Optimal Design of Physical and Numerical Experiments for Computer Code Calibration

Adama Barry, François Bachoc, Sarah Bouquet, Miguel Munoz Munoz Zuniga, Clémentine Prieur

Optimal Design of Physical and Numerical Experiments for Computer Code Calibration

Adama BARRY[a,b], François BACHOC[a], Sarah BOUQUET[b], Miguel MUNOZ ZUNIGA[b,*], Clémentine PRIEUR[c]

[a] *Institut de Mathématiques de Toulouse, Université Paul Sabatier, 31400 Toulouse, France*
[b] *IFP Énergies Nouvelles, 92852 Rueil-Malmaison, France*
[c] *Université Grenoble Alpes, CNRS, Inria, Grenoble INP, LJK, 38000 Grenoble, France*

Abstract

We address the problem of Bayesian calibration of an expensive computer code assumed without model discrepancy. In a calibration process with costly measurement acquisition and high computing time of the computer code, the estimation accuracy and cost mitigation must guide the selection of the design of physical experiments and of numerical experiments. To this end, we propose a hybrid approach to select both designs of physical and numerical experiments with the aim to approximate the posterior density of calibration parameters. We first build an initial Gaussian process emulator which we use to calculate an optimal physical experimental design criterion. Then, after selecting the physical experimental design, we combine physical observations with available computer code evaluations to sequentially add new points to the design of numerical experiments in order to improve the Gaussian process emulator for the calibration purpose. We introduce three new criteria for selecting the design of physical experiments based on posterior density or computer code variation and two new criteria for selecting the design of numerical experiments inspired by the Sequential Uncertainty Reduction (SUR) paradigm. A performance analysis and comparison with state-of-the-art methods is proposed for two test cases and a more realistic one involving the calibration of a harmonic oscillator.

Keywords: Bayesian calibration, design of physical experiments, design of numerical experiments, Gaussian process emulator, information matrix, Kullback-Leibler divergence, history matching, stepwise uncertainty Reduction, simulated annealing.

1. INTRODUCTION

In industry, the development of computer codes is often implemented to study and analyze complex physical phenomena or systems. These computer codes may depend on two types of variables: control/experimental variables and intrinsic parameters. The last ones are often physical

*Corresponding author
Email address: miguel.munoz-zuniga@ifpen.fr (Miguel MUNOZ ZUNIGA)

quantities. The complex nature of these computer codes requires an efficient calibration process, in which adjustment of unknown parameters takes place to improve alignment between computer code outputs and observations of physical phenomena. This calibration process is essential to ensure the accuracy and reliability of the expensive computer code, enabling a more informed understanding of the phenomenon or system of interest. Most of the existing work on Bayesian calibration focuses on building the computer code emulator by selecting the computer simulations to be carried out without worrying about the quality of the physical measurements. Since these measurements are limited by their cost or the difficulty of acquiring them, it would be wise to select them for more effective calibration.Therefore the selection of optimal designs of physical and numerical experiments in the calibration procedure plays a crucial role in achieving robust calibration using Bayesian methods. This paper addresses the problem of calibration through the selection of both the design of physical experiments and of numerical experiments.

Bayesian calibration of computer codes has been the subject of a great deal of research, pioneered by [17], in which the authors introduced the first Bayesian framework for expensive computer code calibration using Gaussian processes. We will refer to this framework as KOH. Their work has paved the way for a deeper understanding of complex computer code calibration. However their approach is known to have certain identifiability issues. [14] developped the KOH framework by exploring the combination of field data and computer simulations for calibration and prediction, highlighting the importance of this fusion for improving model accuracy and better representing the complexity of real phenomena. They also provide a way to circumvent identifiability issues by considering a model discrepancy only when predictions and physical experiments are inconsistent. Subsequently [39] developed an asymptotic theory on the prediction performance of the KOH predictor. Then, from an efficiency point of view, [40] explored computer model fitting via projected kernel calibration, proposing solutions to improve the speed and accuracy of the calibration procedure. Furthermore [10] used the KOH framework to propose an alternative approach that approximates the calibration parameters distribution with knowledge of the physical observations and of the computer code using Kullback-Leibler divergence. The authors have introduced an adaptive two-stage Bayesian optimization algorithm for the sum of squared deviations using a Gaussian process emulator for the computer code, the expected improvement criterion and the information available on the computer code at each iteration. [26] have also proposed a similar approach but approximate the posterior distribution by building a Gaussian process emulator for the likelihood.

The first work on the selection of design of physical experiments dates back to [18], when the authors proposed optimality criteria for estimating the parameters of a linear regression model. These early criteria used a transformation of the Fisher information matrix. The approach was developed and extended by [41, 19, 11]. Subsequently, criteria such as EID-optimality and EIT-optimality were

proposed by [28]. In other papers, authors proposed even more robust criteria ([32]), asymptotic optimality for ED-optimality criterion ([29]) and asymptotic properties for the linear response case ([31]). [1] have proposed a new criterion using the Kullback-Leibler divergence between posterior and prior density to quantify the information contained in a design of physical experiments and proposed a version of stochastic gradient descent algorithm for criterion optimization. More recently, [34] proposed a similar criterion that measures the amount of information using Shanonn's entropy reduction from the prior to the posterior within a sequential approach.

Our contributions can be summarized in three key points. First, we introduce a hybrid method for selecting designs of experiments, incorporating both physical and numerical point acquisition, with the goal of calibration. To the best of our knowledge, there has been no work that addresses both the problem of selecting physical and numerical experiments for the calibration of expensive computer codes. Second we propose three criteria to select the design of physical experiments. The first is based on the posterior covariance of calibration parameters, while the second utilizes a posterior error function on calibration parameters estimation. The last criterion is simple and quick to optimize (sequential greedy approach) and uses the variation of the computer code in relation to calibration parameters combined with a space-filling criterion for physical experiments. Regarding the selection of numerical experiments, we adapt a calibration cost function from the literature ([9]), establishing a connection with an approach based on the Kullback-Leibler divergence criterion also drawn from the literature ([10]). Additionally, two new selection criteria for the design of numerical experiments are proposed, inspired by the paradigm of Step-wise Uncertainty Reduction (SUR). These criteria are based on uncertainty measures using the global prediction error of the physical phenomenon or the posterior variance of calibration parameters.
The following workflow provides a scheme for our hybrid methodology.

This paper is organized as follows. Section 2 introduces notations, Gaussian process emulation and model parameter estimation methods. Section 3 deals with adaptive criteria for the selection of numerical experiments. Section 4 focuses on optimality criteria for the selection of physical experiments. Section 5 presents two analytical test cases in 2D and 4D to compare and assess the performance of the physical and numerical design of experiments selection methods, and a real example involving the calibration of a harmonic oscillator to evaluate the performance of the hybrid method. Finally, Section 6 is devoted to the discussion and conclusions of the paper.

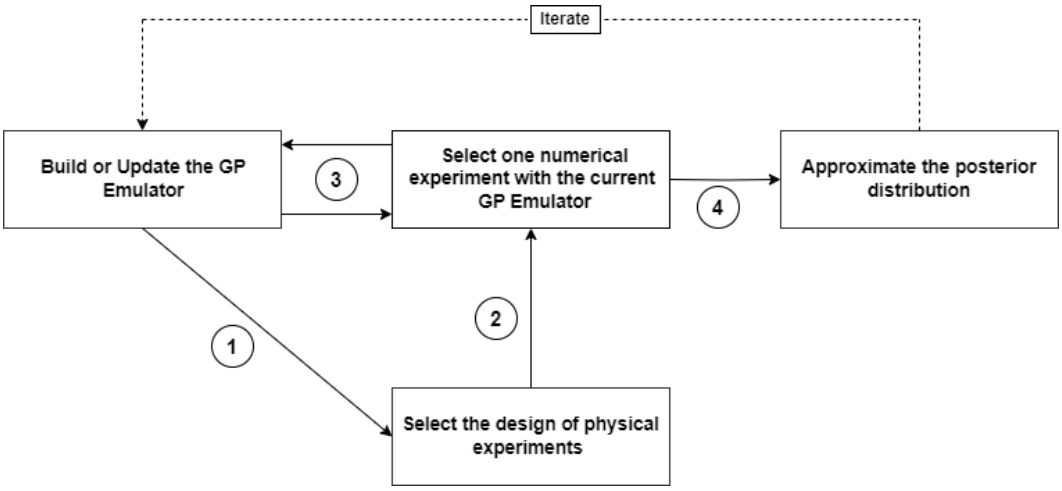

Figure 1: Schematic representation of the proposed hybrid approach for calibration. The process begins by building an initial emulator from a first batch of numerical experiments. After the physical experiments have been selected, they are used in combination with the current emulator to enrich, one point at the time, the design of numerical experiments. At each enrichment the emulator is updated. The final updated version of the emulator is then used to approximate the posterior distribution and estimate calibration parameters. This sequence of steps could be repeated.

2. BACKGROUND

2.1. From prior to posterior : notations and implementation

We consider a computer code that is a parametric function depending on two types of inputs: the control input variables denoted by $x \in \mathcal{X} \subset \mathbf{R}^d$ and a vector of parameters $\theta \in \Theta \subset \mathbf{R}^p$ called calibration parameters. Based on the classical KOH framework, the relation between computer code and physical system is given by the statistical model

$$\mathbf{Y}_{obs}(x) = f_{code}(x, \theta_0) + \varepsilon_x, \tag{1}$$

where $\theta_0 \in \Theta$ is the unknown true vector of calibration parameters and $\varepsilon_x \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ is the measurement error. We assume that the variance σ_ε^2 is known. If it is not the case, it could be considered as an additional parameter handled by our Bayesian framework below. Note that the statistical model (1) is a special case of the KOH model, with model discrepancy $\delta(x) = 0$. In fact, the computer code is supposed to be able to correctly model the physical phenomenon if it is supplied with the true value of the calibration parameter θ_0. This model is also recommended if experimental error can not be distinguised from model discrepancy.

Let $X_{obs} = \left(x^{(1)}, \ldots, x^{(n)}\right)^T$ be the design of physical experiments and $Y_{obs} = \left(y_{obs}^{(1)}, \ldots, y_{obs}^{(n)}\right)^T$ the corresponding physical measurements taken in the field, with $y_{obs}^{(i)} = \mathbf{Y}_{obs}(x^{(i)}), i = 1, \ldots, n$.

Our beliefs or a priori knowledge about the parameters θ_0 can be translated into a probability density called the prior density, which we denote by $\pi : \theta \in \Theta \longmapsto \pi(\theta) \in \mathbf{R}^+$. Using Bayes' theorem, the prior density is updated using physical observations to give the posterior density defined as

$$\pi(\theta \mid Y_{obs}) = \frac{\mathcal{L}(Y_{obs} \mid \theta)\pi(\theta)}{Z} \propto \frac{1}{(2\pi\sigma_\varepsilon^2)^{n/2}} \exp\Big(-\frac{1}{2\sigma_\varepsilon^2} SS(\theta)\Big)\pi(\theta), \tag{2}$$

where $\mathcal{L}(Y_{obs} \mid \theta)$ is the conditional density or the likelihood, $Z = \int_\Theta \mathcal{L}(Y_{obs} \mid \theta)\pi(\theta)d\theta$ is the normalizing constant and $SS(\theta) = \sum_{i=1}^{n} \big(y_{obs}^{(i)} - f_{code}(x^{(i)}, \theta)\big)^2$ the sum of squares of deviations.
The posterior density represents updated knowledge about the calibration parameters given the physical observations. To estimate the calibration parameters posterior density or a summary, we need to sample this distribution by a Markov chain Monte Carlo (MCMC) methods. This requires a large number of calls to the computer code. As the latter is expensive, this approach is not feasible. To reduce the cost, one solution involves the introduction of an emulator. Several emulators have been proposed in the literature, the most common being the Gaussian process (GP) model. GPs are very useful in this context thanks to their multiple advantages: flexibility (covariance kernel choice), fast to run, provide the best unbiased linear predictor and an uncertainty quantifier for each prediction. They are therefore highly suitable for expensive computer code calibration. In [10] and [6], the authors propose using a Gaussian process model to approximate the computer code and thus obtain an approximation of the exact posterior density. Other alternatives are also possible. For example in [26], the authors build a Gaussian process emulator to model the conditional density $\mathcal{L}(Y_{obs} \mid \theta)$ and in [25] a Gaussian process model for the Box-Cox transform of the sum of squares of deviations is adopted.
The accuracy of the posterior density depends on the quality and quantity of physical observations, which are limited by the cost of acquisition. The quality of a design of physical experiments or physical observations is defined as the quantity of information it contains, or the amount of uncertainty it generates. Therefore physical experiments should be selected carefully.

2.2. Gaussian process emulator

In this subsection, we describe Gaussian process emulation of an expensive computer code. A GP is defined as a collection of random variables, any finite number of which have a joint Gaussian distribution ([35]). Formally, a GP is an extension of the familiar Gaussian distribution. We assume that the computer code is a realization of a Gaussian process indexed by the joint space of inputs $\mathcal{X} \times \Theta$ denoted $Y_{code} \sim \mathbf{GP}(m_\beta, k_\psi)$, where

- $m_\beta : (x, \theta) \longmapsto m_\beta(x, \theta) = h(x, \theta)^T\beta$ is the mean function, $\beta \in \mathbf{R}^s$ is the vector of regression parameters and $h(x, \theta) = \big(h_1(x, \theta), \ldots, h_s(x, \theta)\big)^T$ the vector of known regression functions.

- $k_\psi : \big((x,\theta),(x',\theta')\big) \longmapsto k_\psi\big((x,\theta),(x',\theta')\big)$ is the covariance function that encodes our prior about the computer code behavior and ψ represents its hyperparameters. In this paper, we will use the Matérn covariance function with smoothness parameter $\nu = 5/2$

$$k_{5/2,\sigma^2,\phi}(u,u') = \sigma^2\left(1 + \frac{\sqrt{5}\,||\,u-u'\,||}{\phi} + \frac{\sqrt{5}\,||\,u-u'\,||^2}{3\phi^2}\right)\exp\Big(-\frac{\sqrt{5}\,||\,u-u'\,||}{\phi}\Big),$$

with σ^2 the process variance parameter, ϕ the correlation length and $u = (x,\theta)$, $u' = (x',\theta')$ two vectors belonging to $\mathcal{X}\times\Theta$ and $||\,.\,||$ the Euclidean norm.

Let us consider the design of numerical experiments and the corresponding numerical observations:

$$D_M = \Big((x_1,\theta_1),\dots,(x_M,\theta_M)\Big)^T, \quad f_{code}(D_M) = \Big(f_{code}(x_1,\theta_1),\dots,f_{code}(x_M,\theta_M)\Big)^T.$$

We will look at how to select the design of physical experiments X_{obs} and the design of design of numerical experiments in sections 3 and 4. The Gaussian process conditioned by these numerical observations remains Gaussian :

$$Y^M_{code} := \Big[Y_{code} \mid Y_{code}(D_M) = f_{code}(D_M)\Big] \sim \mathbf{GP}\Big(\mu_M, k_M\Big),$$

where μ_M and k_M represent the posterior mean function and the posterior covariance function. Following the properties of Gaussian processes, we have for any $v, v' \in \mathcal{X}\times\Theta$:

$$\begin{aligned}
\mu_M(v) &= \mathbf{E}\Big[Y^M_{code}(v)\Big] = m_\beta(v) + k(v,D_M)\big[k(D_M)\big]^{-1}\big[f_{code}(D_M) - m_\beta(D_M)\big],\\
k_M(v,v') &= \mathbf{Cov}\Big[Y^M_{code}(v), Y^M_{code}(v')\Big] = k(v,v') - k(v,D_M)\big[k(D_M)\big]^{-1}k(D_M,v'),\\
\sigma^2_M(v) &= \mathbf{Var}\Big[Y^M_{code}(v)\Big] = k(v,v) - k(D_M,v)^T\big[k(D_M)\big]^{-1}k(D_M,v),
\end{aligned}$$

where:

$$\begin{aligned}
m_\beta(D_M) &= \Big(m_\beta(x_i,\theta_i)\Big)_{i=1,\dots,M} \quad , \quad k(v,D_M) = \Big(k\big(v,(x_i,\theta_i)\big)\Big)_{i=1,\dots,M},\\
k(D_M,v') &= \Big(k\big((x_i,\theta_i),v'\big)\Big)_{i=1,\dots,M} \quad , \quad k(D_M) = \Big(k\big((x_i,\theta_i),(x_j,\theta_j)\big)\Big)_{i,j=1,\dots,M}.
\end{aligned}$$

Note that the covariance function k depends on parameters $\psi = (\sigma^2,\phi)$, which we have omitted from the notation for the sake of simplicity. We will use the posterior mean function μ_M as a predictor and the posterior variance function σ^2_M to quantify the prediction uncertainty. This predictor is the best linear unbiased predictor in the sense of the mean squared error ([38]).

2.3. Estimation of parameters and posterior density approximation

Let us denote by:

- $m_\beta(X_{obs},\theta) = \Big(m_\beta(x^{(1)},\theta),\dots,m_\beta(x^{(n)},\theta)\Big)^T$ the prior mean prediction on (X_{obs},θ),

- $k(X_{obs}, \theta) = \Big(k((x^{(i)}, \theta), (x^{(j)}, \theta)\Big)_{i,j=1,\dots,n}$ the covariance matrix of (X_{obs}, θ),
- $k\big(D_M, (X_{obs}, \theta)\big) = \Big(k((x_j, \theta_j), (x^{(i)}, \theta)\Big)_{j=1,\dots,M; i=1,\dots,n.}$ the covariance matrix between D_M and (X_{obs}, θ),
- $\mathcal{D} = (f_{code}(D_M)^T, Y_{obs}^T)^T$ the numerical and physical observations.

Following [10], the joint density of the set of observations conditional on model parameters is given by:

$$\mathcal{L}(\mathcal{D} \mid \theta, \beta, \sigma^2, \phi) = \frac{1}{(2\pi)^{(n+M)/2} \mid C \mid^{1/2}} \times \exp\Big\{ -\frac{1}{2}\big[\mathcal{D} - \big(m_\beta(D_M), m_\beta(X_{obs}, \theta)\big)\big]^T C^{-1} \big[\mathcal{D} - \big(m_\beta(D_M), m_\beta(X_{obs}, \theta)\big)\big]\Big\},$$

where

$$C = \begin{bmatrix} k(D_M) & k\big(D_M, (X_{obs}, \theta)\big) \\ k\big(D_M, (X_{obs}, \theta)\big)^T & k\big(X_{obs}, \theta\big) + \sigma_\varepsilon^2 I_n \end{bmatrix}.$$

To estimate the parameters of our emulator, we can maximize the joint density. However, in [10] the authors adopted an alternative, more efficient approach called *the modularization technique* based on the work of [21]. This technique involves estimating the model parameters in two stages. In the first stage, parameters (β, σ^2, ϕ) are estimated by maximizing the marginal density of numerical observations, and then, in the second stage, these estimates are plugged in the conditional density of physical observations to estimate the vector of calibration parameters θ. It is shown in [21] that proceeding this way results in good MCMC sample mixing and it is also shown in [17] that the modularization technique does not damage significantly the estimation of model parameters.

The marginal density of the Gaussian process emulator with parameters (β, σ^2, ϕ) is given by:

$$\mathcal{L}^m\big(f_{code}(D_M) \mid \beta, \sigma^2, \phi\big) = \frac{1}{(2\pi)^{M/2} \mid k(D_M) \mid^{1/2}} \times \exp\Big[-\frac{1}{2}\big(f_{code}(D_M) - m_\beta(D_M)\big)^T k(D_M)^{-1} \big(f_{code}(D_M) - m_\beta(D_M)\big)\Big].$$

Marginal density maximization provides the estimates $(\hat{\beta}, \hat{\sigma}^2, \hat{\phi})$ that are fed back into the conditional density knowing the calibration parameters and numerical observations expressed as follows:

$$\mathcal{L}^c\big(Y_{obs} \mid \theta, f_{code}(D_M)\big) = \frac{1}{(2\pi)^{n/2} \mid k_M\big(X_{obs}, \theta\big) + \sigma_\varepsilon^2 I_n \mid^{1/2}} \times \exp\Big[-\frac{1}{2}\big(Y_{obs} - \mu_M(X_{obs}, \theta)\big)^T \big(k_M\big(X_{obs}, \theta\big) + \sigma_\varepsilon^2 I_n\big)^{-1} \big(Y_{obs} - \mu_M(X_{obs}, \theta)\big)\Big]. \tag{3}$$

where $\mu_M(X_{obs}, \theta)$ and $k_M\big(X_{obs}, \theta\big)$ are respectively the GP posterior mean and posterior covariance matrix, evaluated at (X_{obs}, θ), similarly defined as their prior counterpart at the beginning of section 2.3. We assume that, in the prior, the vector θ and the random vector $Y_{code}(D_M)$ are independent, i.e.

the a priori assumptions about the calibration parameters and the computer code are independent. This allows us to derive an approximation of the posterior density as follows:

$$\begin{aligned}\pi(\theta \mid Y_{obs}, f_{code}(D_M)) &= \frac{\mathcal{L}(Y_{obs}, f_{code}(D_M), \theta)}{\mathcal{L}(Y_{obs}, f_{code}(D_M))} \\ &= \frac{\mathcal{L}^c(Y_{obs} \mid f_{code}(D_M), \theta)\mathcal{L}(f_{code}(D_M))\pi(\theta)}{\int_\Theta \mathcal{L}^c(Y_{obs} \mid f_{code}(D_M), \theta')\mathcal{L}(f_{code}(D_M))\pi(\theta')d\theta'} \\ &= \frac{\mathcal{L}^c(Y_{obs} \mid f_{code}(D_M), \theta)\pi(\theta)}{\int_\Theta \mathcal{L}^c(Y_{obs} \mid f_{code}(D_M), \theta')\pi(\theta')d\theta'} \\ &\propto \mathcal{L}^c(Y_{obs} \mid f_{code}(D_M), \theta)\pi(\theta).\end{aligned} \tag{4}$$

The accuracy of this approximation is highly dependent on the quality of the emulator, and therefore on numerical observations. Hence in the next section, we propose to generate the design of numerical experiments in a goal-oriented and sequential way in order to build efficiently an accurate emulator.

3. OPTIMAL DESIGN OF NUMERICAL EXPERIMENTS

This section describes the selection of the optimal design of numerical experiments. As the computer code is time-consuming, the number of evaluations is limited. Moreover, the quality of the emulator and the accuracy of the calibration depend to a large extent on the quality of the design of numerical experiments. It is therefore essential to choose the numerical experiments carefully. The idea is to adopt a sequential selection approach based on the information available at each iteration. The general principle of sequential selection of numerical experiments is described below:

1. Put a GP prior on the computer code.
2. Update the GP emulator with the available numerical observations.
3. Use a criterion to select the next numerical experiment and evaluate the computer code at this new design point.
4. Given the new evaluation, update the GP emulator.
5. Repeat steps 3-4 until the calibration budget (noted M) is reached.

Next, we present in Sections 3.1 and 3.2 Bayesian approaches from the literature for calibration of computer codes, namely in [10] based on Kullback-Leibler divergence and Bayesian History-Matching. Then in Section 3.3 we leverage two approaches to propose a new criterion. Finally, a new criteria based on the Stepwise Uncertainty Reduction (SUR) paradigm is proposed in Section 3.4.

3.1. Optimal design based on Kullback-Leibler divergence

The Kullback-Leibler divergence is a measure of dissimilarity between two probability distributions. It is denoted as $\mathbf{KL}(p \mid\mid q)$ where q is the reference probability distribution and p is generally the approximate probability distribution, both defined over a set Ω and defined as

$$\mathbf{KL}(p \mid\mid q) = \int_{\Omega} \log\left[\frac{p(x)}{q(x)}\right] p(x)dx.$$

In [10], the authors propose to approximate the posterior density based on the Kullback-Leibler divergence criterion between the true posterior density and the approximated one based on a GP emulator. Thus the design of physical experiments D_M is such that:

$$D_M \in \underset{D\in(\mathcal{X}\times\Theta)^M}{\arg\max}\ \mathbf{KL}\Big[\pi(. \mid Y_{obs}) \| \pi(. \mid Y_{obs}, f_{code}(D))\Big]. \tag{5}$$

Note that this divergence cannot be calculated directly and has no analytical expression. In [10] the following heuristic is used for its computation:

1. Consider a design of the form: $D = D_{M_0} \cup D_{M-M_0}$, where $D_{M_0} \in (\mathcal{X}\times\Theta)^{M_0}$ is the design for building an initial emulator and $D_{M-M_0} \in (X_{obs}\times\Theta)^{M-M_0}$ is the design selected sequentially to reduce the Kullback-Leibler divergence where the control variables are selected among the currently available physical experiments and the computer code parameters are optimally selected

2. Select iteratively $\{\theta_m, m = M_0+1,\dots,M\}$ by solving:

$$\min_{\Theta} SS(\theta), \text{ we recall } SS(\theta) = \sum_{i=1}^{n} \left(y_{obs}^{(i)} - f_{code}(x^{(i)},\theta)\right)^2. \tag{6}$$

Problem (6) results from a term-by-term analysis of the developed expression of the Kullback-Leibler divergence between the posterior distribution and its approximation, and of the interpolation and regularity properties of Gaussian processes considering an a priori uniform distribution. The emulator is built iteratively by solving (6) using the Efficient Global Optimization (EGO) algorithm (see [16]) based on a criterion called Excepted Improvement (EI) combined with some criterion to select $\{x_m \in X_{obs}, m = M_0+1,\dots,M\}$ introduced below. The expression of the EI associated to (6) is given at iteration m as follows:

$$\mathbf{EI}_m(\theta) = \mathbf{E}\Big[(m_m - SS_m(\theta))^+\Big] \in [0, m_m], \tag{7}$$

where $SS_m(\theta) = \|Y_{obs} - Y_{code}^m(X_{obs},\theta)\|^2$, $m_m = \min\{SS(\theta_1),\dots,SS(\theta_m)\}$ denotes the current minimum and we recall that Y_{code}^m is the emulator built using m numerical observations. As we do not have an analytical expression for (7) it is calculated using a Monte Carlo method. We adopt an alternative approach for solving (6) to that adopted in [10]. It consists of varying the Monte Carlo sample size for

calculating the EI_m to carry out the optimization. The method is described in Appendix Appendix C. Finally, at step m the heuristic based on the Kullback-Leibler divergence can be summarized as follows:

$$\theta_{m+1} \in \arg\max_{\Theta} \mathbf{EI}_m(\theta),$$
$$x_{m+1} = \arg\max_{x \in X_{obs}} \mathbf{C}^j_m(x, \theta_{m+1}), j = 1, 2.$$

The two proposed options for the control variables selection criteria are

$$\mathbf{C}^1_m(x, \theta_{m+1}) = \mathbf{V}ar\left[Y^m_{code}(x, \theta_{m+1})\right],$$
$$\mathbf{C}^2_m(x, \theta_{m+1}) = \frac{\mathbf{V}ar\left[Y^m_{code}(x, \theta_{m+1})\right]}{\max\limits_{i=1,\dots,n} \mathbf{V}ar\left[Y^m_{code}(x^{(i)}, \theta_{m+1})\right]} \times \frac{\mathbf{V}ar\left[\mu_m(x, \theta)\right]}{\max\limits_{i=1,\dots,n} \mathbf{V}ar\left[\mu_m(x^{(i)}, \theta)\right]}, \text{ where } \theta \sim \pi(\theta).$$

The first criterion aims to select the point with the highest prediction variance in order to improve the emulator's prediction capability. The second uses the prediction variance combined with the variation of the a posteriori mean with respect to the calibration parameters, with the aim of selecting the point with the least knowledge of the emulator and which provides the most information on θ. Since we select one numerical experiments by iteration, it is necessary to calculate the current minimum differently because the computer code is not evaluated at all the $x^{(i)}, i = 1, \dots, n$ points to calculate $SS(\theta)$. To do this, we consider the posterior exceptation of the sum of the deviations using the emulator as follows:

$$m_m = \min\left\{\mathbf{E}\left[SS_m(\theta_1)\right], \dots, \mathbf{E}\left[SS_m(\theta_m)\right]\right\}$$

where $\mathbf{E}\left[SS_m(\theta)\right] = \mathbf{E}\left[\sum_{i=1}^{n} \left(y^{(i)}_{obs} - \mathbf{Y}^m_{code}(x^{(i)}, \theta)\right)^2\right]$ is estimated from several thousand of GP realizations generated with a Monte Carlo strategy coupled to the Cholesky decomposition of the covariance matrix.

3.2. Optimal design based on Bayesian History Matching

Introduced by [8] for the analysis of expensive computers, History Matching (HM) is a technique developed in the Bayesian computer model literature for finding acceptable inputs to expensive complex models ([8]). The idea is to use a measure of dissimilarity between physical observations and computer code outputs to progressively reduce the parameters domain with respect to prediction accuracy. The main advantage of HM is that it takes into account both model uncertainty and prediction error.

As in [12], there are two approaches to calculate the implausibility of multiple outputs.

- The first is to consider the implausibility per output and retain the maximum which is the worst case. For each output, we have:

$$I_m^{(i)}(\theta) = \frac{\mid y_{obs}^{(i)} - \mu_m(x^{(i)}, \theta) \mid}{\sqrt{\sigma_\varepsilon^2 + \sigma_m^2(x^{(i)}, \theta)}}.$$

  The implausibility metric for the parameter θ is therefore: $I_m(\theta) = \max_{i=1,\dots,n} I_m^{(i)}(\theta)$.

- The second approach is to calculate a multivariate implausibility metric for all outputs. This is defined by the following Mahalanobis distance:

$$I_m(\theta) = \Big(Y_{obs} - \mu_m(X_{obs}, \theta)\Big)^T \Big[k_m\big(X_{obs}, \theta\big) + \sigma_\varepsilon^2 I_n\Big]^{-1} \Big(Y_{obs} - \mu_m(X_{obs}, \theta)\Big)$$

Large implausibilities indicate a parameter set was very unlikely to have produced an output that matched the observational data, given the included uncertainties ([12]). Typically, a threshold T is defined to separate the parameter space into a plausible set and an implausible set. In dimension 1, a threshold $T = 3$ is considered a good choice according to [33]. For multiple outputs, the threshold T can be defined as a quantile of level 95% of a chi-square distribution ([3]). Instead of working only with the implausibility metric, [15] proposed defining a probability of non-implausibility and using entropy of classification based on this probability. The plausible set of parameters $\mathcal{P}$ is formally defined as: $\Big\{\theta \in \Theta : \frac{1}{\sigma_\varepsilon^2} \| Y_{obs} - f_{code}(X_{obs}, \theta) \; ||^2 \leq T\Big\}$, with threshold $T = \chi^2_{95\%}(n)$ is the quantile of level 95% of a chi-square distribution with n degrees of freedom. We then define at step m the probability of non-implausibility as the probability of belonging to the implausible set conditional on D_m (the design of numerical experiments of size m):

$$\begin{aligned} p_m(\theta) &= \mathbf{P}\big(\theta \in \mathcal{P} \mid f_{code}(D_m)\big) \\ &= \mathbf{P}\Big[\big(Y_{obs} - \mathbf{Y}^m_{code}(X_{obs}, \theta)\big)^T \big[k_m\big(X_{obs}, \theta\big) + \sigma_\varepsilon^2 I_n\big]^{-1} \big(Y_{obs} - \mathbf{Y}^m_{code}(X_{obs}, \theta)\big) \leq T\Big] \end{aligned}$$

The probability of non-implausibility will be approximated by Monte Carlo using the conditioned Gaussian process. The entropy of classification, which represents how close we are to certain knowledge and whose expression is given by the following formula:

$$\mathbf{EC}_m(\theta) = -p_m(\theta) \log\big(p_m(\theta)\big) - \big(1 - p_m(\theta)\big) \log\big(1 - p_m(\theta)\big).$$

We can then select the design sequentially so that at iteration m we have

$$D_{m+1} = D_m \cup \big\{(x^{(i)}, \theta_{m+1}), 1 \leq i \leq n\big\} \text{ where } \theta_{m+1} \in \underset{\theta \in \Theta}{\arg\max}\, \mathbf{EC}_m(\theta).$$

In the same way, as in Section 3.1, we use the criteria $\mathbf{C}^j_m, j = 1, 2$ to select x_{m+1} in order to choose one numerical experiment per iteration.

3.3. Optimal design based on Weighted Sum of Squared Deviations

In this section, we propose to permute the two distributions in the Kullback-Leibler divergence formulation (5) of Section 3.1. This makes more sense since the reference distribution becomes the one we want to approximate. We then define D_M as:

$$D_M \in \underset{D\in(\mathcal{X}\times\Theta)^M}{\arg\max}\ \mathbf{KL}\Big[\pi(\theta \mid Y_{obs}, f_{code}(D))\|\pi(\theta \mid Y_{obs})\Big].$$

By adopting a similar reasoning as in [10] (see Appendix Appendix D), we arrive at a modification of the heuristic described in Section 3.1 where in the second step the following calibration cost criterion is used, similar to that proposed in [9]. The second step becomes:

2. Select $\{\theta_{m+1}, m = M_0, \ldots, M-1\}$ of the numerical design of experiments by solving the following optimization problem:

$$\min_{\theta\in\Theta}\quad \big(Y_{obs} - \mu_m(X_{obs},\theta)\big)^T\Big[k_m\big(X_{obs},\theta\big) + \sigma_\varepsilon^2 I_n\Big]^{-1}\big(Y_{obs} - \mu_m(X_{obs},\theta)\big).$$

Thus at each step m of the algorithm, we have:

$$\theta_{m+1} \in \underset{\theta\in\Theta}{\arg\max}\,\mathbf{WSS}_m(\theta),$$
$$x_{m+1} = \underset{x\in X_{obs}}{\arg\max}\,\mathbf{C}_m^j(x,\theta_{m+1}), j = 1,2,$$

where $\mathbf{WSS}_m(\theta) = \|Y_{obs} - \mu_m(X_{obs},\theta)\|_{W_m(\theta)}$, with $W_m(\theta) = k_m\big(X_{obs},\theta\big) + \sigma_\varepsilon^2 I_n$ and $\mathbf{C}_m^j, j = 1,2$ are the criteria defined in Section 3.1. Note that the same criterion ($\mathbf{WSS}_m$) is used to calculate the implausibility metric in the Bayesian History Matching. This establishes the connection between the Kullback-Leibler divergence between the posterior distribution and its approximation ([10]), the Weighted Sum of Squared Deviations criterion ([9]) and the Bayesian History Matching ([12]).

3.4. Optimal design based Stepwise Uncertainty Reduction paradigm

The principle of the Stepwise Uncertainty Reduction (SUR) approach is based on anticipating the impact of the choice of a point to be add to numerical experiments on the uncertainty of the quantity of interest. This anticipated uncertainty is estimated by the expected value of the future uncertainty and computed using Gaussian process regression. Depending on the definition given to the measure of uncertainty, many sequential SUR strategies can be designed to infer different quantity of interest ([7]). As a measure of uncertainty for calibration at step m, we propose the quantities defined as follows:

$$U_m = \mathrm{Tr}\Big[\mathbf{C}ov\big(\theta \mid Y_{obs}, f_{code}(D_m)\big)\Big] \text{ or } U_m = \|Y_{obs} - \mu_m\big(X_{obs},\hat{\theta}_m\big)\|_{W_m(\hat{\theta}_m)},$$

where $\hat{\theta}_m = \mathbf{E}\big[\theta \mid Y_{obs}, f_{code}(D_m)\big]$ is the posterior mean of calibration parameters, $\|u\|_V = u^T V^{-1} u$ is the Mahalanobis distance and we recall $W_m(\theta) = k_m\big(X_{obs},\theta\big) + \sigma_\varepsilon^2 I_n$. The choice of the trace of the

posterior covariance matrix is one option. E.g., one could choose the determinant to take correlations into account.

The SUR criteria for calibration can therefore be defined as:

$$\mathbf{J}_m(x,\theta) = \mathbf{E}_m\big[U_{m+1} \mid x_{m+1} = x, \theta_{m+1} = \theta\big], \text{ where } \mathbf{E}_m(.) = \mathbf{E}\big[. \mid f_{code}(D_m)\big].$$

The calibration algorithm based on the $\mathbf{J}_m$ criterion consists in enriching the design at step m by (x_{m+1}, θ_{m+1}) such that:

$$(x_{m+1}, \theta_{m+1}) \in \underset{(x,\theta)\in\mathcal{X}\times\Theta}{\arg\min}\ \mathbf{J}_m(x,\theta). \tag{8}$$

In practice we reduce (8) to the discrete optimization problem defined as:

$$(x_{m+1}, \theta_{m+1}) \in \underset{(x,\theta)\in X_{obs}\times\Theta}{\arg\min}\ \mathbf{J}_m(x,\theta). \tag{9}$$

We can justify the optimization on the $X_{obs} \times \Theta$ domain rather than on $\mathcal{X} \times \Theta$ by the motivation to learn the relationship between the computer code and the calibration parameters only for the physical experiments, since the posterior distribution uses points $(x, \theta) \in X_{obs} \times \Theta$.

It is well known that computing a SUR criterion can be very time-consuming. We use the technique of importance sampling (see [36]) to approximate it and also propose a metamodeling approach for the optimization of the integral of an expensive function such as problem (9) (see Appendix Appendix C for more details). It should be noted that, given the potential irregular form of the SUR criteria, a good specification of the Gaussian process model (choice of the a priori mean function and the a priori covariance function) and a sufficient number of observations of the criterion will be necessary for the effectiveness. In this context, we opt for an evolutionary optimization algorithm despite the higher time cost involved which has to be mitigated with the effectiveness of the sequential approach.

4. OPTIMAL DESIGN OF PHYSICAL EXPERIMENTS

This section is dedicated to the one-off selection of the design of physical experiments. One-off selection is justified by feasibility in the field. Indeed, it is more practical to make all the physical measurements at the same time and then move on to the computer code calibration procedure. The aim is to select the design of physical experiments that optimizes a certain quality criterion, generally based on the amount of information or the uncertainty on calibration parameters that its choice induces. In the current statistical literature, numerous criteria are used to measure the quality of a design of physical experiments, which can be grouped into two categories: the first based on the Fisher information matrix and the second based on the posterior distribution. In Section 4.1 we recall the definition of criteria based on the information matrix which will be part of the compared strategies in the numerical section. Then in Section 4.2 we recall the definition of a criterion based on the posterior

distribution and propose two new criteria in this family. Finally, in Section 4.3 we introduce a new simple criterion based on computer code variation and fast to evaluate.

4.1. Criteria based on information matrix

Let us recall the non-linear model of physical observation in vector form:

$$\mathbf{Y}_{obs}(X) = f_{code}(X, \theta) + \varepsilon_X, \text{ with } \varepsilon_X \sim \mathcal{N}(0, \sigma_\varepsilon^2 I_n).$$

We assume differentiability with respect to parameter θ. The information matrix is a common measure of the information contained in a design of physical experiments. The Fisher information matrix for the design X at θ is the $p \times p$ matrix defined as follows:

$$\Big[\mathrm{M}(X, \theta)\Big]_{l,k} = \frac{1}{\sigma_\varepsilon^2} \sum_{i=1}^{n} \frac{\partial f_{code}(x^{(i)}, \theta)}{\partial \theta_l} \frac{\partial f_{code}(x^{(i)}, \theta)}{\partial \theta_k}, \text{ for } l, k = 1, \ldots, p.$$

We then define the general form of the criteria based on the Fisher information matrix as

$$\mathbf{C}_\psi^{Minf}(X) = \int_\Theta \psi(\mathrm{M}(X, \theta)) d\theta,$$

where $\psi : \mathbf{M} \in \mathcal{S}_p^+(\mathbf{R}) \mapsto \mathbf{R}$ is a function that transforms a matrix into a scalar, with $\mathcal{S}_p^+$ the set of positive definite symmetric matrices. A classic example of this function is the determinant.
We can also consider the Bayesian information matrix, taking into account a prior information. It can be written as:

$$\mathrm{M}_b(X, \theta) = \mathrm{M}(X, \theta) + \mathrm{M}_0(\theta),$$

where $\mathrm{M}_0(\theta) = \left(\frac{\partial^2}{\partial \theta_l \partial \theta_k} \log \pi(\theta)\right)_{l,k=1,\ldots,p.}$ is called the precision matrix.
The Bayesian version of the criterion is then formulated as follows:

$$\mathbf{C}_\psi^{bMinf}(X) = \int_\Theta \psi(\mathrm{M}_b(X, \theta)) \pi(\theta) d\theta.$$

Criteria based on the Fisher information matrix are abundant in the literature. The most common is the ED-optimality criterion proposed by [41] and extended by [32]. The use of the Fisher information matrix in linear models is justified by its relationship with the asymptotic covariance matrix of the parameter estimators. However, it has a number of inconvenient, the first of which is its locality. In fact, the information matrix does not take into account the non-linearity of the model and uses a local linear approximation. The second is that the transformation of the information matrix does not fully represent the information contained in the physical design of experiments, but rather a summary of it.

4.2. Criteria based on the posterior distribution criteria

The most widely used criterion, based on the posterior distribution, is the Kullback-Leibler divergence criterion proposed by [1]. The higher the Kullback-Leibler divergence between the prior density

and the posterior density, the more informative the design of physical experiments is considered to be. In our context, it is defined as follows:

$$\mathbf{C}_{KL}(X) = \int_{\Theta} \Big[\int_{\mathbf{R^n}} \int_{\Theta} \log \frac{\pi(\theta \mid Y_{sim})}{\pi(\theta)} \pi(\theta \mid Y_{sim}) d\theta \pi(Y_{sim} \mid \theta_0) dY_{sim} \Big] \pi(\theta_0) d\theta_0, \tag{10}$$

where Y_{sim} is a realization of the random vector $Y_{obs}(X, \theta_0) = f_{code}(X, \theta_0) + \varepsilon_X$, knowing θ_0 and $\varepsilon_X \sim \mathcal{N}(0, \sigma_\varepsilon^2 I_n)$.

We propose two new criteria based on the posterior distribution:

- The first focuses on the posterior covariance of calibration parameters as a quantifier of the quality and quantity of information provided by a design of physical experiments.

$$\mathbf{C}_{\psi}^{cov}(X) = \int_{\Theta} \Big[\int_{\mathbf{R^n}} \psi\Big(\mathrm{Cov}\big(\theta \mid Y_{sim}\big) \Big) \pi(Y_{sim} \mid \theta_0) dY_{sim} \Big] \pi(\theta_0) d\theta_0, \tag{11}$$

  where $\psi : \mathbf{M} \in \mathcal{S}_p^+(\mathbf{R}) \to \mathbf{R}$.

- The second considers a measure of the average overall error involved in choosing a design of physical experiments.

$$\mathbf{C}_{\phi}^{loss}(X) = \int_{\Theta} \Big[\int_{\mathbf{R}^n} \int_{\Theta} \phi(\theta, \theta_0) \pi(\theta \mid Y_{sim}) d\theta \pi(Y_{sim} \mid \theta_0) dY_{sim} \Big] \pi(\theta_0) d\theta_0, \tag{12}$$

  where $\phi : \Theta^2 \to \mathbf{R}^+$ is a loss function. An example of a loss function is the Euclidean distance such that $\phi(\theta, \theta_0) = \|\theta - \theta_0\|^2$.

4.3. Computer code variation and space-filling criterion

We also introduce here a sequential criterion for selecting physical experiments. This is motivated by the need for a criterion that is practical and quick to optimise, unlike the previous criteria. The idea behind this is to consider only the behavior of the computer code as a function of the calibration parameter. In addition, we would like the experiments chosen not to be concentrated in a single area but to be fairly well distributed in the experimental space. Hence the idea of considering an additional repartition criterion. For iteration $k = 1, \ldots, n$, we select

$$x^{(k+1)} \in \underset{x \in \mathcal{X} \setminus X_k}{\arg\max}\, \mathbf{C}_k^{CVMm}(x) := \Big[\int_{\Theta} \big[f_{code}(x, \theta) - E_x \big]^2 d\theta \Big] \times \min_{x^{(i)} \in X_k} \|x - x^{(i)}\|,$$

where $E_x = \int_{\Theta} f_{code}(x, \theta) d\theta$ and $X_k = \{x^{(1)}, \ldots, x^{(k)}\}$. The first term of the criterion is used to evaluate the variation of the computer code as a function of θ and the second term is used to measure the proximity of the design at iteration k after the addition of the candidate point. This allows us to choose a space filling design that maximizes the variation of the computer code.

4.4. *Practical consideration on the criterion robustness and optimization*

To increase the robustness of the previous one-shot criteria, we can apply the max-min principle introduced by [32] for the ED-optimality criterion. This consists of considering the worst case according to the nature of the criterion. Indeed, considering the average over all the possible values of the parameters is not robust to the a priori uncertainty of the parameters. Consequently, if the criterion is to be maximized, the worst case corresponds to the minimum over all the possible values of the calibration parameters, and vice versa. Formally, if we have:

$$\mathbf{C}(X) = \int_{\Theta} \mathcal{C}(X, \theta) d\theta.$$

A robust version for this criterion is

$$\mathbf{C}(X) = \max_{\Theta} \mathcal{C}(X, \theta) \text{ or } \min_{\Theta} \mathcal{C}(X, \theta),$$

according to the nature of the criterion $\mathcal{C}(X, \theta)$. We specify that the robust versions will not be used in our numerical experiments.

The computation of the optimality criteria, except for the **CVMm** criterion, is done by importance sampling (details can be found in the Appendix Appendix A). Since their optimization is very time-consuming, we use a combination of the simulated annealing algorithm and a forward greedy optimization algorithm. The forward greedy algorithm provides a better initial solution for simulated annealing, enabling it to find a near-optimal design in relatively few iterations. Our implemented version of the simulated annealing algorithm generates neighbours through line perturbation. Both algorithms can be found in the Appendix Appendix B. Next, we present a numerical study on analytical and real cases to compare and assess the performance of all these methods.

5. NUMERICAL STUDY

5.1. *DOPE strategies evaluation*

5.1.1. *4D test case*

In this section, an analytical example is considered to illustrate the performance of the criteria to select the design of physical experiments. The test function playing the role of computer code is defined as follows:

$$\begin{aligned} f_{code} : [-1,1]^2 \times [0,1]^2 &\to \mathbf{R} \\ (x_1, x_2, \theta_1, \theta_2) &\longmapsto 2x_1 \exp\left(-8\theta_1 x_1^2 - 12\theta_2 x_2^2\right), \end{aligned}$$

This function is cobbled together to create an informative area and a non-informative area of the experimental domain $\mathcal{X} = [-1,1]^2$. An area is considered non-informative when the computer code as

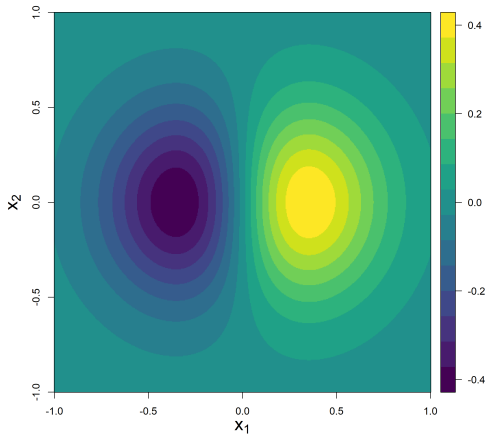

Figure 2: Real physical phenomenon.

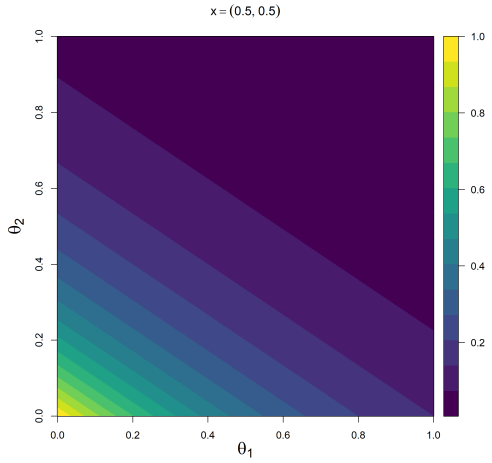

Figure 3: Computer code for $x = (0.5, 0.5)$.

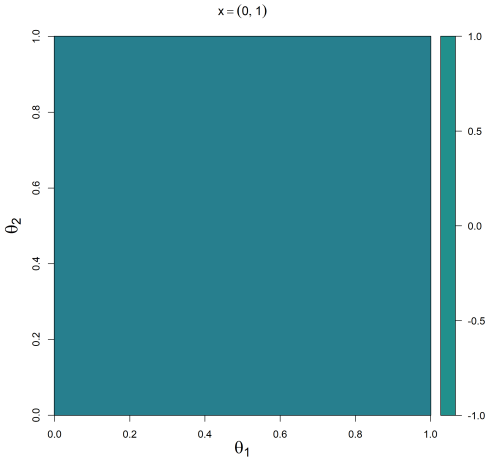

Figure 4: Computer code for $x = (0, 1)$.

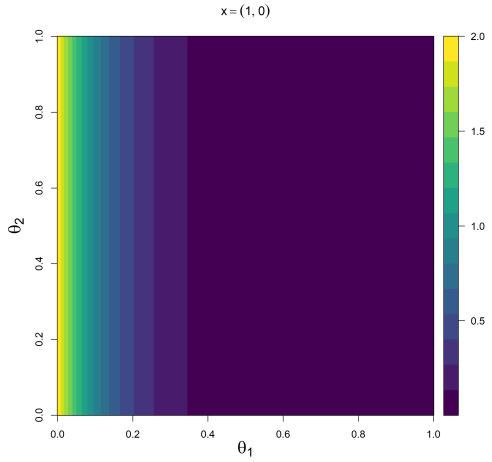

Figure 5: Computer code for $x = (1, 0)$.

a function of θ is almost constant. The true value of the calibration parameters is set to $\theta_0 = (1/2, 1/3)$. Figure 2 shows the behavior in the x-space of the physical phenomenon. Figures 3, 4, and 5 show the computer code as a function of θ for the points $x \in \{(0.5, 0.5), (0, 1), (0, 1)\}$. Here we can see that the points $x = (1, 0)$ and $x = (0.5, 0.5)$ are informative while the point $x = (0, 1)$ is non-informative.

5.1.2. Building the GP emulator

In an industrial application, with a costly to evaluate computer code, an emulator is required in order to reduce, in particular, the computational burden of the criterion optimization. To mimic this real context, we will build an initial GP emulator of the 4D toy example using the R package **DiceKriging** ([37]). The GP mean is chosen to be constant and the covariance function is of type Matérn 5/2. The size of the design of numerical experiments is fixed at $M = 60$. We begin with a Gaussian Process (GP) emulator constructed using observations from an initial set of numerical experiments of size $M_0 = 20$, selected via LHS-maximin. This emulator is improved using observations from $M - M_0$ numerical experiments chosen sequentially using the GP prediction variance. GP hyper-parameters are estimated by maximum likelihood using the BFGS algorithm. The quality of the GP emulator is measured using the predictivity coefficient metric defined as

$$R^2 = 1 - \frac{\sum_{j=1}^{N} \left(f_{code}(x_j, \theta_j) - \mu^M(x_j, \theta_j)\right)^2}{\sum_{j=1}^{M} \left(f_{code}(x_j, \theta_j) - \bar{f}_{code}\right)^2}, \text{ where } \bar{f}_{code} = \frac{1}{N} \sum_{j=1}^{N} f_{code}(x_j, \theta_j).$$

After training the model, we find a predictivity coefficient $R^2 = 76\%$. The role of the initial GP emulator is to enable the optimality criteria to be computed. To do this, it is important that the initial GP emulator has an acceptable coefficient of predictivity ($R^2 \sim 60\%$ for example). That is why we used half of the budget to sequentially select and add points where the predictive variance is high. Another way would be to use the Integrated Mean Squared Error (IMSE) acquisition function (see [38]). Then, taking into account modeling uncertainty in the optimality criteria allows us to have a more relevant design of physical experiments.

5.1.3. Illustration of DOPE strategies

Using the GP emulator, we use the following strategies to select the design of $n = 20$ physical experiments.

- **LHS-maximin**: is the reference strategy for choosing the design of physical experiments by Latin Hybercube Sampling Maximin method in experimental space $\mathcal{X}$; a space-filling method maximizing the minimum distance between pairs of design points.
- **DET**: this strategy consists in selecting the design of physical experiments by optimizing the criterion based on the information matrix defined in Section 4.1 with the function $\psi(M) = \det(M)$.
- **TR**: strategy based on the information matrix defined in Section 4.1 with the function $\psi(M) = \mathrm{tr}(M)$.
- **KL**: strategy based on the Kullback-Leibler divergence criterion defined in (10).

- **SOV**: strategy using the criterion based on the posterior covariance of the calibration parameters defined in (11), taking the function $\psi(M) = \text{tr}(M)$ (i.e. sum of a variances).
- **MSE**: strategy using the criterion based on the posterior covariance of the calibration parameters defined in (11), taking the loss function $\phi(\theta, \theta_0) = \|\theta - \theta_0\|_2^2$ corresponding to the squared error loss.
- **CVMm**: the lowest-cost strategy using the criterion of computer code variation combined with the repartition of the design introduced in Section 4.3.

The graphics in Figure 6 show the repartition of physical experimental designs in the experimental domain for each selection criterion, and in the background we have the variation of the computer code with respect to parameters $V_{code} : x \in \mathcal{X} \longmapsto V_{code}(x) = \int_\Theta \left[f_{code}(x,\theta) - \left(\int_\Theta f_{code}(x,\theta) d\theta \right) \right]^2 d\theta$, to reveal its information area. From Figure 6, we note that the strategies based on the information matrix (**DET** and **TR**) place certain experimental points outside the informative zone, contrarily to the other strategies (**MSE**, **SOV**, **KL** and **CVMm**). However, concerning **MSE**, **SOV** and **KL** strategies, we note a proximity between some of the selected points. Finally, the **CVMm** strategy offers on that example the best repartition with all the points selected in the informative zone which make sense since by definition this criteria is tailored to add points were V_{code} is maximum coupled with an inter-point distance criterion.

5.1.4. Performances of DOPE strategies

To compare and assess the performance of the selection criteria, with each criterion and for two noise levels, we run $L = 30$ times the associated DOPE optimization. The noise levels options are $\sigma_\varepsilon = 5\%\sqrt{V_f}$ or $\sigma_\varepsilon = 10\%\sqrt{V_f}$, with $V_f = \int_{\mathcal{X}} \left[f_{code}(x,\theta_0) - \int_{\mathcal{X}} f_{code}(x,\theta_0) dx \right]^2 dx$. We vary the noise level in order to see its impact on the performance of the criteria. For the performance assessment, we use the following metrics.

- The Mean Squared Error: $\text{MSE} = \frac{1}{L} \sum_{l=1}^{L} \|\theta_0 - \hat{\theta}^{(l)}\|^2$, where θ_0 is the true value of the calibration parameter vector, $\|.\|$ is the Euclidean distance and $\hat{\theta}^{(l)} = \mathbf{E}\left[\theta \mid Y_{obs}^{(l)}\right]$ is the parameters posterior mean given the simulated physical observations $Y_{obs}^{(l)} = f_{code}(X^*, \theta_0) + \varepsilon^{(l)}, l = 1, \ldots, L$ with X^* the optimal design of physical experiments obtained using one of the strategies.
- The Average Length of Credible Interval: $\text{ALCI} = \frac{1}{pL} \sum_{1=1}^{L} \|\hat{\theta}_{sup}^{(l)} - \hat{\theta}_{inf}^{(l)}\|_1$, where $\|u\|_1 = \sum_{i=1}^{p} \mid u_i \mid$ for $u = (u_1, \ldots, u_p)^T \in \mathbf{R}^p$ and $\hat{\theta}_{sup}^{(l)}$ and $\hat{\theta}_{inf}^{(l)}$ are respectively the upper bounds and the lower bounds of $\text{IC}_{90\%}$ the credible interval of level 90%, where $\text{IC}_{90\%}$ is such that $\int_{\text{IC}_{90\%}} \pi(\theta \mid Y_{obs}^{(l)}) d\theta = 0.9$.
- The Coverage Rate: $\text{CR} = \%\theta_0 \in \text{IC}_{90\%}$, where $\text{IC}_{90\%}$ is the credible interval of level 90%.

	$\sigma_\varepsilon^2 = 0.015$			$\sigma_\varepsilon^2 = 0.030$		
	MSE	ALCI	CR	MSE	ALCI	CR
LHS-maximin	0.2647563	0.5842220	100	0.2850127	0.5885662	96.67
CVMm	0.2167279	0.4696518	100	0.2417130	0.5176395	100
DET	0.2211990	0.5379337	83.33	0.2728839	0.6230096	80
TR	0.2202395	0.4859374	96	0.2722238	0.5762567	80
KL	0.2203272	0.4928854	96.67	0.2701196	0.5863462	93
MSE	0.1951027	0.4408275	100	0.2374873	0.5135838	100
SOV	0.2149957	0.4685487	100	0.2534645	0.5360873	100

Table 1: Performance metrics values for DOPE strategies.

Table 1 shows the metrics values for the two noise levels considered. It can be noted that according to the Mean Squared Error (MSE) metric, the **MSE** optimality criterion is the one that performs best for both noise levels. This is not surprising since the criterion is ultimately aimed at minimizing this metric. In second place comes the **SOV** optimality criterion. According to the average length of the credible interval (ALCI) metric and the average coverage rate (CR) metric, the optimality criterion based on the Mean Squared Error (**MSE**) and that of the variation of the code and the design repartition (**CVMm**) performs the best. We note the poor performance in terms of coverage rate (CR) of the two optimality criteria based on the Fisher information matrix, i.e. the **DET** and **TR** optimality criteria. Finally, the cheapest optimality criterion (**CVMm**) performs well overall.

5.2. DONE strategies evaluation

We will illustrate and compare the following strategies to select the design of numerical experiments.

- **LHSCal**: designates the strategy based on a space-filling design generated by the Maximin Latin Hypercube sampling method on $\mathcal{X} \times \Theta$.
- **KLCal**: is a sequential selection strategy using the Kullback-Leibler divergence approach described in Section 3.1.
- **EntropyCal**: this sequential strategy use the entropy of classification based on the probability of implausibility described in Section 3.2.
- **WSSCal**: sequential selection using the weighted sum of square criterion defined in Section 3.3.

- **SURCal1**: sequential selection strategy using the step wise uncertainty reduction criterion based on the trace of posterior covariance of calibration parameters introduced in Section 3.4.

- **SURCal2**: sequential selection strategy using the step wise uncertainty reduction criterion based on the uncertainty of prediction of physical phenomenon by the GP emulator introduced in Section 3.4.

5.2.1. 2D test case: illustration of DONE strategies

The considered test function is defined as follows: $f_{code} : (x, \theta) \in [0,1] \times [0,1] \mapsto x \cos(5\pi\theta x) \in \mathbf{R}$. We use it to illustrate and graphically analyze the distribution of the design of numerical experiments in the experimental domain for each approach. The true value of the calibration parameter is set equal to $\theta_0 = 0.8$. The uniform prior distribution is chosen for the calibration parameter on the interval $[0,1]$. The physical experimental design $X_{obs} = \left(0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1\right)^T$ and the physical observations are generated according to (2) with a noise variance $\sigma_\varepsilon^2 = 0.04^2$. The size of the initial design of numerical experiments chosen by LHS-maximin is set to $M_0 = 20$ and that of the second design of size $M_1 = M - M_0 = 20$ is selected sequentially using one of the strategies described above. A prior Gaussian process with constant mean and Matérn $5/2$ covariance function is chosen for the computer code emulation and its hyperparameters are estimated by maximum likelihood. Figure 7 shows the real physical phenomenon corresponding to the computer code with the true value of the parameter, $\theta_0 = 0.8$, and the noisy physical observations. Figure 8 shows the different trajectories of the computer code for values of parameter θ chosen in a regular grid on $[0,1]$.
Figure 9 shows the repartition of numerical experiments in the joint space $\mathcal{X} \times \Theta$. We observe that **KLCal** strategy selects $\{\theta_m\}_{m=21,\ldots,40}$ around the true value θ_0. Strategies **KLCal, WSSCal** and **ENTCal** select mainly points (red points in Figure 9) with first component at the boundary, that is with $x^{(i)} = 1$ or $x^{(i)} = 0$, whereas the **SUR** strategies (**SURCal1** and **SURCal2**) give less importance to the boundaries. In terms of coverage of the joint space $\mathcal{X} \times \Theta$ the **SUR** strategies do best after **LHSCal**.
Figure 10 shows a comparison of the posterior distributions of the calibration parameters and their approximation for each strategy. It appears that the five proposed strategies give better approximations of the posterior distribution than **LHSCal**.

5.2.2. Performance of DONE strategies

We use the following metrics to appreciate and compare the calibration strategies on the 2D example above.

- The Kullback-Leibler divergence: $\mathbf{KL} = \int_\Theta \log\left[\frac{\pi(\theta \mid Y_{obs}, f_{code}(D_M))}{\pi(\theta \mid Y_{obs})}\right] \pi(\theta \mid Y_{obs}) d\theta$.

- The Predictive Mean Square Error: $\textbf{PMSE} = \int_{\mathcal{X}} \big(Y(x) - f_{code}(x, \hat{\theta}_M)\big)^2 dx$, where $\hat{\theta}_M = \mathbf{E}\big[\theta \mid Y_{obs}, f_{code}(D_M)\big]$.

We repeat the calibration experiment for each strategy $L = 30$ times and calculate the two previous performance metrics. The box-plots in Figure 11 show the distribution of values for the two metrics. According to the **KL** and **PMSE** metrics, the **SUR** and **KLCal** strategies outperform the **LHSCal, WSSCal** and **ENTCal** strategies. Among the sequential strategies, **WSSCal** performed poorly and we note in particular the non-robustness of the **KLCal** strategy with the presence of several outliers.

5.3. Hybrid strategy: harmonic oscillator example

We are interested in the study of a harmonic oscillator (Figure (12)). The number of physical experiments consisting of measuring the position at a time t of the harmonic oscillator to which an object of mass m is attached is limited to $n = 15$. The physical system described by its position satisfies the following second-order differential equation:

$$mY''(m,t) + \theta_2 Y'(m,t) + \theta_1 Y(m,t) = \eta(t), \tag{13}$$

where:

- $Y(m,t)$ represents the position of the object of mass m at time t.
- $\eta(t) = \cos(2t)$ is the external force at time t,
- (m,t) are the control variables, where m takes its values in $\mathcal{M} = \{5, 10, 20, 50, 100, 150\}$, i.e. only objects of mass $m \in \mathcal{M}$ are available and $t \in [0, 100]$ is the observation time,
- $\theta_0 = (\theta_1, \theta_2) = (7, 5)$ are physical constants. θ_1 denotes the stiffness of the spring and θ_2 is the damping coefficient.

We also have a computer code that describes the movement as a function of time of a harmonic oscillator for any value of mass $m \in [1, 150]$, spring stiffness $\theta_1 \in [0, 10]$ and damping coefficient $\theta_2 \in [0, 10]$. This computer code is represented by

$$f_{code} : (m, t, \theta_0, \theta_1) \in [1, 150] \times [0, 100] \times [0, 10]^2 \mapsto f_{code}(m, t, \theta_0, \theta_1) \in \mathbb{R}.$$

Its provides the general solution to the differential equation (13) for any given inputs $(m, t, \theta_0, \theta_1)$. Figure 12 shows the trajectories of the harmonic oscillator as a function of time for the objects of the physical experiment. We can note the low variability for large mass values. For the experiment, we set the variance of the measurement noise at $\sigma_\varepsilon^2 = 0.025$ and the size of the design of numerical experiments to $M = 200$ and that of the initial design to $M_0 = 100$. A constant mean and a Matérn

5/2 covariance function are chosen for the GP emulator building. Half of the initial budget $M_0/2$ is used to build a GP emulator using an LHS-maximin design, and then the prediction variance criterion is used to sequentially enrich the initial design of numerical experiments with the remaining $M_0/2$ points, updating the emulator each time. This GP emulator is used to compute the optimality criteria.

5.3.1. Performance of two hybrid strategies

Three strategies are considered to tackle the calibration problem.

- **LHS-LHSCal**: We select the design of physical experiments by LHS-maximin on the restricted domain $[0, 100] \times \mathcal{M}$ where $\mathcal{M} = \{5, 10, 20, 50, 100, 150\}$ is the discrete set and then we select the design of numerical experiments by LHS-maximin on $[1, 150] \times [0, 100] \times [0, 10]^2$.
- **HybridCal1**: We use a combination of the **SOV** optimality criterion, for selecting the design of physical experiments, and **SURCal1** strategy for selecting the design of numerical experiments. The **SOV** optimality criterion has been optimized using the simulated annealing algorithm for discrete variables presented in the Appendix Appendix B.
- **HybridCal2**: The second hybrid strategy combines the **CVMm** criterion and the **SURCal1** criterion. The advantage of this strategy is its execution time, due to the speed of the **CVMm** criterion.

We use both strategies to solve the calibration problem $L = 35$ times and calculate each time the following metrics:

- The Mean Squared Error: $\mathbf{MSE} = \|\theta_0 - \hat{\theta}_M\|^2$, where $\hat{\theta}_M = \mathbf{E}\big[\theta \mid Y_{obs}, f_{code}(D_M)\big]$.
- The Predictive Mean Squared Error: $\mathbf{PMSE} = \int_{\mathcal{X}} \big(Y(x) - f_{code}(x, \hat{\theta}_M)\big)^2 dx$.

The first metric evaluates calibration accuracy, while the second one evaluates prediction accuracy. Results are presented in Figure 13.

The **HybridCal1** strategy appears as the best strategy from a calibration but also a prediction point of view. However its executation time is quite important. It seems that the **HybridCal2** strategy is a good compromise between accuracy and execution time. We particularly recommend it in a context where the size of the design of physical experiments to be selected is large and the dimension of the experimental domain large, making optimization by simulated annealing difficult.

6. CONCLUSIONS, DISCUSSIONS AND OUTLOOK

In this paper we have introduced a hybrid methodology for selecting the design of physical and numerical experiments for the calibration of expensive computer codes without discrepancy. The

proposed methodology is divided into three phases. The first one consists in building an initial GP emulator for computing the optimality criteria. The second one is to select the optimal physical design of experiments. The optimization algorithm used for this purpose combines simulated annealing and a forward greedy algorithm. This phase results in the most representative posterior distribution of the uncertainty of the calibration parameters for a fixed physical measurement budget. The final phase involves improving the initial GP emulator by sequentially selecting numerical experiments using a suitable criterion, and finally approximating the posterior distribution of the calibration parameters. This sequential selection improves the accuracy of the posterior distribution approximation.

The potential of the criteria to select the optimal design of physical experiments and those for sequential selection of the design of numerical experiments is demonstrated on analytical and a harmonic oscillator test case. The first numerical application on a 4D test case involves analyzing the performance of the proposed optimality criteria and those in the literature based on the Fisher information matrix. The results show that criteria based on the posterior distribution (**MSE** and **SOV**) and those based on computer code variation (**CVMm**) outperform those based on the Fisher information matrix (**DET** and **TR**) and the Maximin Latin Hypercube. Most interesting is the good performance of the **CVMm** criterion, which yields informative experimental designs well distributed in the experimental space, in addition to its optimization speed. The second numerical application on a 2D test case compares the performance of sequential criteria for selecting numerical experiments. Our empirical results demonstrate the good performance of the Kullback-Leibler divergence criterion proposed in [10] and the sequential uncertainty reduction criteria. A final numerical experiment is carried out on a real harmonic oscillator case. Comparative analyses of two hybrid strategies, one combining the **SOV** criterion and **SUR1**, and the other combining the **CVMm** criterion and **SUR1**, reveals significant advantages of the hybrid approach over the reference case, which consists in selecting the physical and numerical experimental designs by Maximin Latin Hypercube. These results encourage the adoption of hybrid strategies for computer code calibration, particularly in contexts where the number of physical measurements and computer code evaluations is limited.

The optimal physical design of experiments construction cost remains high despite the combination of two optimization algorithms (forward optimization algorithm and simulated annealing) to reduce the computation time. For this reason, it would be interesting to explore other iterative approaches to solve the optimization problem. The second optimization challenge that could improved is that of the **SUR** criteria. Indeed, the GP based strategy introduced in the second paragraph of appendix Appendix C is moderately suitable due to the non-stationary behavior of the criteria. Therefore, other metamodels may be more appropriate. Another possibility would be to test the Monte Carlo optimization algorithm described in the first paragraph of appendix Appendix C.

Acknowledgement

This research was conducted with the support of the consortium in Applied Mathematics CIROQUO, gathering partners in technological research and academia in the development of advanced methods for Computer Experiments.

References

[1] A. Abellan and B. Noetinger. Optimizing subsurface field data acquisition using information theory. *Mathematical geosciences*, 42:603–630, 08 2010. doi: 10.1007/s11004-010-9285-6.

[2] M. Abtini. Plans prédictifs à taille fixe et séquentiels pour le krigeage. *Thèse de doctorat de l'Ecole Centrale de Lyon*, 2018.

[3] I. Andrianakis, I. R. Vernon, N. McCreesh, T. J. McKinley, J. E. Oakley, R. N. Nsubuga, M. Goldstein, and R. G. White. Bayesian History Matching of Complex Infectious Disease Models Using Emulation: A Tutorial and a Case Study on HIV in Uganda. *PLOS Computational Biology*, 11 (1):1–18, January 2015. doi: 10.1371/journal.pcbi.1003. URL https://ideas.repec.org/a/plo/pcbi00/1003968.html.

[4] F. Bachoc, M. Ehler, and M. Gräf. Optimal configurations of lines and a statistical application. *Advances in Computational Mathematics*, 43:113 – 126, 2015. URL https://api.semanticscholar.org/CorpusID:36915671.

[5] M. Bayarri, J. Berger, R. Paulo, J. Sacks, A. Kottas, and J. Tu. A framework for validation of computer models. *Technometrics*, 49:138–154, 05 2007. doi: 10.1198/004017007000000092.

[6] M. Carmassi, P. Barbillon, M. Chiodetti, M. Keller, and E. Parent. Bayesian calibration of a numerical code for prediction. *Journal de la société française de statistique*, 160(1):1–30, 2019.

[7] C. Chevalier, D. Ginsbourger, V. Picheny, J. Bect, E. Vazquez, and Y. Richet. Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics*, 56(4):455–465, 2014. ISSN 00401706, 15372723. URL http://www.jstor.org/stable/24587032.

[8] P. S. Craig, M. Goldstein, A. H. Seheult, and J. A. Smith. Pressure matching for hydrocarbon reservoirs: A case study in the use of bayes linear strategies for large computer experiments. In C. Gatsonis, J. S. Hodges, R. E. Kass, R. McCulloch, P. Rossi, and N. D. Singpurwalla, editors, *Case Studies in Bayesian Statistics*, pages 37–93, New York, NY, 1997. Springer New York. ISBN 978-1-4612-2290-3.

[9] X. Dai and P. Chien. Another look at statistical calibration: A non-asymptotic theory and prediction-oriented optimality, 2018. URL https://arxiv.org/abs/1802.00021.

[10] G. Damblin, P. Barbillon, M. Keller, A. Pasanisi, and Ã. Parent. Adaptive numerical designs for the calibration of computer codes. *SIAM/ASA Journal on Uncertainty Quantification*, 6(1): 151âĂŞ179, Jan 2018. ISSN 2166-2525. doi: 10.1137/15m1033162. URL http://dx.doi.org/10.1137/15M1033162.

[11] V. V. Fedorov. Convex design theory 1. *Statistics*, 11:21–43, 1980.

[12] P. Gardner, C. E. Lord, and R. J. Barthorpe. Sequential Bayesian History Matching for Model Calibration. American Society of Mechanical Engineers, 11 2019. doi: {10.1115/vvs2019-5149}. URL https://eprints.whiterose.ac.uk/158406/. Accessed on 2024/04/09.

[13] R. J. Glauber. Time dependent statistics of the Ising model. *Journal of Mathematical Physics*, 4(2): 294–307, 1963.

[14] D. Higdon, M. Kennedy, J. C. Cavendish, J. A. Cafeo, and R. D. Ryne. Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing*, 26(2):448–466, 2004. doi: 10.1137/S1064827503426693. URL https://doi.org/10.1137/S1064827503426693.

[15] P. B. Holden, N. R. Edwards, J. Hensman, and R. D. Wilkinson. Abc for climate: dealing with expensive simulators, 2015.

[16] 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. URL https://api.semanticscholar.org/CorpusID:263864014.

[17] M. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society Series B*, 63:425–464, 02 2001. doi: 10.1111/1467-9868.00294.

[18] J. Kiefer. Optimum experimental designs. *Journal of the Royal Statistical Society. Series B (Methodological)*, 21(2):272–319, 1959. ISSN 00359246. URL http://www.jstor.org/stable/2983802.

[19] J. Kiefer. General Equivalence Theory for Optimum Designs (Approximate Theory). *The Annals of Statistics*, 2(5):849 – 879, 1974. doi: 10.1214/aos/1176342810. URL https://doi.org/10.1214/aos/1176342810.

[20] S. Kirkpatrick, C. Jr. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220 (4598):671–680, 1983.

[21] F. Liu, M. Bayarri, and J. Berger. Modularization in bayesian analysis, with emphasis on analysis of computer models. *Bayesian Analysis*, 4, 03 2009. doi: 10.1214/09-BA404.

[22] A. McHutchon. Differentiating gaussian processes. 2013. URL https://api.semanticscholar.org/CorpusID:6039717.

[23] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. doi: 10.1063/1.1699114. URL http://link.aip.org/link/?JCP/21/1087/1.

[24] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 06 1953. ISSN 0021-9606. doi: 10.1063/1.1699114. URL https://doi.org/10.1063/1.1699114.

[25] M. Munoz Zuniga and D. Sinoquet. Optimization and bayesian approaches for model calibration. application to oil and gas fiel management. Presented at Ateliers de Modélisation de l'Atmosphère (AMA), 2019.

[26] J. Oakley and B. Youngman. Calibration of complex computer simulators using likelihood emulation. *Technometrics*, 59, 03 2014. doi: 10.1080/00401706.2015.1125391.

[27] A. O'Hagan. Bayesian analysis of computer code outputs: A tutorial. *Reliability Engineering and System Safety*, 91(10):1290–1300, 2006. ISSN 0951-8320. doi: https://doi.org/10.1016/j.ress.2005.11.025. URL https://www.sciencedirect.com/science/article/pii/S0951832005002383. The Fourth International Conference on Sensitivity Analysis of Model Output (SAMO 2004).

[28] L. Pronzato. *Synthèse d'expériences robustes pour modéles à paramètres incertains*. PhD thesis, 1986. URL http://www.theses.fr/1986PA112260. Thèse de doctorat dirigée par Walter, ÃL'ric Informatique Paris 11 1986.

[29] L. Pronzato. One-step ahead adaptive D-optimal design on a finite design space is asymptotically optimal. *Metrika*, 71(2):219–238, 2010. doi: 10.1007/s00184-008-0227-y. URL https://hal.science/hal-00396975. available electronically on SpringerLink: http://www.springerlink.com/openurl.asp?genre=article&id=doi:10.1007/s00184-008-0227-y.

[30] L. Pronzato and W. G. Müller. Design of computer experiments: space filling and beyond. *Statistics and Computing*, 22:681–701, 2012.

[31] L. Pronzato and E. Thierry. Sequential experimental design and response optimisation. *Statistical Methods and Applications*, 11:277–292, 2003. URL https://hal.science/hal-01002348.

[32] L. Pronzato and E. Walter. Robust experiment design via stochastic approximation. *Mathematical Biosciences*, 75(1):103–120, 1985. ISSN 0025-5564. doi: https://doi.org/10.1016/0025-5564(85)90068-9. URL https://www.sciencedirect.com/science/article/pii/0025556485900689.

[33] F. Pukelsheim. The three sigma rule. *The American Statistician*, 48(2):88–91, 1994. doi: 10.1080/00031305.1994.10476030. URL https://www.tandfonline.com/doi/abs/10.1080/00031305.1994.10476030.

[34] T. Rainforth, A. Foster, D. R. Ivanova, and F. B. Smith. Modern bayesian experimental design, 2023.

[35] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006. ISBN 026218253X.

[36] C. Robert and G. Casella. *Monte Carlo statistical methods*. Springer Verlag, 2004.

[37] O. Roustant, D. Ginsbourger, and Y. Deville. DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. *Journal of Statistical Software*, 51(1):1–55, 2012. URL https://www.jstatsoft.org/v51/i01/.

[38] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and Analysis of Computer Experiments. *Statistical Science*, 4(4):409 – 423, 1989. doi: 10.1214/ss/1177012413. URL https://doi.org/10.1214/ss/1177012413.

[39] R. Tuo and C. F. Jeff Wu. A theoretical framework for calibration in computer models: Parametrization, estimation and convergence properties. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):767–795, 2016. doi: 10.1137/151005841. URL https://doi.org/10.1137/151005841.

[40] R. Tuo and C. F. J. Wu. Prediction based on the kennedy-o'hagan calibration model: asymptotic consistency and other properties, 2018.

[41] V. V. Fedorov. *Theory of Optimal Experiments Designs*. 01 1972.

Appendix A. Computing the DOPE criteria using the initial emulator

Since the evaluation of the simulation code is very time-consuming, we decided to build a Gaussian process emulator to compute the optimality criteria. Let us denote by $Y_{code}^{M_0}$ the Gaussian process emulator with mean function μ_{M_0} and covariance function k_{M_0} built using the evaluations of the computer code on D_{M_0} an initial design of numerical experiments.

Approximating criteria based on information matrix.. These criteria are approximated by the Fisher information matrix as follows:

$$\Big[\mathbf{M}(X,\theta)\Big]_{l,k} \approx \frac{1}{\sigma_\varepsilon^2}\sum_{i=1}^{n}\frac{\partial \mu_{M_0}(x^{(i)},\theta)}{\partial \theta_l}\frac{\partial \mu_{M_0}(x^{(i)},\theta)}{\partial \theta_k} \text{ for } l,k=1,\dots,p.$$

Note that we can have the explicit formula for the derivation of the posterior mean function of the Gaussian process emulator, thanks to the derivation of covariance function (see [22]).

Approximating criteria based on the posterior distribution.. Two steps are used to approximate the criteria based on the posterior distribution:

1. First, we use the emulator to simulate physical observations for a possible value $\theta_0 \in \Theta$ and for a design of physical experiments X:

$$Y(X,\theta_0) = \mu_{M_0}(X,\theta_0) + \Big[k_{M_0}\big(X,\theta_0\big) + \sigma_\varepsilon^2 I_n\Big]^{1/2}\varepsilon,$$

   where $\varepsilon \sim \mathcal{N}(0, I_n)$.

2. Secondly, we use importance sampling (see [36]) to approximate the integrals (10), (11), and (12), exploiting independence between the prior and the emulator distribution. We draw $\{(\varepsilon^{(l)},\theta_0^{(l)}), l = 1,\dots,L\}$ samples with independent components and respective marginals $\mathcal{N}(0, I_n)$ and $\pi(\theta)$. The $\{\theta_k, k = 1,\dots,K\}$ are sampled from a uniform distribution on Θ. Samples of physical observations are then defined as $Y^{(l)} = \mu_{M_0}(X,\theta_0^{(l)}) + \Big[k_{M_0}\big(X,\theta_0^{(l)}\big) + \sigma_\varepsilon^2 I_n\Big]^{1/2}\varepsilon^{(l)}, l = 1,\dots,L$. For each $(Y^{(l)},\theta_k)$, weights are defined as

$$w_k^{(l)} = \frac{\pi\big(\theta_k \mid Y^{(l)}, f_{code}(D_{M_0})\big)}{\sum_{j=1}^{K}\pi\big(\theta_j \mid Y^{(l)}, f_{code}(D_{M_0})\big)},$$

   where $\pi(. \mid Y^{(l)}, f_{code}(D_{M_0}))$ is the posterior distribution approximated with (4). We define $\tilde{\pi}(\theta_k) = \pi(\theta_k)/\sum_{j=1}^{K}\pi(\theta_j)$, with π the prior distribution. Criteria are then approximated as follows:

   - Kullback-Leibler criterion:

$$\mathbf{C}_{kl}(X) = \int_\Theta \Big[\int_{\mathbf{R}^n}\int_\Theta \log\frac{\pi(\theta \mid Y)}{\pi(\theta)}\pi(\theta \mid Y)d\theta\pi(Y)dY\Big]\pi(\theta_0)d\theta_0 \approx \frac{1}{L}\sum_{l=1}^{L}\sum_{k=1}^{K} w_k^{(l)}\log\frac{w_k^{(l)}}{\tilde{\pi}(\theta_k)}.$$

   - Posterior covariance criteria:

$$\mathbf{C}_\psi^{cov}(X) = \int_\Theta \Big[\int_{\mathbf{R}^n}\psi\Big(\mathrm{Cov}\big(\theta \mid Y\big)\pi(Y)dY\Big]\pi(\theta_0)d\theta_0 \approx \frac{1}{L}\sum_{l=1}^{L}\psi\Big(\sum_{k=1}^{K} w_k^{(l)}(\theta_k - \bar{\theta}_{post}^l)(\theta_k - \bar{\theta}_{post}^l)^T\Big),$$

     where $\bar{\theta}_{post}^l = \sum_{k=1}^{K} w_k^{(l)}\theta_k$ is the posterior mean.

- Posterior error criterion:

$$\mathbf{C}_{\phi}^{loss}(X) = \int_{\Theta}\Big[\int_{\mathbf{R}^n}\int_{\Theta}\phi(\theta,\theta_0)\pi(\theta \mid Y)d\theta\pi(Y)dY\Big]\pi(\theta_0)d\theta_0 \approx \frac{1}{L}\sum_{l=1}^{L}\sum_{k=1}^{K} w_k^{(l)}\phi(\theta_k,\theta_0^{(l)}).$$

Approximating criteria based on computer code. We replace the computer code with the emulator, taking into account modeling errors. More precisely the criteria:

$$\mathbf{C}_k^{cvMm}(x) = \int_{\mathbf{R}}\int_{\Theta}\big[\mu_{M_0}(x,\theta) + \sigma_{M_0}(x,\theta)\varepsilon_x - E_x\big]^2 d\theta d\varepsilon \times \min_{x^{(i)}\in X_k} || \, x - x^{(i)} \, ||,$$

with $E_x = \int_{\Theta}\mu_{M_0}(x,\theta)d\theta$ and $X_k = \{x^{(1)},\ldots,x^{(k)}\}$ the design at step k, has the following Monte Carlo based aproximation:

$$\mathbf{C}_k^{cvMm}(x) \approx \frac{1}{L}\sum_{l=1}^{L}\big[\mu_{M_0}(x,\theta_l) + \sigma_{M_0}(x,\theta_l)\varepsilon_l - \tilde{E}_x\big]^2 \times \min_{x^{(i)}\in X_k} || \, x - x^{(i)} \, ||,$$

with $\theta_1,\ldots,\theta_L$ i.i.d. $\mathcal{N}(0,1)$ samples and $\varepsilon_1,\ldots,\varepsilon_L$ i.i.d. samples from the uniform on Θ, independent from each other, and with $\tilde{E}_x = \frac{1}{L}\sum_{l=1}^{L}\mu_{M_0}(x,\theta_l)$.

Appendix B. Optimization algorithm for the DOPE

Forward Optimization Algorithm. Introduced in [2], the forward optimization algorithm is a global optimization algorithm under certain conditions called submodularity conditions. However, it provides a local optimum when the submodularity property is not verified. For iteration $k \in \{1,\ldots,n\}$, with n the size of the DOPE, we determine

$$x_{k+1}^* \in \arg\max_{x\in\mathcal{X}\setminus X_k} \mathbf{C}(X_k \cup \{x\}), \tag{B.1}$$

and we update the design matrix $X_{k+1} = X_k \cup \{x_{k+1}^*\}$. The final solution corresponds to X_n.
The advantage of this algorithm is that the criterion can be evaluated in parallel on a grid by writing the problem (B.1) in discrete form:

$$x_{k+1}^* \in \arg\max_{x\in G_k\setminus X_k} \mathbf{C}(X_k \cup \{x\}) \text{ where } G_k \subset \mathcal{X} \text{ is a grid.}$$

Solving optimization problem with a variant of Simulated Annealing. We use a variant of simulated annealing to solve optimization problems of the form:

$$X^* \in \arg\max_{X\in\mathcal{X}^n} \mathbf{C}(X), \text{ with } \mathbf{C} \text{ an optimality criterion.}$$

We propose to solve this very time consuming optimization problem by a combination of Forward Optimization Algorithm and Simulated Annealing. The two main ingredients are: first, to use the Forward

Optimization Algorithm to find a local optimum that will serve as the initial design for simulated annealing; second, to draw a neighborhood by perturbing a line at each iteration of Simulated Annealing. The idea of line perturbation, inspired from [2], is based on the fact that it is very difficult to reach the optimal solution by perturbing the entire $n \times d$ matrix. The advantage of simulated annealing is that it requires neither gradient nor Hessian calculations. Algorithm 1 below summarizes the procedure.

Algorithm 1: Simulated Annealing Optimization Algorithm

Input: $\mathcal{X}$ the experimental domain, $\mathbf{C}$ the optimality criterion, n the number of physical experiments, T_0 the initial temperature, k_{max} the maximum number of iterations (we have set $k_{max} = 1000 \times n$). and X_0 the initial matrix (provided by the Forward Optimization Algorithm).

1. **While** $0 \leq k \leq k_{max}$ **do**:

   - Generate by line perturbation X_{prop} a neighbour of X_k such that:

$$\forall i = 1, \ldots, n \quad X_{prop}(i,) = \begin{cases} X_k(i,) & \text{if } i \neq i^* \\ x_v \in \mathcal{V}(X_k(i,)) & \text{otherwise,} \end{cases}$$

     where i^* is the remainder of the Euclidean division of k by n (take $i^* = n$ when the remainder is zero), and $\mathcal{V}(x)$ represents a neighbor of x belonging to $\mathcal{X}$. We use neighborhood generation by Gaussian perturbation $\mathcal{V}(x) = \{x_v \in \mathcal{X}, x_v \sim \mathcal{N}(x, \sigma_{SA}^2 I_d)\}$, where σ_{SA}^2 is the variance hyper-parameter, whose square root is set equal to twenty percent of the minimum length of the intervals constituting the block containing the Θ domain.

   - Evaluate degradation $\Delta\mathbf{C}_k = \mathbf{C}(X_k) - \mathbf{C}(X_{prop})$.
   - Calculate the acceptance probability using the Metropolis scheme $p = \min(e^{-\Delta\mathbf{C}_k / T_k}, 1)$.
   - Generate $u \sim \mathcal{U}_{[0,1]}$.
   - Accept-reject step:

$$X_{k+1} \longleftarrow \begin{cases} X_{prop} & \text{if } p \geq u. \\ X_k & \text{else.} \end{cases}$$

   - Update the temperature $T_{k+1} = cT_{k-1}$ with $0 < c < 1$.
   - Update $k \longleftarrow k + 1$.

   End While.

Output: $X_{k_{max}}$

For a fixed initial acceptance probability P_0, the initial temperature T_0 is defined as: $T_0 = -\Delta\mathbf{C} / \log P_0$, with $\Delta\mathbf{C}$ chosen as the 90% quantile of a set of function degradations computed from a

set of perturbations of the initial solution (X_0).

Simulated Annealing in presence of one or more discrete variables. To solve the optimization problem when one of the experimental variables is discrete, the mass in our example of Section 5.3, we need to adapt the simulated annealing algorithm by defining a proposal probability distribution for discrete variables.

Proposal probability distribution for discrete variables Let $S = \{x_1, \dots, x_N\}$ be the set of possible values for the discrete variables. We define $p_{i,j}$ the probability of proposing x_j as a neighbor of state x_i as follows:

- for $i = 3, \dots, N-2$: $p_{i,j} = \begin{cases} 1/7 & \text{if } j \in \{i-2, i, i+2\}, \\ 2/7 & \text{if } j \in \{i-1, i+1\}, \\ 0 & \text{else;} \end{cases}$
- for $i \in \{1, 2, N, N-1\}$:

$$p_{1,j} = \begin{cases} 1/4 & \text{if } j \in \{1,3\} \\ 1/2 & \text{if } j = 2 \\ 0 & \text{else} \end{cases} \qquad p_{2,j} = \begin{cases} 1/3 & \text{if } j \in \{1,3\}, \\ 1/6 & \text{if } j \in \{2,4\}, \\ 0 & \text{else} \end{cases}$$

$$p_{N-1,j} = \begin{cases} 1/3 & \text{if } j \in \{N-2, N\}, \\ 1/6 & \text{if } j \in \{N-3, N-1\}, \\ 0 & \text{else} \end{cases} \qquad p_{N,j} = \begin{cases} 1/4 & \text{if } j \in \{N, N-2\}, \\ 1/2 & \text{if } j = N-1, \\ 0 & \text{else.} \end{cases}$$

More details for the harmonic oscillator test case For that example, the experimental mass can take values in $\mathcal{M} = \{m_1 := 5, m_2 := 10, m_3 := 20, m_4 := 50, m_5 := 100, m_6 := 150\}$, while time is continuously valued in $\mathcal{T} = [0, 150]$. The neighborhood diagram for the discrete mass variable is drawn in Figure B.14.

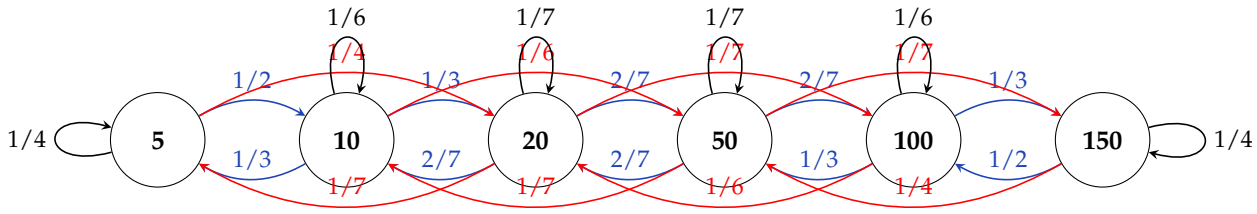

Figure B.14: Neighborhood diagram for the mass.

The corresponding probability matrix and design matrix are:

$$\mathbf{P}_m = \begin{bmatrix} 1/4 & 1/2 & 1/4 & 0 & 0 & 0 \\ 1/3 & 1/6 & 1/3 & 1/6 & 0 & 0 \\ 1/7 & 2/7 & 1/7 & 2/7 & 1/7 & 0 \\ 0 & 1/7 & 2/7 & 1/7 & 2/7 & 1/7 \\ 0 & 0 & 1/6 & 1/3 & 1/6 & 1/3 \\ 0 & 0 & 0 & 1/4 & 1/2 & 1/4 \end{bmatrix}, \quad X = \begin{bmatrix} t^{(1)} & m^{(1)} \\ t^{(2)} & m^{(2)} \\ \vdots & \vdots \\ t^{(n)} & m^{(n)} \end{bmatrix} \in (\mathcal{T} \times \mathcal{M})^n.$$

For a mass value $m_i \in \mathcal{M}$ and an observation time $t \in \mathcal{T}$, the set of neighbours is:

$$\mathcal{V}((t, m_i)) = \{(t_v, m_v) \in \mathcal{T} \times \mathcal{M} : t_v \sim \mathcal{N}(t, \sigma_{SA}^2) \text{ and } m_v = m_j \in \mathcal{M} \text{ with probability } p_{i,j} = [\mathbf{P}_m]_{i,j}\}.$$

Appendix C. Optimization algorithm for the DONE

Gradual Monte-Carlo Optimization on Grid. We propose an algorithm inspired by the one introduced in [4], particularly useful when the objective function is defined from a Monte Carlo approximation. We formalize the optimization problem as follows:

$$\theta^* \in \arg\max_{\theta \in \Theta} f_{obj}(\theta, L), \text{ with } L \text{ is the Monte-Carlo sample size,} \tag{C.1}$$

We now describe the two step algorithms we propose to solve (C.1). Let $G_1 = [\theta_1, \dots, \theta_{N_1}] \subset \Theta$ be a grid of size N_1.

1. Evaluate the objective function on G_1 with a Monte-Carlo size $L = L_1$ and determine the sub-grid $G_2 = [\theta_{(1)}, \dots, \theta_{(N_2)}] \subset G_1$ such that:

$$f_{obj}(\theta, L_1) \leq f_{obj}(\theta_{(i)}, L_1) \quad \forall i = 1, \dots, N_2 \text{ and } \forall \theta \in G_1 \setminus G_2.$$

2. Evaluate the objective function on G_2 with a Monte-Carlo size $L = L_2 > L_1$ and determine the solution $\tilde{\theta}^* \in \arg\max_{\theta \in G_2} f_{obj}(\theta, L_2)$.

The advantage of this optimization algorithm is its simplicity of implementation and its speed of execution due to the possibility of evaluating the objective function in parallel. For our implementation we have set $N_1 = 10^{p+1}$, $N_2 = \max\{100, 10^{p-1}\}$ where $p = \dim(\Theta)$, $L_1 = 10^3$ and $L_2 = 10^4$.

Bayesian Optimisation of the integral of an expensive function. For the sake of generality, we consider $I : x \in \mathcal{X} \mapsto \int_{\mathcal{X}} g(x, y)dy \in \mathbf{R}$, where $g : (x, y) \in \mathcal{X} \times \mathcal{X} \longmapsto g(x, y) \in \mathbf{R}$ is an expensive to evaluate function. The goal is to solve the following optimization problem:

$$\arg\min_{x \in \mathcal{X}} I(x) = \int_{\mathcal{X}} g(x, y)dy. \tag{C.2}$$

The idea is to build a GP emulator for g from which we deduce a GP emulator of I. To model the function I, we place a prior on the expensive function g as a realization of a Gaussian process $Z \sim \mathbf{GP}(\mu^g, k^g)$. The observation vector is denoted by $g(D_n) = \Big[g((x_1, y_1)), \ldots, g((x_n, y_n))\Big]$, where $D_n = \Big((x_1, y_1), \ldots, (x_n, y_n)\Big)$ is the design of experiments. The approximation of $I(x)$ is given by the Gaussian process conditioned on $g(D_n)$: $\hat{I}_n \sim \mathbf{GP}\big(\tilde{\mu}_n, \tilde{k}_n\big)$. Thanks to the properties of Gaussian distributions, the posterior mean function and the posterior covariance function $\tilde{\mu}_n, \tilde{k}_n$ are given by:

$$\begin{aligned}
\tilde{\mu}_n(x) = \mathbf{E}\big[I(x) \mid g(D_n)\big] &= \int_{\mathcal{X}} \mathbf{E}\big[Z(x,u) \mid g(D_n)\big] du = \int_{\mathcal{X}} \mu_n^g(x,u) du, \\
\tilde{k}_n(x, x^{'}) = \mathbf{Cov}\big[I(x), I(x^{'}) \mid g(D_n)\big] &= \int_{\mathcal{X}} \int_{\mathcal{X}} \mathbf{Cov}\big(Z(x,u), Z(x^{'}, v) \mid g(D_n)\big) du dv \\
&= \int_{\mathcal{X}} \int_{\mathcal{X}} k_n^g\big((x,u),(x^{'},v)\big) du dv.
\end{aligned}$$

If $\tilde{\mu}_n(x)$ and $\tilde{k}_n(x, x^{'})$ cannot be computed analytically, it is possible to use Monte Carlo approximation. Now at iteration n:

- Select (x_{n+1}, y_{n+1}) as follows:
  - $x_{n+1} \in \arg\max_{\mathcal{X}} \tilde{\sigma}_n^2(x)$ or $x_{n+1} \in \arg\max_{\mathcal{X}} EI_n(x)$, with $EI_n(x) = \mathbf{E}\big[(\tilde{m}_n - \hat{I}_n(x))^+\big]$, with $\tilde{m}_n = \min_{i=1,\ldots,n} \tilde{\mu}_n(x_i)$ the current minimum;
  - $y_{n+1} \in \arg\max_{\mathcal{X}} \sigma_n^2(y, x_{n+1})$.
- Update the design of experiments $D_{n+1} = D_n \cup \{(x_{n+1}, y_{n+1})\}$;
- Update the evaluations $g(D_{n+1}) = g(D_n) \cup \{g(x_{n+1}, y_{n+1})\}$ and the GP emulator on $g(D_{n+1})$.

Since $\hat{I}_n(x) \sim \mathcal{N}(\tilde{\mu}_n(x), \tilde{\sigma}_n^2(x))$, we have the following analytical expression (see [16]):

$$EI_n(x) = (\tilde{m}_n - \tilde{\mu}_n(x))\Phi\Big(\frac{\tilde{m}_n - \tilde{\mu}_n(x)}{\tilde{\sigma}_n(x)}\Big) + \tilde{\sigma}_n(x)\phi\Big(\frac{\tilde{m}_n - \tilde{\mu}_n(x)}{\tilde{\sigma}_n(x)}\Big),$$

where Φ, ϕ are respectively the cumulative distribution function and the probability distribution function of the standardized Gaussian distribution.

Appendix D. Weighted Sum of Square criterion

Recall that our objective is to select the design of experiments $D_M = \{(x_i, \theta_i)\}_{i=1}^M$. To do this, we consider the Kullback-Leibler divergence between the two densities, this time with the one we're trying to approximate as the reference density. Note that this amounts to permuting the two densities in the [10] criterion defined in equation (5). Consider

$$\mathbf{KL}\Big[\pi\big(. \mid Y_{obs}, f_{code}(D_M)\big) \mid\mid \pi(. \mid Y_{obs})\Big] = \int_{\Theta} \log\Big[\frac{\pi(\theta \mid Y_{obs}, f_{code}(D_M))}{\pi(\theta \mid Y_{obs})}\Big] \pi(\theta \mid Y_{obs}, f_{code}(D_M)) d\theta. \tag{D.1}$$

Our aim is to minimize criterion defined in (D.1). We now provide heuristic arguments to explain how this criterion is related to the cost function defined in [9, Section 4] written as a Mahalanobis norm of predicted deviations:

$$\|Y_{obs} - \mu_M(X_{obs}, \theta)\|^2_{W_M(\theta)}, \text{ where } W_M(\theta) = k_M\big(X_{obs}, \theta\big) + \sigma_\varepsilon^2 I_n. \tag{D.2}$$

The decomposition of equation (D.1) leads to:

$$\mathbf{KL}\Big[\pi\big(. \mid Y_{obs}, f_{code}(D_M)\big) \mid \pi(. \mid Y_{obs})\Big] = C_z + \int_\Theta \big(C_1 - C_3^M(\theta) + C_2(\theta) - C_4^M(\theta)\big)\pi(\theta \mid Y_{obs}, f_{code}(D_M))d\theta, \tag{D.3}$$

$$\begin{aligned}
\text{with } C_1 =& n\log(\sigma_\varepsilon), \quad C_2(\theta) = \frac{1}{2\sigma_\varepsilon^2}\big(Y_{obs} - f_{code}(X_{obs}, \theta)\big)^T\big(Y_{obs} - f_{code}(X_{obs}, \theta)\big),\\
C_3^M(\theta) =& \frac{1}{2}\log\big[\mid k_M\big(X_{obs}, \theta\big) + \sigma_\varepsilon^2 I_n \mid\big],\\
C_4^M(\theta) =& \frac{1}{2}\big(Y_{obs} - \mu_M(X_{obs}, \theta)\big)^T\big(k_M\big(X_{obs}, \theta\big) + \sigma_\varepsilon^2 I_n\big)^{-1}\big(Y_{obs} - \mu_M(X_{obs}, \theta)\big),
\end{aligned}$$

and $C_z = \log(\frac{Z}{Z^M})$, where Z, Z_M are respectively the normalization constants of posterior density and its approximation. We adopt the same reasoning as in [10]. The constants C_z do not offer any options for selecting numerical experiments. First, to reduce the Kullback-Leibler divergence we focus on the integral in equation (D.3). We have an integration of terms weighted by the posterior density approximation $\pi(\theta \mid Y_{obs}, f_{code}(D_M))$. The C_1 and $C_2(\theta)$ terms do not depend on the design of numerical experiments D_M. Small values of $C_3^M(\theta)$ and $C_4^M(\theta)$ correspond to large values of the posterior density approximation (see Eq. (4) and (3) in Section 2.3), and therefore to a reduction of the integral. Second, the idea of heuristics is to consider the designs of numerical experiments of the form $D_M \subset X_{obs} \times \Theta$ which contain the numerical experimental points of type $(x^{(i)}, \theta), i = 1, \ldots, n$, where θ is selected in the area where the posterior density approximation takes large values. The choice of points $\{x_m \in X_{obs}, m = 1, \ldots, M\}$ results in a reduction of the Kullback-Leibler divergence, as we will see. Indeed, the predictor given by the Gaussian process emulator being an exact interpolator then $k_M(X_{obs}, \theta) = 0$ and $\mu_M(X_{obs}, \theta) = f_{code}(X_{obs}, \theta)$, for values of θ in D_M and this remains almost true ($k_M(X_{obs}, \theta') \approx 0$) for all θ' in the neighborhood of θ thanks to the regularity properties of Gaussian processes. Therefore for elements of D_M the differences $C_1 - C_3^M(\theta)$ and $C_2(\theta) - C_4^M(\theta)$ cancel out and the Kullback-Leibler divergence is reduced. This finally leads to sequentially select the θ with the purpose of maximizing the approximation of the posterior density. This gives the following optimization problem:

$$\max_{\theta \in \Theta} \quad \pi\big(\theta \mid Y_{obs}, f_{code}(D_M)\big).$$

By applying the logarithm, we obtain the optimization problem:

$$\min_{\theta\in\Theta} \quad \big(Y_{obs} - \mu_M(X_{obs},\theta)\big)^T \Big[k_M\big(X_{obs},\theta\big) + \sigma_\varepsilon^2 I_n\Big]^{-1} \big(Y_{obs} - \mu_M(X_{obs},\theta)\big) - \log(\pi(\theta)).$$

Using a uniform prior $\pi(\theta) = \mathcal{U}_\Theta$, the problem becomes:

$$\min_{\theta\in\Theta} \quad \big(Y_{obs} - \mu_M(X_{obs},\theta)\big)^T \Big[k_M\big(X_{obs},\theta\big) + \sigma_\varepsilon^2 I_n\Big]^{-1} \big(Y_{obs} - \mu_M(X_{obs},\theta)\big).$$

We thus retrieve the cost function defined in [9, Section 4] recalled in (D.2). The way this cost function is introduced in [9] is different. They prove that their problem of prediction oriented calibration is equivalent to finding the minimizer of the model discrepancy under a reproducing kernel Hilbert space (RKHS) norm.

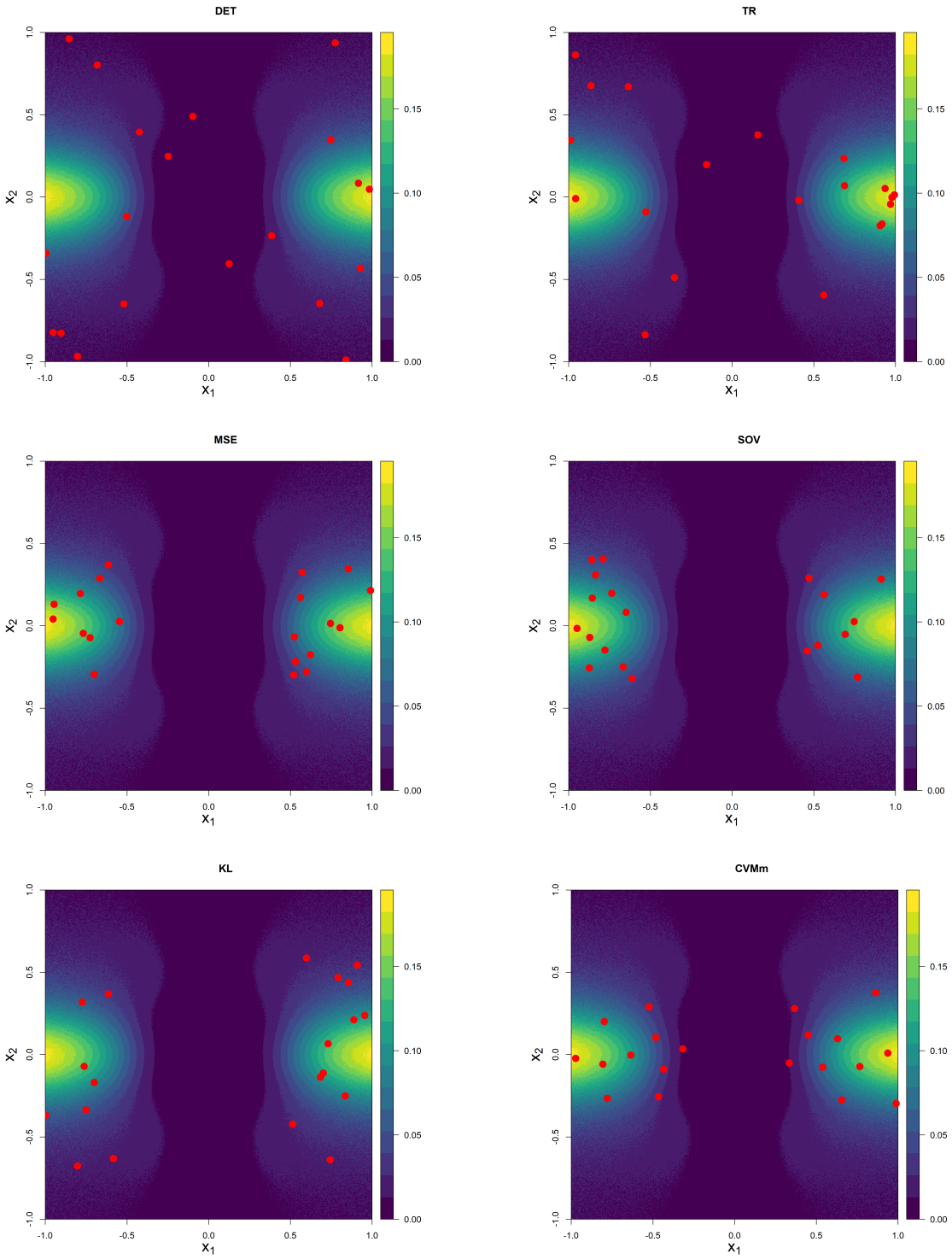

Figure 6: Designs of physical experiments selected using the **DET, TR, RMSE, SOV, KL and CVMm** criterion. In the background is the computer code variation graphic showing the zones where the points are placed by each strategy.

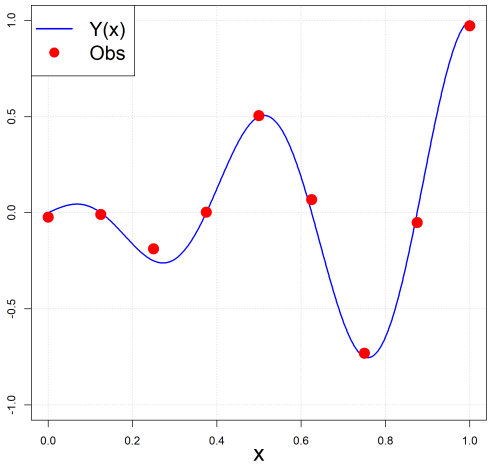

Figure 7: Physical phenomena and observations.

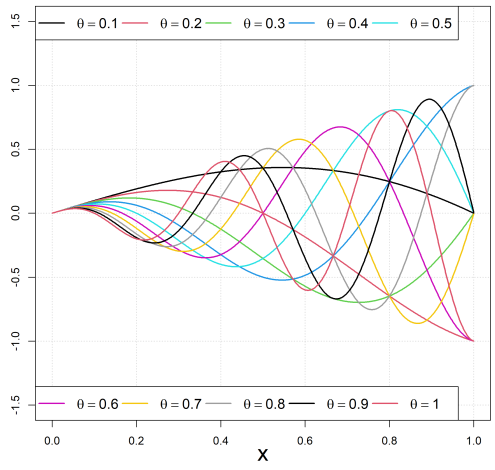

Figure 8: Computer code for
$\theta \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$.

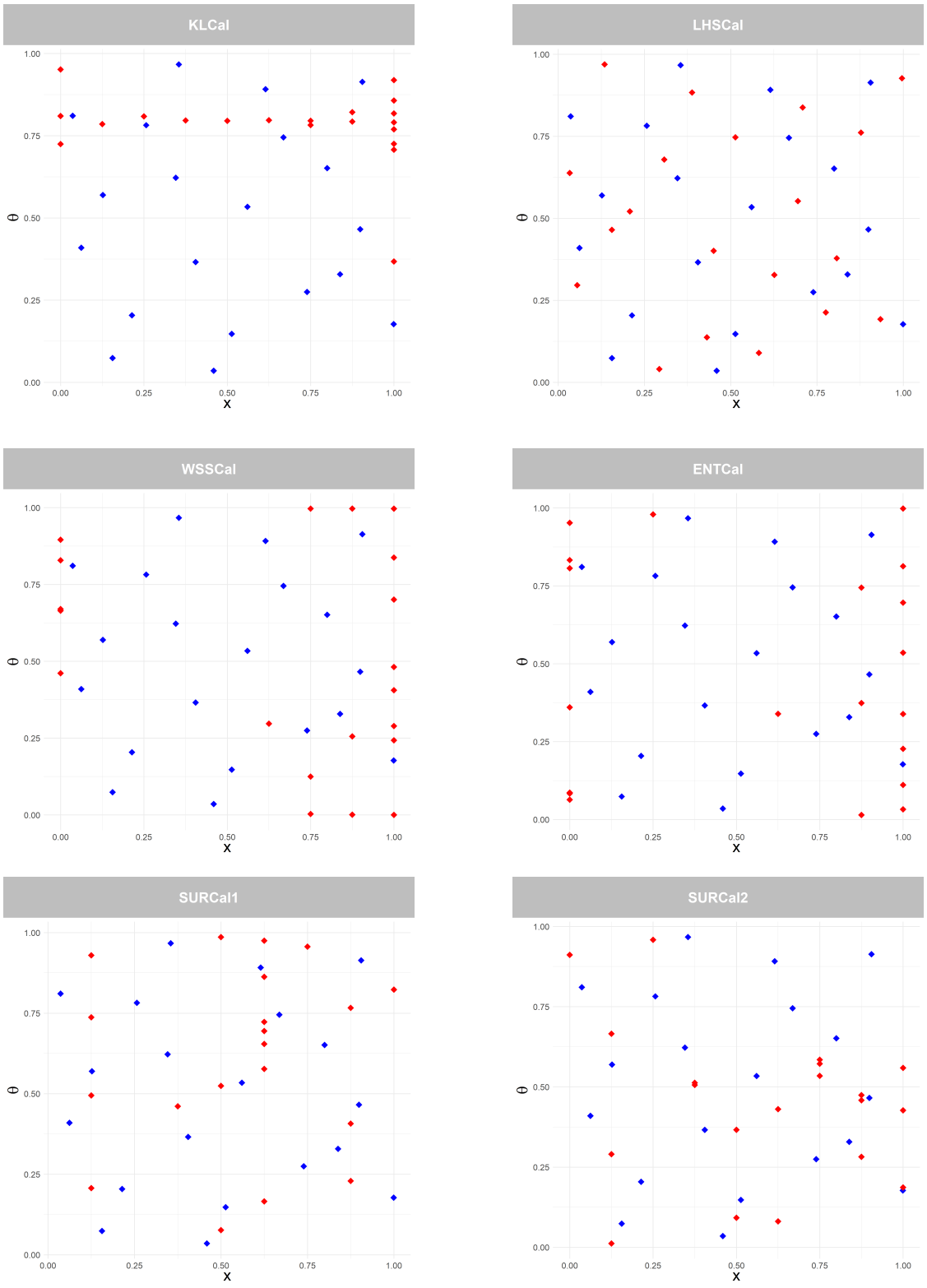

Figure 9: Design of numerical experiments for each strategy. Initial LHS-maximin design (blue dots) and sequential design (red dots). Note that for the **LHSCal** strategy, the design of numerical experiments is selected in one-shot and not sequentially.

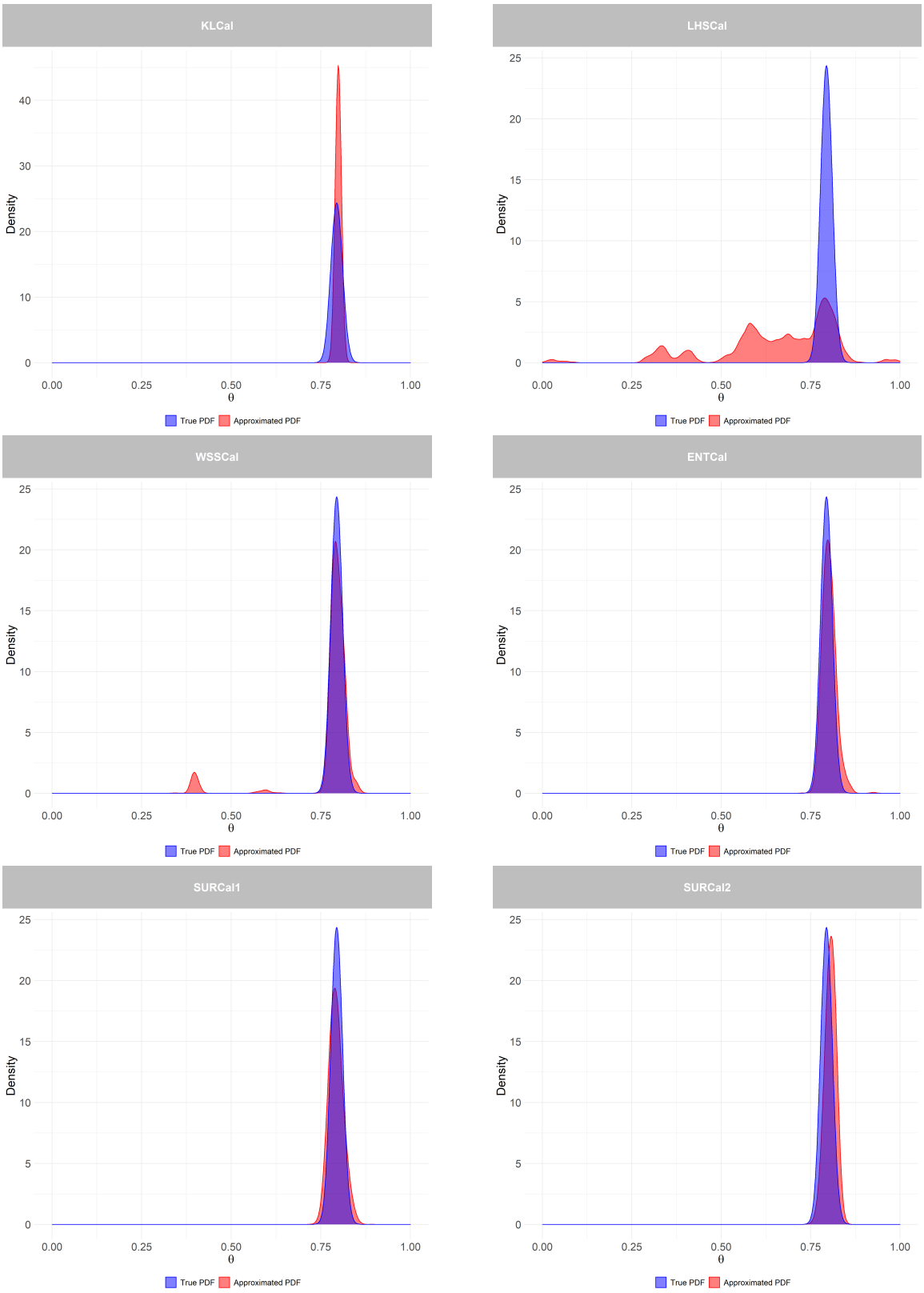

Figure 10: Comparison of the posterior distribution and its approximation for each strategy.

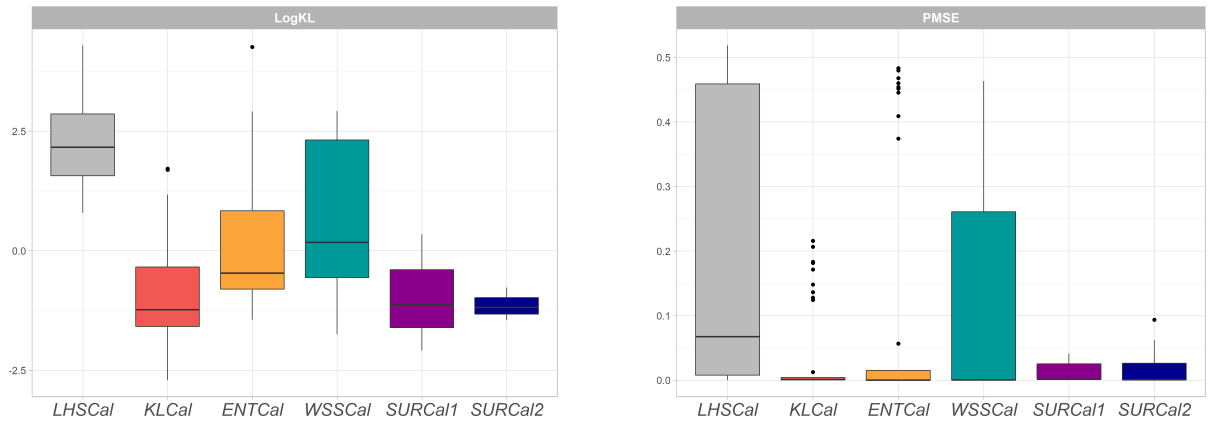

Figure 11: KL and PMSE values for DONE strategies.

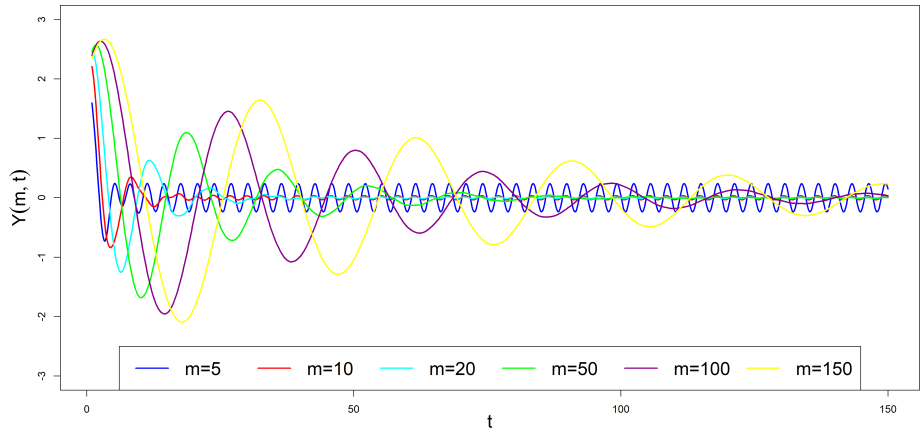

Figure 12: Oscillator trajectories for $m \in \mathcal{M}$.

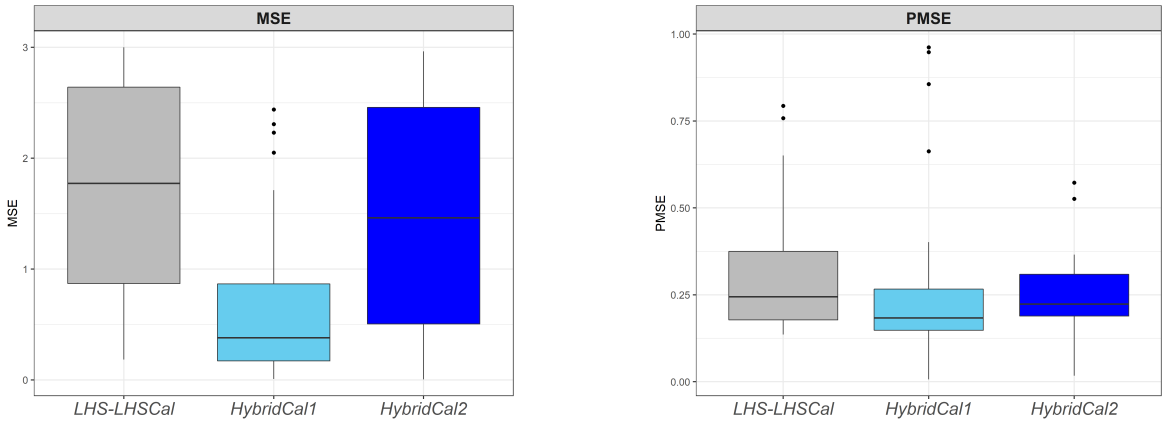

Figure 13: Performance of the hybrid strategies.