}||_1 = \sum_{m=1}^{d} k_m \leq n$. This set of polynomials has a size of

$$M(n, d) = \frac{(n + d)!}{n!d!} \tag{5}$$

M highlights the famous "curse of dimensionality", which underlines the dramatical increase of the number of terms of the series with d and n. For high values of n and d, the computational cost in time and in memory can become problematic. From this moment, the multi-index $\mathbf{k}$ is not used anymore and is replaced by a scalar index j. The polynomials are indexed from 1 to $M(n, d)$ by increasing truncation degree. The order between polynomials with the same truncation degree is not important. Moreover, the

dependence on n and d is sometimes omitted to simply denote the number of chaos terms M. In the approach, the standard computational PC model $\tilde{f}$ is then written

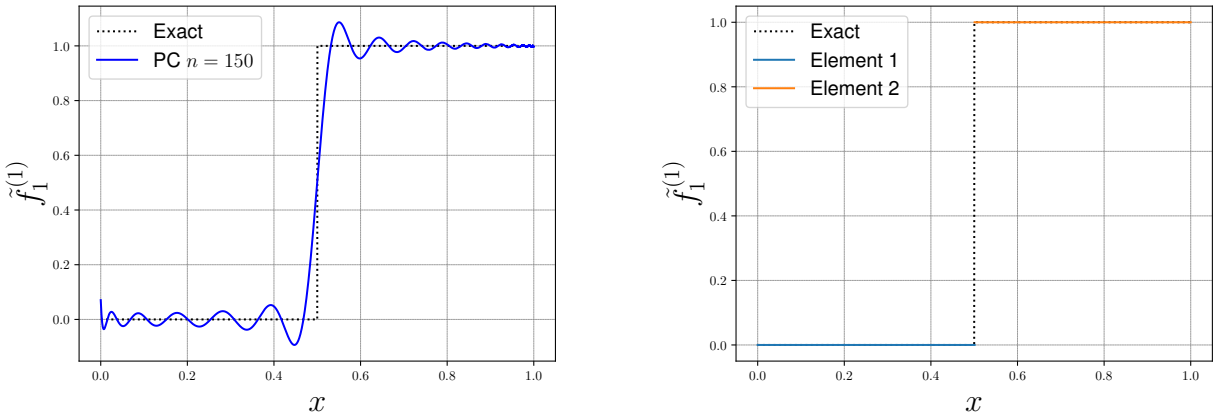
$$\hat{f}(\mathbf{x}) = \sum_{j=1}^{M} c_j \Psi_j(\mathbf{x}) \tag{6}$$

PC model $\tilde{f}$ converges fast when the QOI $\tilde{f}$ is smooth. However, as stated above, the model is not accurate when the QOI is irregular or discontinuous. Figure 1 underlines this loss of accuracy for a $1D$ piecewise constant QOI $\tilde{f}^{(1)}$

$$\tilde{f}_1^{(1)}(x) = \begin{cases} -1 & \text{if } x < \frac{1}{2} \\ 1 & \text{otherwise} \end{cases} \tag{7}$$

where x follows an uniform law on $[0, 1]$.

The Gibbs phenomenon induces that even by increasing the truncation degree, the oscillations close to the discontinuity are not vanishing. In this example, a piecewise surrogate model would have been very accurate, approximating only the two constant parts of the function independently with 0^{th}-degree local polynomial models.



(a) (b)

Figure 1: (a) Plot of $\tilde{f}_1^{(1)}$ (dashed lines) and of its PC surrogate model $\hat{f}_1^{(1)}$ (plain lines) with $n = 150$. (b) Plot of $\tilde{f}_1^{(1)}$ (dashed lines) and of its piecewise polynomials model (called Multi-element model below).

2.3 Non-intrusive methods

Before introducing ME techniques, three NI methods are recalled. NI methods aim at finding the chaos coefficients of the model by running simulations on a set of observations called the training set or the training observations. In the whole article, the number of training observations is denoted N and the set of input parameters taken in the training observations is denoted $\mathcal{T} = \{\mathbf{x}_i\}_{i=1}^{N}$.

Pseudo-spectral projection [29, 14], one of the stochastic collocation method, uses the orthogonality of the chosen chaos polynomials basis to find the expression of the coefficients with an integral. Quadrature rules are then used to evaluate the integral. This implies that the training observations are structured. The structure can correspond to the Gauss-Legendre grid or the Smolyak sparse grids [24] for example. In the context of this article, the ME approach based on clustering implies that the observations are unstructured, and the use of Pseudo-spectral projection method is not possible.

Two NI methods are able to get the chaos coefficients with an unstructured observation grid : interpolation on unstructured grid and linear regression. *Interpolation* on unstructured grid is developed in [15]. The surrogate model is forced to be equal to the real value of the QOI at each training observation. This method is used in [10, 9].

In the present study, Ordinary Least Square (OLS) method, a linear regression method, is used to find the coefficients. Contrary to interpolation on unstructured grid, *regression* methods [2, 4, 8] do not imply that the surrogate model is equal to the real value of the QOI on the training observations. The aim of OLS is to minimize the euclidean distance between the model value and the real value on the N training observations

$$\mathbf{C}_{opt} = \underset{C}{\operatorname{argmin}} \parallel \mathbf{AC} - \mathbf{B} \parallel^2 \tag{8}$$

where $\mathbf{C} \in \mathbb{R}^M$ is the vector containing the unknown chaos coefficients, $\mathbf{A} \in \mathcal{M}_{N,M}$ is the design matrix and $\mathbf{B} \in \mathbb{R}^N$ is the vector of predicted values. For the particular case of finding coefficients of a polynomials model : $A_{ij} = \Psi_j^{(d)}(\mathbf{x}_i)$, $B_i = \tilde{f}(\mathbf{x}_i)$ and $C_j = c_j$

The problem is equivalent to solve this matrix equation, usually referred to as the normal equations

$$\mathbf{A}^T\mathbf{AC} = \mathbf{A}^T\mathbf{B} \tag{9}$$

The solution is therefore

$$\mathbf{C}_{opt} = \mathbf{A}^+ B \tag{10}$$

where $\mathbf{A}^+ = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$.

The number of terms of the series M (determined in the case of finding the chaos coefficients by the chosen truncation degree n and the dimension d through equation (5)) is a crucial variable for regression. Compared to the number of training observations N, if M is too low, the regression is poorly accurate (usually called "underfitting" or "bias" in the ML field). If M is too high, what is called "overfitting" or "variance" in the ML field can occur. Overfitting is the fact that the model is accurate on the training observation but do not generalize well on other observations, thus leading to a poorly accurate model in between the training observations. These aspects can be taken into account by testing all regressions with the truncation degrees between the chosen n_{min} and n_{max} and selecting the optimal model as the one minimizing a cross validation error.

The considered error has to be an error between the surrogate model values and the real values of the QOI evaluated *on the training observations*. In that way, this error can be used when the algorithm is running to select the most suitable degree of a model and more generally the best hyperparameters to use in an range of values. Using the Mean Square Error (MSE) between the local surrogate model values and the values of the QOI on the training observations would give very poor information of the model accuracy if overfitting occurs. In that way, cross-validation is used. The Predicted Residual Sum of Squares (PRESS) error (based on the Leave-One-Out error) used in [4] is really efficient in the context of PC models learned with linear regression. PRESS error is defined by

$$\delta = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{f}^{(i)}(\mathbf{x}_i) - \tilde{f}(\mathbf{x}_i)\right)^2 \tag{11}$$

where $\hat{f}^{(i)}$ is the surrogate model learned on the set $\mathcal{T}\backslash\{\mathbf{x}_i\}$.

Fortunately, in the context of OLS, a method exists to express analytically every term of the sum without having to train a new model on the set $\mathcal{T}\backslash\{\mathbf{x}_i\}$ for every observation $\mathbf{x}_i$. This formula [21], which only needs the model trained on the observations of $\mathcal{T}$, leads to

$$\delta = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\hat{f}(\mathbf{x}_i) - \tilde{f}(\mathbf{x}_i)}{1-h_i}\right)^2 \tag{12}$$

where h_i is the i^{th} diagonal term of the matrix $\mathbf{H} = \mathbf{A}\mathbf{A}^+$ known as the "Hat matrix" or the "projection matrix". This error correctly gives information on the accuracy of the local surrogate models, even when overfitting occurs.

To compute $\mathbf{A}^+$ and the PRESS error, the Singular Value Decomposition (SVD) of $\mathbf{A}$ is used [12]. If $\mathbf{A}$ is a $N \times M$ matrix, finding the SVD of $\mathbf{A}$ consists in finding a $N \times N$ orthogonal matrix $\mathbf{U}$, a $N \times M$ diagonal rectangular matrix $\mathbf{\Sigma}$ and a $M \times M$ orthogonal matrix V so that

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{13}$$

The columns of $\mathbf{U}$ are called "left singular vectors" of $\mathbf{A}$, they are the eigenvectors of $\mathbf{A}\mathbf{A}^T$. The columns of $\mathbf{V}$ are called "right singular vectors" of $\mathbf{A}$, they are the eigenvectors of $\mathbf{A}^T\mathbf{A}$. The diagonal terms of $\mathbf{\Sigma}$ are the so-called "singular values" of $\mathbf{A}$, which are the eigenvalues of $\mathbf{A}\mathbf{A}^T$. SVD decomposition is not unique, but what is usually done is to take the one which leads to the singular values ordered by decreasing value in the diagonal of $\mathbf{\Sigma}$. If the rank of $\mathbf{A}$ is equal to r, $\mathbf{A}$ has r non-zero singular values.

SVD is really useful to compute efficiently $\mathbf{A}^+$. It is proven in [12] that

$$\mathbf{A}^+ = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^T \tag{14}$$

where $\mathbf{\Sigma}^+$ is the diagonal rectangular $M \times N$ matrix with the inverse of the non-zero singular values on the diagonal and zeros elsewhere.

SVD is also useful to compute the "Hat matrix" and then the PRESS error. Indeed, we have

$$\mathbf{H} = \mathbf{A}\mathbf{A}^+ = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{\Sigma}^+\mathbf{U}^T = \mathbf{U}_r\mathbf{U}_r^T \tag{15}$$

where $\mathbf{U}_r$ is the square matrix containing the first r rows and columns of $\mathbf{U}$. The diagonal terms of $\mathbf{H}$, needed for the computation of the PRESS error, are then equals to

$$h_{i,i} = \sum_{j=1}^{r}\left[(U_r)_{i,j}\right]^2 \tag{16}$$

Not all the rows and the columns of $\mathbf{U}$ and $\mathbf{V}$ have to be computed to get $\mathbf{A}^+$ and $\mathbf{H}$. Taking only the r first columns of $\mathbf{U}$, $\mathbf{\Sigma}$ and the first r rows of $\mathbf{V}^T$ is known as "compact SVD".

Algorithm 1 underlines the pseudo-code to compute OLS and the PRESS error.

Note : in all the pseudo-codes of this article, $X \leftarrow$ value means that variable X gets the given value. Moreover, a ":" used as an index of a vector or a matrix denotes all the rows or all the columns of the item.

If $\mathbf{A}$ is ill-conditioned, the user can ignore a set of singular value less than a given threshold, thus considering $\hat{r} \leq r$ singular values instead of r.

Algorithm 1 Pseudo-code of the computation of OLS and PRESS error

$\mathbf{A} \leftarrow N \times M$ design matrix
$\mathbf{B} \leftarrow$ vector of size N containing the values of the QOI on the training observations
$\mathbf{U}$, $\mathbf{\Sigma}$, $\mathbf{V} \leftarrow SVD(\mathbf{A})$ (compact form)
$\mathbf{\Sigma}^+ \leftarrow \min(N,M) \times \min(N,M)$ matrix with $\frac{1}{\sigma_{i,i}}$ on its diagonal and zeros elsewhere
$\mathbf{X} \leftarrow \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^T B$
$\mathbf{H} \leftarrow \sum_{j=1}^{r} \left[(U)_{i,j}\right]^2$ (Here $\mathbf{H}$ is the vector containing the diagonal terms of the "Hat matrix" and not the matrix itself)
$\mathbf{Y} = \mathbf{A}\mathbf{X}$
$\delta \leftarrow \frac{1}{N} \sum_{i=1}^{N} \left(\frac{Y_i - B_i}{1 - H_i}\right)^2$
Return $\mathbf{X}$, δ

Let's consider that $N \geq M$ without loss of generality (if $M > N$ the operations can be conducted on the transpose of the matrix). In the most time expensive case, $r = M$. Time complexity of a matrix product between a $N \times M$ matrix A and a $M \times P$ matrix B is in $\mathcal{O}(NMP)$. In that way, time complexity of calculating $\mathbf{X}$ is in $\mathcal{O}(M^3 + M^2N + MN) = \mathcal{O}(M^2N)$. To the author knowledge, the complexity of compact SVD is $\mathcal{O}(M^2N)$. Time complexity of the algorithm of OLS and PRESS Error is then $\mathcal{O}(M^2N)$.

In [8], the authors state that a standard random sampling often lead to suboptimal approximations, and show the existence of a special sampling leading to nearly optimal solutions. However, as it is described in section 2.4, in the current paper, regression is used on subsets of the sampling, which induces that for every approximation, the user does not have the control on the position of the training observations which are used in the training of the local surrogate models.

2.4 Multi-element approach

As stated above in section 2, to approximate an irregular or a discontinuous QOI, one solution is to use a ME method. ME methods consist into partitioning the parameter space $\mathcal{X}$ into N_E non-overlapping subsets E_e called elements

$$\mathcal{X} = \bigcup_{e=1}^{N_E} E_e \tag{17}$$

$$E_{e_1} \bigcap E_{e_2} = \emptyset \text{ if } e_1 \neq e_2 \tag{18}$$

Once $\mathcal{X}$ is partitioned, a local surrogate model $\hat{f}_e$ is obtained in every element E_e with the training observations located in this element. The global surrogate model is then a piecewise model composed of every local model. The global piecewise model is written

$$\hat{f}(\mathbf{x}) = \sum_{e=1}^{N_E} \hat{f}_e(\mathbf{x}) \mathcal{I}_{E_e}(\mathbf{x}) = \sum_{e=1}^{N_E} \left(\sum_{j=1}^{M_e} c_{ej} \Psi_j^{(d)}(\mathbf{x}) \right) \mathcal{I}_{E_e}(\mathbf{x}) \tag{19}$$

with M_e the number of multivariate monomials of the local PC model $\hat{f}_e$. $\mathcal{I}_{E_e}$ is the indicator function of the element E_e satisfying

$$\mathcal{I}_{E_e}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in E_e \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

In numerical applications, the classifier plays the role of the indicator function. Once the global model is trained, to evaluate the model on a new point, the classifier detects in which E_e the observation lies to

know which local model to use. Every coefficient of the local polynomials models $\hat{f}_e$ is obtained with the OLS algorithm on the training observations located in E_e.

3 ME-ACD algorithm

3.1 Abstract spaces

In the following algorithm, different abstract spaces are considered:

- $\mathcal{P}$: the original d-dimensional parameter space of points of coordinates $(p_1, ..., p_d)$.
- $\mathcal{X}$: the mapped d-dimensional parameter space containing points of coordinates $(x_1, ..., x_d)$. This space is the one referred to as the parameter space.
- $\mathcal{F}$: the $d+1$-dimensional space containing points of coordinates $(x_1, ..., x_d, \tilde{f}(\mathbf{x}))$.
- $\mathcal{D}$: the $d+1$-dimensional space containing points of coordinates $(\tilde{f}(\mathbf{x}), \frac{\partial \tilde{f}}{\partial x_1}(\mathbf{x}), ..., \frac{\partial \tilde{f}}{\partial x_d}(\mathbf{x}))$.

In that way, an observation can be considered in those different spaces. The objective is to split the parameter space $\mathcal{X}$ into a partition of N_E elements E_e. To do this, agglomerative clustering is used. The results of the clustering are different according to the space the observations areconsidered in. The space has to be chosen so that the clustering efficiently splits the parameter space both in the case of a discontinuous QOI and in the case of a continuous but irregular QOI.

3.2 Step 1 - Sampling of the parameter space

The first step of the algorithm consists in properly sampling the parameter space $\mathcal{X}$ to get a training set $\mathcal{T} = \{\mathbf{x}_i\}_{i=1}^N$ of N training observations. One aim of ME-ACD is to work on every sampling, so that it could be used with experimental data. In that way, an adaptive sampling is not used, conversely to other methods [10, 9]. In this article, Sobol quasi-random sequences [25] are used. Sobol sequences, as Latin Hypercube Sampling, aims to sample a $d-$dimensional space leaving less sparse zones in the hypercube than purely random Monte-Carlo sampling. More information on the Sobol sequences can be found in [11]. The initial direction numbers used in the article are the one suggested on the following website : https://web.maths.unsw.edu.au/ fkuo/sobol/ (last accessed : April 2022).

With this strategy, the unit hypercube is sampled and the corresponding $\mathbf{p}_i$ of input parameters of training observation $\mathbf{x}_i$ can be obtained by inverting formula (3). In some cases, a slight modification on the sampling can be performed for the observations located on the hyperfaces of the unit hypercube. They can be translated of a distance $\epsilon = 10^{-3}$ perpendicularly to the hyperface (ϵ is added to every components of every $\mathbf{x}_i$ which are equals to 0 and is subtracted to every components equals to 1). This avoids having an infinitely far point for normal distribution for example. Figure 2 shows a Monte-Carlo sampling and a Sobol sequence for a $2D$ example. 500 observations of $\mathbf{p}$ with p_1 following an uniform law on $[1, 1.5]$ and p_2 following a normal law of mean 10 and of standard deviation 5 sampled with the two strategies are represented in $\mathcal{X}$ and in $\mathcal{P}$. Monte Carlo and Sobol samplings are very close to each other, but the latter is elaborated to guarantee a lower discrepancy.

Once the sampling is performed, the QOI is evaluated on every training observations of $\mathcal{T}$ to get a vector $\mathbf{B} \in \mathbb{R}^N$ with :

$$B_i = \tilde{f}(\mathbf{x}_i) = f(\mathbf{p}_i) \tag{21}$$

Time complexity of the computation of a Sobol sequence is in $\mathcal{O}(N \times d)$. Time complexity of evaluating the QOI is independent of the algorithm.

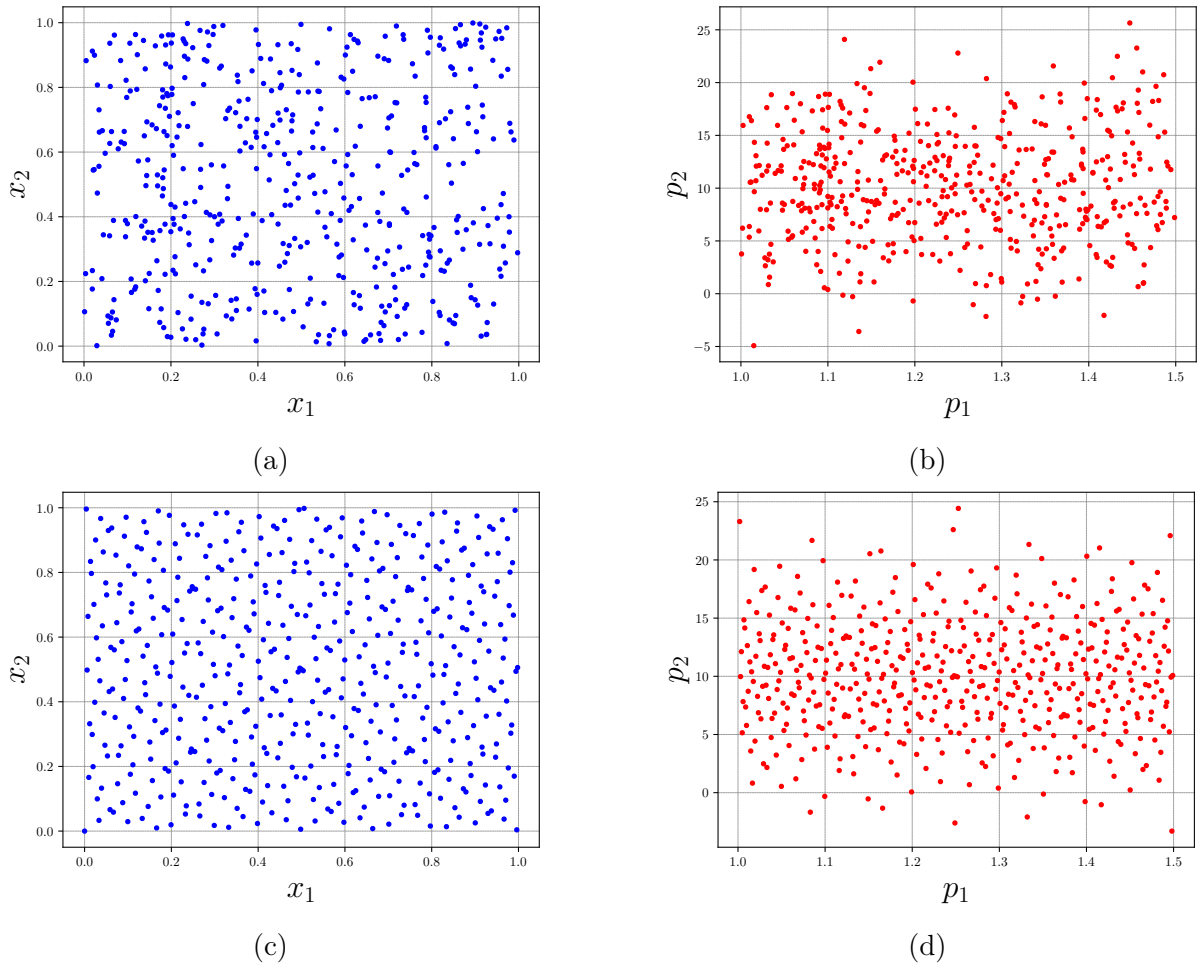


Figure 2: $N_{train} = 500$ observations generated by 2 sampling strategies. (a) Monte-Carlo sampling in $\mathcal{X}$. (b) Mapping of the Monte-Carlo sampling in $\mathcal{P}$. (c) Sobol sampling in $\mathcal{X}$. (d) Mapping of the Sobol sampling in $\mathcal{P}$.

3.3 Step 2 - Computation of the global design matrix

In Step 4, the parameter space is partitioned and OLS with a multidimensional polynomial model of degree varying from n_{min} to n_{max} is used on every element to find the chaos coefficients of the local models. A major advantage of using OLS in this approach is that a single design matrix has to be computed once. This matrix will be referred to as the global design matrix. When OLS is called to learn the coefficients of a local model, only a submatrix of this global design matrix is used. The rows of the submatrix are the rows corresponding to training observations located in the element. Its columns are the first $M(n, d)$ columns corresponding to the chosen truncature degree n. This global matrix, denoted $\mathbf{A}$, is a $N \times M(n_{max}, d)$ matrix, $M(n_{max}, d)$ being the number of terms of the polynomial series of degree n_{max} (see formula (5)). The global design matrix terms are written

$$A_{i,j} = \Psi_j(\mathbf{x}_i) \tag{22}$$

As a reminder, j is a scalar index used to denote the multidimensional canonical polynomial Ψ_j. For theses polynomials :

$$\Psi_j = \prod_{m=1}^{d} X_m^{k_m} \tag{23}$$

$$n = \sum_{m=1}^{d} k_m \tag{24}$$

with $n \in [0, n_{max}]$ being the degree of Ψ_j.

Computing this global matrix in a naive way would result in recomputing every components $(\mathbf{x}_i)_l^{k_l}$ for the calculus of every $\Psi_j(\mathbf{x}_i)$ thus leading to a complexity in $\mathcal{O}(d^2)$ for the computation of every $\Psi_j(\mathbf{x}_i)$. Computing the global matrix would then have a complexity in $\mathcal{O}\Big(N \times M(n_{max}, d) \times d^2\Big)$.

Instead of using this naive computation, for every training observations $\mathbf{x}_i$, we can store the values of every $(\mathbf{x}_i)_m^{k_m}$ in a $d \times n_{max}$ matrix $D^{(i)}$ with $D^{(i)}_{m,l} = (\mathbf{x}_i)^l_m$. Every term $A_{i,j}$ of the global design matrix is then the product of terms of $D^{(i)}$:

$$A_{i,j} = \prod_{m=1}^{d} D^{(i)}_{m,k_m} \tag{25}$$

With this computation scheme, the computation of a matrix $D^{(i)}$ is in $\mathcal{O}(d \times n_{max})$. The computation of the term $\Psi_j(\mathbf{x}_i)$ with these matrices is in $\mathcal{O}(d)$. Time complexity of computing the global matrix is then in $\mathcal{O}\Big(N \times d \times n_{max} + N \times M(n_{max}, d) \times d\Big) = \mathcal{O}\Big(N \times M(n_{max}, d) \times d\Big)$.

Algorithm 2 is the pseudo-code of the global design matrix computation with this method.

Algorithm 2 Pseudo-code of the computation of the global design matrix

```
d ← dimension of the parameter space
𝒯 ← input parameters of training observations sampled in Step 1
n_max ← maximal wanted degree of the local models
M ← (n_max+d)! / (n_max! d!)
Initiate a N × M matrix A with zeros
for i ∈ [1, N] do
    x ← 𝒯_i
    Initiate a d × n_max matrix D with zeros
    for l ∈ [1, d] do
        for m ∈ [1, n_max] do
            D_{l,m} ← x_l^m
        end for
    end for
    for j ∈ [1, M] do
        k ← multidimensional degree corresponding to the mapped scalar degree j
        a = 1
        for l ∈ [1, d] do
            a ← a × D_{l,k_l}
        end for
        A_{i,j} ← a
    end for
end for
```

3.4 Step 3 - Estimation of the derivatives

In Step 4, agglomerative clustering is used on the observations considered as points of $\mathcal{D}$. Therefore, the values of the derivatives of the QOI on the unstructured training points grid need to be estimated. To estimate these values, the first order Taylor's theory is employed. For two different observations $\mathbf{x}$ and $\mathbf{x}'$

$$\tilde{f}(\mathbf{x}) \approx \tilde{f}(\mathbf{x}') + \frac{\partial \tilde{f}}{\partial x_1}(\mathbf{x}') \times (x_1 - x_1') + ... + \frac{\partial \tilde{f}}{\partial x_d}(\mathbf{x}') \times (x_d - x_d') \tag{26}$$

In that way, the values of the derivatives of the QOI on a given point $\mathbf{x}'$ can be obtained by using OLS algorithm, solving the previous formula with its nearest neighbours. $\{\mathbf{x^i}\}_{i=1}^{N_{neigh}}$ denoting the set of the N_{neigh} nearest neighbours of $\mathbf{x}'$, the following system has to be solved

$$\begin{pmatrix} x_1^1 - x_1' & \cdots & x_d^1 - x_d' \\ \vdots & & \vdots \\ x_1^{N_{neigh}} - x_1' & \cdots & x_d^{N_{neigh}} - x_d' \end{pmatrix} \begin{pmatrix} \frac{\partial \tilde{f}}{\partial x_1}(\mathbf{x}') \\ \vdots \\ \frac{\partial \tilde{f}}{\partial x_d}(\mathbf{x}') \end{pmatrix} = \begin{pmatrix} \tilde{f}(\mathbf{x^1}) - \tilde{f}(\mathbf{x}') \\ \vdots \\ \tilde{f}(\mathbf{x^{N_{neigh}}}) - \tilde{f}(\mathbf{x}') \end{pmatrix} \tag{27}$$

The observations considered as neighbours of $\mathbf{x}'$ have to be determined. A first idea would be to consider as neighbours the observations corresponding to the nearest points in $\mathcal{X}$. Nevertheless, if the QOI is discontinuous, the calculus of the gradient between two observations which are from different sides of the discontinuity is going to be highly altered as the QOI is not derivable in this zone. In that way, the N_{neigh} closest neighbours are going to be determined as the nearest neighbours of $\mathbf{x}'$ in the space $\mathcal{F}$, so that in this calculus, only points on the same side of the discontinuity are going to be considered as neighbours. The Ball tree algorithm [17] is used to search the neighbours. To determine the best number of neighbours, OLS algorithm is effectuated for N_{neigh} varying from k_{min} to k_{max} and the one minimizing the PRESS error is selected.

Now the observations can be represented in the space $\mathcal{D}$ and the agglomerative clustering to test the different partitions of the space can be used with these points. Algorithm 3 is the pseudo-code for Step 3.

This step contains the construction of a Ball tree on the training set considered as N points of a $d+1$-dimensional space. Construction time of the ball tree is in $\mathcal{O}(Nlog(N))$ [17]. According to author knowledge, query time in this ball tree is in $\mathcal{O}(d\log(N))$, so the time of all the queries for nearest neighbours is in $\mathcal{O}(dN\log(N))$. This step also contains the construction of a design matrix for every observations for a total time complexity in $\mathcal{O}(N \times dk_{max})$, and $k_{max} - k_{min} + 1$ regressions with submatrices of size $N_{neigh} \times d$ of that matrix. If k_{min} and k_{max} are proportional to the dimension d, time complexity of the $k_{max} - k_{min} + 1$ regressions for a an observation is in $\mathcal{O}(d^4)$. Time complexity of this step is more likely to be dominated by the regressions and is in $\mathcal{O}(N \times d^4)$.

3.5 Step 4 - Partition of the space and training of the local models

This step is the core of ME-ACD. It consists in testing different partitions of the parameter space $\mathcal{X}$ using agglomerative clustering and to select the one providing the most accurate local models. Partitions with $N_E \in [|N_{E_{min}}, N_{E_{max}}|]$ elements are considered leading to $N_p = N_{E_{max}} - N_{E_{min}} + 1$ tested partitions. In this section, as several partitions are considered, an element is denoted $E_{p,e}$, where $p \in [|1, N_p|]$ is the index of the partition with $N_E = N_{E_{min}} + p - 1$ elements and $e \in [|1, N_E|]$ is the index of the element in this partition. $E_{1,e}$ correspond to the elements of the partition with $N_E = N_{E_{min}}$ elements and $E_{N_p,e}$ to the elements of the partition with $N_E = N_{E_{max}}$ elements.

Agglomerative clustering follows a "bottom-up" approach : each point of the considered space begins with its own cluster and at each step, the 2 closest clusters are merged together, until the wanted number

Algorithm 3 Pseudo-code of Step 3

$d \leftarrow$ dimension of the parameter space
$\mathcal{T} \leftarrow$ input parameters of training observations sampled in Step 1
$\mathbf{B} \leftarrow$ values of the QOI of the training observations computed in Step 1
Initiate a $N \times d$ matrix D with zeros
Construct the Ball tree with training data considered in $\mathcal{F}$ (using information of $\mathcal{T}$ and of $\mathbf{B}$)
$\mathbf{J} \leftarrow N \times k_{max}$ matrix where every row contains the indices of the k_{max} nearest neighbours of the observation of index the index of the row sorted from the nearest to the furthest.
for $i \in [1, N]$ **do**
  $\mathbf{x} \leftarrow \mathcal{T}_\mathbf{i}$
  $\mathbf{I} \leftarrow J_{i,:}$
  Initiate a $k_{max} \times d$ matrix $\mathbf{A}$ and a vector $\mathbf{F}$ of length k_{max} with zeros
  for $m \in [1, k_{max}]$ **do**
    $F_m \leftarrow B_i - B_{I_m}$
    for $l \in [1, d]$ **do**
      $A_{m,l} \leftarrow x_l - \mathcal{T}_{I_m,l}$
    end for
  end for
  Initiate a $d \times (k_{max} - k_{min} + 1)$ matrix C and a vector $\mathbf{E}$ of length $k_{max} - k_{min} + 1$ with zeros
  for $m \in [k_{min}, k_{max}]$ **do**
    $C_{:,m-k_{min}+1}, \delta_m = OLS(A_{1:m,:}, F_{1:m})$
  end for
  $k_{opt} \leftarrow \underset{m \in [k_{min}, k_{max}]}{\text{argmin}} \ \delta_m$
  $D_{i,:} \leftarrow C_{:,k_{opt}}$
end for
return(D)

of clusters is reached. Ward clustering is used. In this clustering, the two clusters having the minimal sum of square of the distance between pairs of points (one from one cluster and the other from the other cluster) are merged at each step. To take advantage of the bottom-up approach, only the partition with $N_{E_{min}}$ elements is made. The partitions from $N_{E_{min}} + 1$ to $N_{E_{max}}$ corresponds to the last steps of the agglomerative clustering with $N_{E_{min}}$ elements, and are also accessible. From the partition with $N_E = N_{E_{max}}$ elements, two of the elements are merged to get the partition with $N_{E_{max}} - 1$ elements and so on until the partition with $N_E = N_{E_{min}}$ elements is reached. In that way, elements of partitions with $N_E < N_{E_{max}}$ elements are unions of the elements of the partition with $N_{E_{max}}$ elements. A compact way to store this information is to give a label to every training observations from 1 to $N_{E_{max}}$ corresponding to the index of the element $E_{N_p,e}$ the corresponding points belongs (partition with $N_{E_{max}}$ elements). As elements $E_{p,e}$ of a given partition are union of elements $E_{N_p,e}$, an element can be stored as a list of the indices corresponding to elements $E_{N_p,e}$ which compose it.

The information of all the partitions obtained by clustering can be stored in :

- a vector $\mathcal{L}$ of size N containing the labels for every training observations, labels corresponding to the index e of the element $E_{N_p,e}$ (maximal partition) they belong.
- N_p lists of elements $\mathcal{E}_p$, one by tested partition, containing N_E elements $E_{p,e}$. As $E_{p,e}$ is the union of some elements $E_{N_p,e'}$, one element is represented as a list of labels, labels corresponding to the index e' of the elements $E_{N_p,e'}$ which compose it.

Figure 3 illustrates this idea in a tree like structure diagram. In this example, $N_{E_{min}} = 2$ and $N_{E_{max}} = 4$. Elements are represented as circles and every step of agglomerative clustering is a depth of the tree. In the example, $\mathcal{L}$ is a vector of size N containing labels from 1 to 4 and they are 3 lists :

- $\mathcal{E}_1$ is a list containing the elements $E_{1,e}$ being stored as $E_{1,1} = [1]$ and $E_{1,2} = [2, 3, 4]$
- $\mathcal{E}_2$ is a list containing the elements $E_{2,e}$ being stored as $E_{2,1} = [1]$, $E_{2,2} = [2, 3]$ and $E_{2,3} = [4]$
- $\mathcal{E}_3$ is a list containing the elements $E_{3,e}$ being stored as $E_{3,1} = [1]$, $E_{3,2} = [2]$, $E_{3,3} = [3]$ and $E_{2,4} = [4]$

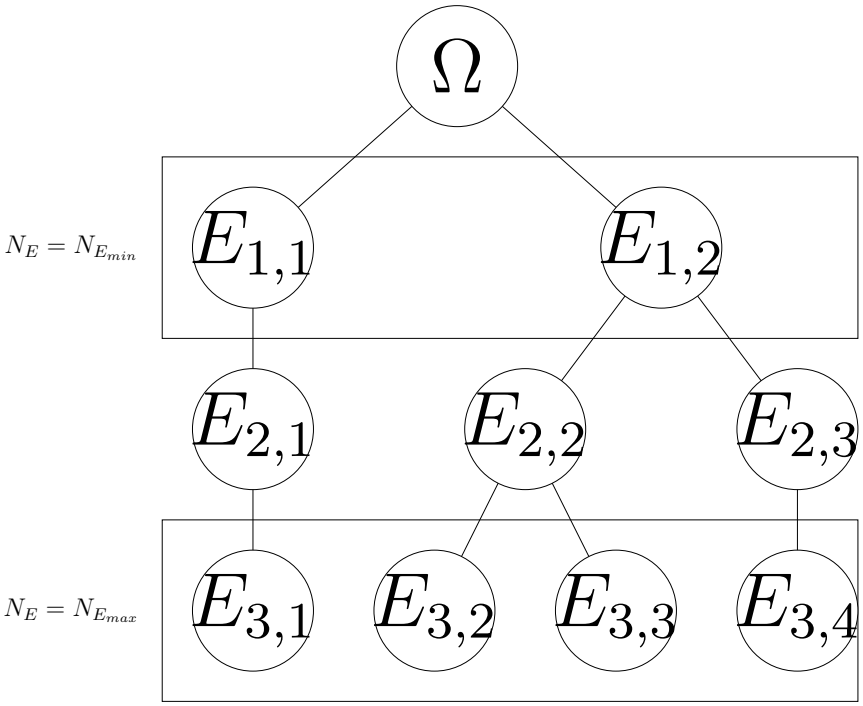


Figure 3: Tree-like diagram giving an example of agglomerative clustering. Clusters are represented by circles and steps from $N_E = 1$ to $N_{E_{max}} = 4$ are displayed ; $N_{E_{min}} = 2$ in this example.

With this data structure, for a given partition, the observations located in an element $E_{p,e}$ are the ones with a label $\mathcal{L}$ equals to one of the indices in the list which store the information of $E_{p,e}$. For example, an observation located in $E_{1,2}$ in the above example has a label of 2, 3 or 4 (corresponding to the elements of the partition with $N_E = N_{E_{max}}$ elements).

Applying the clustering step on observations considered in different spaces give different results. Making a clustering in $\mathcal{X}$ would not have any interest. Making a clustering of the observations considered as points in $\mathcal{F}$ would efficiently split the parameter space for a discontinuous QOI, but would not be efficient in the case of a continuous irregular QOI. An irregular behaviour of a QOI is due to a high variation of its derivatives. The idea is then to make clusters in which the QOI has "close" derivatives' values. In that way, agglomerative clustering on the observations considered in $\mathcal{D}$ is used. Therefore, the space is split according to discontinuities, as the value of the QOI is present in the coordinates of a point in $\mathcal{D}$, and according to irregular behaviours, as the values of the derivatives are also present in the coordinates. Figure 4 represents the plots of observations of two $1D-$QOIs $\tilde{f}_1^{(1)}$ (see equation (7)) and $\tilde{f}_2^{(1)}$ (see equation (28) below) considered in the spaces $\mathcal{D}$ and $\mathcal{F}$. $\tilde{f}_2^{(1)}$ is an irregular QOI defined by

$$\tilde{f}_2^{(1)}(x) = 10 \tanh\left(10(x - \frac{1}{2})\right) \tag{28}$$

As $d = 1$ for the examples of the figure, the considered sampling is an uniform $1D$ grid with a constant step. Figure 4 highlights how the clustering is performed in space $\mathcal{D}$ (subfigures (b) and (d)), and how it permits the use of local models of lower degrees (as shown by subfigures (a) and (c)) in both examples.

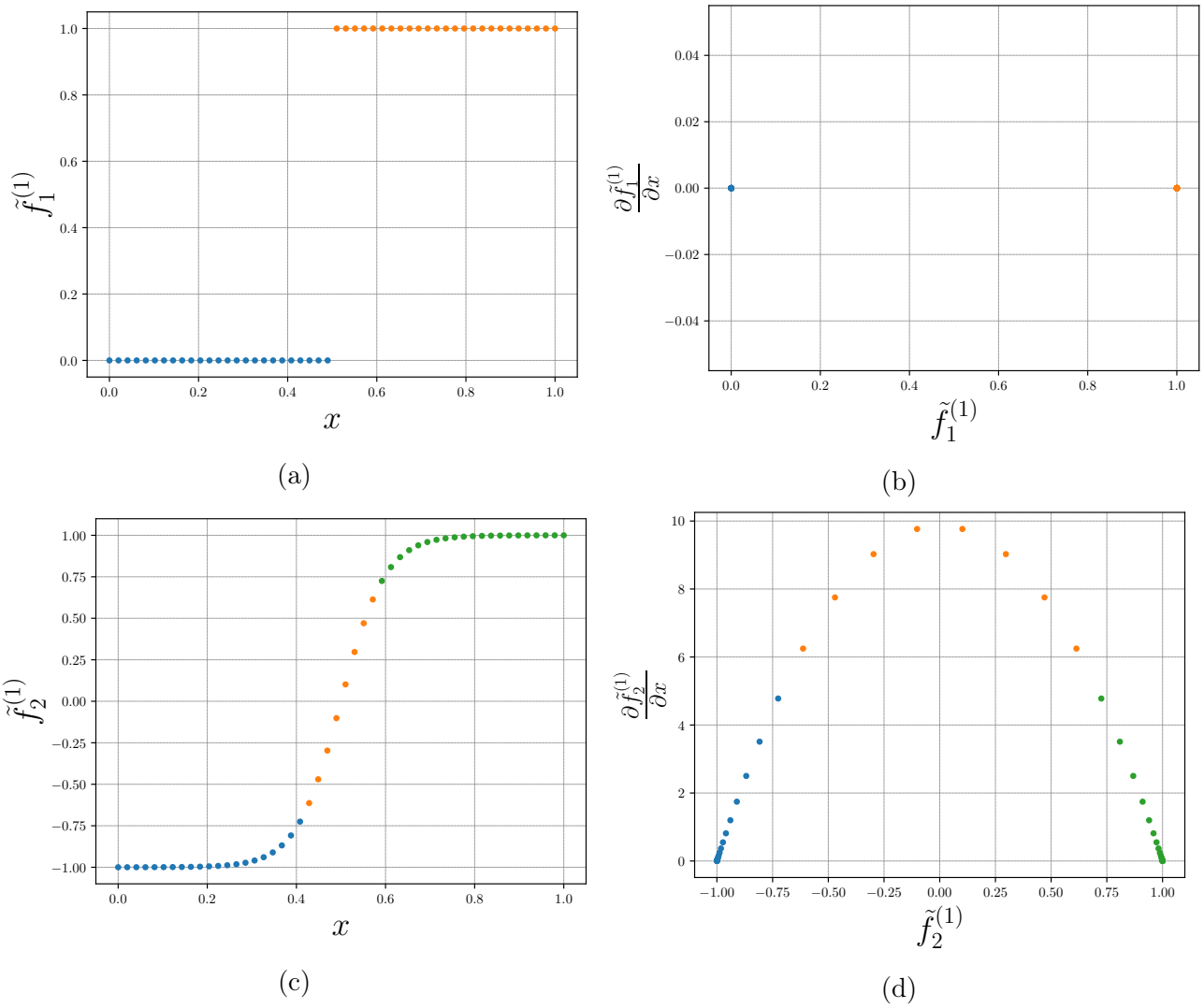


Figure 4: Plots of 50 observations of two $1D-$QOIs $\tilde{f}_1^{(1)}$ and $\tilde{f}_2^{(1)}$ considered in the spaces $\mathcal{F}$ and $\mathcal{D}$. The different colors of the observations stand for different clusters. (a) Observations of $\tilde{f}_1^{(1)}$ represented in $\mathcal{F}$. (b) Observations of $\tilde{f}_1^{(1)}$ represented in $\mathcal{D}$. (c) Observations of $\tilde{f}_2^{(1)}$ represented in $\mathcal{F}$. (d) Observations of $\tilde{f}_2^{(1)}$ represented in $\mathcal{D}$.

One can add a connectivity diagram between the training observations represented in $\mathcal{X}$ before the clustering (as an optional step between step 3 and step 4). With a connectivity diagram, only two clusters which have at least one observation connected together can be merged. This connectivity reduces the chances to gather two clusters which are far from one another in $\mathcal{X}$. If the user decide to use connectivity, one way to construct the diagram is to connect an observation with its $k_{connect}$ nearest neighbours in $\mathcal{X}$, $k_{connect}$ being a hyperparameter to choose. The choice of using or not a connectivity diagram is modelled by the boolean $isConnect$.

A low accuracy of the gradient calculus of step 3 can alter the partition but if a point is miscalculated with this method, it can be seen as an outlier in $\mathcal{D}$ and is more likely to be isolated by the partition approach, however creating unnecessary elements. Some action could then be taken to remove the observation from the training set without counting a supplementary element. In the scope of this article, no

outlier removing actions are taken.

Once the N_p partitions have been found, the one leading to the most accurate model has to be selected. For every partitions, local models are trained on the elements. The selection criterion of the optimal partition is based on the PRESS errors of the local models. The partition of $N_E \in [N_{E_{min}}, N_{E_{max}}]$ elements is considered. For each of its elements $E_{p,e}$, a local surrogate model is trained using the OLS on the observations located in the element to find the coefficients of the chaos series. As stated in 2.3, the optimal truncation degree of the polynomials model is obtained minimising PRESS errors. For every local surrogate model, regressions are carried out with a truncation degree from n_{min} to n_{max}. The selected truncation degree is the one minimizing the PRESS error $\delta_{p,e,n}$ of the considered local surrogate model $\tilde{f}_e$ trained with the observations of $E_{p,e}$ of the considered partition p with N_E elements with a truncation degree n :

$$\delta_{p,e} = \min_{n \in [n_{min}, n_{max}]} \delta_{p,e,n} \tag{29}$$

This error $\delta_{p,e}$ represents the accuracy of the local surrogate model $\tilde{f}_e$ trained on the observations located in $E_{p,e}$ and having an optimal degree between n_{min} and n_{max}. The error δ_p, image of the accuracy of the considered partition p with N_E elements, is defined as the maximal value of the errors $\delta_{p,e}$ of all the local models trained on its elements with an optimal degree :

$$\delta_p = \max_{e \in [1, N_E]} \delta_{p,e} \tag{30}$$

To get the optimal partition, we first consider the one with $N_{E_{min}}$ elements. Local models for every of its elements are trained with optimal degree. From one partition to the next one, only one element is split into two, going down the agglomerative clustering tree. In that way, only the local models on the two new elements can be trained at each step, the other ones having already be trained for previous partitions. During the different partitions of the space, a test can be added : if the selection error δ_p of the current tested partition is lower than a given threshold δ_{min}, a suitable partition is already found and there is no need to look for others with a lowest δ_p error. This test is referred as the early stop.

As stated above in section 2.3, one major advantage of using OLS in this approach is that the global design matrix $\mathbf{A}$, of size $(N \times M(n_{max}, d))$, is only computed once, at Step 2. When a regression on a element is wanted, only a sub-matrix composed of the lines corresponding to the observations located in the element and of the columns corresponding to the considered truncation degree (the first $M(n, d)$ columns for a given degree n) is used.

Algorithm 4 is the pseudo-code for Step 4.

Time complexity of this step is given by the construction of the agglomerative clustering with $N_{E_{min}}$ elements and by all the regressions. Ward clustering time complexity is in $\mathcal{O}(d \times N^3)$. In a case without early stop, $N_{E_{min}} + 2 \times \left(N_{E_{max}} - N_{E_{min}}\right)$ local models are trained. Every training needs $n_{max} - n_{min} + 1$ regressions to determine the local models with optimal degree, dominated by the one with $n = n_{max}$. Time complexity of all these regressions is then in $\mathcal{O}\Bigg(\left[N_{E_{min}} + 2 \times \left(N_{E_{max}} - N_{E_{min}}\right)\right] \times \Big(\max\left(N, M(n_{max}, d)\right) \times \min\left(N, M(n_{max}, d)\right)^2\Bigg)$.

Algorithm 4 Pseudo-code of Step 4

$d \leftarrow$ dimension of the parameter space
$N_{E_{min}} \leftarrow$ minimal number of elements
$N_{E_{max}} \leftarrow$ maximal number of elements
$N_p = N_{E_{max}} - N_{E_{min}} + 1$
$\delta_{min} \leftarrow$ minimal partition error for a potential early stop of the search of partitions
$\mathbf{B} \leftarrow$ values of the QOI on the training observations computed in Step 1
$\mathbf{A} \leftarrow$ global design matrix computed at Step 2
$\mathbf{D} \leftarrow$ derivatives of the QOI on the training observations computed at Step 3
$\mathcal{L}, \mathcal{E}_1, ..., \mathcal{E}_{N_p} \leftarrow$ Results of ward clustering with $N_{E_{min}}$ elements on the observations considered as points of $\mathcal{D}$ (using information of $\mathbf{B}$ and $\mathbf{D}$)
Initiate a variable δ_{opt} to a huge value
Initiate three void lists C, δ and *done_elements*
for $p \in [1, N_p]$ **do**
  $N_E = N_{E_{min}} + p - 1$
  for $e \in [1, N_E]$ **do**
    if the list e of $\mathcal{E}_p$, $E_{p,e}$, is not already in *done_elements* **then**
      $E_{p,e}$ is appended to *done_elements*
      $I \leftarrow$ indices of $\mathcal{L}$ where the items are equals to one of the labels contained in $E_{p,e}$
      $\mathbf{b} \leftarrow$ subvector of $\mathbf{B}$ with the rows of indices in I
      Initiate a void list c and initiate a vector $\tilde{\delta}$ of size $n_{max} - n_{min} + 1$ with zeros
      for $n \in [n_{min}, n_{max}]$ **do**
        $\mathbf{a} \leftarrow$ submatrix of $\mathbf{A}$ with the $M(n, d)$ first column and the rows of indices in I
        the result of the *OLS* with matrices $\mathbf{a}$ and $\mathbf{b}$ is appended to c and the related PRESS error is stored as $\tilde{\delta}_{n-n_{min}+1}$
      end for
      $n_{opt} \leftarrow \underset{n \in [n_{min}, n_{max}]}{\text{argmin}} \ \tilde{\delta}$
      $c_{n_{opt}-n_{min}+1}$ is appended to C
      $\delta_{n_{opt}-n_{min}+1}$ is appended to δ
    end if
  end for
  $\delta_p \leftarrow$ maximum of elements of δ corresponding to the elements of the partition p
  if $\delta_p \leq \delta_{opt}$ **then**
    $N_{E_{opt}} \leftarrow N_E$
    $\delta_{opt} = \delta_p$
    if $\delta_p \leq \delta_{min}$ **then**
      Break the loops
    end if
  end if
end for
the coefficients of the local models are the coefficients in C corresponding to the elements of the partition with $N_{E_{opt}}$ elements

3.6 Step 5 - Training of a classifier

Once the most suitable partition is obtained, with its set of N_E elements and N_E local surrogate models, a classifier is needed. Indeed, for further model evaluations, the first step is to determine in which element E_e the evaluation is, to know which local model to use to perform the evaluation. The use of a local model corresponding to a wrong element leads to huge errors, polynomial models being potentially highly

divergent far from their training observations. To model the action of the indicator function denoted $\mathcal{I}_{E_e}$ in formula (19), a classifier is used.

The user is free to choose any supervised classification method. In this article, Multi Layer Perceptron (MLP) classifier is chosen. MLP classifier is an artificial neural network used in a purpose of classifying observations labelled with class numbers. In Step 4, a label corresponding to the element in the partition with $N_E = N_{E_{max}}$ is given to every training observation. In Step 5, training observations are relabelled with this time a class number equal to the index of the element in the partition with N_E elements, N_E being the optimal number of elements found at the end of Step 4.

In all the below applications, 2 hidden layers of 50 neurons are taken. For the hidden layers, Rectified Linear Unit activation function is used. For the output layer, Softmax function is used. Cross-entropy cost function is minimised. The training of the classifier is done using backpropagation with Adam method [13] (with the classical decay values of $\beta_1 = 0.9$ and $\beta_2 = 0.999$, a mini-bacth size of 200 observations and a max number of iterations over the entire training set of $n_{epochs} = 200$ with an early stop if the weights have converged). Several classifiers are trained with different learning rates. The training which maximises the number of well classified training observations, giving the best accuracy being the fraction of well classified observations. In the article, a naive method testing all N_l learning rates is used. No regularisation term is added as classification overfitting is not a problem in the approach (all the training observations have to be well-classified to avoid using a wrong local model).

To the author knowledge, time complexity of one optimisation step with feedforward, backpropagation and Adam bias momentum correction is in $\mathcal{O}\Big((d \times n_{neurons} + n_{neurons}^2 + n_{neurons} \times N_E) \times N\Big)$, with $n_{neurons}$ being the number of neurons in the hidden layers (50 in our case). In that case, the learning of one classifier is in in $\mathcal{O}(n_{epochs} \times n_{neurons}^2 \times N_{train})$, where $n_{epochs} = 200$ in the worst case is the number of steps of optimisation going on all the training set. So the learning of all classifiers is in $\mathcal{O}(N_l \times n_{epochs} \times n_{neurons}^2 \times N)$.

The input of the classifier is a given point in $\mathcal{X}$ (a set of value of input parameters) which need to be evaluated by the piecewise model. This point is denoted the evaluation point. The output is an array of N_E components containing the probabilities that this evaluation point is located in the element E_e. Therefore, a test can be added to check if the evaluation point lies in an indecision zone : if the highest probability is lower than θ, $\theta \in [0, 1]$ being a threshold to choose, the evaluation point is considered as located in an indecision zone. In this case, the point is not evaluated by the model and is ignored. θ is a trade-off parameter. The greater θ is, the more prudence is taken, but the more evaluation points close to the boundaries of the elements are ignored. With $\theta = 0$, no evaluation point is ignored but the ones located in the indecision zone are going to alter the accuracy of the model. By increasing θ, an evaluation point is more likely to be ignored if it is potentially affecting the accuracy. After a given value, rising θ induces that some evaluation points are ignored even if they are not really altering the accuracy. In [10], the minimal multi-element method presents a similar procedure, using the resolution of the sampling strategy to ignore points lying within a distance of the boundary lower than this resolution. θ is a parameter which can be tuned after the training of the model.

3.7 Summary of ME-ACD algorithm

The numerical parameters and hyperparameters of the algorithm are :

- N : the number of observations generated in the Sobol sequence. It determines the size of the training set.
- $N_{E_{min}}$: the minimal number of wanted elements.

- $N_{E_{max}}$: the maximal number of wanted elements.
- $N_{neigh-min}$: the minimal number of neighbours in $\mathcal{F}$ used in the estimation of the derivatives.
- k_{max} : the maximal number of neighbours in $\mathcal{F}$ used in the estimation of the derivatives.
- n_{min} : the minimum chaos degree a local surrogate model can have
- n_{max} : the maximum chaos degree a local surrogate model can have.
- δ_{min} : an optional threshold to early stop the search of partition if one suitable partition has already been found.
- A set of N_l learning rates to get the optimal classifier.
- $isConnect$: a boolean indicating if a connectivity diagram is used.
- (Optional) $k_{connect}$: number of connected neighbours an observation have in the optional connectivity diagram

For the classifier, hyperparameters are given in section 3.6. The caution threshold $\theta \in [0, 1]$ of the classifier is a post-processing parameter. Once the model is obtained with its attached classifier, the user is still free to tune θ without launching the training once again.

For agglomerative clustering, research of the nearest neighbours and MLP classifiers, the Scikit-learn toolkit has been used [18].

In term of time complexity :

- Step 1 : $\mathcal{O}(N \times d)$ + $N\times$ time complexity for computing a value of f
- Step 2 : $\mathcal{O}\Big(N \times M(n_{max}, d) \times d\Big)$
- Step 3 : $\mathcal{O}(N \times d^4)$ (if k_{min} and k_{max} are set to be proportional to d)
- Step 4 : $\mathcal{O}\left(\left[N_{E_{min}} + 2 \times \left(N_{E_{max}} - N_{E_{min}}\right)\right] \times \left(\max\left(N, M(n_{max}, d)\right) \times \min\left(N, M(n_{max}, d)\right)^2\right)\right)$
- Step 5 : $\mathcal{O}(N_l \times n_{epochs} \times n_{neurons}^2 \times N)$

The dominant step is dependent of the choice of the hyperparameters and of the dimension of the QOI.

4 Applications

In this section, performances of ME-ACD are demonstrated on a set of test functions. To evaluate these performances, the possible causes of errors between the piecewise surrogate model and the QOI it approximates have to be identified. These errors can come from 3 phenomenons :

- **Underfitting (or bias) :** the optimal degrees of the local polynomials models are too low to approximate the QOI on the elements. Increasing N, n_{max} and/or $N_{E_{max}}$ could solve the issue.
- **Overfitting (or variance) :** as the partitions of the parameter space and the optimal degrees of the local models are selected with the PRESS error, overfitting is not likely to occur. If it still occurs, lowering n_{min} and/or increasing $N_{E_{max}}$ could solve the issue.

- **Misclassification :** as the polynomial models are highly divergent far from their training observations, if the evaluation point is misclassified in a wrong element, the wrong local model is used and the accuracy is highly altered. As a reminder, the caution threshold θ is here to avoid these errors, ignoring the point when the probability of being in the element given by the classifier is strictly lower than it. Increasing N or adding training observations close to the boundaries of the elements and/or modifying the classifier parameters could solve the issue. Increasing θ makes the algorithm stricter and makes it ignoring more evaluation points which are more likely to be responsible of misclassification errors.

For the classifier, the learning rate is the one within $[10^{-m}, 2 \times 10^{-m}, 5 \times 10^{-m}]$, $\forall m \in [1, 7]$ given the classifier maximising the accuracy on the training set itself (the optimal classifier between $N_l = 21$ is therefore selected). The threshold is set to $\theta = 0.85$, which means that the evaluation point is not ignored by the caution procedure if the classifier is sure at more than 85% that it is located in a given element.

The performances of the algorithm are compared to the ones of the standard PC model of optimal degrees between a given n_{min-PC} and n_{max-PC} and of Gradient Boosted Trees (GBT) algorithm trained on the same test observations. In fact, in the applications, the standard PC model is obtained with the ME-ACD method with $N_{E_{min}} = N_{E_{max}} = 1$. To set the 3 hyperparameters of GBT : the learning rate, the number of trees and the depth of the trees ; a validation set of $N_{val} = 10^5$ observations for every training set is sampled. A brute force search is realised for every training set : every combination of hyperparameters is tried to train a GBT model and the one minimizing the MSE between the true value of the QOI and the value of the given GBT model on the validation observations is chosen. In the applications, the learning rates is located in $[10^{-m}, 2 \times 10^{-m}, 5 \times 10^{-m}]$, $\forall m \in [1, 7]$, the number of trees in $[100, 200, 300, 400, 500]$ and the maximum depth of the trees in $[2, 3, 4, 5]$.

To evaluate the precision of the algorithms, a test set of $N_{test} = 10^5$ observations is randomly sampled. This test set is different from the validation ones used to tune the GBT models, but remains the same for a given convergence study. Conversely, the validation sets used to tune the hyperparameters of GBT models is different for every different training set (for every different N of the convergence study).

The precision of each model is given by E, the MSE between the true value of the QOI and the value given by the predictions of the model on the test set. To see the impact of the caution parameter, E_θ is also studied, which is the same error but evaluated with the N_θ test observations kept by the caution procedure of the ME-ACD model with $\theta = 85\%$. The value $V_{\theta=0.85} = \frac{N_{\theta=0.85}}{N_{test}}$, fraction of the kept test observations by the caution procedure, is given for each N of the convergence study.

4.1 Discontinuous functions

In this section, a set of 3 different discontinuous functions taking $d = 2$ input parameters are studied :

$$\tilde{f}^{hyperplane} = \begin{cases} 1 & \text{if } x_1 + 0.3x_2 - 0.5 > 0 \\ -1 & \text{otherwise} \end{cases} \tag{31}$$

This function presents a discontinuity which is a line of equation $x_1 + 0.3x_2 - 0.5 = 0$.

$$\tilde{f}^{hypersphere} = \begin{cases} 1 & \text{if } (x_1 - 0.5)^2 + (x_2 - 0.5)^2 < (0.25)^2 \\ -1 & \text{otherwise} \end{cases} \tag{32}$$

This function presents a discontinuity which is a circle of center $(0.5, 0.5)$ and of radius 0.25.

$$\tilde{f}^{multiple} = \begin{cases} 3 & \text{if } (x_1)^2 + (x_2)^2 < (0.25)^2 \\ 2 & \text{else if } x_1 < 0.5 - 0.3x_2 \\ 1 & \text{else if } (x_1 - 1)^2 + (x_1 - 1)^2 > 0.3^2 \\ 0 & \text{otherwise} \end{cases} \tag{33}$$

This function presents 3 discontinuities, two fractions of circles in the corners of the unit hypercube and the same line as in $\tilde{f}^{hyperplane}$.

For these test functions, optimal elements are clearly defined. In that way, a true label can be provided indicating in which element a test observation is located. With these labels, the accuracy of the classifier, denoted F and representing the fraction of well-classified test observations, can be computed. For the next sections, as the Multi-element procedure is also used to study continuous irregular behaviour with polynomials series of fewer degrees, the optimal partition is not clearly defined and the computation of this accuracy is not possible.

Figure 5 shows 1000 training observations labelled by the clustering step of ME-ACD and represented in the parameter space for the 3 test functions. This first visualisation highlights that in that dimension, ME-ACD method is able to detect discontinuities and to partition the parameter space according to these discontinuities.

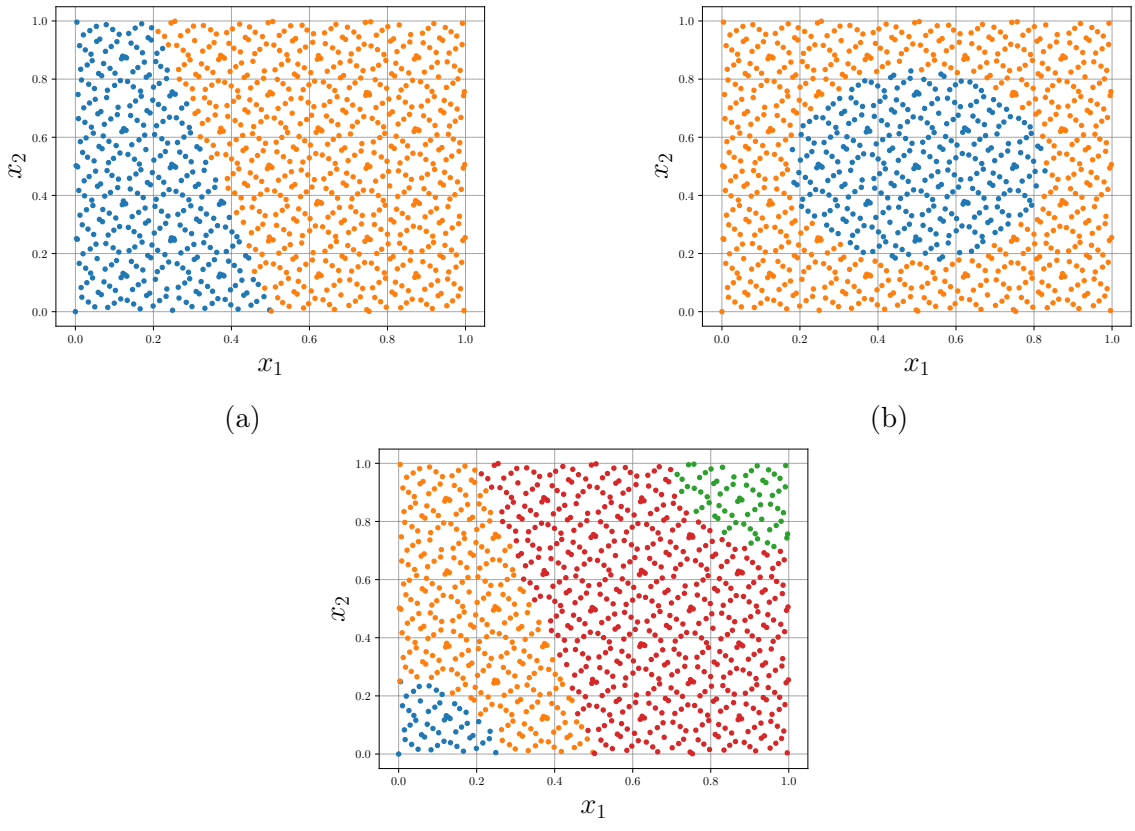


Figure 5: Visualisation of 1000 labelled training observations represented in the $2D$ parameter space $\mathcal{X}$ partitioned for the 3 test functions. Different colors stands for different elements obtained with the clustering. (a) Visualisation of $\tilde{f}^{hyperplane}$. (a) Visualisation of $\tilde{f}^{hypersphere}$. (a) Visualisation of $\tilde{f}^{multiple}$.

Figure 6 gathers all convergence plots of the mean square error E, of the accuracy F of the ME-ACD classifier and of the fraction of kept observations by the caution procedure of ME-ACD on the 3 test

	PC	ME-ACD
N	$[100, 2000]$	
$N_{E_{min}}$	1	1
$N_{E_{max}}$	1	10
k_{min}	not relevant	6
k_{max}	not relevant	10
n_{min}	0	0
n_{max}	20	0
δ_{min}	10^{-5}	10^{-5}
learning rate	see section 4	
$isConnect$	False	False
θ	not relevant	0.85

Table 1: Hyperparameters and parameters selected for ME-ACD and PC for the convergence study of $\tilde{f}^{hyperplane}$, $\tilde{f}^{hypersphere}$ and $\tilde{f}^{multiple}$.

functions on the test set. Table 1 contains the hyperparameters and the parameters selected for PC and for ME-ACD for these convergence studies. Hyperparameters for GBT are the best from the brute-force search described above in section 4 for every N of the convergence studies.

The convergence plots of E (Figures 6a, Figure 6c and Figure 6e) highlight that ME-ACD method is able to give a more accurate model than the 2 other methods, but error is still relatively high. As the functions are piecewise constant and the model is also piecewise constant(as $n_{min} = n_{max} = 0$), the cause of the errors for the ME model are missclassification errors. As F is close to 99% for every N, around 1% of the observations are responsible for all the error. They are the observations located close to the discontinuities.

Figure 7 shows this time a convergence plot of the MSE on the N_θ test observations kept by the caution procedure for every N. With $\theta = 85\%$, V_θ is around 90% for every N (shown by Figure 6b, Figure 6d and Figure 6f). In most of the case, all the problematic observations are removed by the caution procedure, leading to errors of magnitude 10^{-30}, stating that no remaining observation is missclassified, or leading to errors of magnitude 10^{-5}, highlighting than only around 10 of the $N_\theta \approx 10^4$ kept test observations are missclassified. With $\theta = 85\%$, around 9% of the test observations are ignored despite the fact they are not leading to any error. The caution procedure is then able to detect the problematic observations and the user is free to act consequently to avoid these missclassification errors.

Even better classifier results (F and V_θ) could have been obtained with another sampling. Here, Sobol sampling is used to model a general case, leading to a homogeneous sampling no matter the potential discontinuities. In [9], an adaptive sampling is developed. This sampling is able to get more observations close to discontinuities. If the user is free to sample the parameter space, more observations close to the discontinuities should lead to an even better accuracy of the classifier and to an even more efficient caution procedure.

Figure 8 shows the optimal degrees for the PC models in the convergence studies for the 3 test functions. The corresponding number of terms of the series is also shown on the left in parenthesis. With increasing number of training observations, PC is able to select higher optimal degrees without suffering from overfitting. Compared to these PC models, local polynomials models of the ME method have only one term and lead to a better accuracy. The economy in number of terms of the PC series induced by using lower degrees local model would be even more marked at higher dimensions.

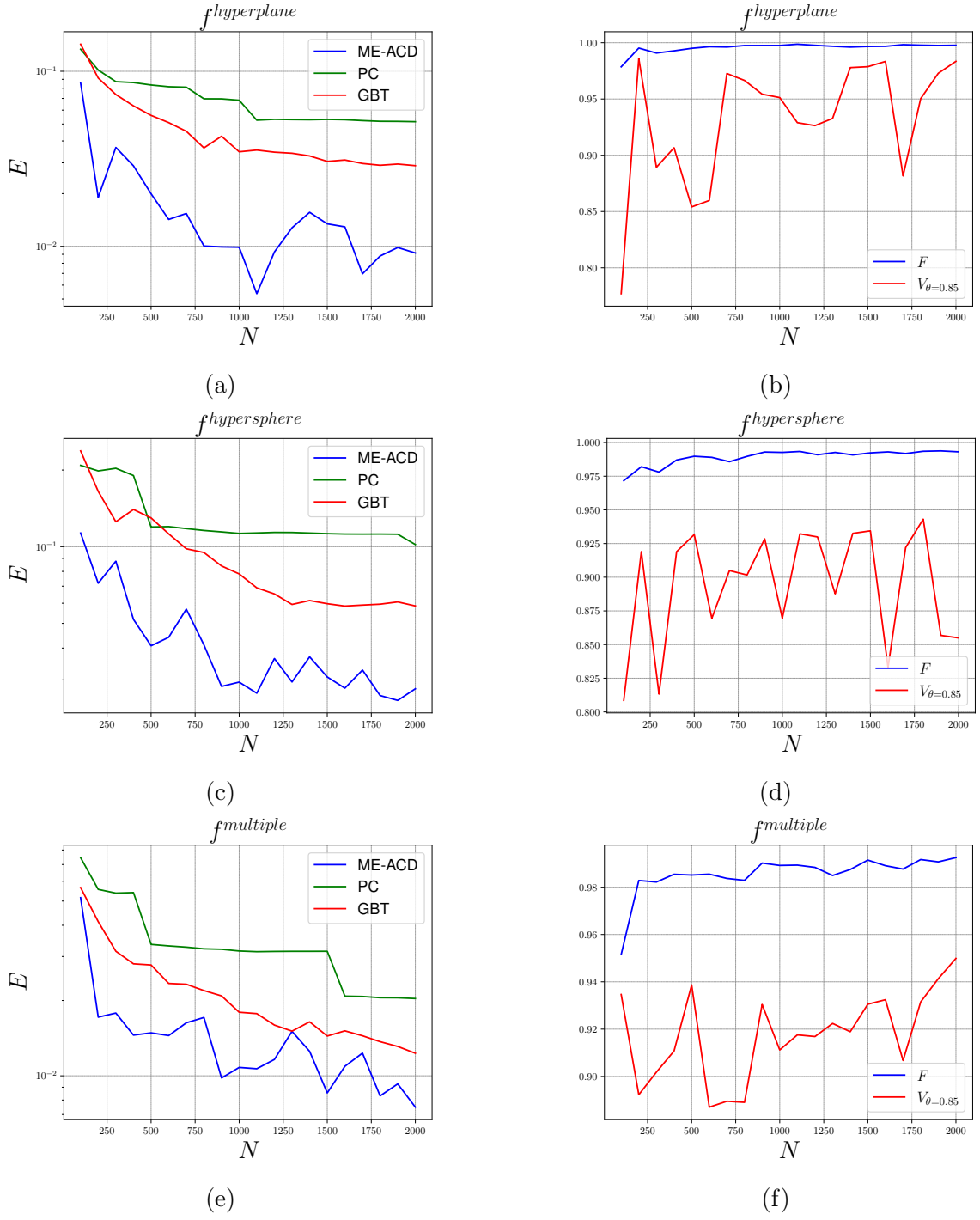


Figure 6: On the left, convergence plots of the MSE between the values of the models and the true value of the 3 discontinuous functions on the test set. (a) $\tilde{f}^{hyperplane}$. (c) $\tilde{f}^{hypersphere}$. (e) $\tilde{f}^{multiple}$. On the right, convergence plots of the accuracy of the classifier of ME-ACD and of the kept fraction of observations by the caution procedure of the ME-ACD, on the test set. (b) $\tilde{f}^{hyperplane}$. (d) $\tilde{f}^{hypersphere}$. (f) $\tilde{f}^{multiple}$.

4.2 Discontinuity which runs partially through the parameter space

In this subsection, the following QOI is studied :

$$\tilde{f}^{partial} = \begin{cases} 0.5x_2 - 0.25 & \text{if } x_1 >= 0.5 \\ -0.5x_2 + 0.25 & \text{otherwise} \end{cases} \tag{34}$$

This piecewise function presents a discontinuity which "runs partially" through the parameter space. The function is linear on each side of the hyperplane $x_1 = 0.5$ and so has the geometry of an hyperplane.

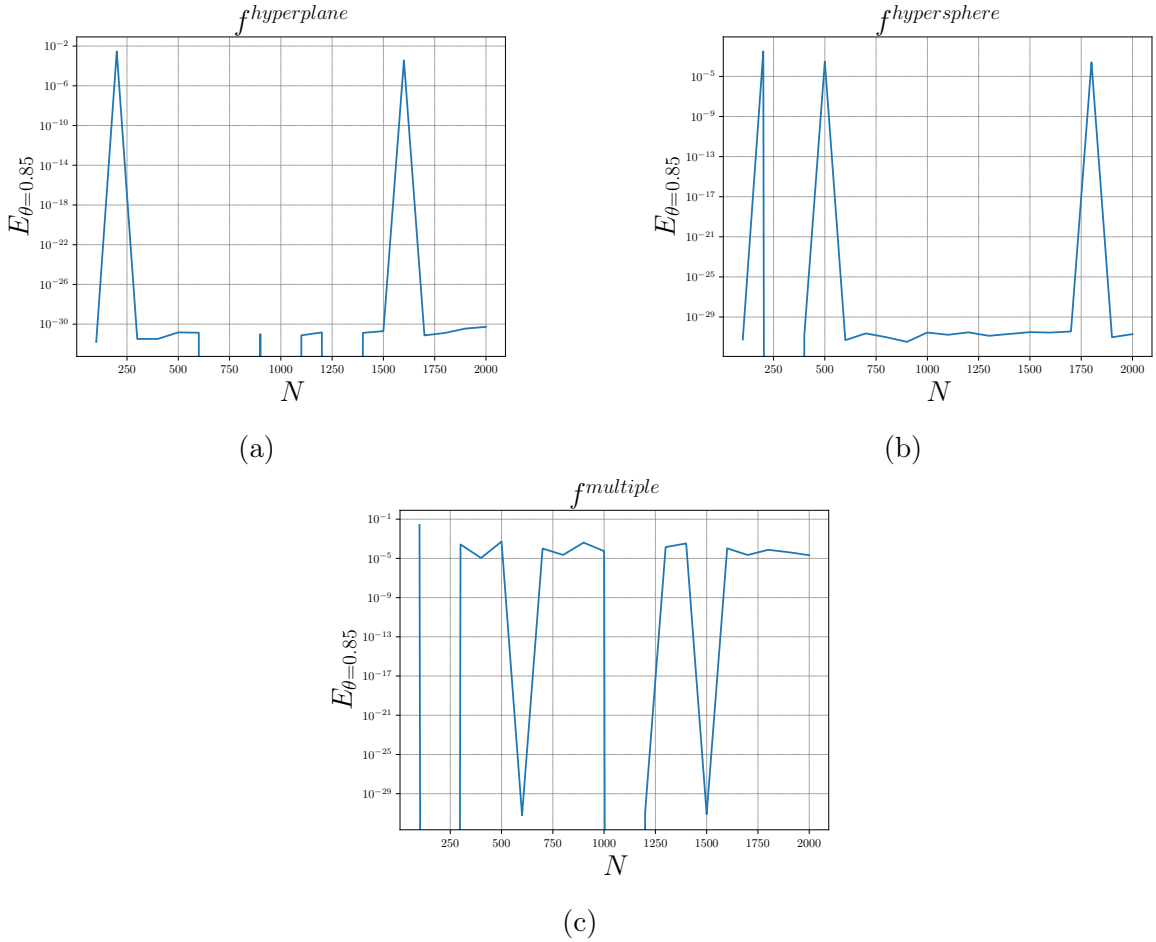


Figure 7: Convergence plots of the MSE between the values of the models and the true value, on the $N_{\theta=0.85}$ of the of $N_{test} = 10^5$ tests observations kept by the caution procedure set observations at each N. (a) Convergence plot of the MSE for $\tilde{f}^{hyperplane}$ on the kept observations. (b) Convergence plot of the MSE for $\tilde{f}^{hypersphere}$ on the kept observations. (c) Convergence plot of the MSE for $\tilde{f}^{multiple}$ on the kept observations.

	PC	ME-ACD
N	$[100, 2000]$	
$N_{E_{min}}$	1	1
$N_{E_{max}}$	1	10
k_{min}	not relevant	6
k_{max}	not relevant	10
n_{min}	0	0
n_{max}	20	10
δ_{min}	10^{-5}	10^{-5}
learning rate	see section 4	
$isConnect$	False	False
θ	not relevant	0.85

Table 2: Hyperparameters and parameters selected for ME-ACD and PC for the convergence study of $\tilde{f}^{partial}$.

However, when x_2 becomes close to 0.5, the gap between the two surfaces is so thin that numerically, it becomes more and more regular (although the discontinuity is still here mathematically). To the author knowledge, none of the previous multi-element methods permits to split the parameter space in presence

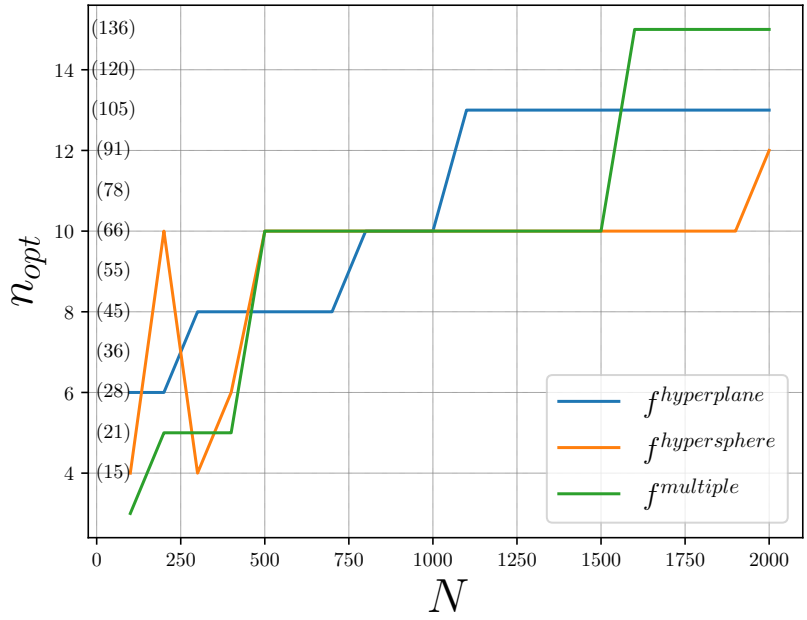


Figure 8: Evolution of the optimal degrees of PC models in the convergence studies for the 3 disconitnuous functions $\tilde{f}^{hyperplane}$, $\tilde{f}^{hypersphere}$ and $\tilde{f}^{multiple}$. The corresponding number of terms of the series $M(n, d = 2)$ is shown on the left in parenthesis.

of a discontinuity which runs partially through it. As stated in [9], this is a problematic for some of the modern methods in the multi-element domain.

Figure 9, containing results obtained with the parameters and hyperparameters gathered in Table 2, shows similar results for the convergence study of this function as for the discontinuous functions presented at the previous subsection. ME-ACD is then able to handle discontinuities running partially through the parameter space in the same way as any other discontinuity. Conversely to polynomial annihilation used in [1, 10], agglomerative clustering of the observations in space $\mathcal{D}$ is not affected by thinner gap close to the boundaries of elements.

4.3 Bifurcation function

One major advantage of ME-ACD is that it is able to accurately model continuous regular functions, continuous irregular functions and discontinuous functions. The adaptation of ME-ACD to model solutions of Ordinary Differential Equation (ODE) is under development. This adaptation will then be suitable to approximate solutions of ODE with the apparition of bifurcations (thus leading to irregularities and discontinuities with respect to the input parameters).

In this subsection, the following QOI is studied :

$$\tilde{f}^{bif-t}(x_1, x_2) = (1 + x_2)\tanh(\frac{1}{2}(3x_1 - 1)(1 + x_2)t) \tag{35}$$

this function is the solution at time t of the following ODE :

$$\dot{y} = -p_1(y^2 - p_2^2) \tag{36}$$

with $y(t = 0) = 0$, p_1 following an uniform distribution law on $[-\frac{1}{2}, 1]$ and p_2 following an uniform distribution law on $[1, 2]$. The transformation to get the corresponding $\mathbf{x}$ appears in the form of the mapped

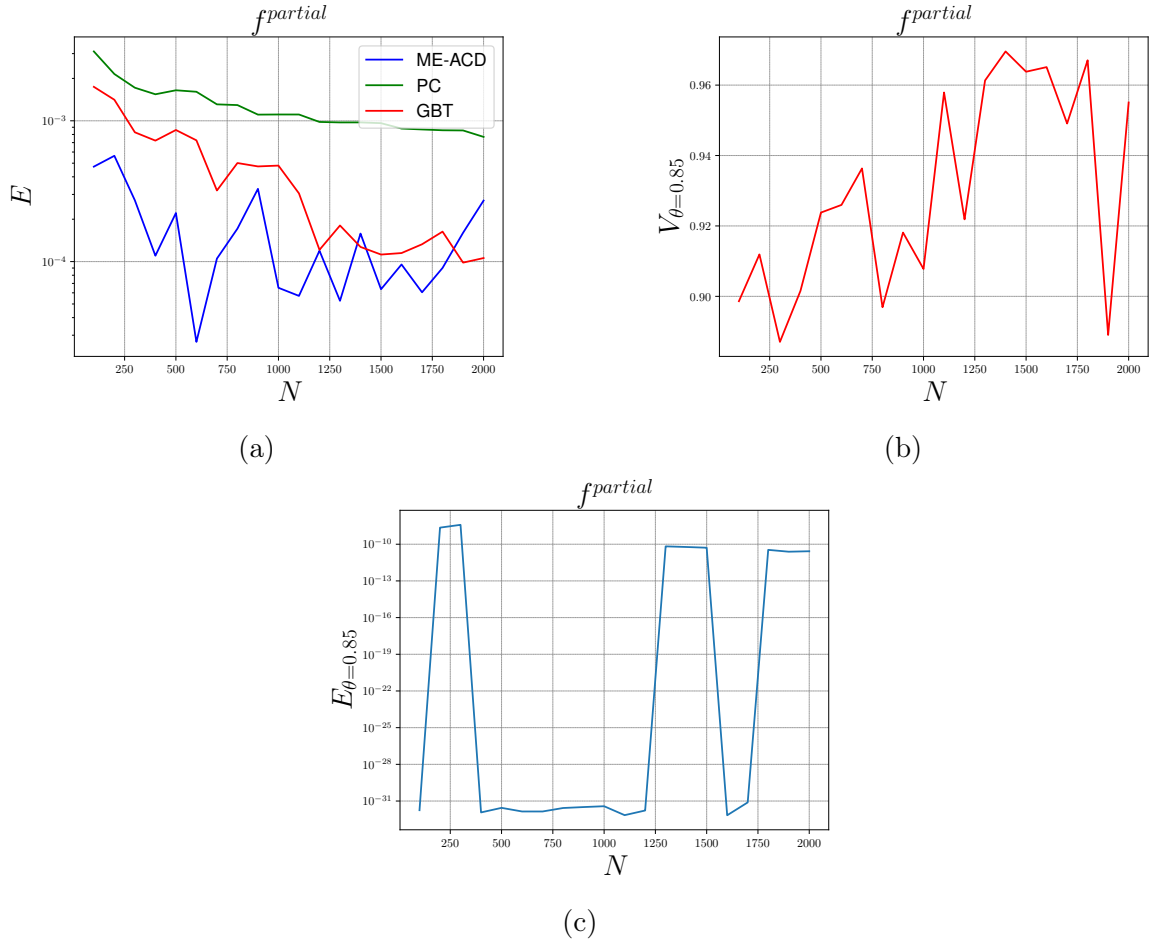


Figure 9: (a) Convergence plot of the MSE for $\tilde{f}^{partial}$ on its test set. (b) Convergence plot of the accuracy of the ME-ACD classifier and fraction of kept observations by the caution procedure of ME-ACD for $\tilde{f}^{partial}$ on its test set. (c) Convergence plot of the MSE between the values of the models and the true value on the $N_{\theta=0.85}$ tests observations kept by the caution procedure at each N for $\tilde{f}^{partial}$.

function. This system models the apparition of a bifurcation in a system. The two equilibria of the system (if we ignore the punctual case $p_1 = 0$) are $-p_2$, stable if $p_1 < 0$ and p_2, stable if $p_1 > 0$.

At the first instant, the solution is regular in function of $\mathbf{x}$. When t rises, the solution is more and more irregular with respect to $\mathbf{x}$. Asymptotically ($t \to \infty$), the solution presents a discontinuity which is a line of equation $p_1 = 0$ ($x_1 = \frac{1}{3}$), symbol of a bifurcation with respect to p_1 (x_1).

The 3 considered instants are : $t = 1$ (regular), $t = 10$ (irregular) and $t = 1000$ (discontinuous). Figure 10 shows the variation of the function with x_1 for $x_2 = 1$ (the value of x_2 leading to the widest gap).

Similarly to previous sections, Figure 11 shows the results of convergence studies of the MSE on all the test set and of the kept fraction of test observations by the caution procedure of ME-ACD. Figure 12 underlines the convergence of the MSE $E_{\theta=0.85}$ on the kept observations. The hyperparameters and the parameters used for the functions in these studies are given by Table 3.

At $t = 1$, the early stop of ME-ACD makes the algorithm not using the ME procedure, thus being equivalent to PC, but with a lower n_{max}. PC is then outmatching ME-ACD is this case. With a similar n_{max}, both methods would have been identical. As $N_{E_{opt}} = 1$ for all N, 100% of the test set is always kept by the caution procedure.

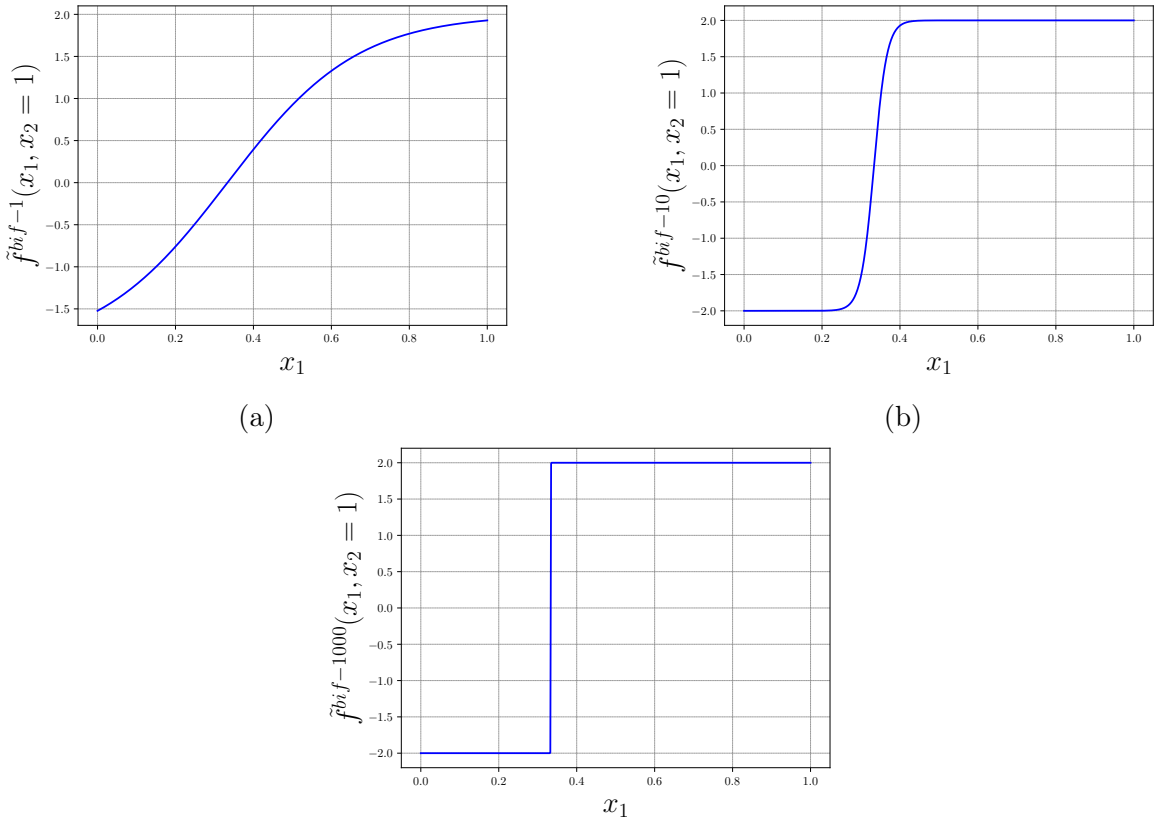


Figure 10: Plot of $\tilde{f}^{bif-t}(x_1, x_2 = 1)$ vs x_1 at 3 different instants : $t = 1$ (regular), $t = 10$ (irregular) and $t = 1000$ (discontinuous).

	PC	ME-ACD
N	$[100, 2000]$	
$N_{E_{min}}$	1	1
$N_{E_{max}}$	1	10
k_{min}	not relevant	6
k_{max}	not relevant	10
n_{min}	0	0
n_{max}	20	10
δ_{min}	10^{-5}	10^{-5}
learning rate	see section 4	
$isConnect$	False	True
$k_{connect}$	not relevant	10
θ	not relevant	0.85

Table 3: Hyperparameters and parameters selected for ME-ACD and PC for the convergence study of $\tilde{f}^{bif-1}$, $\tilde{f}^{bif-10}$ and $\tilde{f}^{bif-1000}$.

At $t = 10$, ME-ACD split the parameter space into 3 or more elements to use local models of lower degrees. ME-ACD is able to accurately give an approximation of a continuous irregular QOI using lower degrees local models. As shown in Figure 12b, the caution procedure permits to have an even more accurate model, but as the QOI is not discontinuous, missclassification errors are less altering the accuracy.

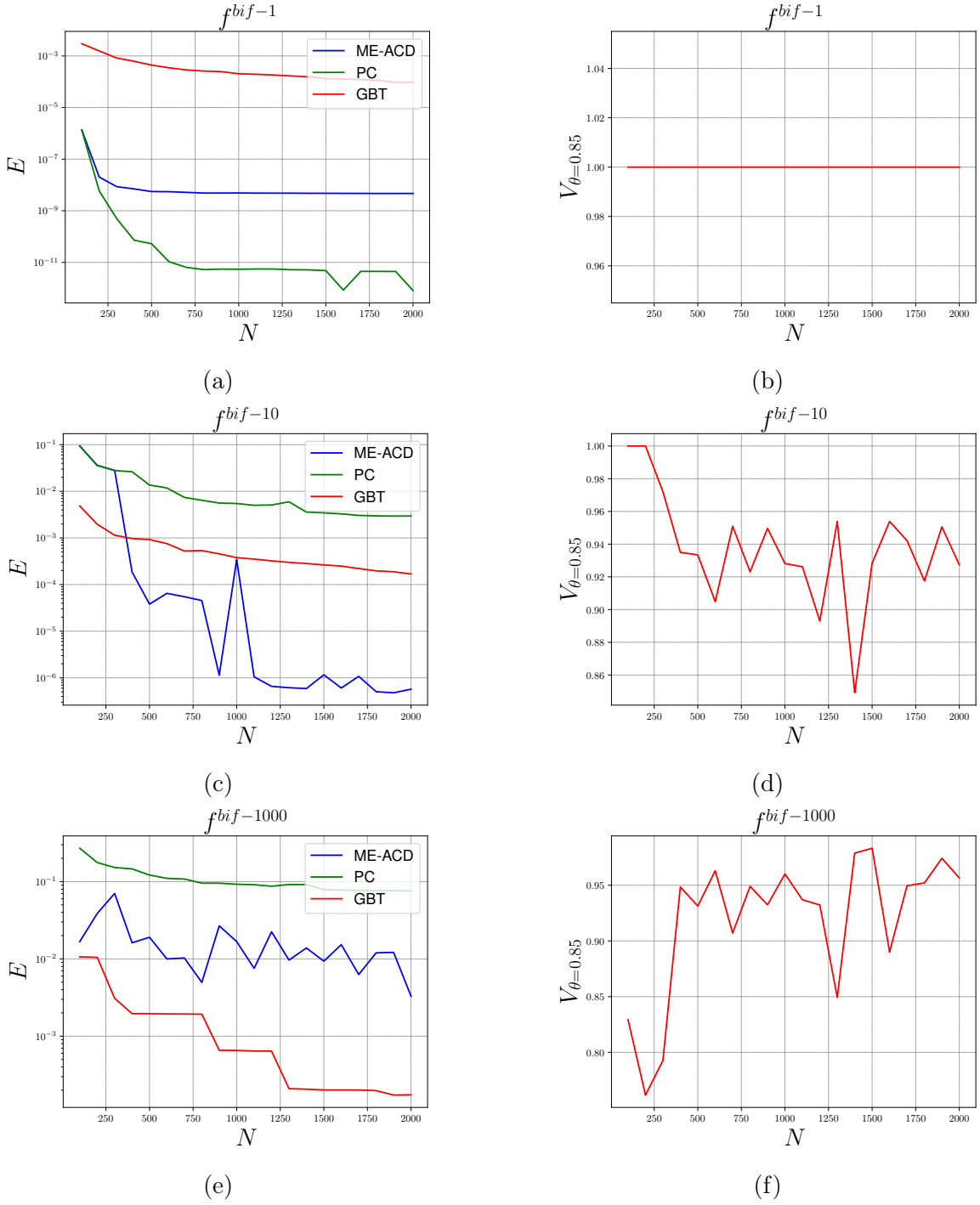


Figure 11: On the left, convergence plots of the MSE between the values of the models and the true value of the 3 discontinuous functions on the test set. (a) $\tilde{f}^{bif-1}$. (c) $\tilde{f}^{bif-10}$. (e) $\tilde{f}^{bif-1000}$. On the right, convergence plots of the kept fraction of observations by the caution procedure of the ME-ACD on the test set. (b) $\tilde{f}^{bif-1}$. (d) $\tilde{f}^{bif-10}$. (f) $\tilde{f}^{bif-1000}$.

At $t = 1000$, ME-ACD is behaving quasi-identically to the case of $\tilde{f}^{hyperplane}$. Only a few training observations close to the quasi-discontinuity have a QOI value different from p_2 or $-p_2$. These training observations alters the accuracy of the local models, but there are not enough of them to make a third element. If the caution procedure enables to get rid of a lot of missclassification errors and getting a relatively accurate model, the problematic training observations alter the accuracy of the local models. This explains why the error is decreasing very slowly, as Sobol sampling does not have high chances to sample new training observations in the very thin quasi-discontinuity zone. With a low N, no observation

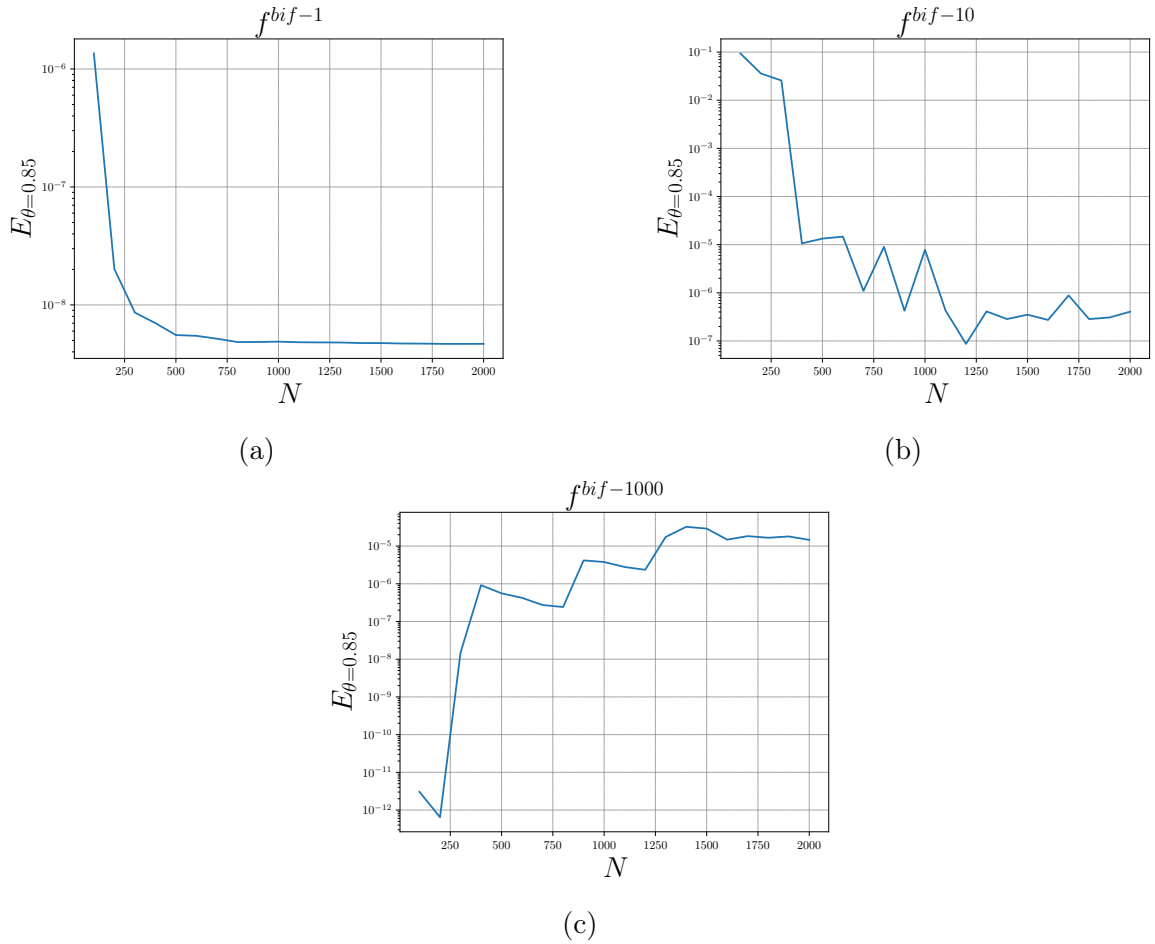


Figure 12: Convergence plots of the MSE between the values of the models and the true value of the 3 functions, on the $N_{\theta=0.85}$ of the of $N_{test} = 10^5$ tests observations kept by the caution procedure set observations at each N. (a) Convergence plot of the MSE for $\tilde{f}^{bif-1}$ on the kept observations. (b) Convergence plot of the MSE for $\tilde{f}^{bif-10}$ on the kept observations. (c) Convergence plot of the MSE for $\tilde{f}^{bif-1000}$ on the kept observations.

is located in this zone, and a 2 elements model with two hyperplanes is given by ME-ACD, explaining why the accuracy is better with a low N. Generating more observations close to the quasi-discontinuity with an adaptive sampling or ignoring these points seeing them at outliers should solve the issue.

ME-ACD is then able to give an accurate model in the three cases of a regular QOI, a continuous irregular QOI and a discontinuous QOI, which makes it a model able to approximate a lot of QOIs and a good candidate to approximate solutions of ODE with potential bifurcations.

4.4 Genz function in dimension 5

In this section, a 5−dimensional function a QOI from the Genz family [7], depending on $d = 5$ input parameters is studied :

$$\tilde{f}^{genz}(\mathbf{x}) = \begin{cases} 0 & \text{if one of the } x_i > 0.5 \\ \exp(\sum_{i=1}^{5} x_i) & \text{otherwise} \end{cases} \tag{37}$$

Table 4 gathers all selected hyperparameters and parameters, and Table 5 shows the performances of the 3 methods with $N = 50000$ training observations. Once again, ME-ACD is able to give a relatively accurate model and is able to reduce missclassification errors with the caution procedure, ignoring test observations located in the indecision zone. For this application, ME-ACD and PC use the same global

	PC	ME-ACD
N	50000	
$N_{E_{min}}$	1	1
$N_{E_{max}}$	1	5
k_{min}	not relevant	15
k_{max}	not relevant	25
n_{min}	0	0
n_{max}	10	10
δ_{min}	10^{-5}	10^{-5}
learning rate	see section 4	
$isConnect$	not relevant	True
$k_{connect}$	not relevant	30
θ	not relevant	0.85

Table 4: Hyperparameters and parameters selected for ME-ACD and PC for the convergence study of $\tilde{f}^{genz}$

	PC	GBT	ME-ACD
E	0.15415751	0.01737417	0.02062906
$E_{\theta=0.85}$	not relevant		0.00182077
$V_{\theta=0.85}$	not relevant		99.651%

Table 5: Performances of the 3 methods for the $5-d$ QOI f^{genz} trained with $N = 50000$ observations. Performances are evaluated on a test set of $N_{test} = 10^5$ test observations.

design matrix having $M(n_{max} = 10, d = 5) = 3003$ columns, but ME-ACD leads to a far better accuracy with its ME procedure. Optimal degree of PC is equal to 9.

5 Conclusion and discussions

A new NI-ME method, ME-ACD, is developed in this paper. This method is able to efficiently partition the parameter space in order to construct a piecewise chaos polynomial model in the case of a discontinuous QOI, as the methods relying on Bayesian inference [22] or polynomial annihilation edge detection [10]. ME-ACD is also able to partition the space in presence of a discontinuity which runs partially through the parameter space, which was not possible with previous methods to the author knowledge. Moreover, the method permits to study a continuous but irregular QOI, which needs a lot of chaos modes to be accurately approximate by the standard PC models. ME-ACD model is composed of local models of lower degrees, taking the same logic as ME-PCM [5] but with a partition following the irregular behaviours. The method has been successfully tested on several discontinuous and continuous irregular QOIs from $d = 2$ to $d = 5$. From the results of the paper, the method is general and is very promising to approximate a large set of QOIs. The method will be tested on other kind of theoretical QOIs and on practical applications in the near future. On the majority of the discontinuous and irregular applications presented in the paper, ME-ACD models are more accurate than standard PC and GBT models. Moreover, ME-ACD can lead to physical interpretation, partitioning the space into elements which could be interpreted. With the use of a neural network classifier, the model is also able to determine which evaluation lie in a zone in between the given elements, and is so able to be cautious with these points, ignoring them instead of giving a value which risks to be inaccurate due to misclassification errors. ME-ACD has been developed in the context of UQ, but as other NI PC methods, it can be used in another context as a new regression method, with inputs which are not representing uncertaintites.

ME-ACD presents limits which will be the subject of future researches. The partition performed by the ME-ACD algorithm is sensitive to the chosen hyperparameters and to the sampling strategy. In that way, the partition and the classification may lack of robustness according to the studied QOI and to the chosen sampling. One of the aims of ME-ACD is to work on every sampling. However, ME method may fail to give a suitable partition leading to an accurate model with the chosen sampling and/or hyperparameters. According to the QOI, some zones of the parameter space may need more observations than others. With a general sampling, sparsity can lead to selection of nearest neighbours from different sides of the discontinuity, leading to poorly accurate gradient estimation. Moreover, sparsity in some zones can lead to observations which are isolated from the others in $\mathcal{D}$, not being enough to constitute a proper element with an accurate local model, like in the case of the QOI $\tilde{f}^{bif-1000}$ studied in section 4.3, and therefore spoiling the accuracy of another element. Finally, this sparsity can alter the accuracy of the classification step, leading to missclassification errors undetected by the caution procedure. For now, even if these drawbacks were present, ME-ACD have been tested successfully on theoretical QOIs depending on a low or moderate number of input parameters, up to $d = 5$. For higher dimensions, the notion of nearest neighbours which is used for the gradient calculus and for the connectivity diagram, might become less "meaningful", as highlighted in [3]. ME-ACD is therefore expected to be even less robust as the dimension increases. Moreover, as the number of input parameters rises, the value of the QOI, which is one of the $d + 1$ considered coordinate in $\mathcal{F}$ or in $\mathcal{D}$, is expected to have less impact in the selection of the nearest neighbours and in the agglomerative clustering step. The risk of selecting nearest neighbours from different sides of a discontinuity is therefore expected to increase, altering the calculus of the derivatives and then of the partition. First tests have been made with QOIs depending on $d = 9$ input parameters. The partition step worked on a simple constant piecewise QOI with $d = 9$, but failed with a more complex one where an element represented only 0.2% of the parameter space. These first tests showed that the method is promising with dimensions $d \in [5, 10]$, even if the lack of robustness persists and seems exacerbated. Further works will therefore consist in adapting the method for higher dimensions and finding a sampling strategy suitable to make the algorithm more robust, if the user has control over sampling. With such improvements, the use of local models with lower degrees would be promising to approximate higher dimensionality examples ($d > 5$). Finally, ME-ACD is also going to be adapted to approximate time-dependent functions being the solution of an ODE.

6 CRediT authorship contribution statement

Nicolas Vauchel: Writing, investigation, reviewing and editing.
Éric Garnier: Supervision, Rewiewing.
Thomas Gomez: Supervision, Rewiewing.

7 Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

8 Acknowledgements

The authors wish to thank ONERA and Hauts-de-France region for their funding.

References

[1] Rick Archibald, Anne Gelb, and Jungho Yoon. Polynomial fitting for edge detection in irregularly sampled signals and images. *SIAM journal on numerical analysis*, 43(1):259–279, 2005.

[2] Marc Berveiller, Bruno Sudret, and Maurice Lemaire. Stochastic finite element: a non intrusive approach by regression. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, 15(1-3):81–92, 2006.

[3] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In *International conference on database theory*, pages 217–235. Springer, 1999.

[4] Géraud Blatman and Bruno Sudret. Adaptive sparse polynomial chaos expansion based on least angle regression. *Journal of Computational Physics*, 230(6):2345–2367, 2011.

[5] Jasmine Foo, Xiaoliang Wan, and George Em Karniadakis. The multi-element probabilistic collocation method (me-pcm): Error analysis and applications. *Journal of Computational Physics*, 227(22):9572–9595, 2008.

[6] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[7] Alan Genz. Testing Multidimensional Integration Routines. In *Proc. Of International Conference on Tools, Methods and Languages for Scientific and Engineering Computation*, pages 81–94, New York, NY, USA, 1984. Elsevier North-Holland, Inc.

[8] Ling Guo, Akil Narayan, and Tao Zhou. Constructing least-squares polynomial approximations. *SIAM Review*, 62(2):483–508, 2020.

[9] Yous V Halder, Benjamin Sanderse, and Barry Koren. An adaptive minimum spanning tree multielement method for uncertainty quantification of smooth and discontinuous responses. *SIAM Journal on Scientific Computing*, 41(6):A3624–A3648, 2019.

[10] John D Jakeman, Akil Narayan, and Dongbin Xiu. Minimal multi-element stochastic collocation for uncertainty quantification of discontinuous functions. *Journal of Computational Physics*, 242:790–808, 2013.

[11] Stephen Joe and Frances Y Kuo. Notes on generating sobol sequences. *ACM Transactions on Mathematical Software (TOMS)*, 29(1):49–57, 2008.

[12] Dan Kalman. A singularly valuable decomposition: the svd of a matrix. *The college mathematics journal*, 27(1):2–23, 1996.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Guang Lin, Xiaoliang Wan, Chau-Hsing Su, and George Em Karniadakis. Stochastic computational fluid mechanics. *Computing in Science & Engineering*, 9(2):21, 2007.

[15] Akil Narayan and Dongbin Xiu. Stochastic collocation methods on unstructured grids in high dimensions via interpolation. *SIAM Journal on Scientific Computing*, 34(3):A1729–A1752, 2012.

[16] Andre Nataf. Determination des distribution don't les marges sont donnees. *Comptes rendus de l'Académie des Sciences*, 225:42–43, 1962.

[17] Stephen M Omohundro. *Five balltree construction algorithms*. International Computer Science Institute Berkeley, 1989.

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

[19] Prashant Rai. *Sparse low rank approximation of multivariate functions–Applications in uncertainty quantification*. PhD thesis, Ecole Centrale de Nantes (ECN), 2014.

[20] Murray Rosenblatt. Remarks on a multivariate transformation. *The annals of mathematical statistics*, 23(3):470–472, 1952.

[21] Gilbert Saporta. *Probabilités, analyse des données et statistique*. Editions Technip, 2006.

[22] Khachik Sargsyan, Cosmin Safta, Bert Debusschere, and Habib Najm. Uncertainty quantification given discontinuous model response and a limited number of model runs. *SIAM Journal on Scientific Computing*, 34(1):B44–B64, 2012.

[23] Khachik Sargsyan, Cosmin Safta, Habib N Najm, Bert J Debusschere, Daniel Ricciuto, and Peter Thornton. Dimensionality reduction for complex models via bayesian compressive sensing. *International Journal for Uncertainty Quantification*, 4(1), 2014.

[24] Sergei Abramovich Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. In *Doklady Akademii Nauk*, volume 148, pages 1042–1045. Russian Academy of Sciences, 1963.

[25] Il'ya Meerovich Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 7(4):784–802, 1967.

[26] Bruno Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability engineering & system safety*, 93(7):964–979, 2008.

[27] Xiaoliang Wan and George Em Karniadakis. An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *Journal of Computational Physics*, 209(2):617–642, 2005.

[28] Norbert Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938.

[29] Dongbin Xiu. Efficient collocational approach for parametric uncertainty analysis. *Commun. Comput. Phys*, pages 293–309, 2007.

[30] Dongbin Xiu. Fast numerical methods for stochastic computations: a review. *Communications in computational physics*, 5(2-4):242–272, 2009.

[31] Dongbin Xiu and Jan S Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM Journal on Scientific Computing*, 27(3):1118–1139, 2005.

[32] Dongbin Xiu and George Em Karniadakis. The wiener–askey polynomial chaos for stochastic differential equations. *SIAM journal on scientific computing*, 24(2):619–644, 2002.