



Sequential Design of Mixture Experiments with an Empirically Determined Input Domain and an Application to Burn-up Credit Penalization of Nuclear Fuel Rods

François Bachoc, Théo Barthe, Thomas Santner, Yann Richet

► To cite this version:

François Bachoc, Théo Barthe, Thomas Santner, Yann Richet. Sequential Design of Mixture Experiments with an Empirically Determined Input Domain and an Application to Burn-up Credit Penalization of Nuclear Fuel Rods. Nuclear Engineering and Design, 2021, 374, pp.111034. 10.1016/j.nucengdes.2020.111034 . hal-02535987

HAL Id: hal-02535987

<https://hal.science/hal-02535987>

Submitted on 7 Apr 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Sequential Design of Mixture Experiments with an Empirically Determined Input Domain and an Application to Burn-up Credit Penalization of Nuclear Fuel Rods

François Bachoc^{*1}, Théo Barthe², Thomas Santner³, and Yann Richet⁴

¹Institut de Mathématiques de Toulouse, 118 route de Narbonne, 31062, Toulouse, France.

²Atos, Les Espaces St Martin, 6 Impasse Alice Guy, 31300 Toulouse, France

³The Ohio State University, 1958 Neil Avenue, Columbus, Ohio 43210, United States

⁴Institut de Radioprotection et de Sûreté Nucléaire, 31 Avenue de la Division Leclerc, 92260 Fontenay-aux-Roses, France

Abstract

This paper presents methodologies for solving a common nuclear engineering problem using a suitable mathematical framework. Besides its potential for more general applications, this abstract formalization of the problem provides an improved robustness to the solution compared to the empirical treatment used in industrial practice of today. The essence of the paper proposes a sequential design for a stochastic simulator experiment to maximize a computer output $y(\mathbf{x})$. The complications present in applications of interest are (1) the input \mathbf{x} is an element of an unknown subset of a positive hyperplane and (2) $y(\mathbf{x})$ is measured with error. The training data for this problem are a collection of historical inputs \mathbf{x} corresponding to runs of a physical system that

^{*}Corresponding author. Institut de Mathématiques de Toulouse, 118 route de Narbonne, 31062 Toulouse, France. francois.bachoc@math.univ-toulouse.fr

is linked to the simulator and the associated $y(\boldsymbol{x})$. Two methods are provided for estimating the input domain. An extension of the well-known efficient global optimization (EGO) algorithm is presented to solve the optimization problem. An example of application of the method is given in which patterns of the “combustion rate” of fissile spent fuel rods are determined to maximize the computed k-effective taken to be the “criticality coefficient”.

KEY WORDS: Expected Improvement; Gaussian process interpolator; Simplex; Stochastic simulation; Empirically determined input domain.

1 Introduction

Some important issues in safety assessment are the selection of the measures used to prevent accidental events. In nuclear safety applications, there are at least two competing ways of modeling accidental events: worst-case identification (and then prevention of this case), and the probabilistic containment of consequences. Depending on the precise application, the opportunity to choose one or the other method of modeling can rely on practical considerations, but should be consistent within the whole safety study framework and include the mixture of information from many fields (say, for instance, seismology, structural mechanics, nuclear core cooling, neutronics, and radiology).

Ordinarily industrial applications are in greater control than environment ones, where unknown external conditions can modify critical process elements that affect safety. Indeed, sophisticated mathematical models of industrial safety studies are ordinarily regarded as reliable descriptions of their performance in the real-world. Thus, the design principle used in many industrial safety studies is to avoid the worst-case scenarios identified by a mathematical model of the industrial process. However when the complexity of the safety study increases and the mathematical model is sensitive to uncertain parameters, the prevention of mathematically-based unacceptable events becomes more and more problematic.

A typical example of increasing system complexity occurs when the homogeneity of a crit-

ical materials' density, its mixing phases, its temperature, or any other spatially-dependent properties can not be guaranteed to be the fixed value assumed by a homogeneous model of the system. To be a more accurate approximation of reality, the homogeneous model of a critical component must be replaced by a heterogeneous one. However, it is typically far more difficult to determine the worst case performance of a system having heterogeneous components than one having homogeneous subsystems.

This paper proposes methodology for the construction of heterogeneity-based models to provide a worst-but-credible-case in a nuclear fuel storage application. Also known as “burn-up credit”, our assessment takes into account the burning (and thus reactivity loss) of fissile fuel rods to relax some storage constraints in order to reduce the risk from using multiple over-constrained storages. It should be noticed that even if this problem is solved in this paper for systems that contain a single one-dimensional heterogeneity (say, along fuel rod lengths), it is already useful for many other applications such as radial powder densities and lengthwise density in pipes, among others. Furthermore, the methodologies proposed here may well be extended to larger dimensions in the same mathematical framework.

To be now more formal, consider a real-valued simulator having functional input $x(t)$. The input $x(t)$ is assumed to be *positive* and *continuous* with argument t having domain which is a bounded interval that we take to be $[0, 1]$ without loss of generality. In our application n input curves to the simulator, $x_1(t), \dots, x_n(t)$, were generated from a set of n runs of a physical system that is described below in more detail. For each input $\mathbf{x} = \{x(t)\}_{t \in [0, 1]}$ the simulator computes a $y(\mathbf{x})$ that is corrupted by measurement error.

Two other features of the input curves will be noted. First, the functions $x_i(t)$, $i = 1, \dots, n$, are measured only at a finite grid of d values. This grid is assumed to be the *same* for all functions, say, $0 \leq t_1 < \dots < t_d \leq 1$. An interpolation scheme would be applied to the $\{x_i(t)\}_i$ to achieve a common input domain if the raw data did not satisfy it. Second, by dividing each component of $\mathbf{x} = (x(t_1), \dots, x(t_d))^T$ by its mean $\bar{x} = \frac{1}{d} \sum_{j=1}^d x(t_j)$ it is assumed that $\sum_{j=1}^d x(t_j) = d$ for all inputs. Thus each \mathbf{x} has positive elements and is an

element of the d -hyperplane $\{w_1, \dots, w_d; \sum_{j=1}^d w_j = d\}$; informally \mathbf{x} is said to belong to the positive d -hyperplane. Equivalently, \mathbf{x} has discrete values whose *average* is one. Because of their origin, this paper will also refer to the $d \times 1$ vector \mathbf{x} as a curve or function.

The goals of this paper are two-fold. First, it characterizes curves \mathbf{x} that are “consistent” with the historical $\mathbf{x}_1, \dots, \mathbf{x}_n$ from physical system runs. This objective can be viewed as estimating the valid input subset of the positive d -hyperplane, a set which is denoted \mathcal{X} . Second, an $\mathbf{x}^* \in \mathcal{X}$ is sought that maximizes $y(\mathbf{x})$ over $\mathbf{x} \in \mathcal{X}$.

To make this general setup concrete, consider an example that will be analyzed in Section 4. Spent fuel rods retrieved from a nuclear reactor are inspected. Let $x(t)$ be the “combustion rate” (in megawatt-days/ton) measured at vertical position t along the fuel rod, where the rod has been scaled so that $t \in [0, 1]$. Figure 1 is a cartoon that illustrates a spent fuel rod and the corresponding combustion rate energy $x(t)$. In this application there are $n = 3,158$ spent fuel rods (and their combustion rate curves) of standard nuclear power plants, available from Cacciapouti and Volkinburg (1997). The combustion rate is measured at $d = 18$ equally spaced points along its input domain; thus the associated \mathbf{x}_i is an element of the positive 18-hyperplane. For each combustion rate $\mathbf{x} = (x(t_1), \dots, x(t_{18}))$, an associated *criticality coefficient* $y(\mathbf{x})$ is the output quantity of interest: if $y(\mathbf{x}) < 1$, the rod is called “subcritical”; if $y(\mathbf{x}) = 1$, the rod is “critical”; and if $y(\mathbf{x}) > 1$, the rod is termed “super-critical”.

A depletion calculation was made for each of the $d = 18$ zones using the criticality numerical simulation package CRISTAL (CRISTAL, 2018) which includes a macroscopic cross-section calculation for each zone, followed by a Monte-Carlo k-effective calculation of the whole system to compute the criticality coefficient for an entire combustion rate curve $\mathbf{x} = (x(t_1), \dots, x(t_{18}))$. This process produces a noisy version of $y(\mathbf{x})$. In addition, CRISTAL can be costly to run, although in this application the evaluations are approximately 15 minutes each. Let $y_i = y(\mathbf{x}_i)$ denote the computed criticality coefficient for the i^{th} combustion rate function, $i = 1, \dots, 3,158$.

The literature contains a number of investigations that provide additional information about Nuclear Engineering and Statistical optimization of stochastic simulators. Cacuci (2010) provides additional background on nuclear engineering and on numerical simulation in this context. Several papers have studied prediction of deterministic simulator outputs having non-rectangular input regions. Stinstra et al. (2003) and Draguljić et al. (2012) consider cases where codes have bounded polygonal input regions. The article de Klerk (2008) and the references therein review optimization of a deterministic function defined on a simplex. Loeppky et al. (2013) performed global sensitivity analysis on the output from a deterministic computer simulator whose input is the simplex. Picheny et al. (2013) estimate sequentially the quantile of a function $y(\mathbf{X})$ with random inputs \mathbf{X} when $y(\mathbf{x})$ is observed with measurement error.

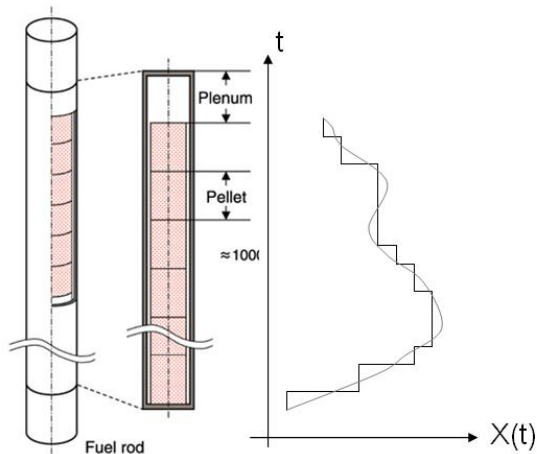


Figure 1: Schematic of a spent fuel rod and its combustion rate curve.

The remainder of this paper is organized as follows. Section 2 reviews the Efficient Global Optimization (EGO) of Jones et al. (1998) for minimizing an unknown $y(x)$: $\mathcal{X} \mapsto \mathbb{R}$ over a rectangular \mathcal{X} and modifications of EGO for cases when $y(x)$ is measured with noise. Section 3 introduces two methods for identifying a set of curves \mathbf{x} that are compatible with the training inputs. One method, given in Subsection 3.1, uses expert opinion and

a second method, described in Subsection 3.2, uses a kernel density estimation approach. Finally, Section 4 gives two examples; the first is an easily understood analytic application which is used to observe the performance of the proposed methodology while the second is a determination of configurations of spent fuel rods in nuclear power reactors that are associated with high criticality settings.

2 Sequential Optimization

Once the mathematical framework is defined, the following practical constraints of our simulation must be accounted for in its solution: a d -dimensional input space is used to approximate the functional burn-up profile, and the identification of a penalizing and credible burn-up profile should be achievable. This problem is one of global optimization. Even though numerous algorithms have been proposed to solve such problems, we will focus on the ones relevant to our practical calculation issues: each simulation output relies on a deterministic depletion calculation based on $\mathbf{x} = (x(t_1), \dots, x(t_{18}))$ which is followed by a time-consuming Monte-Carlo determination of the reactivity ($y(\mathbf{x})$ being the k -effective). These computational characteristics motivate us to choose an optimization method with the smallest possible number of objective function evaluations and which also allows a noisy objective function (as the k -effective is estimated with Gaussian measurement error).

2.1 The Expected Improvement (EI) Algorithm

First the case of minimizing an unknown but deterministic $y: \mathcal{X} \mapsto \mathbb{R}$ will be discussed; then modifications will be given to accommodate the case of $y(\mathbf{x})$ measured with error. In either case, problems of maximizing $y(\mathbf{x})$ can be solved by applying the methods below to minimizing the negative of $y(\mathbf{x})$.

The (expected improvement) sequential design strategies of Schonlau et al. (1998) and Jones et al. (1998), called *Efficient Global Optimization* (EGO) algorithms, were introduced

to find any $x_{\min} \in \arg \min_{\mathbf{x} \in \mathcal{X}} y(\mathbf{x})$ where $y(\mathbf{x}): \mathcal{X} \mapsto \mathbb{R}$ is deterministic. EGO is initiated by computing $y(\mathbf{x})$ on a space-filling set of n inputs, say \mathcal{D}_n , so that starting information about $y(\cdot)$ is available over a wide, if not dense, subset of the input space. Let \mathbf{y}^n denote the vector of outputs corresponding to the initial n -point experimental design. At each stage EGO adds one input \mathbf{x} to the previous design and the associated $y(\mathbf{x})$.

The reference Schonlau et al. (1998) assumed $y(\mathbf{x})$ can be modelled as a draw from a hierarchical Gaussian process plus regression. Letting $Y(\mathbf{x})$ denote this process, its distribution is specified in two stages. Given process parameters $Y(\mathbf{x})$ is assumed to have mean $f(\mathbf{x})^\top \boldsymbol{\beta}$ where $f(\mathbf{x})$ is known and $\boldsymbol{\beta} \in \mathbb{R}^p$ is unknown, and $Z(\mathbf{x}) \equiv Y(\mathbf{x}) - f(\mathbf{x})^\top \boldsymbol{\beta}$ is assumed to be a stationary Gaussian process with zero mean and process variance σ^2 . Unconditionally, $\boldsymbol{\beta}$ is assumed to have a uniform prior distribution over \mathbb{R}^p . Suppose that n evaluations of $y(\mathbf{x})$ have been made at $\mathbf{x}_1, \dots, \mathbf{x}_n$; let \mathbf{y}^n denote the $n \times 1$ vector of these evaluations. Also let \mathbf{Y}^n denote the prior associated with \mathbf{y}^n .

The EGO algorithm is based on the fact that when σ^2 and the correlation parameters are known, inference about $y(\mathbf{x}_0)$ is obtained from the posterior distribution

$$[Y(\mathbf{x}_0) \mid \mathbf{Y}^n = \mathbf{y}^n] \sim N(\hat{y}(\mathbf{x}_0), s^2(\mathbf{x}_0)), \quad (1)$$

where $\hat{y}(\mathbf{x}_0) = \mathbf{f}_0^\top \hat{\boldsymbol{\beta}} + \mathbf{r}_0^\top \mathbf{R}^{-1}(\mathbf{y}^n - \mathbf{F}\hat{\boldsymbol{\beta}})$ is the best linear unbiased predictor (BLUP) of $y(\mathbf{x}_0)$, $\hat{\boldsymbol{\beta}} = (\mathbf{F}^\top \mathbf{R}^{-1} \mathbf{F})^{-1} \mathbf{R}^{-1} \mathbf{F} \mathbf{y}^n$, and

$$s^2(\mathbf{x}_0) = \sigma^2 \left\{ 1 - \mathbf{r}_0^\top \mathbf{R}^{-1} \mathbf{r}_0 + \mathbf{h}_0^\top (\mathbf{F}^\top \mathbf{R}^{-1} \mathbf{F})^{-1} \mathbf{h}_0 \right\} \quad (2)$$

is the mean squared prediction error of $\hat{y}(\mathbf{x}_0)$. Here $\mathbf{h}_0 = \mathbf{f}_0 - \mathbf{F}^\top \mathbf{R}^{-1} \mathbf{r}_0$ while the vectors $\mathbf{f}_0 = f(\mathbf{x}_0)$ is $p \times 1$, \mathbf{r}_0 is $n \times 1$ vector of correlations of $Y(\mathbf{x}_0)$ with $Y(\mathbf{x}_i)$ at each of the n training data inputs, \mathbf{R} is the $n \times n$ matrix of correlations of all pairs of training data, and \mathbf{F} is the $n \times p$ matrix of regression functions for the n training data. See Chapter 3 of Santner et al. (2018).

Let

$$y_{\min}^n = \min_{i=1,\dots,n} y(\mathbf{x}_i)$$

be the smallest value of $y(\mathbf{x})$ among the n previous evaluations. To find an \mathbf{x} with smaller value of $y(\mathbf{x})$ than y_{\min}^n , the EGO algorithm defines the amount of improvement in $y(\cdot)$ at \mathbf{x} to be *zero* if $y(\mathbf{x}) \geq y_{\min}^n$ and to be the difference $y_{\min}^n - y(\mathbf{x})$ otherwise, i.e.,

$$\text{Improvement at } \mathbf{x} = \begin{cases} y_{\min}^n - y(\mathbf{x}), & y_{\min}^n - y(\mathbf{x}) > 0 \\ 0, & y_{\min}^n - y(\mathbf{x}) \leq 0. \end{cases} \quad (3)$$

Equation (3) is an “in principle” quantity because $y(\mathbf{x})$ is *unknown* but the distribution of

$$I_n(\mathbf{x}) = \begin{cases} y_{\min}^n - Y(\mathbf{x}), & y_{\min}^n - Y(\mathbf{x}) > 0 \\ 0, & y_{\min}^n - Y(\mathbf{x}) \leq 0 \end{cases}, \quad (4)$$

given the current data \mathbf{Y}^n is a stochastic version of (3) which can be analyzed for any $\mathbf{x} \in \mathcal{X}$. In particular, the conditional expectation of $I_n(\mathbf{x})$ given \mathbf{Y}^n is termed the *expected improvement*. It is straightforward to show that $E[I_n(\mathbf{x}) \mid \mathbf{Y}^n] = 0$ for any \mathbf{x} which is a training data input and otherwise is

$$\begin{aligned} E[I_n(\mathbf{x}) \mid \mathbf{Y}^n] &= s(\mathbf{x}) \left\{ \frac{y_{\min}^n - \hat{y}(\mathbf{x})}{s(\mathbf{x})} \Phi \left(\frac{y_{\min}^n - \hat{y}(\mathbf{x})}{s(\mathbf{x})} \right) + \phi \left(\frac{y_{\min}^n - \hat{y}(\mathbf{x})}{s(\mathbf{x})} \right) \right\} \\ &= (y_{\min}^n - \hat{y}(\mathbf{x})) \Phi \left(\frac{y_{\min}^n - \hat{y}(\mathbf{x})}{s(\mathbf{x})} \right) + s(\mathbf{x}) \phi \left(\frac{y_{\min}^n - \hat{y}(\mathbf{x})}{s(\mathbf{x})} \right), \end{aligned} \quad (5)$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the $N(0, 1)$ distribution and density function, respectively. Inspection shows that (5) is “large” for those \mathbf{x} having either

- a predicted value at \mathbf{x} that is much smaller than the best minimum computed so far, i.e., $\hat{y}(\mathbf{x}) \ll y_{\min}^n$, or

- there is much uncertainty about the predicted value of $y(\mathbf{x})$, i.e., $s(\mathbf{x})$ is large relative to $|y_{\min}^n - \hat{y}(\mathbf{x})|$.

In sum, the EGO algorithm update to \mathcal{D}_n selects an $\mathbf{x}_{n+1} \in \mathcal{X}$ to maximize $E[I_n(\mathbf{x}) | \mathbf{Y}^n]$. Then EGO sets $\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{\mathbf{x}_{n+1}\}$, $\mathbf{Y}^{n+1} = ((\mathbf{Y}^n)^\top, Y(\mathbf{x}_{n+1}))^\top$, increments n , and continues with the next update. The algorithm is typically stopped after a fixed budget has been exhausted for $y(\mathbf{x})$ evaluations or when the maximum expected improvement is “small”. When EGO stops sampling, it predicts \mathbf{x}_{\min} to be that member of the current set of inputs at which $y(\cdot)$ has been evaluated, say $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, to satisfy

$$y(\hat{\mathbf{x}}_{\min}) = \min_{i=1, \dots, N} y(\mathbf{x}_i). \quad (6)$$

The article by Picheny et al. (2013) and the references therein discuss extensions of EGO to sequentially identify an $\mathbf{x} \in \arg \min y(\mathbf{x})$ when $y(\mathbf{x})$ observations contain measurement error, i.e., the observed value at \mathbf{x} is

$$y^s(\mathbf{x}) = y(\mathbf{x}) + \epsilon(\mathbf{x}),$$

where $\epsilon(\mathbf{x})$ is a white noise process with variance τ^2 . The main complication is that no exact $y(\mathbf{x})$ evaluation is available. However $y(\mathbf{x})$ can be approximated from the observed data by

$$\hat{y}(\mathbf{x}) = E \{Y(\mathbf{x}) | Y^s(\mathbf{x}_1), \dots, Y^s(\mathbf{x}_n)\}$$

where $Y(\mathbf{x})$ is the GP described at the beginning of this section and $Y^s(\mathbf{x})$ is the sum of the $Y(\mathbf{x})$ process and a white noise process. A simple alternative approximation of the current minimum in (6) is used in the examples of Section 4. The method estimates $\min_{i=1, \dots, n} y(\mathbf{x}_i)$ by $\min_{i=1, \dots, n} \hat{y}(\mathbf{x}_i) - 2\tau$. The quantity -2τ makes the estimator downward-biased and promotes exploration. With this estimation, the expected improvement in (5) is maximized to select the next observation point.

In this paper the problem of maximizing $y(\mathbf{x})$ over \mathbf{x} in an unknown \mathcal{X} is solved by identifying \mathcal{X} using one of the two methods described in Section 3. Then the problem

$$\mathbf{x}^* \in \operatorname{argmax}_{\mathbf{x} \in \mathcal{X}} y(\mathbf{x}) \quad (7)$$

is solved by the stochastic version of the EGO algorithm with the following adjustments. Because $\mathbf{x} = (x_1, \dots, x_d) \in \mathcal{X}$ satisfy $\sum_{j=1}^d x_j = d$, the Gaussian process model is built on the $d-1$ dimensional image, E , of \mathcal{X} given by the linear transformation $\mathbf{\Lambda}$ in Loeppky et al. (2013); then $E = \mathbf{\Lambda}\mathcal{X}$ is $d-1$ dimensional. The expected improvement is defined for $\mathbf{x} \in E$ by equation (5) with the stochastic adjustments given in the previous paragraph.

3 Empirical Determination of the Input Domain

Beyond searching for penalizing cases through the optimization described above, the critical problem in our approach is the identification of a credible set of penalized parameters. More precisely, the optimization domain \mathcal{X} was considered to be known in Section 2, while the objective of Section 3 is to “estimate” \mathcal{X} , from the observed inputs $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathcal{X}$. In most industrial applications a simple hypercube may be considered for the optimization domain, and even the boundaries may be reasonably easily assumed (for instance a density of water between 0 and 1). But as described in the introduction, the parameters focused in our framework are more complex and linked to each other (as a discrete sampling of functional parameters, continuous by nature). So the relevant set for such parameters has to be defined and standardized gauged by expert knowledge or empirical experience.

This section describes two methods for identifying a set of positive input curves $\mathbf{x} = (x_1, \dots, x_d)$ which satisfy $\sum_{j=1}^d x_j = d$ and that are “near” the historical set of curves, $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,d})$, $i = 1, \dots, n$. These constructions recognize that the historical curves form a skeleton of the total set of curves that should be considered as the input domain for the optimization problems considered in this paper. The two methods do not make the

assumption that there is a common theoretical domain that is to be estimated; indeed, they need not produce the same extension of the set of historical curves. Acknowledging this ambiguity, informally we use the notation \mathcal{X} to denote the input space. The first approach introduced in this section defines \mathcal{X} by constraints based on a mixture of expert knowledge of the physical system and/or graphical analysis of $\mathbf{x}_1, \dots, \mathbf{x}_n$. The second approach defines \mathcal{X} as a kernel density estimate formed from the coefficients of the projections of the historical \mathbf{x}_i onto a basis of spline functions.

3.1 Defining \mathcal{X} Using Expert Knowledge and/or Empirical Experience

The first approach identifies \mathcal{X} to be positive \mathbf{x} curves using constraints determined by expert knowledge and empirical experience. The latter uses a visual analysis of the n historical curves. As an example, the following four constraints based on the historical data were used in the examples of Section 4.

1. **Bound Constraints** at each of the d component positions of \mathbf{x}

$$\min_{i=1,\dots,n} (x_{i,j}) - \epsilon \leq x_j \leq \max_{i=1,\dots,n} (x_{i,j}) + \epsilon \quad (8)$$

where $j \in \{1, \dots, d\}$ and $\epsilon \geq 0$ is a user-specified tolerance level.

2. **Bounds on Incremental Changes** in consecutive components of \mathbf{x}

$$\min_{i=1,\dots,n} [x_{i,j+1} - x_{i,j}] - \epsilon \leq x_{j+1} - x_j \leq \max_{i=1,\dots,n} [x_{i,j+1} - x_{i,j}] + \epsilon \quad (9)$$

for all $j \in \{1, \dots, d-1\}$ where $\epsilon > 0$ a user-specified tolerance level.

3. **Constraints on Maximum Variation of \mathbf{x}**

$$\max_{j=j_1,\dots,j_2} |x_{j+1} - x_j| \leq \max_{i=1,\dots,n} \max_{j=j_1,\dots,j_2} |x_{i,j+1} - x_{i,j}| + \epsilon, \quad (10)$$

where $1 \leq j_1 < j_2 \leq d$ and $\epsilon \geq 0$ are user-specified.

4. Constraints on Maximum Total Variation of \mathbf{x}

$$\sum_{j=j_1}^{j_2} |x_{j+1} - x_j| \leq \max_{i=1, \dots, n} \sum_{j=j_1}^{j_2} |x_{i,j+1} - x_{i,j}| + \epsilon, \quad (11)$$

where $1 \leq j_1 < j_2 \leq d$ and $\epsilon \geq 0$ are user-specified. In other cases, more general linear or non-linear constraints such as

$$\mathbf{A} \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x} \end{bmatrix} \leq \mathbf{b}$$

for $i = 1, \dots, n$ could be used.

3.2 Defining \mathcal{X} by Projections onto a Basis Set

Projecting $x(t)$ onto the Set of Spline Basis Functions

The references Ramsay (2006) and Muehlenstaedt et al. (2017) provide an introduction to spline basis functions. Let \mathbb{N} denotes the set of positive integers. Briefly, a spline basis of order m , $m \in \mathbb{N}$, is a set of functions $B_{i,m} : [0, 1] \rightarrow \mathbb{R}^+$, for $i = 1, \dots, K$ where $K \in \mathbb{N}$ is the number of spline functions. Here m is called the order of the spline. Figure 2 illustrates the notation.

The *projection* of a given positive real-valued function $x: [0, 1] \rightarrow \mathbb{R}^+$ onto $\{B_{i,m}(t)\}_{i=1}^K$ is the function

$$\hat{x}^{(m,K)} = \hat{x}^{(m,K)}(t) = \frac{\tilde{x}^{(m,K)}(t)}{(1/d) \sum_{j=1}^d \tilde{x}^{(m,K)}(t_j)}$$

where

$$\tilde{x}^{(m,K)}(t) = \sum_{k=1}^K \alpha_k^* B_{k,m}(t), \quad \text{and } (\alpha_1^*, \dots, \alpha_K^*) \in \underset{\alpha \in \mathbb{R}^K}{\operatorname{argmin}} \int_0^1 \left(x(t) - \sum_{i=1}^K \alpha_i B_{i,m}(t) \right)^2 dt \quad (12)$$

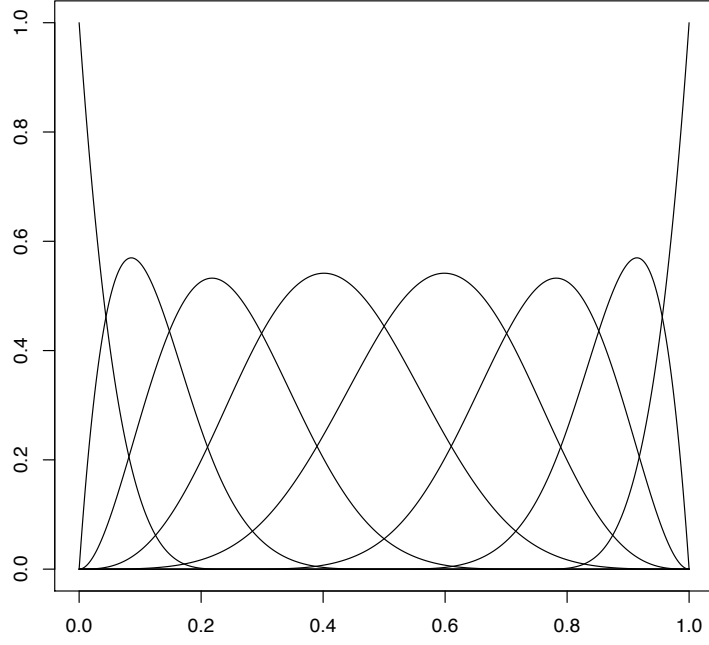


Figure 2: $K = 8$ spline functions of order 5; each spline can be identified by the location of its maximum value.

which shows that $\hat{x}^{(m,K)}$ has the form

$$\hat{x}^{(m,K)}(t) = \sum_{k=1}^K \hat{\alpha}_k B_{k,m}(t). \quad (13)$$

The coefficients $\boldsymbol{\alpha}^* = (\alpha_1^*, \dots, \alpha_K^*)$ have an explicit expression as the least square solution to (12) and hence $\hat{\boldsymbol{\alpha}} = (\hat{\alpha}_1, \dots, \hat{\alpha}_K)$ is straightforward to obtain.

Recall that the observed data for the i^{th} curve is the vector $\mathbf{x}_i = (x_i(t_1), \dots, x_i(t_d))^\top$, $i = 1, \dots, n$. To apply (12) to the function $\mathbf{x} = \mathbf{x}_i$, let ϕ_i denote the spline interpolating function satisfying $\phi_i(t_1) = x_i(t_1), \dots, \phi_i(t_d) = x_i(t_d)$. Then $\boldsymbol{\alpha}^*$ corresponding to \mathbf{x}_i is obtained from (12) by replacing $x_i(t)$ by $\phi_i(t)$. (In the Section 4 examples, $\phi_i(t)$ is obtained by the R function `splinefun` in the package `splines`.)

Kernel Density Estimation

Let $\hat{\boldsymbol{\alpha}}^{(i)} = (\hat{\alpha}_1^{(i)}, \dots, \hat{\alpha}_K^{(i)})$ denote the $(\hat{\alpha}_1, \dots, \hat{\alpha}_K)$ in (13) for the i^{th} input curve \mathbf{x}_i ,

$i = 1, \dots, n$. Consider the following kernel density estimation procedure based on the set $\{\hat{\alpha}^{(1)}, \dots, \hat{\alpha}^{(n)}\}$. Following the approach of Perrin et al. (2018), let $\phi(\cdot)$ denote the probability density function of the univariate standard Normal distribution. Given K and positive scale factors $\lambda = (\lambda_1, \dots, \lambda_K)$, let

$$\rho_{\lambda_1, \dots, \lambda_K}(\alpha) = \rho_{\lambda}(\alpha) = \frac{1}{n} \sum_{i=1}^n \prod_{k=1}^K \frac{1}{\lambda_k} \phi\left(\frac{\alpha_k - \hat{\alpha}_k^{(i)}}{\lambda_k}\right) \quad (14)$$

define a function from \mathbb{R}^K to \mathbb{R}^+ where $\alpha = (\alpha_1, \dots, \alpha_K)$. It is straightforward to check that $\rho_{\lambda}(\alpha)$ has integral one over \mathbb{R}^K . Intuition suggests that given a scaling $\lambda \in (0, \infty)^K$, $\rho_{\lambda}(\alpha)$ is *large* for choices of α that are compatible with the *set of observed input curves*.

In this paper the scale parameters $\lambda_1, \dots, \lambda_K$ are selected by cross validation using

$$\hat{\lambda} = (\hat{\lambda}_1, \dots, \hat{\lambda}_K)^{\top} \in \underset{(\lambda_1, \dots, \lambda_K) \in (0, \infty)^K}{\operatorname{argmax}} \sum_{i=1}^n \log(\rho_{\lambda}^{-i}(\hat{\alpha}^{(i)})) \quad (15)$$

where ρ_{λ}^{-i} is obtained from (14) by removing $\hat{\alpha}^{(i)}$ from the set $\{\hat{\alpha}^{(i)}\}_{i=1}^n$ (and decrementing n to $n-1$). Thus $\rho_{\hat{\lambda}}(\alpha)$ can be viewed as a kernel density estimator most compatible with the coefficients $\{\hat{\alpha}^{(1)}, \dots, \hat{\alpha}^{(n)}\}$ from $\mathbf{x}_1, \dots, \mathbf{x}_n$. Thus the value of $\rho_{\hat{\lambda}}(\alpha)$ is used to quantify the level of “realism” of curves having form $\sum_{k=1}^K \alpha_k B_{k,m}$ to the observed $\mathbf{x}_1, \dots, \mathbf{x}_n$.

Threshold Selection

In the following, $\hat{\rho}(\alpha) = \rho_{\hat{\lambda}}(\alpha)$ denotes the estimated compatibility function in equations (14) and (15). To select discretized curves \mathbf{x} most compatible with the training data, we choose $T > 0$ so that $\alpha \in \mathbb{R}^K$ is considered compatible with $\mathbf{x}_1, \dots, \mathbf{x}_n$ if and only if $\hat{\rho}(\alpha) \geq T$.

The value T is chosen as follows. For $\alpha \in \mathbb{R}^K$, let $x_{\alpha}(t) = \sum_{k=1}^K \alpha_k B_{k,m}(t)$; given $\Delta > 0$, select a $T > 0$ so that any α which satisfies

$$\left(\int_0^1 [x_{\alpha}(t) - x_{\hat{\alpha}^{(i)}}(t)]^2 dt \right)^{1/2} \leq \Delta \quad (16)$$

for at least one $i \in \{1, \dots, n\}$ also satisfies $\widehat{\rho}(\boldsymbol{\alpha}) \geq T$. Inspection of (14) shows that $\widehat{\rho}(\boldsymbol{\alpha}) \geq T$ holds provided, for some $i \in \{1, \dots, n\}$,

$$\frac{1}{n} \prod_{k=1}^K \frac{1}{\widehat{\lambda}_k} \phi \left(\frac{\alpha_k - \widehat{\alpha}_k^{(i)}}{\widehat{\lambda}_k} \right) \geq T \quad (17)$$

and (16) holds for this $\boldsymbol{\alpha}$ and i . The expression on the right hand side of (17) is the limiting value of (14) corresponding to the case where $\widehat{\boldsymbol{\alpha}}^{(i)}$ is infinitely distant from all $\{\widehat{\boldsymbol{\alpha}}^{(\ell)}\}_{\ell \neq i}$ and where $\left(\int_0^1 [x_{\boldsymbol{\alpha}}(t) - x_{\widehat{\boldsymbol{\alpha}}^{(i)}}(t)]^2 dt \right)^{1/2} \leq \Delta$. Based on the above observations, the selected threshold \widehat{T} is defined as

$$\widehat{T} = \min_{\substack{\boldsymbol{\alpha} \in \mathbb{R}^K \\ \left(\int_0^1 x_{\boldsymbol{\alpha}}(t)^2 dt \right)^{1/2} \leq \Delta}} \frac{1}{n} \prod_{k=1}^K \frac{1}{\widehat{\lambda}_k} \phi \left(\frac{\alpha_k}{\widehat{\lambda}_k} \right). \quad (18)$$

In practice, the calculation of \widehat{T} in (18) is straightforward since one can precompute the Gram matrix with $(i, j)^{th}$ element $\int_0^1 B_{i,m}(t) B_{j,m}(t) dt$.

Let $\mathbf{x}(\boldsymbol{\alpha})$ denote the $d \times 1$ vector with j^{th} element $\left[\sum_{k=1}^K \alpha_k B_{k,m}(t_j) \right]$ for $j = 1, \dots, d$. As for the historical data, scaling $\mathbf{x}(\boldsymbol{\alpha})$ by the average of its components, i.e., by $\bar{x}(\boldsymbol{\alpha}) = (x_1(\boldsymbol{\alpha}) + \dots + x_d(\boldsymbol{\alpha}))/d$ results in a positive point on the d -hyperplane (when $\boldsymbol{\alpha}$ has positive components). To select $\boldsymbol{\alpha}$ compatible with $\widehat{\rho}(\boldsymbol{\alpha}) > T$, compute

$$\boldsymbol{\alpha}^* \in \underset{\substack{\boldsymbol{\alpha} \in [0, \infty)^K \\ \widehat{\rho}(\mathbf{x}(\boldsymbol{\alpha})/\bar{x}(\boldsymbol{\alpha})) \geq \Delta}}{\operatorname{argmax}} y(\mathbf{x}(\boldsymbol{\alpha})/\bar{x}(\boldsymbol{\alpha})). \quad (19)$$

The optimization problem corresponds to minimizing a function where the constraints can be tested with negligible cost. This optimization takes place in the K -dimensional space of the $\boldsymbol{\alpha}'$ s.

4 Worked Examples

The rationale behind our approach, based on either expert knowledge or empirical experience, relies on the availability of a sufficient amount of relevant data. The present example explicitly requires the axial burn-up profile database for pressurized water reactors available through OECD Nuclear Energy Agency Data Bank (Cacciapouti and Volkinburg, 1997) which is taken as the reference "historical" data.

First, the two methods of Section 3 will be applied to identify a set of combustion rate curves \mathbf{x} that are consistent with those of spent fuel rods from nuclear plants, the "historical" curves for both examples. Then the optimization method of Section 2 will be applied to a simple analytic function where the answer and the performance of the optimization procedure is straightforward to understand. Finally, the example introduced in Section 1 in which it is desired to maximize the criticality coefficient for spent fuel is considered. This process is termed Burn-up Credit Penalization in the nuclear industry.

4.1 Exploratory Analysis of the Fuel Rod Data

In this application, there are $n = 3,158$ discretized combustion rate curves $\mathbf{x}_1, \dots, \mathbf{x}_{3158}$, each of which has been measured at the (same) $d = 18$ vertical measurement points $(0, 1/17, \dots, 16/17, 1)$. Recall that the curves have been normalized so that $\sum_{j=1}^{18} x_{i,j} = 18$, for $i = 1, \dots, 3,158$, or equivalently to have an average combustion rate equal to one. Figure 3 shows 50 representative curves from the population of curves; all 50 curves show a common 'vertical-horizontal-vertical' shape which is true of the majority of curves. A small minority of the population have a more complex character.

Because each run of the *CRISTAL* code for this application required only fifteen minutes, sufficient budget was available that the code was run for all 3,158 input functions. Figure 4 plots the 50 curves yielding the *lowest* values of the criticality coefficient, $y(\mathbf{x}_i)$ (between 0.86149 and 0.86665), and the 50 curves yielding the *largest* values of the criticality coefficient

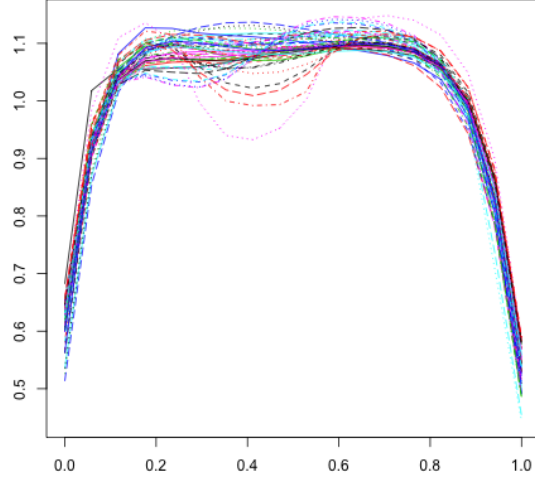


Figure 3: Fifty representative combustion rate curves from the population of 3,158 discretized historical curves.

(between 0.92758 and 0.94123). Visually, it is plain that rods which are evenly burned over t , i.e., which have constant $x(t)$, are safest in the sense of having small $y(\mathbf{x})$ values while rods that have burned unevenly are more hazardous.

4.2 Forming \mathcal{X}

Section 3 described two methods for defining the domain of curves \mathbf{x} having representative combustion rates. The first method combines graphical and numerical explanatory data analysis (EDA) with expert knowledge about the features of burned fuel rods; the second method selects \mathbf{x} which are “near” to the body of basis representatives of the original 3,158 curves.

Defining \mathcal{X} by EDA and Expert Knowledge

Based on visualization of the curves in Figure 3, curves that satisfy the following constraints are considered to have representative combustion rates.

- **Bound Constraints** in (8): set $\epsilon = 0.05$ for the values of the first and last measured combustion rate, i.e., $x_{i,1}$ and $x_{i,18}$;

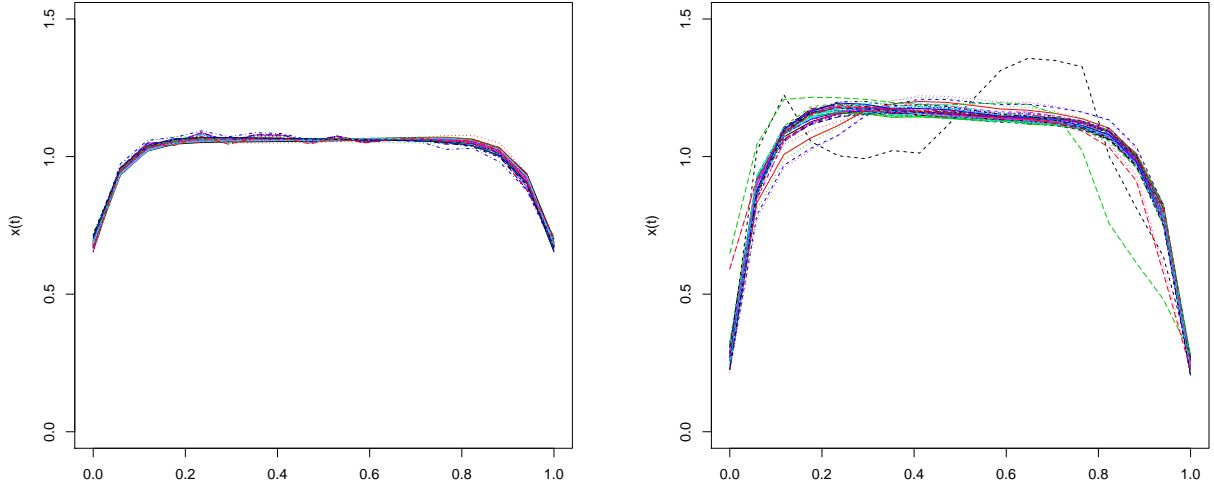


Figure 4: In the fuel rod application of Subsection 4.1, the 50 historical curves \mathbf{x}_i with the smallest (left panel) and the largest (right panel) outputs $y(\mathbf{x}_i)$.

- **Bounds on Incremental Changes** in (9): set $\epsilon = 0.03$ for each of the increments $|x_{i,j+1} - x_{i,j}|$, $j = 1, 2, 16$, and 17 ;
- **Constraints on Maximum Variation of \mathbf{x}** in (10): set $\epsilon = 0.03$, $j_1 = 3$, and $j_2 = 16$;
- **Constraints on Maximum Total Variation of \mathbf{x}** in (11): set $\epsilon = 0.1$, $j_1 = 3$, and $j_2 = 16$.

All 3,158 historical curves satisfy these four constraints by definition and are thus part of the \mathcal{X} domain defined by this criterion.

Defining \mathcal{X} as a Kernel Density Estimate

This application of Section 3.2 takes $K = 8$ spline functions of order $m = 5$ (and are constructed using the R package `splines` with knot sequence $(0, 0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1, 1)$ and option `monoH.FC`). Figure 2 plots the resulting set of spline functions $\{B_{k,5}(t)\}_{k=1}^8$ over $t \in [0, 1]$. As described in Section 3.2, each of the 3,158 discretized curves \mathbf{x} can be approximated by a spline $\sum_{k=1}^8 \hat{\alpha}_k^{(i)} B_{k,5}(t)$. This representation results in a dimension reduction from 18 to 8, and provides a good fit of the 3,158 curves. Figure 5 plots the original and

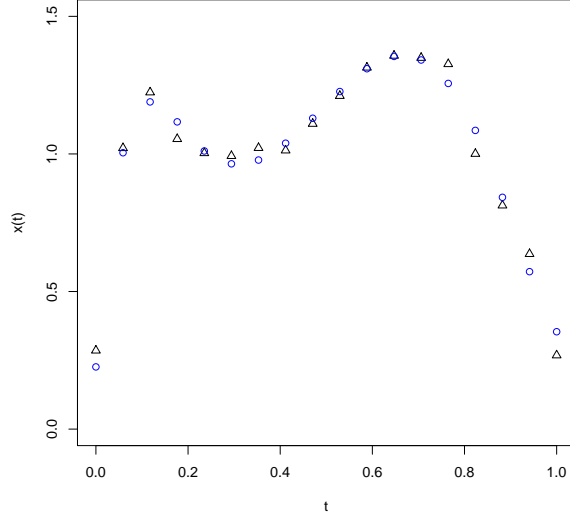


Figure 5: Original (black triangles) and spline approximation (blue circles) for the curve $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,18})$ having the largest mean square difference compared with its spline approximation, among the 3,158 historical curves.

spline approximation for the curve \mathbf{x}_i having *largest mean square difference* from its spline approximation, among the 3,158 historical curves.

Kernel density estimation is performed as described in Section 3.2. The window vector obtained is $(\hat{\lambda}_1, \dots, \hat{\lambda}_8) \approx (0.018, 0.019, 0.019, 0.018, 0.017, 0.015, 0.021, 0.017)$. To illustrate, Figure 6 plots the density of the first marginal probability density function of the (8-dimensional) probability density function $\rho_{\hat{\lambda}_1, \dots, \hat{\lambda}_8}(\alpha_1)$ together with the histogram of $\hat{\alpha}_1^{(1)}, \dots, \hat{\alpha}_1^{(3,158)}$.

The threshold value is selected as described in Section 3.2, where $\Delta = 0.05$ is chosen, and is $\hat{T} = 54.86$. Figure 7 provides a visual insight of the domain $\{\boldsymbol{\alpha}: \hat{\rho}(\boldsymbol{\alpha}) \geq \hat{T}\}$. Coefficient vectors $\hat{\boldsymbol{\alpha}}^{(i_1)}$ and $\hat{\boldsymbol{\alpha}}^{(i_2)}$ are considered for two of the historical inputs, where $\hat{\boldsymbol{\alpha}}^{(i_1)}$ is numerically distant from the remaining $\{\hat{\boldsymbol{\alpha}}^{(i)}\}_{i \neq i_1}$, while $\hat{\boldsymbol{\alpha}}^{(i_2)}$ has closer neighbors. As a consequence $\hat{\rho}(\hat{\boldsymbol{\alpha}}^{(i_1)}) > \hat{\rho}(\hat{\boldsymbol{\alpha}}^{(i_2)})$: The value of $\hat{\rho}(\boldsymbol{\alpha})$ is plotted, for $\boldsymbol{\alpha}$ belonging to the segment with endpoints $\hat{\boldsymbol{\alpha}}^{(i_1)} - 0.1 (\hat{\boldsymbol{\alpha}}^{(i_2)} - \hat{\boldsymbol{\alpha}}^{(i_1)})$ and $\hat{\boldsymbol{\alpha}}^{(i_1)} + 1.1 (\hat{\boldsymbol{\alpha}}^{(i_2)} - \hat{\boldsymbol{\alpha}}^{(i_1)})$. One observes that the parts of the segment close to $\hat{\boldsymbol{\alpha}}^{(i_1)}$ and $\hat{\boldsymbol{\alpha}}^{(i_2)}$ correspond to admissible $\boldsymbol{\alpha}$'s, while the middle of the segment corresponds to inadmissible $\boldsymbol{\alpha}$'s. Recall that Δ is user selected and

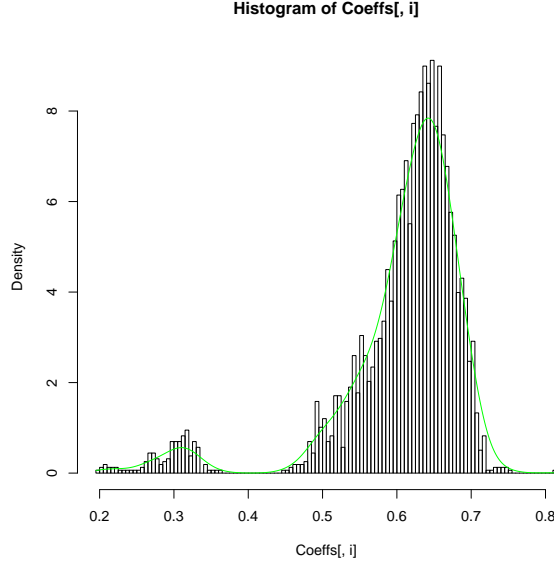


Figure 6: Plot of the first marginal probability density function of the (8-d) probability density function $\rho_{\hat{\lambda}_1, \dots, \hat{\lambda}_8}$ (in green), together with the histogram of the first components $\hat{\alpha}_1^{(1)}, \dots, \hat{\alpha}_1^{(3,158)}$ from the spline kernel basis representations of the 3,158 curves.

that decreasing it increases the threshold and vice versa.

4.3 Optimization of an Analytical Function

This subsection maximizes an analytical function $y(\mathbf{x})$ over the domain of the simplex determined from the $n = 3,158$ historical curves from the nuclear power industry. Both the Expert Knowledge methodology and the Kernel Density approximation will be illustrated to identify this input space. The analytical function to be maximized is

$$y_a(\mathbf{x}) = -\|\mathbf{x} - \mathbf{x}_0\|_2 - \sin(3\|\mathbf{x} - \mathbf{x}_0\|_2)^2, \quad (20)$$

where \mathbf{x}_0 is a fixed one of the $n = 3,158$ historical curves and $\|\mathbf{w}\|_2 = (\mathbf{w}^\top \mathbf{w})^{1/2}$ for a column vector \mathbf{w} . The unique maximizer of $y_a(\mathbf{x})$ in (7) is $\mathbf{x} = \mathbf{x}_0$ with optimal value $y_a(\mathbf{x}_0) = 0$. The function $y_a(\mathbf{x})$ is observed with additive Gaussian noise having mean zero and variance 0.0005^2 . This mild Monte Carlo noise in the $y(\mathbf{x})$ function mimics that present in the second example.

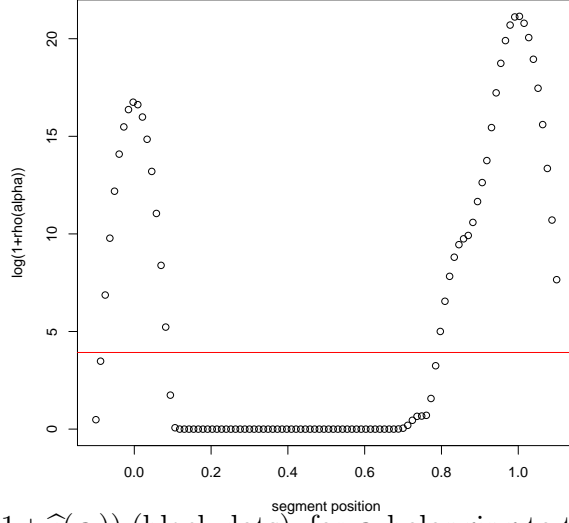


Figure 7: Values of $\log(1 + \widehat{\rho}(\alpha))$ (black dots), for α belonging to the segment with endpoints obtained from a isolated historical coefficient vector and a non-isolated one. The red line is the logarithm of (1 plus the threshold).

The goal is to assess whether the expected improvement algorithm is able to converge to the global maximizer for both the expert knowledge domain in (7) or the domain defined using kernel density approximation of (19).

EI Optimization of (20) over \mathcal{X} Determined by Expert Knowledge

The set of training curves used for the analytic function consisted of 100 curves selected by a space-filling design among the 3,158 historical input vectors. The maximum value of $y_a(\mathbf{x})$ among the training data is approximately -0.75 . Then 50 additional discretized curves were selected from \mathcal{X} using the EI/expert knowledge procedure. The maximum $y_a(\mathbf{x})$ increased to approximately -0.17 using the 50 additional curves.

Figure 8 provides a visual understanding of this performance by plotting three curves that illustrate the performance of the proposed procedure. The first curve is \mathbf{x}_0 which denotes the true global maximizer; the second curve, denoted \mathbf{x}_{init} , is the maximizer of $y_a(\mathbf{x})$ among the 100 initial curves; the third curve, denoted \mathbf{x}_{EI} , is the maximizer of $y_a(\mathbf{x})$ among the 50 curves obtained by expected improvement. Observe that \mathbf{x}_{EI} is, visually, significantly closer to \mathbf{x}_0 than is \mathbf{x}_{init} , which suggests the convergence of the expected improvement procedure.

This example also shows that the admissible set in (7) is amenable to maximization in practice.

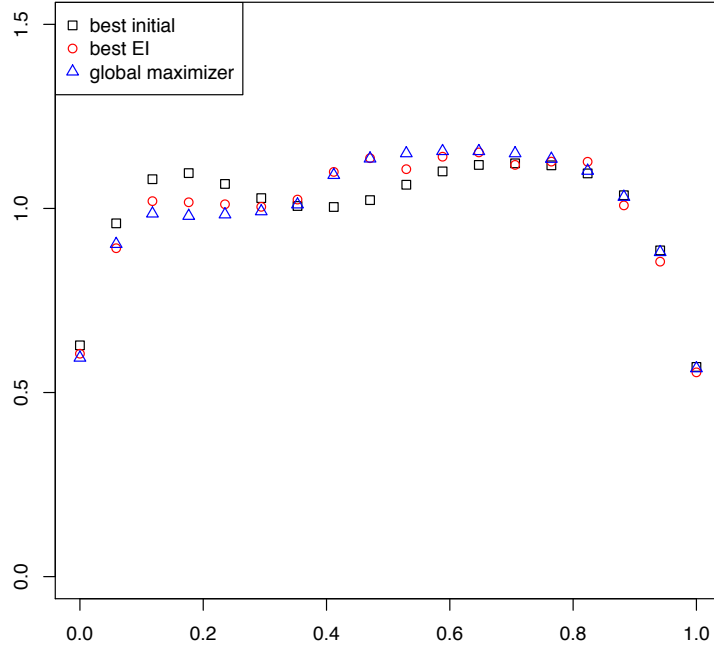


Figure 8: Three input curves for the analytic optimization problem (20): the true global $y_a(\mathbf{x})$ maximizer; the $y_a(\mathbf{x})$ maximizer among the 100 initial curves; the $y_a(\mathbf{x})$ maximizer among the 50 curves added by EI/expert knowledge based on (8)-(11).

EI Optimization of (20) over \mathcal{X} Determined by Kernel Density Estimation

In this case, while the analytical function $y_a(\mathbf{x})$ is (20), the curve \mathbf{x}_0 is now given by $(x_0(t_1), \dots, x_0(t_d))$, with

$$x_0(t) = \frac{\sum_{i=1}^K \alpha_{0,i} B_{i,m}(t)}{\frac{1}{d} \sum_{j=1}^d \sum_{i=1}^K \alpha_{0,i} B_{i,m}(t_j)}$$

where α_0 is one of the 3,158 $\hat{\alpha}^{(1)}, \dots, \hat{\alpha}^{(3158)}$. Thus in the optimization problem (19), the global maximizer curve is $\alpha^* = \alpha_0$. Noisy observations of $y_a(\mathbf{x})$ are obtained as above.

The initial random design of 100 curves was obtained similarly to that for the expert knowledge procedure above; the sequentially added 50 curves were obtained by the EI/kernel density estimation procedure. The conclusions are similar to those for the EI/expert knowledge procedure. Namely, among the initial 100 training data curves, the maximum value of $y_a(\mathbf{x})$ is approximately -0.71 which occurs at \mathbf{x}_{init} . With the 50 additional curves obtained

by expected improvement, this maximum increases to approximately -0.11 at \mathbf{x}_{EI} . Figure 9 plots the three curves \mathbf{x}_{init} , \mathbf{x}_{EI} and \mathbf{x}_0 .

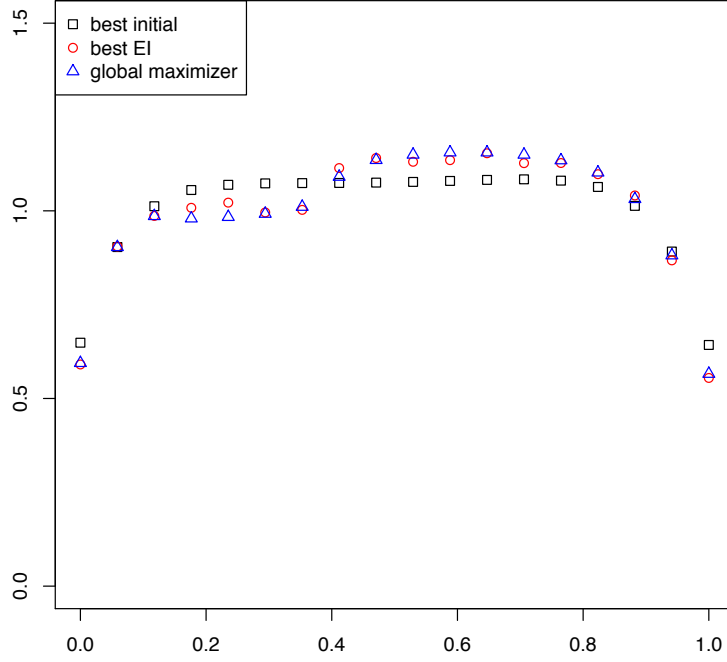


Figure 9: Three input curves for the analytic optimization problem (20): the true global $y_a(\mathbf{x})$ maximizer; the $y_a(\mathbf{x})$ maximizer among the 100 initial curves; the $y_a(\mathbf{x})$ maximizer among the 50 curves added by EI/kernel density estimation.

4.4 Optimizing Burn-up Credit Penalization

In this subsection the expected improvement procedure for solving (7) or (19) is carried out similarly as in Section 4.3 with the analytical function $y_a(\mathbf{x})$ replaced by **CRISTAL** code function $y(\mathbf{x})$ evaluations.

For both methods of identifying the valid input space, \mathcal{X} , the EI algorithm was carried out starting from a Gaussian Process model based on 100 observed values of $y(\mathbf{x})$. For EI/kernel density via the optimization problem (19), these observed values corresponded to a subset $\{\hat{\alpha}^{(i_1)}, \dots, \hat{\alpha}^{(i_{100})}\}$ of $\{\hat{\alpha}^{(1)}, \dots, \hat{\alpha}^{(3,158)}\}$. This subset was selected by the following space-filling procedure. First, 100 barycenters were computed from a K means clustering algorithm applied to $\{\hat{\alpha}^{(1)}, \dots, \hat{\alpha}^{(3,158)}\}$. Then, the $\{\hat{\alpha}^{(i_1)}, \dots, \hat{\alpha}^{(i_{100})}\}$ closest to these

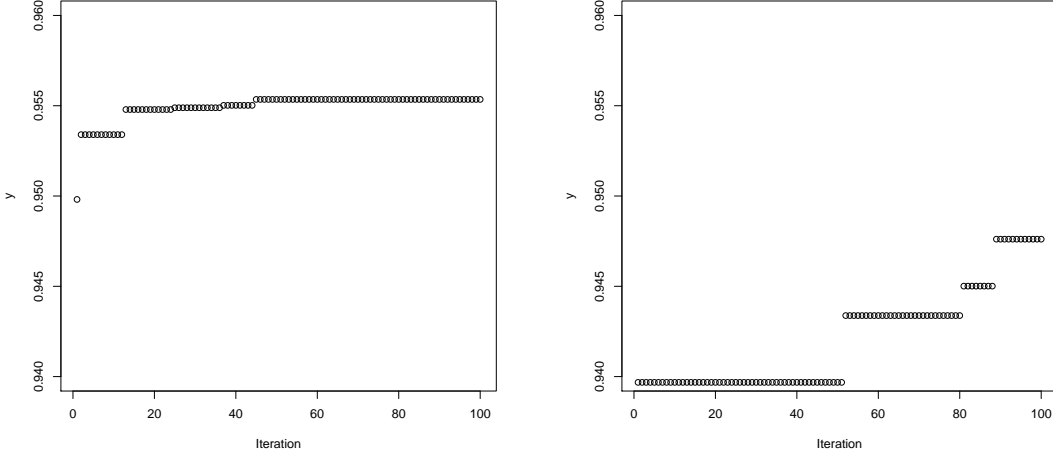


Figure 10: The cumulative maxima, $\max \{m_{\text{hist}}, y(\mathbf{x}_{EI,1}), \dots, y(\mathbf{x}_{EI,i})\}$, as a function of the iteration index i for EI based on the expert knowledge constraints (left panel) and on kernel density estimation (right panel). The symbol m_{hist} denotes the maximum value of $y(\mathbf{x})$ among the 100 training data input curves.

barycenters were selected. The **CRISTAL** code was run 100 times to compute the corresponding $y(\mathbf{x}(\hat{\boldsymbol{\alpha}}^{(i_1)})/\bar{\mathbf{x}}(\hat{\boldsymbol{\alpha}}^{(i_1)}))$, \dots , $y(\mathbf{x}(\hat{\boldsymbol{\alpha}}^{(i_{100})})/\bar{\mathbf{x}}(\hat{\boldsymbol{\alpha}}^{(i_{100})}))$.

For EI/expert knowledge method, a subset $\{\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_{100}}\}$ of $\{\mathbf{x}_1, \dots, \mathbf{x}_{3,158}\}$ was obtained, using the same space-filling procedure as above. The corresponding $y(\mathbf{x}_{i_1}), \dots, y(\mathbf{x}_{i_{100}})$ were selected from the historical data base.

The number of initial values for expected improvement, 100, was hence selected for two reasons. First reason is that 100 initial observations allows an interpretable comparison between the results of EI using expert knowledge versus kernel density estimation methods. The second reason was based on computational budget considerations. In the future, the Burn-up Credit code is expected to become more complex and costly to evaluate. It was extrapolated that the value 100 satisfies future budget constraints and suggests the performance of the two input determination methods.

EI optimization of Burn-up Credit Penalization

The maximum of $\{y(\mathbf{x}_1), \dots, y(\mathbf{x}_{3,158})\}$ is equal to 0.94123. This illustrates the maximization performance using only the historical data base (although these values contain a

small Monte Carlo noise). Starting with 100 training inputs and their corresponding $y(\mathbf{x})$ values and then running 100 iterations of the expected improvement procedure yields new curves $\mathbf{x}_{EI,1}, \dots, \mathbf{x}_{EI,100}$ for both the EI/expert knowledge and EI/kernel density estimation procedures. The maximum of $y(\mathbf{x}_{EI,1}), \dots, y(\mathbf{x}_{EI,100})$ is 0.95535 for the EI/expert knowledge procedure and is 0.94761 for the EI/kernel density estimation procedure. Hence, the admissible set obtained from the expert knowledge is larger, so to speak, than that obtained from the kernel density estimation procedure, and allows for larger values of $y(\mathbf{x})$. This is possibly due to the choices of the tolerance values ϵ and of the distance Δ (see Section 3). One may also notice that, in essence, the EI/expert knowledge procedure allows for a larger search space for optimization, as it does not project the curves onto a lower dimensional space.

To show the effectiveness of the two EI procedures, Figure 10 plots the cumulative maxima of $y(\mathbf{x})$, including that based on the training data, for the 100 EI iterations. The convergence appears to be relatively fast when \mathcal{X} is determined by expert knowledge. When \mathcal{X} is determined by kernel density estimation, additional iterations of expected improvement would likely result in a further improvement of $y(\mathbf{x})$.

Finally Figure 11 plots the three curves \mathbf{x}_{hist} , $\mathbf{x}_{\text{expert}}$, and \mathbf{x}_{kde} , where \mathbf{x}_{hist} corresponds to the maximum of the historical values $\{y(\mathbf{x}_1), \dots, y(\mathbf{x}_{3,158})\}$ and $\mathbf{x}_{\text{expert}}$ (resp. \mathbf{x}_{kde}) corresponds to the maximum of $y(\mathbf{x}_{EI,1}), \dots, y(\mathbf{x}_{EI,100})$ for the EI/expert knowledge (resp. EI/kernel density estimation) procedure. The deviation from $\mathbf{x}_{\text{expert}}$ and \mathbf{x}_{kde} to \mathbf{x}_{hist} is moderate but non-negligible. We also observe that \mathbf{x}_{kde} is smoother than $\mathbf{x}_{\text{expert}}$, which is a feature of the spline decomposition.

5 Summary and Discussion

This paper proposes methods to identify a complicated input domain which is known to be a subset of the simplex, based on observational historical data that are known to belong to the

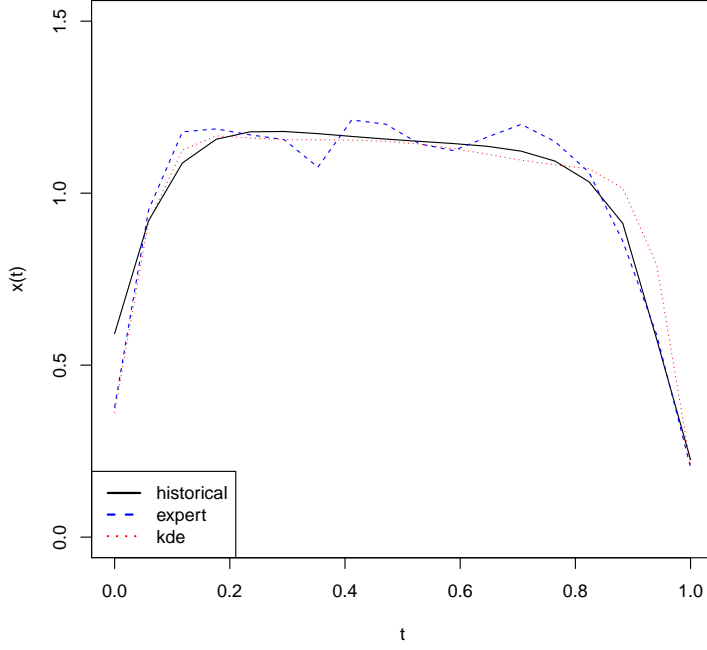


Figure 11: Curves that maximize the Burn-up Credit Penalization: (1) among the 3,158 historical curves; (2) among the 100 curves added by the EI/expert knowledge procedure; (3) among the 100 curves added by the EI/kernel density estimation procedure.

input domain. It also shows how a variant of the EGO algorithm for deterministic output can be applied to optimize the mean output of a stochastic simulator over this domain. The expected improvement function is maximized over an input region of positive Lebesgue measure by applying a linear transformation of the simplex to a lower-dimensional space.

The application of these methods to a large validated historical database of burn-up profiles is an original proposal to solve the problem of burn-up credit in nuclear safety assessment. It has to be compared with current approaches, such as ones that use pre-defined profiles to check the sub-criticality of burned assemblies. But in a wider framework, where it tends to be difficult to pre-define reference profiles (like in mixed oxides fuels), a more general approach like the one presented here should be more robust.

We mention two problems that have not been addressed in this paper but are topics for future research. The first problem stems from the frequently-occurring need in climate science and other scientific areas to build adaptively an input domain from training data.

Climate models consist of submodels for surface temperatures, wind magnitude, wind velocity, sea surface temperatures and other interacting constituents that determine the climate. These submodels must all be computable and verifiable. The bounds on the input domain where all the composite models can be simultaneously run is *unknown* and can be complex. Thus the problem of identifying the input region is one of sequential design. A series of inputs is identified with the resulting attempted model run being successful or not. These data are used to estimate the input domain. The second problem is the determination of sensitivity analysis tools for the mean of a stochastic simulator when the input domain is an estimated subset of the simplex. The research of Loeppky et al. (2013) who developed global sensitivity tools for deterministic simulator output defined on the simplex, is a starting point for this more complicated scenario.

ACKNOWLEDGMENTS

This research was conducted with the support of the Chair in Applied Mathematics OQUAIDO, gathering partners in technological research (BRGM, CEA, IFPEN, IRSN, Safran, Storengy) and academia (CNRS, Ecole Centrale de Lyon, Mines Saint-Étienne, University of Grenoble, University of Nice, University of Toulouse) in the development of advanced methods for Computer Experiments.

The authors would like also to thank the Isaac Newton Institute for Mathematical Sciences for support and hospitality during the programme on *Uncertainty Quantification* when work on this paper was undertaken and The Statistical and Applied Mathematical Sciences Institute for support and hospitality during the program *Model Uncertainty: Mathematical and Statistical*. Finally, this work was supported by: EPSRC grant numbers EP/K032208/1 and EP/R014604/1. This research was also sponsored, in part, by the National Science Foundation under Agreements DMS-0806134 and DMS-1310294 (The Ohio State University). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science

Foundation.

References

- R.J. Cacciapouti and S. Van Volkinburg. Axial burnup profile database for pressurized water reactors. Technical Report USCD1219 ZZ-PWR-AXBUPRO-SNL, OECD Nuclear Energy Agency Data Bank, 1997.
- Dan Gabriel Cacuci. *Handbook of Nuclear Engineering: Vol. 1: Nuclear Engineering Fundamentals; Vol. 2: Reactor Design; Vol. 3: Reactor Analysis; Vol. 4: Reactors of Generations III and IV; Vol. 5: Fuel Cycles, Decommissioning, Waste Disposal and Safeguards*, volume 2. Springer Science & Business Media, 2010.
- CRISTAL. Criticality Calculation Package, v2.0.2,. OECD Nuclear Energy Agency Data Bank, NEA-1903, 2018. URL <http://www.cristal-package.org>.
- Etienne de Klerk. The complexity of optimizing over a simplex, hypercube or sphere: A short survey. *Central European Journal of Operations Research*, 16(2):111–125, 2008.
- Danel Draguljić, Thomas J. Santner, and Angela M. Dean. Non-collapsing spacing-filling designs for bounded polygonal regions. *Technometrics*, 54:169–178, 2012.
- D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.
- Jason L. Loeppky, Brian J. Williams, and Leslie M. Moore. Global sensitivity analysis for mixture experiments. *Technometrics*, 55:68–78, 2013.
- Thomas Muehlenstaedt, Jana Fruth, and Olivier Roustant. Computer experiments with functional inputs and scalar outputs by a norm-based approach. *Statistics and Computing*, 27(4):1083–1097, 2017.
- Guillaume Perrin, Christian Soize, and N Ouhbi. Data-driven kernel representations for sampling with an unknown block dependence structure under correlation constraints. *Computational Statistics & Data Analysis*, 119:139–154, 2018.

- Victor Picheny, David Ginsbourger, Yann Richet, and Gregory Caplin. Quantile-based optimization of noisy computer experiments with tunable precision. *Technometrics*, 55(1): 2–13, 2013.
- James O Ramsay. *Functional data analysis*. Wiley Online Library, 2006.
- Thomas J. Santner, Brian J. Williams, and William I. Notz. *The Design and Analysis of Computer Experiments, Second Edition*. Springer Verlag, New York, 2018.
- M. Schonlau, W. J. Welch, and D. R. Jones. Global versus local search in constrained optimization of computer models. In N. Flournoy, W. F. Rosenberger, and W. K. Wong, editors, *New Developments and Applications in Experimental Design*, volume 34, pages 11–25. Institute of Mathematical Statistics, 1998.
- Erwin Stinstra, Dick den Hertog, Peter Stehouwer, and Arjen Vestjens. Constrained maximin designs for computer experiments. *Technometrics*, 45(4):340–346, 2003.