



Adaptive calibration of a computer code with time-series output

Guillaume Perrin

► **To cite this version:**

Guillaume Perrin. Adaptive calibration of a computer code with time-series output. Reliability Engineering and System Safety, Elsevier, In press. hal-01794807

HAL Id: hal-01794807

<https://hal-upec-upem.archives-ouvertes.fr/hal-01794807>

Submitted on 17 May 2018

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

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

Adaptive calibration of a computer code with time-series output

G. Perrin^a

^a*CEA/DAM/DIF, F-91297, Arpajon, France*

Abstract

Simulation plays a major role in the conception, the optimization and the certification of complex systems. Of particular interest here is the calibration of the parameters of computer models from high-dimensional physical observations. When the run times of these computer codes is high, this work focuses on the numerical challenges associated with the statistical inference. In particular, several adaptations of the Gaussian Process Regression (GPR) to the high-dimensional or functional output case are presented for the emulation of computer codes from limited data. Then, an adaptive procedure is detailed to minimize the calibration parameters uncertainty at the minimal computational cost. The proposed method is eventually applied to two applications that are based on dynamic simulators.

Key words:

Bayesian framework, Computer experiment, dynamic simulator, Gaussian process, multi-output

1. Introduction

Following the increase of the computational resources and a series of breakthroughs in the modelling of physical phenomena, simulation has become a major tool for conception, optimisation and certification of complex systems. Computer codes are therefore introduced to predict the behaviour of these systems. The codes we consider in this work are deterministic, which means that running them with the same inputs produces the same output, and are treated as "black boxes". Such codes are generally based on two kinds

Email addresses: guillaume.perrin2@cea.fr (G. Perrin)

of inputs. On the one hand, the system parameters are used to characterize the conditions of the experiment, such as the temperature, the pressure, the dimensions of the systems, the boundary conditions, and so on. On the other hand, the calibration parameters refer to the physical or numerical quantities that need to be defined for the codes to be run. These parameters are supposed to be independent of the system parameters, and generally correspond to the parameters of physical laws (one can think to the laws of crack propagation, turbulence, thermal conductivity for instance). To make the codes predictive, these calibration parameters have to be carefully adjusted from measurements that are representative of the working conditions of the system.

The Bayesian framework of Kennedy and O’Hagan [16] to the calibration of computer codes is applied in this work. The calibration parameters are supposed to be random, and we search their posterior distribution given some measured data. Such a posterior distribution characterizes the residual uncertainty about the calibration parameters, and can eventually be propagated to associate credible intervals to the prediction given by the calibrated code.

The computation of such a posterior distribution is generally based on a huge number of code evaluations [5, 15]. When the computational cost associated with one evaluation of the computer code is high (between several minutes to several days CPU), surrogate models have to be introduced to emulate the time-demanding computer code in the inversion procedure. To this end, following [25] and [26], we assume a Gaussian process prior distribution for the code output, conditional on several hyper-parameters. The idea is, then, to update this prior distribution using a limited number of code evaluations. Under not too restrictive conditions on the parametric structure of the Gaussian prior, the posterior prediction is Gaussian and its mean and covariance functions can explicitly be derived.

In principle, such a Bayesian formulation can be applied to any multivariate output code. But in practice, two kinds of simplification hypotheses are generally introduced. On the one hand, in association with some basis representations (such as the principal component analysis) to condense the statistical properties of the high-dimensional or functional outputs, extensions of the univariate case can be proposed by considering the reduced outputs as independent random variables [12, 3, 6]. On the other hand, it is very convenient to assume that the input and the output dependence structure can be separated in the definition of the covariance function. Indeed,

taking advantage of the Kronecker structure of the covariance matrices, this strongly reduces the computational costs associated with the construction of the emulator [27, 23, 2, 8, 13]. The first objective of this work is to compare these two approaches in terms of computational efficiency and flexibility to solve the inference problem.

The posterior distribution of the calibration parameters aggregates at least three sources of uncertainty: the choice of the prior distribution of the calibration parameters, the experimental errors, and the substitution of the computer code by its random emulator. Whereas it is generally not possible to work on the two first sources of uncertainty, the third source of uncertainty can be reduced by updating the Gaussian emulator using additional code evaluations. Iterative procedures can therefore be proposed to minimize the dispersion of the calibration parameters at a minimal computational cost. Proposing criteria for such an adaptive calibration is the second objective of this paper.

The outline of this work is as follows. Section 2 presents the theoretical framework of the Bayesian calibration. The construction of emulators for multi-outputs computer codes and the criterion we propose for the adaptive calibration are then explained in Section 3. At last, the efficiency of the method is illustrated on two examples in Section 4.

2. Bayesian calibration

2.1. General framework

In this work, we are interested in the time response of a complex system, whose properties (dimensions, densities, boundary conditions) can be characterized by a vector of $d \geq 1$ inputs. We denote by $\mathbf{x} \in \mathbb{X} \subset \mathbb{R}^d$ the vector gathering these inputs, and by $y(\cdot; \mathbf{x}) \in \mathbb{R}$ the associated time response, which is supposed to be discretized on a fixed N_t -dimensional grid, $\mathbb{T} := \{t_1, \dots, t_{N_t}\}$. To analyse this system, we have access to a parametric computer code, $g(\cdot; \mathbf{x}, \boldsymbol{\beta})$, where $\boldsymbol{\beta} \in \mathbb{B} \subset \mathbb{R}^{N_\beta}$ gathers the $N_\beta \geq 1$ parameters that have to be fixed for this code to be run. This computer code is assumed non linear with respect to $\boldsymbol{\beta}$ and computationally expensive, and we suppose that it has no model discrepancy. That is, the computational cost associated with one evaluation of g is high (between several minutes to several days) and there exists $\boldsymbol{\beta}^*$, so that:

$$\mathbf{y}(\mathbf{x}) = \mathbf{g}(\mathbf{x}, \boldsymbol{\beta}^*), \quad (\mathbf{x}, \boldsymbol{\beta}) \in \mathbb{X} \times \mathbb{B}. \quad (1)$$

Here, we denote by $\mathbf{y}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x}, \boldsymbol{\beta})$ the vectors gathering the projection of $y(\cdot; \mathbf{x})$ and $g(\cdot; \mathbf{x}, \boldsymbol{\beta})$ on \mathbb{T} :

$$\mathbf{y}(\mathbf{x}) := (y(t_1; \mathbf{x}), \dots, y(t_{N_t}; \mathbf{x})), \quad \mathbf{g}(\mathbf{x}, \boldsymbol{\beta}) := (g(t_1; \mathbf{x}, \boldsymbol{\beta}), \dots, g(t_{N_t}; \mathbf{x}, \boldsymbol{\beta})). \quad (2)$$

The no-discrepancy assumption can be strong if it is not justified. As no computer code can virtually be said to be a perfect representation of reality, this only means that the predictions provided by the calibrated code are accurate enough (compared to the measurement error) for its intended use. In particular, it can exploit error compensation to maximise its predictive capabilities, and therefore not necessarily perfectly fits reality.

In addition, we assume that the system response has been measured in $N_x \geq 1$ points of \mathbb{X} , which are gathered in $\mathcal{X}^{(N_x)} := \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N_x)}\}$. For all \mathbf{x} in $\mathcal{X}^{(N_x)}$, let $\mathbf{y}_{\text{mes}}(\mathbf{x})$ be the measurement of $\mathbf{y}(\mathbf{x})$. In the following, the subscript "mes" will always refer to the measured quantities. These measurements are supposed to be noisy:

$$\mathbf{y}_{\text{mes}}(\mathbf{x}) := \mathbf{y}(\mathbf{x}) + \boldsymbol{\varepsilon}_{\text{mes}}(\mathbf{x}) = \mathbf{g}(\mathbf{x}, \boldsymbol{\beta}^*) + \boldsymbol{\varepsilon}_{\text{mes}}(\mathbf{x}), \quad (3)$$

where $\boldsymbol{\varepsilon}_{\text{mes}}(\mathbf{x})$ characterizes the experimental error. Based on these N_x measurements of y , the idea of this work is to propose a general method to adjust the value of $\boldsymbol{\beta}^*$ to make computer code g be as predictive as possible.

2.2. Notations

The notations that will be used in this paper are the following:

- a, b will correspond to scalars.
- \mathbf{a}, \mathbf{b} will correspond to vectors.
- \mathbf{A}, \mathbf{B} will correspond to matrices.
- The entries of a vector \mathbf{a} will be denoted by $(\mathbf{a})_i$, and the entries of a matrix \mathbf{A} will be denoted by $(\mathbf{A})_{ij}$.
- \mathbf{A}^T , $\det(\mathbf{A})$ and $\text{Tr}(\mathbf{A})$ will denote the transpose, determinant and trace of a square matrix \mathbf{A} respectively.

- $\mathbf{A} \otimes \mathbf{B}$ will correspond to the Kronecker product of two matrices \mathbf{A} and \mathbf{B} .
- $\text{diag}(\cdot)$ will be the vector operator which arranges the elements of a vector into a diagonal matrix, such that:

$$\forall \mathbf{a} \in \mathbb{R}^N, \text{diag}(\mathbf{a}) = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_N \end{bmatrix}. \quad (4)$$

- $\mathcal{N}(\mathbf{a}, \mathbf{A})$ will correspond to the multidimensional Gaussian distribution, whose mean vector and covariance matrix are given by \mathbf{a} and \mathbf{A} respectively.
- $\text{GP}(\mu, C)$ will correspond to the distribution of a Gaussian process whose mean function is μ , and whose covariance function is C .
- \mathcal{U}_Ω will correspond to the uniform distribution over the space Ω .
- For all $N \geq 1$, \mathbf{I}_N will be the $(N \times N)$ -dimensional identity matrix, such that $(\mathbf{I}_N)_{ij}$ is equal to 1 if $i = j$, and to 0 otherwise.

2.3. Bayesian calibration

In this work, we assume that vector $\boldsymbol{\beta}^*$ is random and we denote by $f_{\boldsymbol{\beta}^*}$ its probability density function (PDF). This PDF is supposed to be known from expert judgment [17]. Hence, calibrating $\boldsymbol{\beta}^*$ amounts to searching the conditional distribution, $f_{\boldsymbol{\beta}^* | \mathbf{y}_{\text{mes}}}$, of

$$\boldsymbol{\beta}^* \mid \mathbf{y}_{\text{mes}}(\mathbf{x}^{(1)}), \dots, \mathbf{y}_{\text{mes}}(\mathbf{x}^{(N_x)}). \quad (5)$$

To do so, we first suppose that the experimental error is random, but that its distribution is known. For the sake of simplicity, it is supposed to be centred and Gaussian in the following:

$$(\boldsymbol{\varepsilon}_{\text{mes}}(\mathbf{x}^{(1)}), \dots, \boldsymbol{\varepsilon}_{\text{mes}}(\mathbf{x}^{(N_x)})) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\text{mes}}), \quad (6)$$

where $\boldsymbol{\Sigma}_{\text{mes}}$ is a known $(N_t N_x \times N_t N_x)$ -dimensional symmetric positive definite matrix. Each evaluation of the true model being computationally expensive, we then assume that for all $(\mathbf{x}, \boldsymbol{\beta})$ in $\mathbb{X} \times \mathbb{B}$, the value of $\mathbf{g}(\mathbf{x}, \boldsymbol{\beta})$

corresponds to a particular realization of a Gaussian random vector, which is denoted by $\mathbf{y}_{\text{meta}}(\mathbf{x}, \boldsymbol{\beta})$. In the following, the subscript "meta" will always refer to the quantities associated with the Gaussian emulator of \mathbf{g} . Let $\boldsymbol{\Sigma}_{\text{meta}}(\boldsymbol{\beta})$ be the covariance matrix of $(\mathbf{y}_{\text{meta}}(\mathbf{x}^{(1)}, \boldsymbol{\beta}), \dots, \mathbf{y}_{\text{meta}}(\mathbf{x}^{(N_x)}, \boldsymbol{\beta}))$. Finally, for all $\boldsymbol{\beta}$ in \mathbb{B} , under the condition that the matrix $\boldsymbol{\Sigma}_{\text{mes}} + \boldsymbol{\Sigma}_{\text{meta}}(\boldsymbol{\beta})$ is invertible, we deduce:

$$f_{\boldsymbol{\beta}^* | \mathbf{y}_{\text{mes}}}(\boldsymbol{\beta}) = \frac{\tilde{f}_{\boldsymbol{\beta}^* | \mathbf{y}_{\text{mes}}}(\boldsymbol{\beta})}{\int_{\mathbb{B}} \tilde{f}_{\boldsymbol{\beta}^* | \mathbf{y}_{\text{mes}}}(\mathbf{b}) d\mathbf{b}}, \quad (7)$$

$$\tilde{f}_{\boldsymbol{\beta}^* | \mathbf{y}_{\text{mes}}}(\boldsymbol{\beta}) := \frac{\exp\left(-\frac{1}{2}\boldsymbol{\delta}(\boldsymbol{\beta})^T (\boldsymbol{\Sigma}_{\text{mes}} + \boldsymbol{\Sigma}_{\text{meta}}(\boldsymbol{\beta}))^{-1} \boldsymbol{\delta}(\boldsymbol{\beta})\right) f_{\boldsymbol{\beta}^*}(\boldsymbol{\beta})}{\det(\boldsymbol{\Sigma}_{\text{mes}} + \boldsymbol{\Sigma}_{\text{meta}}(\boldsymbol{\beta}))^{1/2}}, \quad (8)$$

$$\boldsymbol{\delta}(\boldsymbol{\beta}) := (\mathbf{y}_{\text{mes}}(\mathbf{x}^{(1)}) - \mathbb{E}[\mathbf{y}_{\text{meta}}(\mathbf{x}^{(1)}, \boldsymbol{\beta})], \dots, \mathbf{y}_{\text{mes}}(\mathbf{x}^{(N_x)}) - \mathbb{E}[\mathbf{y}_{\text{meta}}(\mathbf{x}^{(N_x)}, \boldsymbol{\beta})]). \quad (9)$$

A two-step procedure is generally used to compute PDF $f_{\boldsymbol{\beta}^* | \mathbf{y}_{\text{mes}}}$. First, sampling techniques, such as Monte Carlo Markov Chains (MCMC) [24], are used to generate a set of points that are approximately distributed according to the PDF of $\boldsymbol{\beta}^* | \mathbf{y}_{\text{mes}}(\mathbf{x}^{(1)}), \dots, \mathbf{y}_{\text{mes}}(\mathbf{x}^{(N_x)})$. Based on these points, PDF $f_{\boldsymbol{\beta}^* | \mathbf{y}_{\text{mes}}}$ can be constructed from statistical non-parametric approaches [19, 20].

2.4. Uncertainty reduction

According to Eq. (8), we can list at least three sources of uncertainty for $\boldsymbol{\beta}^*$:

- the prior uncertainty that is summarized by PDF $f_{\boldsymbol{\beta}^*}$,
- the experimental uncertainty characterized by $\boldsymbol{\Sigma}_{\text{mes}}$,
- the model uncertainty associated with \mathbf{y}_{meta} .

To reduce the first two sources of uncertainty, more *a priori* information about $\boldsymbol{\beta}^*$ and a more precise experimental protocol are needed, which is generally not possible. On the contrary, several directions can be investigated to minimize the third source of uncertainty at a given total computational cost. Indeed, the model uncertainty depends on the kind of model that is used to replace the expensive computer code, but also on the positions of the evaluation points. This raises at least three questions :

- from a given set of code evaluations, how to construct a relevant emulator of g ?
- how to efficiently compute PDF $f_{\beta^* | \mathbf{y}_{\text{mes}}}$ when N_t is high?
- how to (iteratively) choose the positions of the evaluation points?

3. Adaptive calibration

For the sake of clarity, most of the mathematical justifications of the results presented in this Section have been moved to Appendix.

3.1. Gaussian process-based surrogate models for multi-outputs computer codes

As explained in Introduction, the Gaussian process regression (GPR) is a very popular class of surrogate because of its closed-forms expressions, flexibility and ability to provide prediction uncertainties. It is based on the assumption that function g is a particular realization of a Gaussian process, \mathcal{G} , such that for all $(\mathbf{z}, \mathbf{z}', t, t') \in \mathbb{Z} \times \mathbb{Z} \times \mathbb{T} \times \mathbb{T}$, with $\mathbb{Z} := \mathbb{X} \times \mathbb{B}$,

$$\begin{cases} \mathbb{E}[\mathcal{G}(t; \mathbf{z})] = \mu(t; \mathbf{z}), \\ \text{Cov}(\mathcal{G}(t_i; \mathbf{z}), \mathcal{G}(t_j; \mathbf{z}')) = C(t_i, t_j; \mathbf{z}, \mathbf{z}'), \quad 1 \leq i, j \leq N_t. \end{cases} \quad (10)$$

Here, μ is the mean function, and C is the covariance function, which are *a priori* two unknown functions. Let us assume that g has been evaluated in $N_z \geq 1$ points of \mathbb{Z} , $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(N_z)}\}$. Let $\mathbf{g}_{\text{obs}} := (\mathbf{g}(\mathbf{z}^{(1)}), \dots, \mathbf{g}(\mathbf{z}^{(N_z)}))$ be the $N_t N_z$ -dimensional vector that gathers the time response associated with these N_z evaluations of g , where for each $\mathbf{z} = (\mathbf{x}, \beta)$ in $\mathbb{Z} = \mathbb{X} \times \mathbb{B}$, $\mathbf{g}(\mathbf{z}) := \mathbf{g}(\mathbf{x}, \beta)$. For all \mathbf{z}, \mathbf{z}' in \mathbb{Z} and all $1 \leq i, j \leq N_t$, the following notations are introduced:

- $(\mathbf{R}(\mathbf{z}, \mathbf{z}'))_{ij} := C(t_i, t_j; \mathbf{z}, \mathbf{z}')$,
- $\mathbf{R}_{\text{obs}} := \begin{bmatrix} \mathbf{R}(\mathbf{z}^{(1)}, \mathbf{z}^{(1)}) & \dots & \mathbf{R}(\mathbf{z}^{(1)}, \mathbf{z}^{(N_z)}) \\ \vdots & \ddots & \vdots \\ \mathbf{R}(\mathbf{z}^{(N_z)}, \mathbf{z}^{(1)}) & \dots & \mathbf{R}(\mathbf{z}^{(N_z)}, \mathbf{z}^{(N_z)}) \end{bmatrix}$,
- $\mathbf{S}(\mathbf{z}) := [\mathbf{R}(\mathbf{z}, \mathbf{z}^{(1)}) \dots \mathbf{R}(\mathbf{z}, \mathbf{z}^{(N_z)})]^T$,
- $\boldsymbol{\mu}(\mathbf{z}) := (\mu(t_1; \mathbf{z}), \dots, \mu(t_{N_t}; \mathbf{z}))$,

- $\boldsymbol{\mu}_{\text{obs}} := (\boldsymbol{\mu}(\mathbf{z}^{(1)}), \dots, \boldsymbol{\mu}(\mathbf{z}^{(N_z)}))$.

Thus, if the matrix \mathbf{R}_{obs} can be inverted,

$$\mathbf{g} \mid \boldsymbol{\mu}, \mathbf{C}, \mathbf{g}_{\text{obs}} \sim \text{GP}(\boldsymbol{\mu}_*, \mathbf{C}_*), \quad (11)$$

$$\begin{cases} \boldsymbol{\mu}_*(\mathbf{z}) = \boldsymbol{\mu}(\mathbf{z}) + \mathbf{S}(\mathbf{z})^T \mathbf{R}_{\text{obs}}^{-1} (\mathbf{g}_{\text{obs}} - \boldsymbol{\mu}_{\text{obs}}), \\ \mathbf{C}_*(\mathbf{z}, \mathbf{z}') = \mathbf{R}(\mathbf{z}, \mathbf{z}') - \mathbf{S}(\mathbf{z})^T \mathbf{R}_{\text{obs}}^{-1} \mathbf{S}(\mathbf{z}'). \end{cases} \quad (12)$$

Under this formalism, for any non-observed point \mathbf{z} in \mathbb{Z} , $\boldsymbol{\mu}_*(\mathbf{z})$ is the best deterministic prediction of $\mathbf{g}(\mathbf{z})$ in the L^2 sense, whereas $\mathbf{C}_*(\mathbf{z}, \mathbf{z})$ can be used to quantify the precision associated with this prediction. If functions $\boldsymbol{\mu}$ and \mathbf{C} are known, the notations of Eq. (8) can be found back by identifying, for all $\boldsymbol{\beta}$ in \mathbb{B} and all $1 \leq n \leq N_x$:

$$\mathbb{E}[\mathbf{y}_{\text{meta}}(\mathbf{x}^{(n)}, \boldsymbol{\beta})] = \boldsymbol{\mu}_*(\mathbf{z}^{(n)}), \quad \mathbf{z}^{(n)} := (\mathbf{x}^{(n)}, \boldsymbol{\beta}), \quad (13)$$

$$\boldsymbol{\Sigma}_{\text{meta}}(\boldsymbol{\beta}) = \begin{bmatrix} \mathbf{C}_*((\mathbf{x}^{(1)}, \boldsymbol{\beta}), (\mathbf{x}^{(1)}, \boldsymbol{\beta})) & \cdots & \mathbf{C}_*((\mathbf{x}^{(1)}, \boldsymbol{\beta}), (\mathbf{x}^{(N_x)}, \boldsymbol{\beta})) \\ \vdots & \ddots & \vdots \\ \mathbf{C}_*((\mathbf{x}^{(N_x)}, \boldsymbol{\beta}), (\mathbf{x}^{(1)}, \boldsymbol{\beta})) & \cdots & \mathbf{C}_*((\mathbf{x}^{(N_x)}, \boldsymbol{\beta}), (\mathbf{x}^{(N_x)}, \boldsymbol{\beta})) \end{bmatrix}. \quad (14)$$

The computational budget that is needed to compute PDF $f_{\boldsymbol{\beta}^* | \mathbf{y}_{\text{mes}}}$, which is defined by Eq. (8), can be divided in two main contributions. On the one hand, the computational cost for the construction of the emulator is mostly driven by the N_z evaluations of expensive function g and the inversion of the $(N_z N_t \times N_z N_t)$ -dimensional matrix \mathbf{R}_{obs} . On the other hand, the computational cost associated with each MCMC step is mostly driven by the inversion of the $(N_x N_t \times N_x N_t)$ -dimensional matrix $(\boldsymbol{\Sigma}_{\text{mes}} + \boldsymbol{\Sigma}_{\text{meta}}(\boldsymbol{\beta}))$. In this general case, if N_t is high (between several dozens to several thousands), this total computational budget is not affordable. To circumvent this problem, two simplifications are generally introduced, whose pros and cons are presented in the next two sections.

3.1.1. Reduced basis approach

If \mathbf{D}_Q is a $(N_t \times Q)$ -dimensional matrix so that $\mathbf{D}_Q^T \mathbf{D}_Q = \mathbf{I}_Q$, the vector $\mathbf{g}_Q := \mathbf{D}_Q^T \mathbf{g}$ corresponds to the projection of \mathbf{g} on the space spanned by the columns of \mathbf{D}_Q . It comes:

$$\mathbf{g}_Q \sim \text{GP}(\boldsymbol{\mu}_Q = \mathbf{D}_Q^T \boldsymbol{\mu}, \mathbf{R}_Q = \mathbf{D}_Q^T \mathbf{R} \mathbf{D}_Q). \quad (15)$$

Let $\boldsymbol{\mu}_{*,Q}$ and $\mathbf{C}_{*,Q}$ be the mean and covariance functions of $\mathbf{g}_Q | \mathbf{g}_{\text{obs}}$, whose computations only require the inversion of an $(N_z Q \times N_z Q)$ -dimensional matrix. For all \mathbf{z} in \mathbb{Z} , if $\mathbf{g}(\mathbf{z}) \approx \mathbf{D}_Q \mathbf{D}_Q^T \mathbf{g}(\mathbf{z})$, it follows that:

$$\tilde{f}_{\boldsymbol{\beta}^* | \mathbf{y}_{\text{mes}}}(\boldsymbol{\beta}) \approx \frac{\exp\left(-\frac{1}{2} \boldsymbol{\delta}_Q(\boldsymbol{\beta})^T (\boldsymbol{\Sigma}_{\text{mes},Q} + \boldsymbol{\Sigma}_{\text{meta},Q}(\boldsymbol{\beta}))^{-1} \boldsymbol{\delta}_Q(\boldsymbol{\beta})\right)}{\det(\boldsymbol{\Sigma}_{\text{mes},Q} + \boldsymbol{\Sigma}_{\text{meta},Q}(\boldsymbol{\beta}))^{1/2}} f_{\boldsymbol{\beta}^*}(\boldsymbol{\beta}), \quad (16)$$

where:

$$\boldsymbol{\delta}_Q(\boldsymbol{\beta}) := (\mathbf{D}_Q^T \mathbf{y}_{\text{mes}}(\mathbf{x}^{(1)}) - \boldsymbol{\mu}_{*,Q}(\mathbf{x}^{(1)}, \boldsymbol{\beta}), \dots, \mathbf{D}_Q^T \mathbf{y}_{\text{mes}}(\mathbf{x}^{(N_x)}) - \boldsymbol{\mu}_{*,Q}(\mathbf{x}^{(N_x)}, \boldsymbol{\beta})), \quad (17)$$

$$\boldsymbol{\Sigma}_{\text{mes},Q} := (\mathbf{D}_Q^T \otimes \mathbf{I}_{N_x}) \boldsymbol{\Sigma}_{\text{mes}} (\mathbf{D}_Q \otimes \mathbf{I}_{N_x}), \quad (18)$$

$$\boldsymbol{\Sigma}_{\text{meta},Q}(\boldsymbol{\beta}) = \begin{bmatrix} \mathbf{C}_{*,Q}((\mathbf{x}^{(1)}, \boldsymbol{\beta}), (\mathbf{x}^{(1)}, \boldsymbol{\beta})) & \cdots & \mathbf{C}_{*,Q}((\mathbf{x}^{(1)}, \boldsymbol{\beta}), (\mathbf{x}^{(N_x)}, \boldsymbol{\beta})) \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{*,Q}((\mathbf{x}^{(N_x)}, \boldsymbol{\beta}), (\mathbf{x}^{(1)}, \boldsymbol{\beta})) & \cdots & \mathbf{C}_{*,Q}((\mathbf{x}^{(N_x)}, \boldsymbol{\beta}), (\mathbf{x}^{(N_x)}, \boldsymbol{\beta})) \end{bmatrix}. \quad (19)$$

This approach requires the definition of \mathbf{D}_Q , $\boldsymbol{\mu}_Q$ and \mathbf{R}_Q . Matrix \mathbf{D}_Q is generally obtained by carrying out a singular value decomposition (SVD) of $\mathbf{G}_{\text{obs}} = [\mathbf{g}(\mathbf{z}^{(1)}) \cdots \mathbf{g}(\mathbf{z}^{(N_z)})]$ [12, 3, 6], or by selecting relevant element of an adapted wavelet basis [1]. In these cases, the elements of \mathbf{g}_Q are usually assumed independent, leading to a diagonal representation for \mathbf{R}_Q . When Q is very small (less than 5 for instance), non-diagonal representations can also be proposed [11], but at the expense of a much higher number of parameters to identify. If the covariance structure of \mathbf{g} seems easier to infer, function \mathbf{R}_Q can also be searched under the form $\mathbf{D}_Q^T \mathbf{R} \mathbf{D}_Q$, where \mathbf{R} would be a particular element of a well-known class of parametric covariance functions (see [25, 26] for further details about classical covariance functions). At last, a constant vector that corresponds to the empirical estimation of the mean of \mathbf{g} is usually chosen for $\boldsymbol{\mu}_Q$:

$$\boldsymbol{\mu}_Q(\mathbf{z}) = \frac{1}{N_z} \sum_{k=1}^{N_z} \mathbf{D}_Q^T \mathbf{g}(\mathbf{z}^{(k)}), \quad \mathbf{z} \in \mathbb{Z}. \quad (20)$$

But once again, more sophisticated representations could be proposed for $\boldsymbol{\mu}_Q$ (or $\boldsymbol{\mu}$), depending on the studied application.

Under this formalism, the computational cost associated with each evaluation of function $\tilde{f}_{\boldsymbol{\beta}^* | \mathbf{y}_{\text{mes}}}$ is now driven by the inversion of an $(N_x Q \times N_x Q)$ -dimensional matrix, which has to be compared to the inversion of an $(N_x N_t \times N_x N_t)$ matrix in the general case. The main difficulty of this approach is the choice of \mathbf{D}_Q when N_z is small. Indeed, the rank of matrix \mathbf{G}_{obs} being less than N_z , at most N_z columns of \mathbf{D}_Q are actually relevant for the description of \mathbf{g} . Thus, the interest of choosing $Q > N_z$ for the construction of the emulator is almost zero if the construction of \mathbf{D}_Q is based on an SVD. Nevertheless, if Q is too small, the difference between \mathbf{g} and $\mathbf{D}_Q \mathbf{D}_Q^T \mathbf{g}(\mathbf{z})$ can be significant, and the approximation given by Eq. (16) may not be true anymore, leading to poor calibration results.

3.1.2. Separable covariance structure

As an alternative to the reduced basis approach, it can be very convenient to suppose that the input and the output dependence structures can be separated in the modelling of function C . This amounts to assuming that there exists two covariance functions C_t and C_z such that:

$$C(t_i, t_j; \mathbf{z}, \mathbf{z}') = C_t(t_i, t_j) C_z(\mathbf{z}, \mathbf{z}'), \quad 1 \leq i, j \leq N_t, \quad \mathbf{z}, \mathbf{z}' \in \mathbb{Z}. \quad (21)$$

Indeed, if the matrices \mathbf{R}_t and \mathbf{R}_z , so that $(\mathbf{R}_z)_{kl} := C_z(\mathbf{z}^{(k)}, \mathbf{z}^{(l)})$ and $(\mathbf{R}_t)_{ij} := C_t(t_i, t_j)$, can be inverted, it follows that:

$$\mathbf{g} \mid \mu, C_t, C_z, \mathbf{g}_{\text{obs}} \sim \text{GP}(\boldsymbol{\mu}_*, c_* \mathbf{R}_t), \quad (22)$$

$$\begin{cases} \boldsymbol{\mu}_*(\mathbf{z}) = \boldsymbol{\mu}(\mathbf{z}) + (\mathbf{G}_{\text{obs}} - \mathbf{M}_{\text{obs}}) \mathbf{R}_z^{-1} \mathbf{r}_z(\mathbf{z}), \\ c_*(\mathbf{z}, \mathbf{z}') = C_z(\mathbf{z}, \mathbf{z}') - \mathbf{r}_z(\mathbf{z})^T \mathbf{R}_z^{-1} \mathbf{r}_z(\mathbf{z}'), \end{cases} \quad (23)$$

where $\mathbf{M}_{\text{obs}} = [\boldsymbol{\mu}(\mathbf{z}^{(1)}) \ \dots \ \boldsymbol{\mu}(\mathbf{z}^{(N_z)})]$ and $\mathbf{r}_z(\mathbf{z}) := (C_z(\mathbf{z}, \mathbf{z}^{(1)}), \dots, C_z(\mathbf{z}, \mathbf{z}^{(N_z)}))$.

Therefore, taking advantage of the Kronecker structure of the covariance matrices, it is possible to construct an interesting emulator of g , whose mean and covariance functions are based on the inversion of the unique $(N_z \times N_z)$ -dimensional matrix \mathbf{R}_z . Closed-forms expressions for this emulator can also be derived when a linear form is proposed for the parametrization of mean function μ (see Appendix). In particular, for all \mathbf{z}, \mathbf{z}' in \mathbb{Z} , if $\boldsymbol{\mu}(\mathbf{z}) = \mathbf{A}\mathbf{h}(\mathbf{z})$, with \mathbf{h} a given vector-valued function,

$$\mathbf{g} \mid \mathbf{h}, C_t, C_z, \mathbf{g}_{\text{obs}} \sim \text{GP}(\boldsymbol{\mu}_*, c_* \mathbf{R}_t), \quad (24)$$

$$\begin{cases} \boldsymbol{\mu}_*(\mathbf{z}) = \mathbf{G}_{\text{obs}} \mathbf{R}_z^{-1} (\mathbf{r}_z(\mathbf{z}) + \mathbf{H}^T (\mathbf{H} \mathbf{R}_z^{-1} \mathbf{H}^T)^{-1} \mathbf{u}(\mathbf{z})), \\ c_*(\mathbf{z}, \mathbf{z}') = C_z(\mathbf{z}, \mathbf{z}') - \mathbf{r}_z(\mathbf{z})^T \mathbf{R}_z^{-1} \mathbf{r}_z(\mathbf{z}') + \mathbf{u}(\mathbf{z})^T (\mathbf{H} \mathbf{R}_z^{-1} \mathbf{H}^T)^{-1} \mathbf{u}(\mathbf{z}), \end{cases} \quad (25)$$

where $\mathbf{u}(\mathbf{z}) := \mathbf{h}(\mathbf{z}) - \mathbf{H} \mathbf{R}_z^{-1} \mathbf{r}_z(\mathbf{z})$ and $\mathbf{H} := [\mathbf{h}(\mathbf{z}^{(1)}) \ \dots \ \mathbf{h}(\mathbf{z}^{(N_z)})]$.

The former results are conditioned by the choice of functions C_z and C_t . In the same manner than for the reduced basis approach, when the maximal available information about \mathbf{g} is a finite set of evaluations, these two functions can be chosen in a general parametric class of symmetric positive definite functions. For instance, C_z can be chosen in the Matern-5/2 class, such that, for all \mathbf{z} and \mathbf{z}' in \mathbb{Z} :

$$C_z(\mathbf{z}, \mathbf{z}') = (1 + \sqrt{5}h + 5/3h^2) \exp(-\sqrt{5}h), \quad h := \|\text{diag}(\boldsymbol{\ell}_z)^{-1}(\mathbf{z} - \mathbf{z}')\|. \quad (26)$$

There, $\|\cdot\|$ is the classical Euclidian norm and the vector $\boldsymbol{\ell}_z$ characterizes the correlation lengths associated with each element of \mathbf{z} . The choice of the class of covariance functions associated with C_t is however more complicated, as there is no reason for the correlation between two code responses at two distinct times to be stationary. To circumvent this difficulty, it is interesting to notice that if the mean function is known ("Simple Kriging") or can be written as $\boldsymbol{\mu}(\mathbf{z}) = \mathbf{A}\mathbf{h}(\mathbf{z})$ ("Universal Kriging"), the mean and the covariance functions of the GPR of \mathbf{g} only depend on \mathbf{R}_t . This means that even not-invertible estimations of \mathbf{R}_t can be proposed to predict the value of $\mathbf{g}(\mathbf{z})$ in a non-observed point $\mathbf{z} \in \mathbb{Z}$. In particular, the matrix

$$\widehat{\mathbf{R}}_t := \frac{1}{N_z} \left(\mathbf{G}_{\text{obs}} - \widehat{\mathbf{M}} \right) \mathbf{R}_z^{-1} \left(\mathbf{G}_{\text{obs}} - \widehat{\mathbf{M}} \right)^T, \quad (27)$$

$$\widehat{\mathbf{M}} := \begin{cases} \mathbf{M}_{\text{obs}} & \text{in the "Simple Kriging" case,} \\ \mathbf{A}_0 \mathbf{H} & \text{in the "Universal Kriging" case,} \end{cases} \quad (28)$$

is not invertible when $N_z < N_t$, but can be seen as the maximum likelihood estimation of \mathbf{R}_t (see Appendix and Property 3 for more justification of this estimation). Here, the matrix \mathbf{A}_0 is an *a priori* choice for matrix \mathbf{A} . But this matrix can also be chosen equal to its posterior estimation, $\mathbf{G}_{\text{obs}} \mathbf{R}_z^{-1} \mathbf{H}^T (\mathbf{H} \mathbf{R}_z^{-1} \mathbf{H}^T)^{-1}$.

Hence, if C_z is chosen in the Matern-5/2 class, the construction of the GPR of \mathbf{g} only requires the identification of $\boldsymbol{\ell}_z$. This is generally done by maximizing the likelihood of getting \mathbf{g}_{obs} given $\boldsymbol{\ell}_z$. However, if \mathbf{R}_t is approximated by $\widehat{\mathbf{R}}_t$, and if $\widehat{\mathbf{R}}_t$ is not invertible, this likelihood is not well defined. As an alternative, we propose to search $\boldsymbol{\ell}_z$ using a cross-validation procedure. To this end, for all $1 \leq k \leq N_z$, we denote by $\boldsymbol{\mu}^{(-k)}(\mathbf{z}; \boldsymbol{\ell}_z)$ the mean function of the GPR of $\mathbf{g}(\mathbf{z})$ that is constructed from the $N_z - 1$ evaluations $\{\mathbf{g}(\mathbf{z}^{(1)}), \dots, \mathbf{g}(\mathbf{z}^{(k-1)}), \mathbf{g}(\mathbf{z}^{(k+1)}), \dots, \mathbf{g}(\mathbf{z}^{(N_z)})\}$ only. The following "Leave-one-out" (LOO) error is then introduced:

$$e^2(\boldsymbol{\ell}_z) := \sum_{k=1}^{N_z} \left\| \boldsymbol{\mu}^{(-k)}(\mathbf{z}^{(k)}; \boldsymbol{\ell}_z) - \mathbf{g}(\mathbf{z}^{(k)}) \right\|^2, \quad (29)$$

so that:

$$\boldsymbol{\ell}_z = \arg \min_{\boldsymbol{\ell} \in (\mathbb{R}^+)^{N_{\beta+d}}} e^2(\boldsymbol{\ell}). \quad (30)$$

Cross-validation procedures can be computationally expensive in the general case. However, in our case, N_z is relatively small as each evaluation of the computer code is expensive. Hence, the evaluation of $\boldsymbol{\mu}^{(-k)}(\mathbf{z}; \boldsymbol{\ell}_z)$ is quick as it only relies on the product of small-dimensional matrices, and so is the evaluation of e^2 .

Once parameter $\boldsymbol{\ell}_z$ has been identified, for all $\boldsymbol{\beta}$ in \mathbb{B} , it follows that:

$$\boldsymbol{\Sigma}_{\text{meta}}(\boldsymbol{\beta}) = \mathbf{R}_{\text{meta}}(\boldsymbol{\beta}) \otimes \widehat{\mathbf{R}}_t, \quad (31)$$

$$\mathbf{R}_{\text{meta}}(\boldsymbol{\beta}) = \begin{bmatrix} c_{\star}((\mathbf{x}^{(1)}, \boldsymbol{\beta}), (\mathbf{x}^{(1)}, \boldsymbol{\beta})) & \cdots & c_{\star}((\mathbf{x}^{(1)}, \boldsymbol{\beta}), (\mathbf{x}^{(N_x)}, \boldsymbol{\beta})) \\ \vdots & \ddots & \vdots \\ c_{\star}((\mathbf{x}^{(N_x)}, \boldsymbol{\beta}), (\mathbf{x}^{(1)}, \boldsymbol{\beta})) & \cdots & c_{\star}((\mathbf{x}^{(N_x)}, \boldsymbol{\beta}), (\mathbf{x}^{(N_x)}, \boldsymbol{\beta})) \end{bmatrix}. \quad (32)$$

Such a separation in the input and the output dependence structures is particularly interesting when $\boldsymbol{\Sigma}_{\text{mes}}$ also presents a separable structure. Indeed, if it exists two symmetric positive definite matrices $\mathbf{R}_{\text{mes},\beta}$ and $\mathbf{R}_{\text{mes},t}$ so that $\boldsymbol{\Sigma}_{\text{mes}} = \mathbf{R}_{\text{mes},\beta} \otimes \mathbf{R}_{\text{mes},t}$, we can define $\mathbf{V}_{\beta}(\boldsymbol{\beta})$ and \mathbf{V}_t as the two invertible matrices that verify:

$$\mathbf{R}_{\text{meta}}(\boldsymbol{\beta}) = \mathbf{V}_{\beta}(\boldsymbol{\beta}) \text{diag}(\boldsymbol{\lambda}_{\beta}) \mathbf{V}_{\beta}(\boldsymbol{\beta})^T, \quad \mathbf{R}_{\text{mes},\beta} = \mathbf{V}_{\beta}(\boldsymbol{\beta}) \mathbf{V}_{\beta}(\boldsymbol{\beta})^T, \quad (33)$$

$$\widehat{\mathbf{R}}_t = \mathbf{V}_t \text{diag}(\boldsymbol{\lambda}_t) \mathbf{V}_t^T, \quad \mathbf{R}_{\text{mes},t} = \mathbf{V}_t \mathbf{V}_t^T. \quad (34)$$

It comes that:

$$\boldsymbol{\Sigma}_{\text{meta}}(\boldsymbol{\beta}) + \boldsymbol{\Sigma}_{\text{mes}} = (\mathbf{V}_{\beta}(\boldsymbol{\beta}) \otimes \mathbf{V}_t) \text{diag}(\boldsymbol{\Lambda}) (\mathbf{V}_{\beta}(\boldsymbol{\beta})^T \otimes \mathbf{V}_t^T), \quad (35)$$

$$\text{diag}(\boldsymbol{\Lambda}) := \text{diag}(\boldsymbol{\lambda}_{\beta}) \otimes \text{diag}(\boldsymbol{\lambda}_t) + \mathbf{I}_{N_x} \otimes \mathbf{I}_{N_t}. \quad (36)$$

We deduce (for the sake of concision, the derivations leading to this expression have been moved to Appendix):

$$\tilde{f}_{\boldsymbol{\beta}^{\star} | \mathbf{y}_{\text{mes}}}(\boldsymbol{\beta}) \propto \frac{\exp(-\frac{1}{2}t^2)}{\det(\mathbf{V}_{\beta}(\boldsymbol{\beta})^T \mathbf{V}_{\beta}(\boldsymbol{\beta}))^{N_t/2} \prod_{l=1}^{N_x N_t} (\Lambda_l)^{1/2}} f_{\boldsymbol{\beta}^{\star}}(\boldsymbol{\beta}), \quad (37)$$

where:

$$t^2 := \sum_{j=1}^{N_x} \left\| \text{diag}(\boldsymbol{\Lambda}_j)^{-1/2} \mathbf{V}_t^{-1} \boldsymbol{\Delta} \mathbf{v}^{(j)}(\boldsymbol{\beta}) \right\|^2, \quad (38)$$

$$\mathbf{v}^{(j)}(\boldsymbol{\beta}) := ((\mathbf{V}_{\beta}(\boldsymbol{\beta})^{-1})_{j1}, \dots, (\mathbf{V}_{\beta}(\boldsymbol{\beta})^{-1})_{jN_x}), \quad (39)$$

$$\boldsymbol{\Delta} := [\boldsymbol{\delta}_1(\boldsymbol{\beta}) \cdots \boldsymbol{\delta}_{N_x}(\boldsymbol{\beta})], \quad \boldsymbol{\delta}_k(\boldsymbol{\beta}) := \mathbf{y}_{\text{mes}}(\mathbf{x}^{(k)}) - \boldsymbol{\mu}_{\star}(\mathbf{x}^{(k)}, \boldsymbol{\beta}), \quad (40)$$

and where $\text{diag}(\mathbf{\Lambda}_1), \dots, \text{diag}(\mathbf{\Lambda}_{N_x})$ are the $(N_t \times N_t)$ -dimensional diagonal matrices so that:

$$\text{diag}(\mathbf{\Lambda}) = \begin{bmatrix} \text{diag}(\mathbf{\Lambda}_1) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \text{diag}(\mathbf{\Lambda}_2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \text{diag}(\mathbf{\Lambda}_{N_x}) \end{bmatrix}. \quad (41)$$

Given the formerly introduced constraint on the dependence structure of the experimental uncertainty, the computational cost associated with the evaluation of $\tilde{f}_{\beta^*|\mathbf{y}_{\text{mes}}}$ is strongly reduced. Indeed, matrices \mathbf{V}_t and $\text{diag}(\mathbf{\Lambda}_t)$ can be calculated once for all, such that for each value of β , the evaluation of $\tilde{f}_{\beta^*|\mathbf{y}_{\text{mes}}}$ only requires the inversion of an $(N_x \times N_x)$ -dimensional matrix, which has to be compared with the inversion of an $(N_x N_t \times N_x N_t)$ -dimensional matrix in the general case, and the inversion of an $(QN_x \times QN_x)$ -dimensional matrix in the reduced basis approach.

Remarks.

- If $\Sigma_{\text{mes}} = \mathbf{0}$, Eq. (38) becomes:

$$t^2 = \text{Tr}(\mathbf{R}_{\text{meta}}(\beta)^{-1} \mathbf{\Delta}^T \mathbf{R}_t^{-1} \mathbf{\Delta}). \quad (42)$$

- Although it is always very convenient from a computational point of view, the separable structure may not be adapted to the emulation of function g . In that case, before trying to emulate g , it is generally interesting to work on a transformation to apply to g to make this assumption more relevant (when it is possible).

3.2. Adaptive calibration

For a given set of evaluations of the computer code, once the surrogate model has been computed, MCMC procedures can be used to approximate $f_{\beta^*|\mathbf{y}_{\text{mes}}}$ (see [24] for further details about these approaches). As presented in Introduction, the dispersion of the posterior distribution is mostly dependent of the prior uncertainty associated with f_{β^*} and the experimental error characterized by Σ_{mes} , but also of the prediction uncertainty controlled by $\beta \mapsto \Sigma_{\text{meta}}(\beta)$. Whereas it is not possible to work on the two first sources of uncertainty, the prediction uncertainty can be reduced by increasing the

number of evaluations of the computer code. In that prospect, following the works achieved in [22], let γ be the cost function defined on \mathbb{B} so that

$$\gamma(\boldsymbol{\beta}) := \sum_{n=1}^{N_x} \|\mathbf{g}(\mathbf{x}^{(n)}, \boldsymbol{\beta}) - \mathbf{g}(\mathbf{x}^{(n)}, \boldsymbol{\beta}^*)\|. \quad (43)$$

Assuming that the computer code has been evaluated for each point of

$$\left\{ (\mathbf{x}^{(n)}, \boldsymbol{\beta}^{(m)}), 1 \leq n \leq N_x, 1 \leq j \leq M \right\}, \quad (44)$$

function γ is known in M values of $\boldsymbol{\beta}$. Based on this information, we can construct its GPR, which is denoted by $\hat{\gamma}_M$. Hence, the dispersion of the posterior distribution of $\boldsymbol{\beta}^*$ can be reduced by adding to the learning set the N_x new input points $(\mathbf{x}^{(1)}, \hat{\boldsymbol{\beta}}), \dots, (\mathbf{x}^{(N_x)}, \hat{\boldsymbol{\beta}})$, where $\hat{\boldsymbol{\beta}}$ is solution of the following expected improvement (EI) based optimization problem (see [14] for more details about the EI criterion):

$$\hat{\boldsymbol{\beta}} := \arg \min_{\boldsymbol{\beta} \in \mathbb{B}} \mathbb{E} \left[\max \left(\min(\gamma(\boldsymbol{\beta}^{(1)}), \dots, \gamma(\boldsymbol{\beta}^{(M)})) - \hat{\gamma}_M(\boldsymbol{\beta}), 0 \right) \right]. \quad (45)$$

In the same spirit, [21] proposed to use this EI formalism to minimize another function inspired by the Kullback-Liebler divergence to aggregate the differences between $\mathbf{g}(\mathbf{x}^{(n)}, \boldsymbol{\beta})$ and $\mathbf{g}(\mathbf{x}^{(n)}, \boldsymbol{\beta}^*)$ for each $1 \leq n \leq N_x$.

However, when the computer code is very expensive, requiring the computer code to be evaluated N_x times for each value of $\boldsymbol{\beta}$ may be too restrictive, as in that case, the number of code evaluations quickly increases. In the same manner, considering a fixed grid in space \mathbb{X} is generally not optimal for the construction of the GPR of \mathbf{g} . In this work, we propose to consider another selection criterion for the new evaluation point, $\mathbf{z}^{\text{new}} := (\mathbf{x}^{\text{new}}, \boldsymbol{\beta}^{\text{new}})$. Two considerations are taken into account to choose this point. First, there is no interest for the surrogate model to be precise in the unlikely regions for $\boldsymbol{\beta}$. Thus, we propose to search $\boldsymbol{\beta}^{\text{new}}$ where $f_{\boldsymbol{\beta}^* | \mathbf{y}_{\text{mes}}}(\boldsymbol{\beta})$ is maximal:

$$\boldsymbol{\beta}^{\text{new}} = \arg \max_{\boldsymbol{\beta} \in \mathbb{B}} f_{\boldsymbol{\beta}^* | \mathbf{y}_{\text{mes}}}(\boldsymbol{\beta}). \quad (46)$$

Then, to minimize the prediction uncertainty in the most likely regions of $\boldsymbol{\beta}^*$, we propose to adopt a Stepwise Uncertainty Reduction (SUR) strategy [4, 7]. The purpose of these approaches is to choose the new evaluation

point in order to minimize the expected value of a well chosen measure of the uncertainty about the search domain. In our case, this uncertainty is characterized by Σ_{meta} . Denoting by $\Sigma_{\text{meta}}^{\mathbf{x}}$ the adaptation of function Σ_{meta} given the new evaluation of the computer code in $(\mathbf{x}, \boldsymbol{\beta}^{\text{new}})$, \mathbf{x}^{new} can therefore be searched where the difference between $\Sigma_{\text{meta}}(\boldsymbol{\beta}^{\text{new}})$ and $\Sigma_{\text{meta}}^{\mathbf{x}}(\boldsymbol{\beta}^{\text{new}})$ is the highest. However, we observed on a series of test cases that better results were obtained when working on the predictive variance rather than on the predictive covariance. Hence, in this work, we propose to choose \mathbf{x}^{new} as the solution of:

$$\mathbf{x}^{\text{new}} := \arg \max_{\mathbf{x} \in \mathbb{X}} \sum_{i=1}^{N_t} (\Sigma_{\text{meta}}(\boldsymbol{\beta}^{\text{new}}))_{ii} - \sum_{i=1}^{N_t} (\Sigma_{\text{meta}}^{\mathbf{x}}(\boldsymbol{\beta}^{\text{new}}))_{ii}. \quad (47)$$

Remarks.

- The searching set for \mathbf{x}^{new} can be reduced in practice to the convex hull of the set of the available experimental data, that is to say to the smallest convex region enclosing all the points in $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N_x)}\}$.
- Under the formalism presented in Section 3.1, it is important to notice that no evaluation of the computer code is needed for the computation of $\Sigma_{\text{meta}}^{\mathbf{x}}$ for any \mathbf{x} in \mathbb{X} .

4. Applications

This section aims at illustrating the proposed adaptive calibration on two applications. The first application is based on simulated data, whereas the second one is based on experimental data.

4.1. Analytical example

4.1.1. Description of the problem

The first application we present can be linked to the identification of the mechanical properties of a structure from a non-destructive test. In that prospect, let us consider the coupled system that corresponds to Figure 1, where :

- y is the horizontal displacement of the beam,
- θ is the rotation angle of the pendulum,

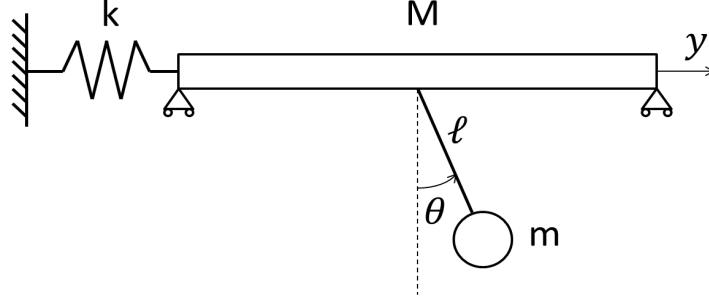


Figure 1: Parameterization of the mechanical system of interest.

- θ_0 is the initial rotation angle of the pendulum,
- $x := g\ell \sin(\theta_0)$ is the potential energy of the system,
- $\beta := (M, k)$ corresponds to the mechanical properties of the rigid beam,
- m and ℓ are the characteristics of the pendulum,
- g is the gravity.

In this application, the gravity and the characteristics of the pendulum are supposed to be known. For the numerical application, we chose $g = 9.81$, $m = 0.5$, $\ell = 2$. We assume that there exists a true value for the mechanical properties of the beam, $\beta^* = (2, 1.2)$. This value has to be inferred from the analysis of the movement of the beam that is induced by the movement of the pendulum. To carry out this calibration of β , we assume that the time evolution of the beam displacement has been measured in N_x initial configurations of the pendulum, $\{x^{(1)}, \dots, x^{(N_x)}\}$, on the fixed time grid $\{t_r = 10r/101, 0 \leq r \leq 100\}$. It follows that $N_t = 101$. Let $y_{\text{mes}}(t; x)$ be the measured beam displacement at time t corresponding to the initial energy x . These measurements at each time step are supposed to be affected by a centred Gaussian measurement error. For all $1 \leq n, m \leq N_x$ and all $1 \leq r, s \leq N_t$, it comes:

$$y_{\text{mes}}(t_r; x_n) \sim \mathcal{N}(y(t_r; x_n), \sigma_{\text{mes}}^2), \quad (48)$$

$$\mathbb{E}[y_{\text{mes}}(t_s; x_m)y_{\text{mes}}(t_r; x_n)] = y(t_s; x_m)y(t_r; x_n) + \delta_{r,s}\delta_{n,m}\sigma_{\text{mes}}^2. \quad (49)$$

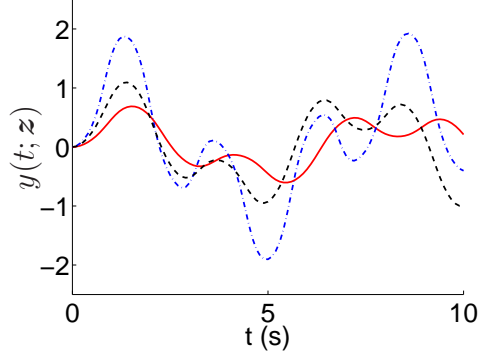


Figure 2: Comparison of the evolution of y with respect to several values of M , k and x . Red continuous line : $(x, M, k) = (10.8, 2.33, 2.29)$. Blue dashed line : $(x, M, k) = (15.0, 0.573, 0.813)$. Black dotted line : $(x, M, k) = (9.62, 1.06, 1.68)$.

There, σ_{mes}^2 is supposed to be known, and $\delta_{n,m}$ is the Kronecker symbol, which is equal to 1 if $n = m$ and to 0 otherwise.

In addition, for all $\mathbf{z} := (x, M, k)$ in $[-g\ell, g\ell] \times \mathbb{R}^{+*} \times \mathbb{R}^{+*}$, the time evolutions of y and θ can be computed as the solutions of the following system of coupled equations:

$$\begin{cases} (M + m)\ddot{y}(t; \mathbf{z}) + ky(t; \mathbf{z}) + m\ell \left(\ddot{\theta}(t; \mathbf{z}) \cos(\theta(t; \mathbf{z})) - \dot{\theta}^2(t; \mathbf{z}) \sin(\theta(t; \mathbf{z})) \right) = 0, \\ \ddot{y}(t; \mathbf{z}) \cos(\theta(t; \mathbf{z})) + \ell \ddot{\theta}(t; \mathbf{z}) + g \sin(\theta(t; \mathbf{z})) = 0, \\ y(t=0; \mathbf{z}) = 0, \quad \theta(t=0; \mathbf{z}) = \theta_0. \end{cases} \quad (50)$$

As no analytic solution exists for this problem, numerical procedures based on a time discretization were used to find the values of $y(t; \mathbf{z})$ and $\theta(t; \mathbf{z})$ at each time step and for each value of \mathbf{z} . A verification analysis was performed to adapt this time discretization such that no distinction is made in the following between y and the true solution of Eq. (50). To illustrate the influence of \mathbf{z} on y , Figure 2 represents the evolution of $y(t; \mathbf{z})$ with respect to t for three values of \mathbf{z} .

Even if the system defined by Eq. (50) is non-linear, it can be interesting to consider its linear approximation, which is defined by:

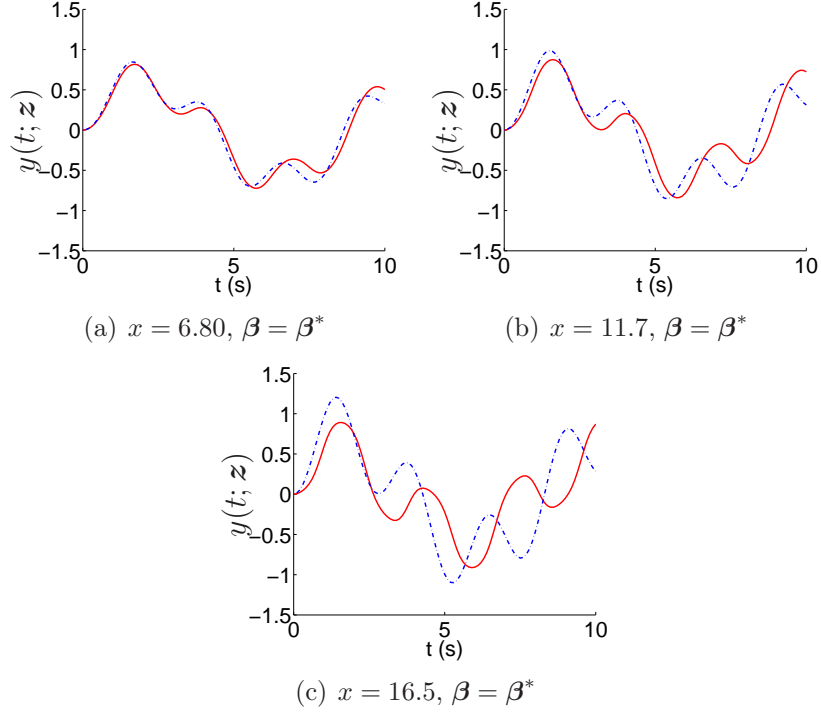


Figure 3: Comparison of the time evolutions of y and y_{lin} for different values of x . Red continuous line : y . Blue dashed line : y_{lin} .

$$\begin{cases} (M + m)\ddot{y}_{\text{lin}}(t; \mathbf{z}) + ky_{\text{lin}}(t; \mathbf{z}) + m\ell\ddot{\theta}_{\text{lin}}(t; \mathbf{z}) = 0, \\ \ddot{y}_{\text{lin}}(t; \mathbf{z}) + \ell\ddot{\theta}_{\text{lin}}(t; \mathbf{z}) + g\theta_{\text{lin}}(t; \mathbf{z}) = 0, \\ y_{\text{lin}}(t = 0; \mathbf{z}) = 0, \quad \theta_{\text{lin}}(t = 0; \mathbf{z}) = \theta_0. \end{cases} \quad (51)$$

Indeed, as the former system of equations is linear, its solutions, which are denoted by y_{lin} and θ_{lin} , can be derived explicitly. By construction, the smaller is the value of x , and the closer y and y_{lin} are likely to be. This can be seen in Figure 3, which compares the evolutions of y and y_{lin} for the same values of β but for three different values of x . Even if the computational time associated with the computation of $\mathbf{y}(\mathbf{z}) = (y(t_1; \mathbf{z}), \dots, y(t_{N_t}; \mathbf{z}))$ is relatively small, it is much higher than the one associated with the computation of $\mathbf{y}_{\text{lin}}(\mathbf{z}) = (y_{\text{lin}}(t_1; \mathbf{z}), \dots, y_{\text{lin}}(t_{N_t}; \mathbf{z}))$.

To identify β^* , $N_z \ll N_t$ values of \mathbf{z} are chosen in \mathbb{Z} , $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(N_z)}\}$.

To optimize the space filling properties of this set, set $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(N_z)}\}$ is chosen as an optimized Latin Hypercube sample (LHS) [10, 9, 18]. Based on the evaluation of y at these N_z points only, two Gaussian process-based surrogate models were constructed.

- The first one is based on the classical reduced-basis approach introduced in Section 3.1.1:

$$\mathbf{y}(\mathbf{z}) \approx \mathbf{y}_{\text{lin}}(\mathbf{z}) + \sum_{q=1}^{N_z} \omega_q(\mathbf{z}) \mathbf{d}_q, \quad (52)$$

where $\{\mathbf{d}_1, \dots, \mathbf{d}_{N_z}\}$ gathers the $N_z < N_t$ eigenvectors of the empirical estimator of the covariance matrix of $\mathbf{y} - \mathbf{y}_{\text{lin}}$ associated with the N_z highest eigenvalues. Matern-5/2 covariance functions are chosen for the projection coefficients $\omega_q(\mathbf{z})$, which are supposed to be centred and statistically independent:

$$\mathbb{E}[\omega_q(\mathbf{z})\omega_p(\mathbf{z}')] = \delta_{qp} \prod_{i=1}^d (1 + \sqrt{5}h_i + 5h_i^2/3) \exp(-\sqrt{5}h_i), \quad h_i = |z_i - z'_i|/\ell_{q,i}, \quad (53)$$

where the parameters $\ell_{1,1}, \dots, \ell_{N_z,d}$ are chosen as their maximum likelihood estimates.

- The second one is based on the separable covariance structure approach presented in Section 3.1.2. The mean function of $\mathbf{y}(\mathbf{z})$ is also chosen equal to $\mathbf{y}_{\text{lin}}(\mathbf{z})$. This latter surrogate is the direct extension of the "Simple Kriging" formalism to the multi-output case. A Matern-5/2 is also chosen for the covariance structure in \mathbf{z} , whereas the covariance structure in time is given by its empirical estimator given by Eq. (27).

Figure 4 compares the relevance of these two surrogate models with respect to the number N_z of code evaluations. In this figure, it can be noticed that, for each value of N_z , the mean-squared errors associated with the separable covariance structure is better, in average. In the following, we therefore focus on this particular emulator of \mathbf{y} . To illustrate the fact that the more code evaluations are available, the more precise is the prediction, Figure 5

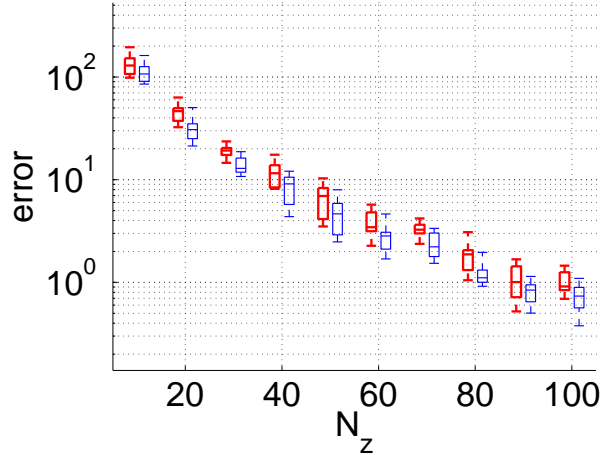


Figure 4: Comparison of the mean-squared errors associated with two kinds of surrogate models for $N_z \in \{10, 20, \dots, 100\}$. Red: surrogate model associated with a dimension-reduction technique. Blue: extension of the Simple Kriging case to the multi-output case. Boxplots are produced from 50 runs with different initial DoE.

compares the capacity of the proposed surrogate models to predict, with a given precision, the value of y in a non-observed value of \mathbf{z} for two values of N_z .

Based on this emulator, it is possible to infer the value of $\boldsymbol{\beta}^*$ at a reasonable computational cost, from a limited number of code evaluations and experimental measurements. As an example, Figure 6 shows the calibration results associated with the former surrogate models for $N_z = 20$ and $N_x = 3$. A uniform prior is chosen for $\boldsymbol{\beta}^*$:

$$f_{\boldsymbol{\beta}^*}(\boldsymbol{\beta}) = \begin{cases} 1/4 & \text{if } \boldsymbol{\beta} \in [0.5, 2.5]^2, \\ 0 & \text{otherwise.} \end{cases} \quad (54)$$

Even for these low values of N_z and N_x , it can be seen that good estimates of $\boldsymbol{\beta}^*$ can be found in each case. If we focus on the 90% credible ellipse for $\boldsymbol{\beta}^*$, it is also possible to quantify the influence of the experimental uncertainties on the estimation of $\boldsymbol{\beta}^*$. Indeed, dividing by 10 the value of σ_{mes} allows us to strongly concentrate the posterior PDF of $\boldsymbol{\beta}^*$ on its true value.

In order to better quantify the robustness of the calibration results, Figure 7 shows the evolution of the distance, $\varepsilon_{\boldsymbol{\beta}}$, between the true value of $\boldsymbol{\beta}^*$ and

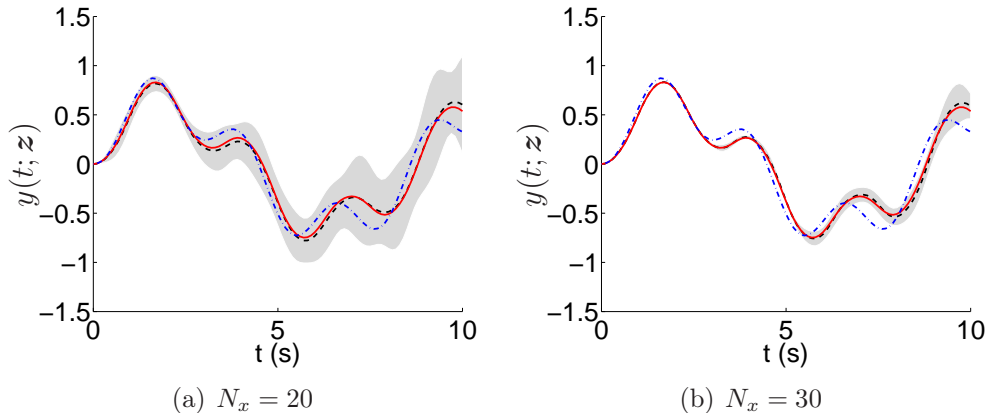


Figure 5: Influence of the number of code evaluations on the prediction precision of the surrogate models. Grey area : 95%–credible interval for the prediction. Black dotted line: *a posteriori* mean prediction. Red continuous line: code output $y(t; \mathbf{z})$. Blue dashed line : *a priori* mean prediction. These predictions are based on a separate covariance structure.

its maximum likelihood estimate, $\boldsymbol{\beta}^{\text{MLE}}$, with respect to different values of N_z , N_x and σ_{mes} :

$$\varepsilon_{\boldsymbol{\beta}} := \|\boldsymbol{\beta}^* - \boldsymbol{\beta}^{\text{MLE}}\|. \quad (55)$$

The boxplots are produced from 50 runs with different initial space filling designs of experiments for the construction of the surrogate models, and different values of x in $[-g\ell, g\ell]$ for the experimental data. In these figures, we verify that, in average, the value of $\varepsilon_{\boldsymbol{\beta}}$ decreases with respect to N_z and N_x , and increases with respect to σ_{mes} . However, for fixed values of σ_{mes} and N_x , the decrease of $\varepsilon_{\boldsymbol{\beta}}$ with respect to N_z is slow. As explained in Section 3.2, this decrease can be accelerated by better choosing the new positions of the code evaluations. Indeed, focusing the new code evaluations in the region of high probability for the true value of $\boldsymbol{\beta}^*$, while uniformly filling the input space of x , can strongly reduce the prediction uncertainties given by the surrogate models, and therefore better concentrate the PDF of $\boldsymbol{\beta}^*$ on its true value. As an illustration, Figure 8 compares the evolutions of distance $\varepsilon_{\boldsymbol{\beta}}$ with respect to N_z , for two enrichment procedures. Whereas the proposed adaptive procedure is represented in red, the results in blue correspond to the case where the surrogate models are based on the evaluation of the computer code on independent N_z -dimensional space filling designs. Hence, for $\sigma_{\text{mes}} = 0.1$ and different values of N_x , it can be seen that working on the position

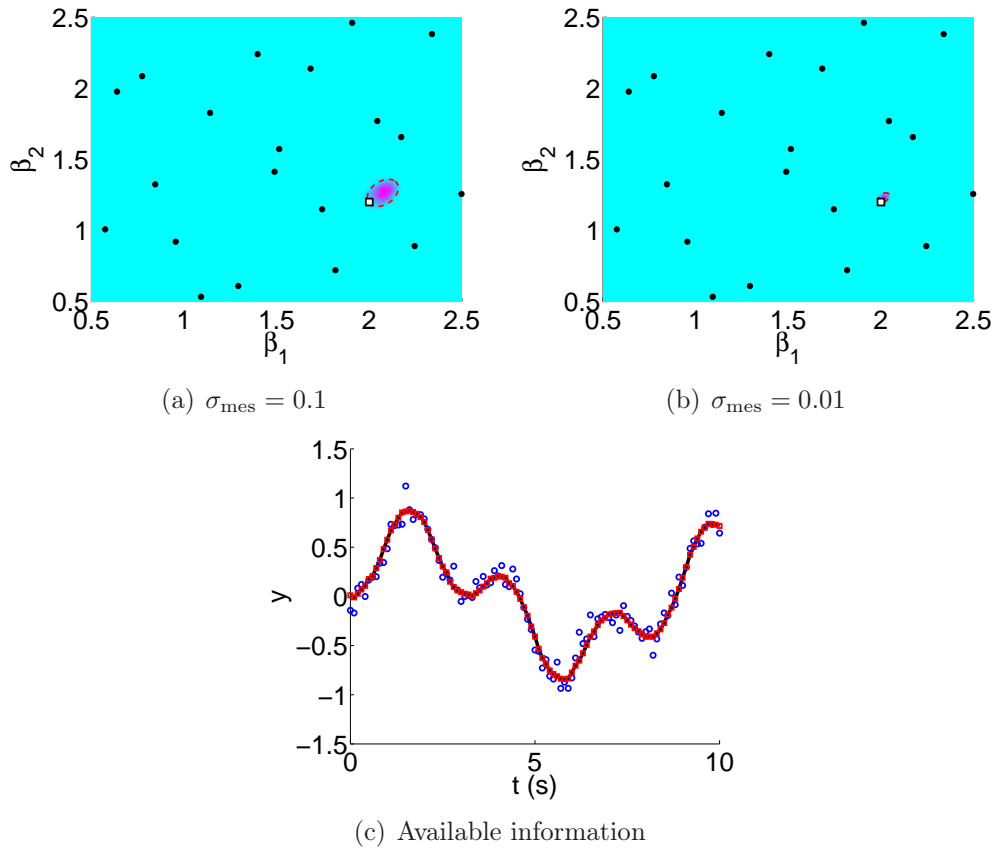


Figure 6: Influence of the experimental uncertainties on the calibration results for $N_z = 20$ and $N_x = 3$. In subfigures (a) and (b), the contour plots correspond to the level sets associated with the posterior PDF of β^* , the black points indicate the values of β where the code has been evaluated to compute the surrogate model (each of these values were associated with different values of x), the white square is the true value of β^* , and the red dotted line defines a 90% credible ellipse, such that the integration of the posterior distribution of β^* on this ellipse is equal to 0.9. For a given value of x and for $\sigma_{\text{mes}} = 0.1$, subfigure (c) compares the true evolution of y (the red squares) to the discrete experimental information (the blue circles) that is available for the calibration of β^* .

of 10 new code evaluations can allow us to divide by two the final values of ε_β . Such a reduction is even higher when $\sigma_{\text{mes}} = 0.01$. Indeed, in that case, the experimental uncertainties are relatively small compared to the prediction uncertainties, such that the precision of the calibration results mostly depends on the prediction uncertainties provided by the surrogate models.

4.2. Industrial example

The second application deals with the calibration of an Eulerian hydrodynamics code developed by CEA. A spherical explosive charge is placed at the geometric center of a spherical containment vessel. Once it is ignited, we measure the transient pressure acting on the inner surface of the containment vessel, which is denoted by y to be coherent with the notations of Section 2. It is worth noticing that the evolution of the pressure is not regular, as several reflections of the shock wave on the wall of the vessel can be observed. $N_x = 5$ experiments were carried out for five different quantities of explosive. This defines the available information for the calibration. Let $x > 0$ denote the radius of the explosive charge, which characterizes the quantity of explosive in the vessel, and let $x^{(1)} < \dots < x^{(5)}$ be the five radii that were used for the five experiments. The time evolutions of the measured inner pressure for these five values of x are represented in Figure 9.

For confidentiality reason, only normalized values are presented in this second application. To simplify the construction of the surrogate models, the time origin of the simulation and the measurements is chosen equal to the time of the first shock reflection.

The experiments are supposed to be discretized on a fixed N_t -dimensional time grid, $\{t_1, \dots, t_{N_t}\}$, with $N_t = 353$. Using the same notations than in Section 2.1, for each value of x , we denote by $\mathbf{y}(x) = (y(t_1; x), \dots, y(t_{N_t}; x))$ and $\mathbf{y}_{\text{mes}}(x) = (y_{\text{mes}}(t_1; x), \dots, y_{\text{mes}}(t_{N_t}; x))$ the vectors that gather respectively the true values and the measured values of the inner pressure at the different time steps. The experiments are affected by measurement uncertainties, which are assumed to be centred, Gaussian and non-stationary in time. These uncertainties are related to the precision of the measurement device, which does not depend on the quantity of explosive that is placed in the vessel. In that case, the global covariance of the experimental measurements, Σ_{mes} , presents the following separate structure in time and space:

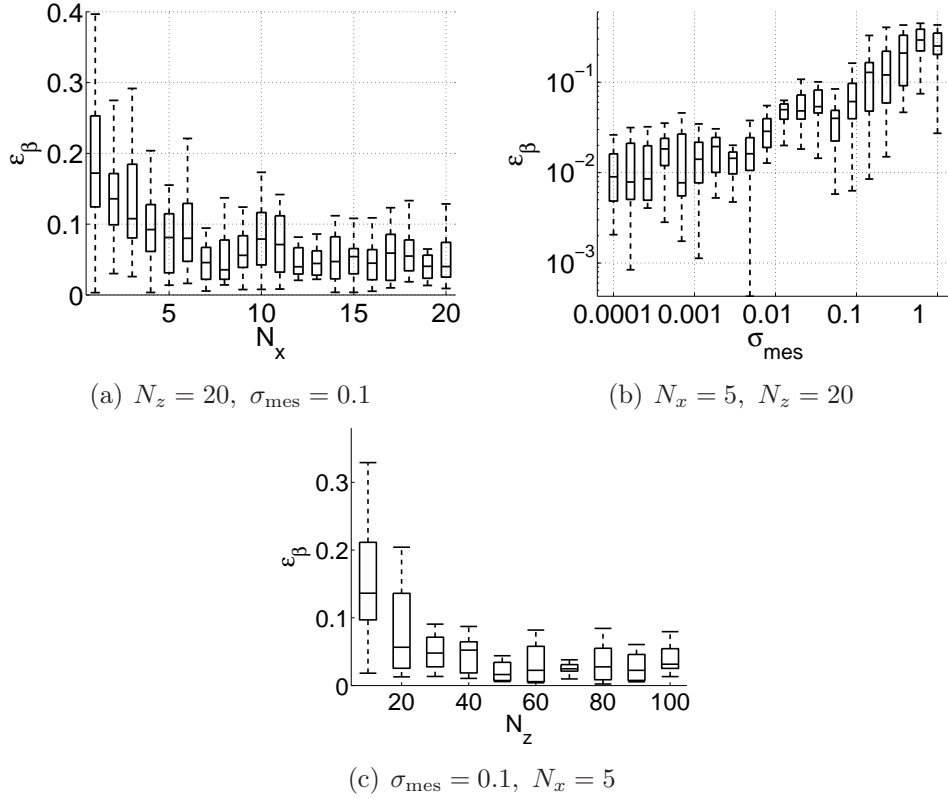


Figure 7: Dependence of the calibration results on the number of code evaluations, N_z , the number of experimental measurements, N_x , and the level of experimental uncertainty, σ_{mes} . The boxplots are produced from 50 repetitions of the calibration procedure. In each case, the initial designs of experiments for the construction of the surrogate models correspond to space filling LHS, whereas the values of x for the experimental data are uniformly drawn in $[-g\ell, g\ell]$.

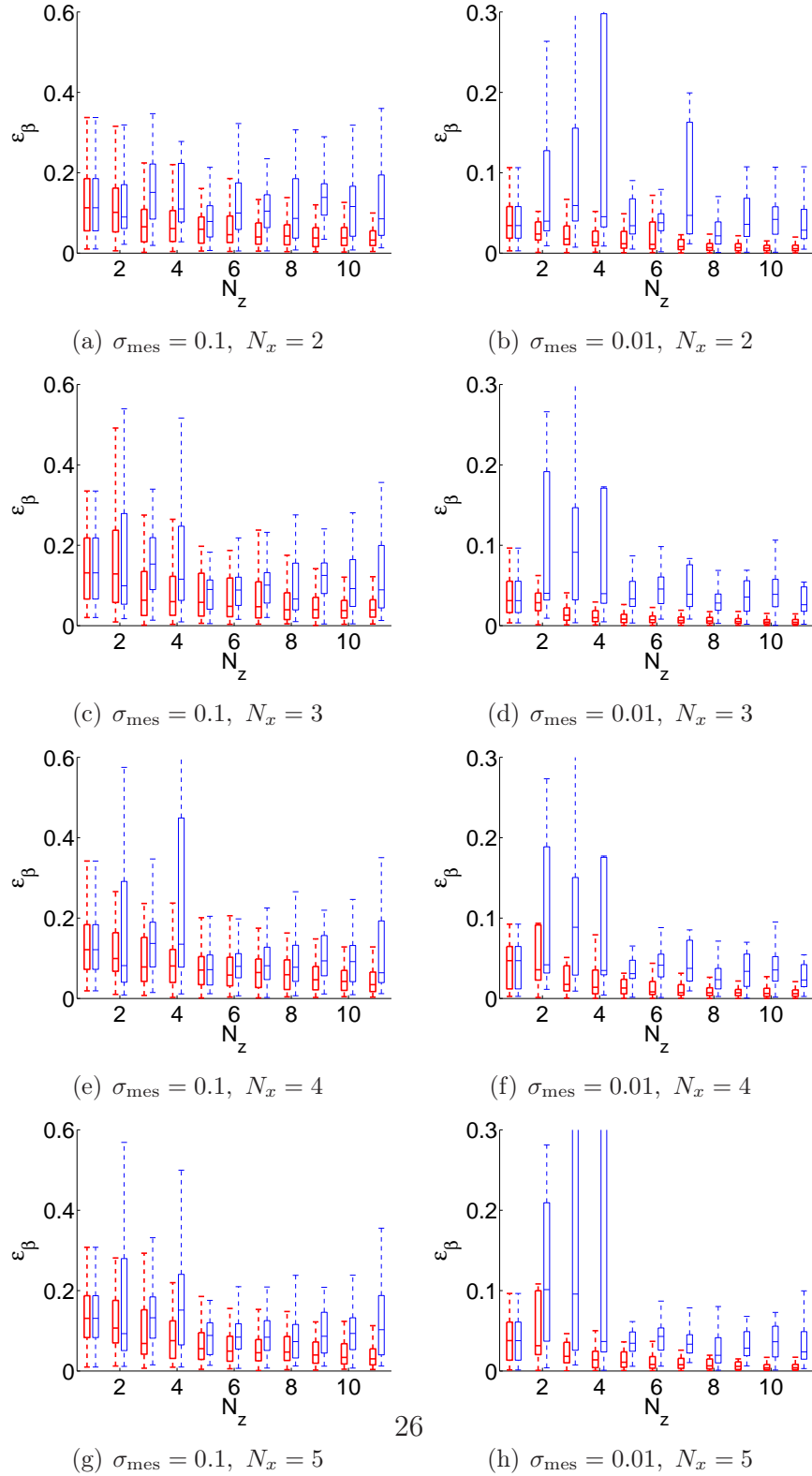


Figure 8: Influence of the positions of the new code evaluations on the calibration results. The boxplots in red correspond to the proposed adaptive method, whereas the boxplots in blue are based on independent N_z -dimensional space filling LHS. Each boxplot is produced from 10 repetitions of the calibration procedure with different initial designs of experiments.

$$\Sigma_{\text{mes}} = \mathbf{I}_{N_x} \otimes \mathbf{R}_{\text{mes},t}, \quad (56)$$

where the $(N_t \times N_t)$ -dimensional matrix $\mathbf{R}_{\text{mes},t}$ is supposed to be known.

To predict the inner pressure, the vessel is modelled by a spherical steel shell, and the explosive is supposed to be a homogeneous sphere. As an illustration, the true containment vessel and its spherical model are represented in Figure 10. The hydrodynamics code is then used to simulate the explosive burns, the internal gas dynamics, and the shock wave propagation. An elastic-plastic law is considered to describe the mechanical deformations of the vessel. The pressure evolution is then deduced from this calculation for a duration that is sufficient to cover the initial blast loading and the principal reverberations inside the vessel. The hydrodynamics code we are considering requires the calibration of three parameters. The first one, β_1 , characterizes the speed of the shock wave. The second one, β_2 , controls the shock intensity. The third one, β_3 , plays on the time attenuation of the shock intensity. The definition domain for these three parameters is supposed to be $[-1, 1]$, such that $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)$ is an element of $\mathbb{B} = [-1, 1]^3$. For any values of x and $\boldsymbol{\beta}$, let $\mathbf{g}(x, \boldsymbol{\beta}) = (g(t_1; x, \boldsymbol{\beta}), \dots, g(t_{N_t}; x, \boldsymbol{\beta}))$ be the vector that gathers the simulated pressure at each element of \mathbb{T} . The individual influence of these three parameters can be seen in Figure 11. This hydrodynamics code is supposed to have no discrepancy in the sense that we assume that it exists a true value for $\boldsymbol{\beta}$, $\boldsymbol{\beta}^*$, such that:

$$\mathbf{y}(x) = \mathbf{g}(x, \boldsymbol{\beta}^*). \quad (57)$$

A uniform prior is once again chosen for $\boldsymbol{\beta}^*$:

$$f_{\boldsymbol{\beta}^*}(\boldsymbol{\beta}) = \begin{cases} 1/8 & \text{if } \boldsymbol{\beta} \in [-1, 1]^3, \\ 0 & \text{otherwise.} \end{cases} \quad (58)$$

We now assume that the hydrodynamics code has been evaluated in N_z values of $\mathbf{z} := (x, \beta_1, \beta_2, \beta_3)$ in $\mathbb{Z} = [0.9 \times \min(x^{(1)}, \dots, x^{(5)}), 1.1 \times \max(x^{(1)}, \dots, x^{(5)})] \times \mathbb{B}$. Given these values, we can construct a Gaussian process-based surrogate model to predict the value of $\mathbf{g}(\mathbf{z})$ for any non-observed point \mathbf{z} . Hence, vector-valued function $\mathbf{z} \mapsto \mathbf{g}(\mathbf{z})$ is supposed to be a particular realization of a Gaussian process, whose mean and covariance functions are denoted by $\boldsymbol{\mu}$ and \mathbf{C} . Unlike the first application, we do

not have access, in this second application, to a simplified modelling of the pressure evolution, which makes the definition of $\boldsymbol{\mu}$ harder. The "Universal Kriging" formalism is therefore adopted, which consists in assuming a linear parametric representation for $\boldsymbol{\mu}$:

$$\boldsymbol{\mu}(\mathbf{z}) = \mathbf{A}\mathbf{h}(\mathbf{z}), \quad \mathbf{h}(\mathbf{z}) = (1, \mathbf{z}), \quad \mathbf{z} \in \mathbb{Z}. \quad (59)$$

There, \mathbf{A} is a $(N_t \times 5)$ -dimensional matrix that needs to be identified from the N_z code evaluations. In addition, a separate structure is proposed for function \mathbf{C} . A Matern-5/2 covariance is chosen for the covariance structure in \mathbf{z} , and the covariance in time is given by its empirical estimator given by Eq. (27). This choice is motivated by two main reasons. First, reduced-basis representations are not well adapted to this kind of functions, which is mostly due to the fact that parameters x and β_1 play on the arrival time of the shock wave. Thus, several dozens of projection functions are required to make these representations relevant. Secondly, as it is explained in Section 3.1.2, considering a separate structure for \mathbf{C} allows us to exploit the separate structure of $\boldsymbol{\Sigma}_{\text{mes}}$ and to reduce the computational cost of the likelihood evaluation. As an illustration of the prediction capability of the surrogate model, Figure 12 compares the true code response and its Gaussian predictor at four randomly chosen values of \mathbf{z} in \mathbb{Z} . This Gaussian predictor is based on the evaluation of the hydrodynamics code in only $N_z = 40$ values of \mathbf{z} , which were uniformly chosen in \mathbb{Z} using a space filling LHS.

Given this surrogate model of \mathbf{g} , we can now search $\boldsymbol{\beta}^*$ using the developments presented in Sections 2 and 3. For this application, as the value of $\boldsymbol{\beta}^*$ is unknown and the experimental uncertainty is small, the following L^2 error is introduced to assess the relevance of its estimation:

$$\varepsilon_{L^2}^2 := \sum_{n=1}^{N_x} \|\mathbf{y}_{\text{mes}}(x^{(n)}) - \mathbf{g}(x^{(n)}, \boldsymbol{\beta}^{\text{MLE}})\|^2, \quad (60)$$

where $\boldsymbol{\beta}^{\text{MLE}}$ is the maximum likelihood estimate of $\boldsymbol{\beta}^*$ given the available experimental data. Two approaches are compared for the identification of $\boldsymbol{\beta}^*$ in the following.

- On the one hand, we call "non adaptative calibration" the calibration of $\boldsymbol{\beta}^*$ that is based on N_z independent evaluations of the hydrodynamics code in N_z points that are uniformly chosen in \mathbb{Z} .

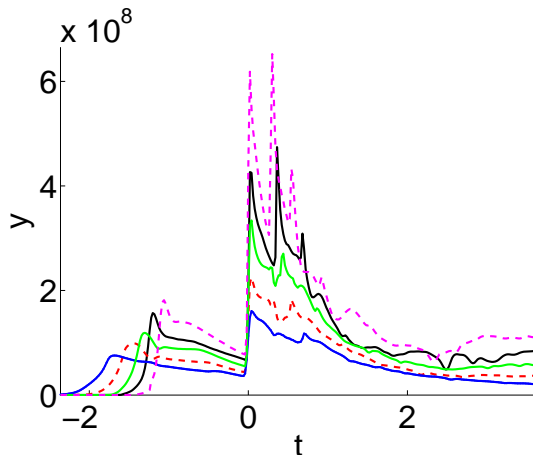


Figure 9: Evolution of the pressure at the inner surface of the containment vessel with respect to the time for the five available measurements. The more explosive, the higher the pressure maximum. For each measure, the time of the first shock reflection is chosen as the time origin.

- On the other hand, we call "adaptive calibration" the calibration of β^* that is based on the concatenation of 20 evaluations of the hydrodynamics code in 20 points that are uniformly chosen in \mathbb{Z} , and $N_z - 20$ additional evaluations of the code in well chosen positions, as it is proposed in Section 3.2.

The calibration results associated with these two approaches are shown in Figure 13. Hence, much less code evaluations are needed to get the same value of $\varepsilon_{L^2}^2$ when considering the proposed adaptive calibration than the non adaptive one. The interest of the proposed procedure can also be seen in Figure 14, which compares the evolution of each measured inner pressure to its 95% credible area using the calibrated code. To lay stress on the relevance of the proposed approach, for all $1 \leq n \leq N_x$, the credible area of the n^{th} experiment is based on all the available experiments but the n^{th} . That is, the credible area of the n^{th} experiment is based on the post-processing of 20 evaluations of the hydrodynamics code in 20 independent realizations of β_{-n}^* , where we denoted by β_{-n}^* the estimator of β^* , whose PDF has been computed using all the available measurements but the n^{th} . In each case, the calibration results are based on the proposed adaptive procedure with a maximum number of code evaluations that is equal to $N_z = 40$.

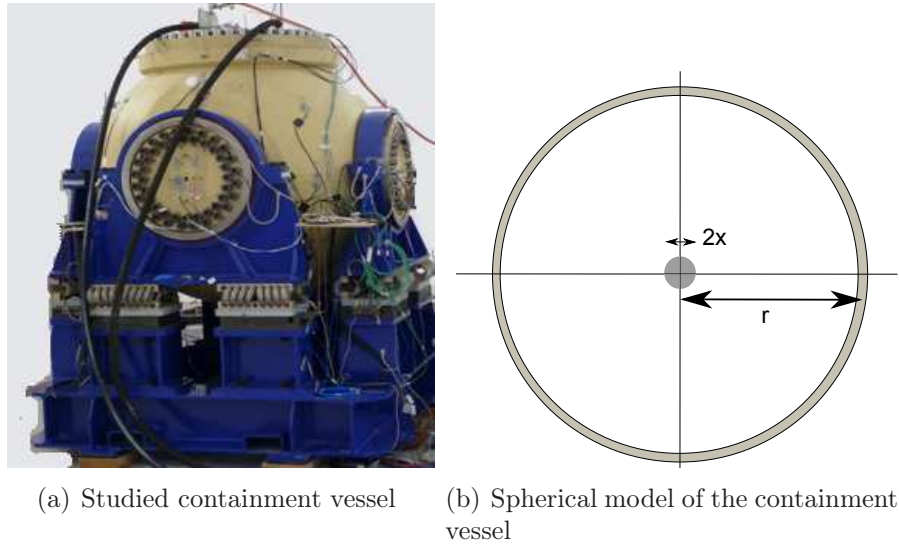


Figure 10: Presentation of the true containment vessel and its spherical model.

5. Conclusion

This work considers the challenging problem of calibrating the parameters of computationally demanding computer models when few high-dimensional physical observations are available. In that prospect, several surrogate models for multi-outputs computer codes have been presented, and an adaptive procedure has been proposed to minimize the uncertainty on the calibration parameters using a reduced number of code evaluations. The efficiency of the proposed approach has eventually been illustrated on two applications. The first one was based on generated data, whereas the second one was based on experimental data.

The main assumption of this work is the fact that the computer code has no discrepancy. If it is not the case, it is important to introduce a model error in the calibration process. However, this model error is generally unknown and has also to be identified from the available experimental data. When only few measurements are available and when surrogate models have to be introduced to emulate the computer code to solve the inference problem, as it is the case in this work, proposing simple but relevant structures for the model error is an interesting perspective for this work.

In the same manner, the fact that the pressure is always positive has not been taken into account in the construction of the surrogate models of the

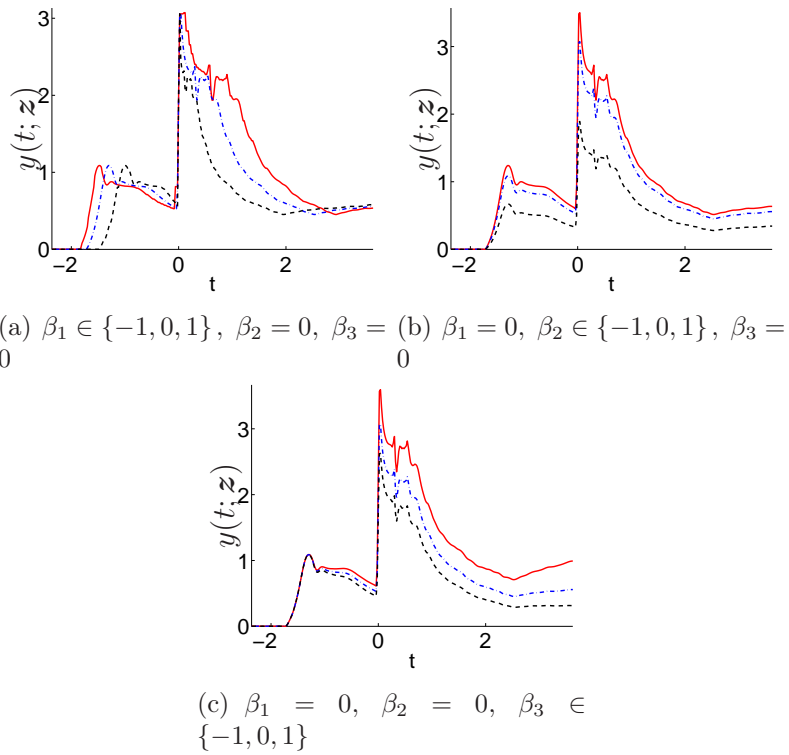


Figure 11: Sensitivity of the code response to the values of β_1 , β_2 and β_3 . In each figure, two components of $\beta = (\beta_1, \beta_2, \beta_3)$ are fixed to 0, whereas the third component is equal to -1 for the dashed black line, to 0 for the blue dotted line, and to 1 for the red continuous line.

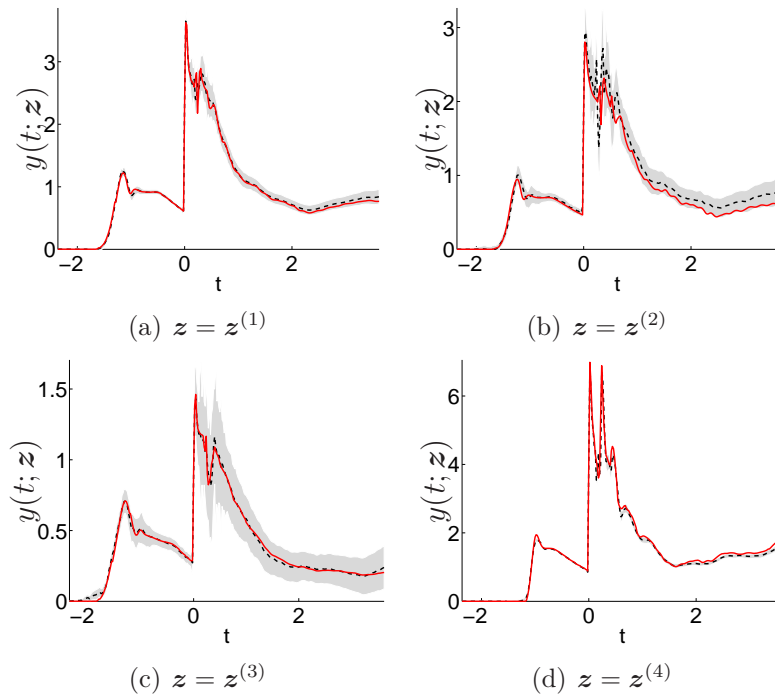


Figure 12: Relevance of the proposed Gaussian process-based surrogate model to predict the value of the simulated inner pressure at four particular non-observed values of z , given $N_z = 40$ code evaluations. The code response is represented in red continuous line. The thick black dotted lines correspond to the mean predictions, whereas the grey areas show the 95% credible intervals for the prediction.

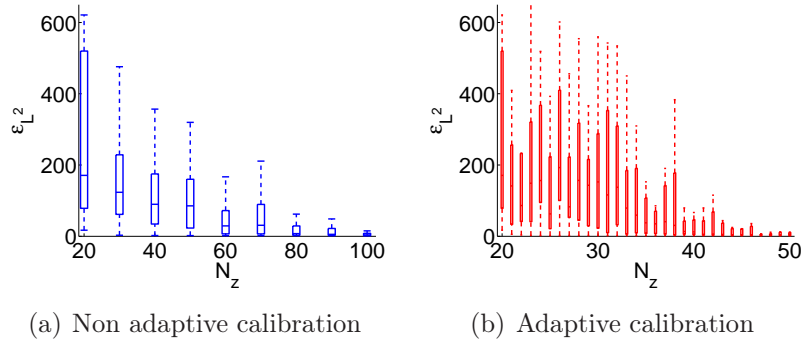


Figure 13: Interest of the adaptive calibration in terms of computational cost reduction for the identification of β^* . The boxplots associated with the "non adaptive calibration" (left) are based on independent N_z -dimensional space filling LHS. The boxplots associated with the "adaptive calibration" (right) correspond to the proposed adaptive method. Each boxplot is produced from 20 repetitions of the calibration procedure with different initial designs of experiments.

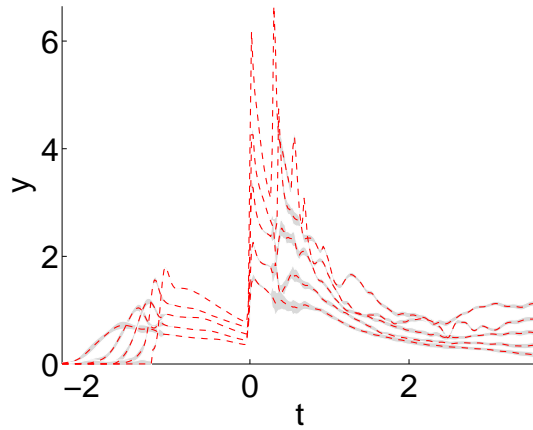


Figure 14: Cross validation prediction of the five available measured evolutions of the pressure. Grey area : 95%–credible interval for the prediction. Red continuous line: experimental data.

second application. Trying to integrate positivity or monotony constraints in the surrogate models seems also promising for future work.

Appendix

□ Proof of Eq. (12)

For all \mathbf{z}, \mathbf{z}' in \mathbb{Z} , we have:

$$(\mathbf{g}(\mathbf{z}), \mathbf{g}_{\text{obs}}) | \mu, C \sim \mathcal{N} \left(\begin{bmatrix} \boldsymbol{\mu}(\mathbf{z}) \\ \boldsymbol{\mu}_{\text{obs}} \end{bmatrix}, \begin{bmatrix} \mathbf{R}(\mathbf{z}, \mathbf{z}) & \mathbf{S}(\mathbf{z}) \\ \mathbf{S}(\mathbf{z})^T & \mathbf{R}_{\text{obs}} \end{bmatrix} \right). \quad (61)$$

By property of the Gaussian conditioning, it comes:

$$\mathbb{E}[\mathbf{g}(\mathbf{z}) | \mu, C, \mathbf{g}_{\text{obs}}] = \boldsymbol{\mu}(\mathbf{z}) + \mathbf{S}(\mathbf{z})^T \mathbf{R}_{\text{obs}}^{-1} (\mathbf{g}_{\text{obs}} - \boldsymbol{\mu}_{\text{obs}}), \quad (62)$$

$$\text{Cov}(\mathbf{g}(\mathbf{z}), \mathbf{g}(\mathbf{z}') | \mu, C, \mathbf{g}_{\text{obs}}) = \mathbf{R}(\mathbf{z}, \mathbf{z}') - \mathbf{S}(\mathbf{z})^T \mathbf{R}_{\text{obs}}^{-1} \mathbf{S}(\mathbf{z}'). \quad (63)$$

□

□ Proof of Eq. 23

For all $1 \leq i, j \leq N_t$ and all \mathbf{z}, \mathbf{z}' in \mathbb{Z} , if $C(t_i, t_j; \mathbf{z}, \mathbf{z}') = C_t(t_i, t_j) C_z(\mathbf{z}, \mathbf{z}')$, then:

- $\mathbf{R}(\mathbf{z}, \mathbf{z}') = C_z(\mathbf{z}, \mathbf{z}') \mathbf{R}_t$,
- $\mathbf{R}_{\text{obs}} = \mathbf{R}_z \otimes \mathbf{R}_t$,
- $\mathbf{S}(\mathbf{z}) = \mathbf{r}_z(\mathbf{z}) \otimes \mathbf{R}_t$.

We deduce:

$$\begin{aligned} \mathbb{E}[\mathbf{g}(\mathbf{z}) | \mu, C_t, C_z, \mathbf{g}_{\text{obs}}] &= \boldsymbol{\mu}(\mathbf{z}) + (\mathbf{r}_z(\mathbf{z}) \otimes \mathbf{R}_t)^T (\mathbf{R}_z \otimes \mathbf{R}_t)^{-1} (\mathbf{g}_{\text{obs}} - \boldsymbol{\mu}_{\text{obs}}), \\ &= \boldsymbol{\mu}(\mathbf{z}) + (\mathbf{r}_z(\mathbf{z})^T \otimes \mathbf{R}_t) (\mathbf{R}_z^{-1} \otimes \mathbf{R}_t^{-1}) (\mathbf{g}_{\text{obs}} - \boldsymbol{\mu}_{\text{obs}}), \\ &= \boldsymbol{\mu}(\mathbf{z}) + (\mathbf{r}_z(\mathbf{z})^T \mathbf{R}_z^{-1}) \otimes \mathbf{I}_{N_t} (\mathbf{g}_{\text{obs}} - \boldsymbol{\mu}_{\text{obs}}), \\ &= \boldsymbol{\mu}(\mathbf{z}) + (\mathbf{G}_{\text{obs}} - \mathbf{M}_{\text{obs}}) \mathbf{R}_z^{-1} \mathbf{r}_z(\mathbf{z}), \end{aligned} \quad (64)$$

$$\begin{aligned}
\text{Cov}(\mathbf{g}(\mathbf{z}), \mathbf{g}(\mathbf{z}') | \mu, C_t, C_z, \mathbf{g}_{\text{obs}}) &= \mathbf{R}(\mathbf{z}, \mathbf{z}') - (\mathbf{r}_z(\mathbf{z}) \otimes \mathbf{R}_t)^T (\mathbf{R}_z \otimes \mathbf{R}_t)^{-1} (\mathbf{r}_z(\mathbf{z}')^T \otimes \mathbf{R}_t), \\
&= \mathbf{R}(\mathbf{z}, \mathbf{z}') - (\mathbf{r}_z(\mathbf{z})^T \otimes \mathbf{R}_t) (\mathbf{R}_z^{-1} \otimes \mathbf{R}_t^{-1}) (\mathbf{r}_z(\mathbf{z}')^T \otimes \mathbf{R}_t), \\
&= (C_z(\mathbf{z}, \mathbf{z}') - \mathbf{r}_z(\mathbf{z})^T \mathbf{R}_z^{-1} \mathbf{r}_z(\mathbf{z}')) \mathbf{R}_t.
\end{aligned} \tag{65}$$

□

Proposition 1. For all \mathbf{z} in \mathbb{Z} , if $\boldsymbol{\mu}(\mathbf{z}) = \mathbf{F}(\mathbf{z})\boldsymbol{\alpha}$, where $\mathbf{F}(\mathbf{z})$ is a $(N_t \times N_a)$ -dimensional matrix, and if $\boldsymbol{\alpha} \sim \mathcal{U}_{\mathbb{R}^{N_a}}$, then:

$$\mathbf{g} \mid \mathbf{F}, C_t, C_z, \mathbf{g}_{\text{obs}} \sim GP(\boldsymbol{\mu}_{**}, \mathbf{R}_{**}), \tag{66}$$

$$\left\{ \begin{array}{l}
\mathbf{F}_{\text{obs}} := (\mathbf{F}(\mathbf{z}^{(1)}) \dots, \mathbf{F}(\mathbf{z}^{(N_x)})), \\
\mathbf{Q}_* := (\mathbf{F}_{\text{obs}}^T \mathbf{R}_z^{-1} \otimes \mathbf{R}_t^{-1} \mathbf{F}_{\text{obs}})^{-1}, \\
\boldsymbol{\alpha}_* := \mathbf{Q}_* \mathbf{F}_{\text{obs}}^T \mathbf{R}_z^{-1} \otimes \mathbf{R}_t^{-1} \mathbf{g}_{\text{obs}}, \\
\boldsymbol{\mu}_{**}(\mathbf{z}) := \mathbf{U}(\mathbf{z})\boldsymbol{\alpha}_* + ((\mathbf{r}_z(\mathbf{z})^T \mathbf{R}_z^{-1}) \otimes \mathbf{I}_{N_t}) \mathbf{g}_{\text{obs}}, \\
\mathbf{U}(\mathbf{z}) := \mathbf{F}(\mathbf{z}) - ((\mathbf{r}_z(\mathbf{z})^T \mathbf{R}_z^{-1}) \otimes \mathbf{I}_{N_t}) \mathbf{F}_{\text{obs}}, \\
\mathbf{Q}_{**}^{-1} := \mathbf{Q}_*^{-1} + c_*^{-1}(\mathbf{z}, \mathbf{z}) \mathbf{U}(\mathbf{z})^T \mathbf{R}_t^{-1} \mathbf{U}(\mathbf{z}), \\
\mathbf{R}_{**}(\mathbf{z}, \mathbf{z})^{-1} := c_*^{-1}(\mathbf{z}, \mathbf{z}) \mathbf{R}_t^{-1} - c_*^{-1}(\mathbf{z}, \mathbf{z}) \mathbf{R}_t^{-1} \mathbf{U}(\mathbf{z}) \mathbf{Q}_{**} \mathbf{U}(\mathbf{z})^T \mathbf{R}_t^{-1} c_*^{-1}(\mathbf{z}, \mathbf{z}).
\end{array} \right. \tag{67}$$

□ **Proof of Proposition 1**

First, we can calculate:

$$\begin{aligned}
&(\mathbf{g}_{\text{obs}} - \mathbf{F}_{\text{obs}}\boldsymbol{\alpha})^T \mathbf{R}_z^{-1} \otimes \mathbf{R}_t^{-1} (\mathbf{g}_{\text{obs}} - \mathbf{F}_{\text{obs}}\boldsymbol{\alpha}) \\
&= \mathbf{g}_{\text{obs}}^T \mathbf{R}_z^{-1} \otimes \mathbf{R}_t^{-1} \mathbf{g}_{\text{obs}} + \boldsymbol{\alpha}^T \mathbf{F}_{\text{obs}}^T \mathbf{R}_z^{-1} \otimes \mathbf{R}_t^{-1} \mathbf{F}_{\text{obs}} \boldsymbol{\alpha} - 2\boldsymbol{\alpha}^T \mathbf{F}_{\text{obs}}^T \mathbf{R}_z^{-1} \otimes \mathbf{R}_t^{-1} \mathbf{g}_{\text{obs}} \\
&= \mathbf{g}_{\text{obs}}^T \mathbf{R}_z^{-1} \otimes \mathbf{R}_t^{-1} \mathbf{g}_{\text{obs}} - \boldsymbol{\alpha}_*^T \mathbf{Q}_*^{-1} \boldsymbol{\alpha}_* + (\boldsymbol{\alpha} - \boldsymbol{\alpha}_*)^T \mathbf{Q}_*^{-1} (\boldsymbol{\alpha} - \boldsymbol{\alpha}_*) \\
&= \mathbf{g}_{\text{obs}}^T (\mathbf{R}_z^{-1} \otimes \mathbf{R}_t^{-1} - \mathbf{R}_z^{-1} \otimes \mathbf{R}_t^{-1} \mathbf{F}_{\text{obs}} \mathbf{Q}_* \mathbf{F}_{\text{obs}}^T \mathbf{R}_z^{-1} \otimes \mathbf{R}_t^{-1}) \mathbf{g}_{\text{obs}} + (\boldsymbol{\alpha} - \boldsymbol{\alpha}_*)^T \mathbf{Q}_*^{-1} (\boldsymbol{\alpha} - \boldsymbol{\alpha}_*).
\end{aligned}$$

Then, $\boldsymbol{\mu}_*(\mathbf{z}) = \mathbf{U}(\mathbf{z})\boldsymbol{\alpha} + ((\mathbf{r}_z(\mathbf{z})^T \mathbf{R}_z^{-1}) \otimes \mathbf{I}_{N_t}) \mathbf{g}_{\text{obs}}$, such that we have:

$$\begin{aligned}
\mathbf{g}(\mathbf{z}) - \boldsymbol{\mu}_*(\mathbf{z}) &= \mathbf{g}(\mathbf{z}) - \boldsymbol{\mu}_{**}(\mathbf{z}) + \boldsymbol{\mu}_{**}(\mathbf{z}) - \boldsymbol{\mu}_*(\mathbf{z}) \\
&= \mathbf{g}(\mathbf{z}) - \boldsymbol{\mu}_{**}(\mathbf{z}) + \mathbf{U}(\mathbf{z})(\boldsymbol{\alpha}_* - \boldsymbol{\alpha}),
\end{aligned}$$

Introducing $\alpha_{**} := \alpha_* + c_*^{-1}(z, z) \mathbf{Q}_{**} U(z)^T \mathbf{R}_t^{-1}(g(z) - \mu_{**}(z))$, it comes:

$$\begin{aligned}
& (g(z) - \mu_*(z))^T c_*^{-1}(z, z) \mathbf{R}_t^{-1}(g(z) - \mu_*(z)) + (\alpha - \alpha_*)^T \mathbf{Q}_*^{-1}(\alpha - \alpha_*) \\
&= (g(z) - \mu_{**}(z))^T c_*^{-1}(z, z) \mathbf{R}_t^{-1}(g(z) - \mu_{**}(z)) + (\alpha - \alpha_*)^T \mathbf{Q}_{**}^{-1}(\alpha - \alpha_*) \\
&\quad + 2(\alpha_* - \alpha)^T U(z)^T c_*^{-1}(z, z) \mathbf{R}_t^{-1}(g(z) - \mu_{**}(z)) \\
&= (g(z) - \mu_{**}(z))^T c_*^{-1}(z, z) \mathbf{R}_t^{-1}(g(z) - \mu_{**}(z)) + 2\alpha_*^T U(z)^T c_*^{-1}(z, z) \mathbf{R}_t^{-1}(g(z) - \mu_{**}(z)) \\
&\quad + \alpha^T \mathbf{Q}_{**}^{-1} \alpha + \alpha_*^T \mathbf{Q}_{**}^{-1} \alpha_* - 2\alpha^T (\mathbf{Q}_{**}^{-1} \alpha_* + U(z)^T c_*^{-1}(z, z) \mathbf{R}_t^{-1}(g(z) - \mu_{**}(z))), \\
&= (g(z) - \mu_{**}(z))^T c_*^{-1}(z, z) \mathbf{R}_t^{-1}(g(z) - \mu_{**}(z)) + (\alpha - \alpha_{**})^T \mathbf{Q}_{**}^{-1}(\alpha - \alpha_{**}) \\
&\quad + \alpha_*^T \mathbf{Q}_{**}^{-1} \alpha_* + 2\alpha_*^T U(z)^T c_*^{-1}(z, z) \mathbf{R}_t^{-1}(g(z) - \mu_*(z)) - \alpha_{**}^T \mathbf{Q}_{**}^{-1} \alpha_{**}.
\end{aligned}$$

Noticing that:

$$\begin{aligned}
\alpha_{**}^T \mathbf{Q}_{**}^{-1} \alpha_{**} &= \alpha_*^T \mathbf{Q}_{**}^{-1} \alpha_* - 2\alpha_*^T U(z)^T c_*^{-1}(z, z) \mathbf{R}_t^{-1}(y(z) - \mu_*(z)) \\
&\quad + (g(z) - \mu_{**}(z))^T c_*^{-1}(z, z) \mathbf{R}_t^{-1} U(z) \mathbf{Q}_{**} U(z)^T \mathbf{R}_t^{-1} c_*^{-1}(z, z) (g(z) - \mu_{**}(z)),
\end{aligned}$$

we get:

$$\begin{aligned}
& (g(z) - \mu_*(z))^T c_*^{-1}(z, z) \mathbf{R}_t^{-1}(g(z) - \mu_*(z)) + (\alpha - \alpha_*)^T \mathbf{Q}_*^{-1}(\alpha - \alpha_*) \\
&= (g(z) - \mu_{**}(z))^T \mathbf{R}_{**}(z, z)^{-1}(g(z) - \mu_{**}(z)) + (\alpha - \alpha_{**})^T \mathbf{Q}_{**}^{-1}(\alpha - \alpha_{**}).
\end{aligned}$$

Let $f_{g(z)|g_{\text{obs}}}$, $f_{g(z)|\alpha, g_{\text{obs}}}$, $f_{\alpha|g_{\text{obs}}}$ and f_{α} be the PDFs of $(g(z)|\mathbf{F}, C_t, C_z, \mathbf{g}_{\text{obs}})$, $(g(z)|\alpha, \mathbf{F}, C_t, C_z, \mathbf{g}_{\text{obs}})$, $(\alpha|\mathbf{F}, C_t, C_z, \mathbf{g}_{\text{obs}})$ and α respectively. By definition, for all \mathbf{y} in \mathbb{R}^{N_t} , we have:

$$\begin{aligned}
f_{g(z)|g_{\text{obs}}}(\mathbf{y}) &= \int_{\mathbb{R}^{N_a}} f_{g(z)|\alpha, g_{\text{obs}}}(\mathbf{y}) f_{\alpha|g_{\text{obs}}}(\mathbf{a}) d\mathbf{a}, \\
&\propto \int_{\mathbb{R}^{N_a}} \exp\left(-\frac{1}{2} \left(\begin{array}{c} (\mathbf{y} - \mu_*(z))^T c_*^{-1}(z, z) \mathbf{R}_t^{-1}(\mathbf{y} - \mu_*(z)) \\ + (\mathbf{a} - \alpha_*)^T \mathbf{Q}_*^{-1}(\mathbf{a} - \alpha_*) \end{array} \right)\right) f_{\alpha}(\mathbf{a}) d\mathbf{a}. \\
&\propto \int_{\mathbb{R}^{N_a}} \exp\left(-\frac{1}{2} \left(\begin{array}{c} (\mathbf{y} - \mu_{**}(z))^T \mathbf{R}_{**}(z, z)^{-1}(\mathbf{y} - \mu_{**}(z)) \\ + (\mathbf{a} - \alpha_{**})^T \mathbf{Q}_{**}^{-1}(\mathbf{a} - \alpha_{**}) \end{array} \right)\right) f_{\alpha}(\mathbf{a}) d\mathbf{a}
\end{aligned} \tag{68}$$

If $\alpha \sim \mathcal{U}_{\mathbb{R}^{N_a}}$, then $f_{\alpha}(\mathbf{a}) \propto 1$, and we eventually deduce:

$$g(z)|\mathbf{F}, C_t, C_z, \mathbf{g}_{\text{obs}} \sim \mathcal{N}(\mu_{**}(z), \mathbf{R}_{**}(z, z)). \tag{69}$$

□

Proposition 2. For all \mathbf{z} in \mathbb{Z} , if $\boldsymbol{\mu}(\mathbf{z}) = \mathbf{F}\mathbf{A}\mathbf{h}(\mathbf{z})$, where \mathbf{F} is a $(N_t \times N_a)$ -dimensional matrix, $\mathbf{h}(\mathbf{z})$ is a M -dimensional vector, and if $\mathbf{A} \sim \mathcal{U}_{\mathcal{M}_{N_a \times M}(\mathbb{R})}$, then:

$$\mathbf{g} \mid \mathbf{F}, \mathbf{h}, C_t, C_z, \mathbf{g}_{obs} \sim GP(\boldsymbol{\mu}_{**}, \mathbf{R}_{**}), \quad (70)$$

$$\begin{cases} \mathbf{H} := [\mathbf{h}(\mathbf{z}^{(1)}) \ \dots \ \mathbf{h}(\mathbf{z}^{(N_z)})], \\ \mathbf{A}_* := (\mathbf{F}^T \mathbf{R}_t^{-1} \mathbf{F})^{-1} \mathbf{F}^T \mathbf{R}_t^{-1} \mathbf{G}_{obs} \mathbf{R}_z^{-1} \mathbf{H}^T (\mathbf{H} \mathbf{R}_z^{-1} \mathbf{H}^T)^{-1}, \\ \mathbf{u}(\mathbf{z}) := \mathbf{h}(\mathbf{z}) - \mathbf{H} \mathbf{R}_z^{-1} \mathbf{r}_z(\mathbf{z}), \\ \boldsymbol{\mu}_{**}(\mathbf{z}) := \mathbf{G}_{obs} \mathbf{R}_z^{-1} \mathbf{r}_z(\mathbf{z}) + \mathbf{F} \mathbf{A}_* \mathbf{u}(\mathbf{z}), \\ u_*(\mathbf{z}) := \mathbf{u}(\mathbf{z})^T (\mathbf{H} \mathbf{R}_z^{-1} \mathbf{H}^T)^{-1} \mathbf{u}(\mathbf{z}) c_*^{-1}(\mathbf{z}, \mathbf{z}), \\ \mathbf{R}_{**}^{-1}(\mathbf{z}, \mathbf{z}) := c_*^{-1}(\mathbf{z}, \mathbf{z}) \mathbf{R}_t^{-1} - c_*^{-1}(\mathbf{z}, \mathbf{z}) \mathbf{R}_t^{-1} \mathbf{F} (\mathbf{F} \mathbf{R}_t^{-1} \mathbf{F}^T)^{-1} \mathbf{F}^T \mathbf{R}_t^{-1} u_*(\mathbf{z}) / (1 + u_*(\mathbf{z})). \end{cases} \quad (71)$$

where $\mathcal{M}_{N_a \times M}(\mathbb{R})$ is the set of all the $(N_a \times M)$ -dimensional real-valued matrices.

□ **Proof of Proposition 2**

If $\mathbf{f}_{obs}(\mathbf{A}) := (\mathbf{F}\mathbf{A}\mathbf{h}(\mathbf{z}^{(1)}), \dots, \mathbf{F}\mathbf{A}\mathbf{h}(\mathbf{z}^{(N_z)}))$, then we have:

$$\begin{aligned} & (\mathbf{g}_{obs} - \mathbf{f}_{obs}(\mathbf{A}))^T \mathbf{R}_z^{-1} \otimes \mathbf{R}_t^{-1} (\mathbf{g}_{obs} - \mathbf{f}_{obs}(\mathbf{A})), \\ &= \text{Tr} \left((\mathbf{G}_{obs} - \mathbf{F}\mathbf{A}\mathbf{H}) \mathbf{R}_z^{-1} (\mathbf{G}_{obs} - \mathbf{F}\mathbf{A}\mathbf{H})^T \mathbf{R}_t^{-1} \right), \\ &= \text{Tr} \left(\begin{aligned} & (\mathbf{G}_{obs} \mathbf{R}_z^{-1} \mathbf{G}_{obs}^T - \mathbf{F} \mathbf{A}_* (\mathbf{H} \mathbf{R}_z^{-1} \mathbf{H}^T) \mathbf{A}_*^T \mathbf{F}^T) \mathbf{R}_t^{-1} \\ & + (\mathbf{A} - \mathbf{A}_*) (\mathbf{H} \mathbf{R}_z^{-1} \mathbf{H}^T) (\mathbf{A} - \mathbf{A}_*)^T \mathbf{F}^T \mathbf{R}_t^{-1} \mathbf{F} \end{aligned} \right). \end{aligned} \quad (72)$$

Moreover, we have:

$$(\mathbf{r}_z(\mathbf{z})^T \mathbf{R}_z^{-1}) \otimes \mathbf{I}_{N_t} (\mathbf{g}_{obs} - \mathbf{f}_{obs}(\mathbf{A})) = (\mathbf{G}_{obs} - \mathbf{F}\mathbf{A}\mathbf{H}) \mathbf{R}_z^{-1} \mathbf{r}_z(\mathbf{z}), \quad (73)$$

such that:

$$\boldsymbol{\mu}_*(\mathbf{z}) = \mathbf{G}_{obs} \mathbf{R}_z^{-1} \mathbf{r}_z(\mathbf{z}) + \mathbf{F} \mathbf{A} \mathbf{u}(\mathbf{z}), \quad (74)$$

$$\mathbf{g}(\mathbf{z}) - \boldsymbol{\mu}_*(\mathbf{z}) = \mathbf{g}(\mathbf{z}) - \boldsymbol{\mu}_{**}(\mathbf{z}) + \mathbf{F}(\mathbf{A}_* - \mathbf{A})\mathbf{u}(\mathbf{z}). \quad (75)$$

It comes:

$$\begin{aligned}
& (\mathbf{g}(\mathbf{z}) - \boldsymbol{\mu}_*(\mathbf{z}))^T c_*^{-1}(\mathbf{z}, \mathbf{z}) \mathbf{R}_t^{-1} (\mathbf{g}(\mathbf{z}) - \boldsymbol{\mu}_*(\mathbf{z})) + (\mathbf{g}_{\text{obs}} - \mathbf{f}_{\text{obs}}(\mathbf{A}))^T \mathbf{R}_z^{-1} \otimes \mathbf{R}_t^{-1} (\mathbf{g}_{\text{obs}} - \mathbf{f}_{\text{obs}}(\mathbf{A})) \\
& \propto (\mathbf{g}(\mathbf{z}) - \boldsymbol{\mu}_{**}(\mathbf{z}))^T c_*^{-1}(\mathbf{z}, \mathbf{z}) \mathbf{R}_t^{-1} (\mathbf{g}(\mathbf{z}) - \boldsymbol{\mu}_{**}(\mathbf{z})) + \text{Tr} \left((\mathbf{A} - \mathbf{A}_*) \widehat{\mathbf{Q}}^{-1} (\mathbf{A} - \mathbf{A}_*)^T \mathbf{F}^T \mathbf{R}_t^{-1} \mathbf{F} \right) \\
& \quad - 2(\mathbf{g}(\mathbf{z}) - \boldsymbol{\mu}_{**}(\mathbf{z}))^T c_*^{-1}(\mathbf{z}, \mathbf{z}) \mathbf{R}_t^{-1} \mathbf{F} (\mathbf{A} - \mathbf{A}_*) \mathbf{u}(\mathbf{z}) \\
& \propto (\mathbf{g}(\mathbf{z}) - \boldsymbol{\mu}_{**}(\mathbf{z}))^T \mathbf{R}_{**}^{-1}(\mathbf{z}, \mathbf{z}) (\mathbf{g}(\mathbf{z}) - \boldsymbol{\mu}_{**}(\mathbf{z})) + \text{Tr} \left((\mathbf{A} - \mathbf{A}_{**}) \widehat{\mathbf{Q}}^{-1} (\mathbf{A} - \mathbf{A}_{**})^T \mathbf{F}^T \mathbf{R}_t^{-1} \mathbf{F} \right),
\end{aligned} \tag{76}$$

where \mathbf{A}_{**} is a constant matrix that does not depend on $\mathbf{g}(\mathbf{z})$ and \mathbf{A} , and $\widehat{\mathbf{Q}}^{-1} := \mathbf{H} \mathbf{R}_x^{-1} \mathbf{H}^T + c_x^{-1}(\mathbf{z}, \mathbf{z}) \mathbf{u}(\mathbf{z}) \mathbf{u}(\mathbf{z})^T$.

Assuming a prior distribution for \mathbf{A} that is constant (improper prior), integrating \mathbf{A} , we eventually get:

$$\mathbf{g}(\mathbf{z}) | \mathbf{F}, \mathbf{h}, C_t, C_x, \mathbf{g}_{\text{obs}} \sim \mathcal{N}(\boldsymbol{\mu}_{**}(\mathbf{z}), \mathbf{R}_{**}(\mathbf{z}, \mathbf{z})). \tag{77}$$

□

Proposition 3. *If:*

- N_z and N_t are two integers such that $N_z \geq N_t$,
- \mathbf{A} is a $(N_z \times N_z)$ -dimensional symmetric positive-definite matrix,
- \mathbf{W} is a $(N_t \times N_z)$ -dimensional matrix such that $\mathbf{W} \mathbf{A}^{-1} \mathbf{W}^T$ can be inverted,
- \mathbf{w} is the vector gathering the elements of \mathbf{W} columnwise, such that $\mathbf{w} = ((\mathbf{W})_{11}, \dots, (\mathbf{W})_{N_t 1}, (\mathbf{W})_{12}, \dots, (\mathbf{W})_{N_t N_z})$,
- $\mathbb{M}^+(\mathbb{R})$ is the set of all the real-valued $(N_t \times N_t)$ -dimensional symmetric positive-definite matrices,
- \mathcal{C} is the cost function such that for all \mathbf{B} in $\mathbb{M}^+(\mathbb{R})$:

$$\mathcal{C}(\mathbf{B}) = \frac{\exp\left(-\frac{1}{2} \mathbf{w}^T \mathbf{A}^{-1} \otimes \mathbf{B}^{-1} \mathbf{w}\right)}{(2\pi)^{N_z N_t / 2} \det(\mathbf{A})^{N_t / 2} \det(\mathbf{B})^{N_z / 2}}, \tag{78}$$

Then:

$$\frac{1}{N_z} \mathbf{W} \mathbf{A}^{-1} \mathbf{W}^T = \arg \max_{\mathbf{B} \in \mathbb{M}^+(\mathbb{R})} \mathcal{C}(\mathbf{B}). \quad (79)$$

□ **Proof of Property 3**

Let \mathbf{D}_A and $\boldsymbol{\ell}_A$ be the orthogonal matrix and the vector that respectively gather the eigenvectors and the eigenvalues of $\mathbf{Z} \mathbf{A}^{-1} \mathbf{Z}^T$:

$$\mathbf{Z} \mathbf{A}^{-1} \mathbf{Z}^T = \mathbf{D}_A \text{diag}(\boldsymbol{\ell}_A) \mathbf{D}_A^T.$$

In the same manner, for any matrix \mathbf{B} in $\mathbb{M}^+(\mathbb{R})$, let \mathbf{D}_B and $\boldsymbol{\ell}_B$ be the orthogonal matrix and the vector that respectively gather the eigenvectors and the eigenvalues of \mathbf{B}^{-1} :

$$\mathbf{B}^{-1} = \mathbf{D}_B \text{diag}(\boldsymbol{\ell}_B) \mathbf{D}_B^T.$$

It comes:

$$\begin{aligned} \mathbf{z}^T \mathbf{A}^{-1} \otimes \mathbf{B}^{-1} \mathbf{z} &= \text{Tr} [(\mathbf{Z} \mathbf{A}^{-1} \mathbf{Z}^T) \mathbf{B}^{-1}], \\ &= \text{Tr} [\mathbf{D}_B^T \mathbf{D}_A \text{diag}(\boldsymbol{\ell}_A) \mathbf{D}_A^T \mathbf{D}_B \text{diag}(\boldsymbol{\ell}_B)], \\ &= \text{Tr} [\mathbf{V} \text{diag}(\boldsymbol{\ell}_A) \mathbf{V}^T \text{diag}(\boldsymbol{\ell}_B)], \\ &= \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} \ell_{A,i} V_{ij}^2 \ell_{B,j}, \end{aligned} \quad (80)$$

where $\mathbf{V} := \mathbf{D}_B^T \mathbf{D}_A$ is an orthogonal matrix.

We deduce:

$$\log(\mathcal{C}(\mathbf{B})) = c + \frac{N_x}{2} \sum_{j=1}^{N_t} \log(\ell_{B,j}) - \frac{1}{2} \sum_{j=1}^{N_t} \ell_{B,j} \sum_{i=1}^{N_t} \ell_{A,i} V_{ij}^2, \quad (81)$$

where c is a constant that do not depend on \mathbf{B} . For all $1 \leq j \leq N_t$, the application $\ell_{B,j} \mapsto \log(\mathcal{C}(\mathbf{B}))$ is concave, and we have:

$$\frac{\partial \log(\mathcal{C}(\mathbf{B}))}{\partial \ell_{B,j}} = \frac{N_x}{2\ell_{B,j}} - \frac{1}{2} \sum_{i=1}^{N_t} \ell_{A,i} V_{ij}^2. \quad (82)$$

Let ℓ_B^* be the vector such that for all $1 \leq j \leq N_t$,

$$\ell_{B,j}^* = \frac{N_x}{\sum_{i=1}^{N_t} \ell_{A,i} V_{ij}^2}. \quad (83)$$

As for all $1 \leq i \leq N_t$, $\sum_{j=1}^{N_t} V_{ij}^2 = 1$, the concavity of the log function gives:

$$\begin{aligned} \log(\mathcal{C}(\mathbf{B})) &\leq \log(\mathcal{C}(\mathbf{D}_B \text{diag}(\ell_B^*)^{-1} \mathbf{D}_B^T)), \\ &\leq c - \frac{N_x}{2} \sum_{j=1}^{N_t} \log \left(\frac{1}{N_x} \sum_{i=1}^{N_t} \ell_{A,i} V_{ij}^2 \right) - \frac{N_x N_t}{2}, \\ &\leq c - \frac{N_x}{2} \sum_{i=1}^{N_t} \log \left(\frac{1}{N_x} \ell_{A,i} \right) \sum_{j=1}^{N_t} V_{ij}^2 - \frac{N_x N_t}{2}, \\ &\leq c - \frac{N_x}{2} \sum_{i=1}^{N_t} \log \left(\frac{1}{N_x} \ell_{A,i} \right) - \frac{N_x N_t}{2}. \end{aligned} \quad (84)$$

Hence, if $\mathbf{D}_B = \mathbf{D}_A$, then $V_{ij} = 1$ if $i = j$ and zero otherwise, such that $\log(\mathcal{C}(\mathbf{D}_B \text{diag}(\ell_B^*)^{-1} \mathbf{D}_B^T)) = c - \frac{N_x}{2} \sum_{i=1}^{N_t} \log \left(\frac{1}{N_x} \ell_{A,i} \right) - \frac{N_x N_t}{2}$ and :

$$\text{diag}(\ell_B^*)^{-1} = \frac{1}{N_x} \text{diag}(\ell_A). \quad (85)$$

Finally, we conclude that:

$$\frac{1}{N_x} \mathbf{D}_A \text{diag}(\ell_A) \mathbf{D}_A^T = \frac{1}{N_x} \mathbf{Z} \mathbf{A}^{-1} \mathbf{Z}^T = \arg \max_{\mathbf{B} \in \mathbb{M}^+(\mathbb{R})} \mathcal{C}(\mathbf{B}). \quad (86)$$

□

□ Proof of Eqs. 35 and 36

If $\mathbf{R}_{\text{mes},\beta}$ and $\mathbf{R}_{\text{mes},t}$ are two symmetric positive definite matrices, there exists two invertible matrices, $\mathbf{V}_\beta(\boldsymbol{\beta})$ and \mathbf{V}_t , and two diagonal matrices, $\text{diag}(\boldsymbol{\lambda}_\beta)$ and $\text{diag}(\boldsymbol{\lambda}_t)$, such that:

$$\mathbf{R}_{\text{meta}}(\boldsymbol{\beta}) = \mathbf{V}_\beta(\boldsymbol{\beta}) \text{diag}(\boldsymbol{\lambda}_\beta) \mathbf{V}_\beta(\boldsymbol{\beta})^T, \quad \mathbf{R}_{\text{mes},\beta} = \mathbf{V}_\beta(\boldsymbol{\beta}) \mathbf{V}_\beta(\boldsymbol{\beta})^T, \quad (87)$$

$$\mathbf{R}_t = \mathbf{V}_t \text{diag}(\boldsymbol{\lambda}_t) \mathbf{V}_t^T, \quad \mathbf{R}_{\text{mes},t} = \mathbf{V}_t \mathbf{V}_t^T. \quad (88)$$

Moreover, by property of the Kronecker product:

$$\begin{aligned}
& (\mathbf{V}_\beta(\boldsymbol{\beta}) \text{diag}(\boldsymbol{\lambda}_\beta) \mathbf{V}_\beta(\boldsymbol{\beta})^T) \otimes (\mathbf{V}_t \text{diag}(\boldsymbol{\lambda}_t) \mathbf{V}_t^T) + (\mathbf{V}_\beta(\boldsymbol{\beta}) \mathbf{V}_\beta(\boldsymbol{\beta})^T) \otimes (\mathbf{V}_t \mathbf{V}_t^T) \\
&= (\mathbf{V}_\beta(\boldsymbol{\beta}) \otimes \mathbf{V}_t) (\text{diag}(\boldsymbol{\lambda}_\beta) \otimes \text{diag}(\boldsymbol{\lambda}_t) + \mathbf{I}_{N_x} \otimes \mathbf{I}_{N_t}) (\mathbf{V}_\beta(\boldsymbol{\beta})^T \otimes \mathbf{V}_t^T) \\
&= (\mathbf{V}_\beta(\boldsymbol{\beta}) \otimes \mathbf{V}_t) \text{diag}(\Lambda) (\mathbf{V}_\beta(\boldsymbol{\beta})^T \otimes \mathbf{V}_t^T),
\end{aligned} \tag{89}$$

which leads to the searched result.

□

□ **Proof of Eq. 37**

$$\begin{aligned}
& \boldsymbol{\delta}(\boldsymbol{\beta})^T (\boldsymbol{\Sigma}_{\text{mes}} + \boldsymbol{\Sigma}_{\text{meta}}(\boldsymbol{\beta}))^{-1} \boldsymbol{\delta}(\boldsymbol{\beta}) \\
&= \boldsymbol{\delta}(\boldsymbol{\beta})^T ((\mathbf{V}_\beta(\boldsymbol{\beta}) \otimes \mathbf{V}_t) \text{diag}(\Lambda) (\mathbf{V}_\beta(\boldsymbol{\beta})^T \otimes \mathbf{V}_t^T))^{-1} \boldsymbol{\delta}(\boldsymbol{\beta}), \\
&= \boldsymbol{\delta}(\boldsymbol{\beta})^T ((\mathbf{V}_\beta(\boldsymbol{\beta})^{-T} \otimes \mathbf{V}_t^{-T}) \text{diag}(\Lambda)^{-1} (\mathbf{V}_\beta(\boldsymbol{\beta})^{-1} \otimes \mathbf{V}_t^{-1})) \boldsymbol{\delta}(\boldsymbol{\beta}), \\
&= \sum_{j=1}^{N_x} \sum_{i=1}^{N_x} \sum_{k=1}^{N_x} (\mathbf{V}_\beta(\boldsymbol{\beta})^{-1})_{ji} (\mathbf{V}_\beta(\boldsymbol{\beta})^{-1})_{jk} \boldsymbol{\delta}_i(\boldsymbol{\beta})^T \mathbf{V}_t^{-T} \text{diag}(\Lambda_j)^{-1} \mathbf{V}_t^{-1} \boldsymbol{\delta}_k(\boldsymbol{\beta}), \\
&= \sum_{j=1}^{N_x} \mathbf{v}^{(j)}(\boldsymbol{\beta})^T \boldsymbol{\Delta}^T \mathbf{V}_t^{-T} \text{diag}(\Lambda_j)^{-1} \mathbf{V}_t^{-1} \boldsymbol{\Delta} \mathbf{v}^{(j)}(\boldsymbol{\beta}), \\
&= \sum_{j=1}^{N_x} \left\| \text{diag}(\Lambda_j)^{-1/2} \mathbf{V}_t^{-1} \boldsymbol{\Delta} \mathbf{v}^{(j)}(\boldsymbol{\beta}) \right\|^2,
\end{aligned} \tag{90}$$

$$\begin{aligned}
& \det(\boldsymbol{\Sigma}_{\text{mes}} + \boldsymbol{\Sigma}_{\text{meta}}(\boldsymbol{\beta})), \\
&= \det((\mathbf{V}_\beta(\boldsymbol{\beta}) \otimes \mathbf{V}_t) \text{diag}(\Lambda) (\mathbf{V}_\beta(\boldsymbol{\beta})^T \otimes \mathbf{V}_t^T)), \\
&= \det(\text{diag}(\Lambda)) \det((\mathbf{V}_\beta(\boldsymbol{\beta})^T \otimes \mathbf{V}_t^T) (\mathbf{V}_\beta(\boldsymbol{\beta}) \otimes \mathbf{V}_t)) \\
&\propto \det(\mathbf{V}_\beta(\boldsymbol{\beta})^T \mathbf{V}_\beta(\boldsymbol{\beta}))^{N_t} \prod_{l=1}^{N_x N_t} \Lambda_l.
\end{aligned} \tag{91}$$

□

References

- [1] M. J. Bayarri, J. O. Berger, J. Cafeo, G. Garcia-Donato, F. Liu, J. Palomo, R. J. Parthasarathy, R. Paulo, J. Sacks, and D. Walsh. Computer model validation with functional output. *The Annals of Statistics*, 35:1874–1906, 2007.
- [2] M. J. Bayarri, J. O. Berger, M. C. Kennedy, A. Kottas, R. Paulo, J. Sack, J. A. Cafeo, C. H. Lin, and J. Tu. Predicting vehicle crash-worthiness: Validation of computer models for functional and hierarchical data. *Journal of the American Statistical Association*, 104:929–943, 2009.
- [3] M. J. Bayarri, V. De Oliveira, and B. Sanso. Objective bayesian analysis of spatially correlated data. *Journal of the American Statistical Association*, 96(456):1361–1374, 2007.
- [4] J. Bect, D. Ginsbourger, L. Li, V. Picheny, and E. Vasquez. Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22, 2012.
- [5] L. M. Berliner. Monte carlo based ensemble forecasting. *Statistics and Computing*, 11, 2001.
- [6] K. Campbell, M. D. McKay, and B. J. Williams. Sensitivity analysis when model outputs are functions. *Reliability Engineering and System Safety*, 91(10):1468–1472, 2006.
- [7] C. Chevalier, J. Bect, D. Ginsburger, E. Vasquez, V. Picheny, and Y. Richet. Fast kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics*, 56(4):455–465, 2014.
- [8] S. Conti and A. O’Hagan. Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, 140:640–651, 2010.
- [9] K.T. Fang, R. Li, and A. Sudjianto. *Design and modeling for computer experiments*. Chapman & Hall, Computer Science and Data Analysis Series, London, 2006.

- [10] K.T. Fang and D.K. Lin. Uniform experimental designs and their applications in industry. *Handbook of Statistics*, 22:131–178, 2003.
- [11] Thomas E. Fricker, Jeremy E. Oakley, and Nathan M. Urban. Multivariate gaussian process emulators with nonseparable covariance structures. *Technometrics*, 55(1):47–56, 2013.
- [12] D. Higdon, J. Gattiker, B. Williams, and M. Rightley. Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, 103(482):570–583, 2008.
- [13] Ying Hung, V. Roshan Joseph, and Shreyes N. Melkote. Analysis of computer experiments with functional response. *Technometrics*, 57(1):35–44, 2015.
- [14] D. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.
- [15] J. P . Kaipio and E. Somersalo. *Statistics and Computational Inverse Problems*. Springer, New York, 2004.
- [16] M.C. Kennedy and A. O’Hagan. Bayesian calibration of computer models. *Journal of the royal statistical society*, 63:425–464, 2001.
- [17] J. M . Marin and C. P. Robert. *Bayesian core*. Springer-Verlag, New York, 2007.
- [18] G. Perrin and C. Cannamela. A repulsion-based method for the definition and the enrichment of optimized space filling designs in constrained input spaces. *Journal de la Société Française de Statistique*, 158(1):37–67, 2017.
- [19] G. Perrin, C. Soize, D. Duhamel, and C. Funfschilling. Identification of polynomial chaos representations in high dimension from a set of realizations. *SIAM J. Sci. Comput.*, 34(6):2917–2945, 2012.
- [20] G. Perrin, C. Soize, and N. Ouhbi. Data-driven kernel representations for sampling with an unknown block dependence structure under correlation constraints. *Journal of Computational Statistics and Data Analysis*, 119:139–154, 2018.

- [21] M.T. Pratola, S.R. Sain, D. Bingham, M. Wiltberger, and E.J. Rigler. Fast sequential computer model calibration of large nonstationary spatial-temporal processes. *Technometrics*, 55(2):232–242, 2013.
- [22] P. Ranjan, M. Thomas, H. Teismann, and S. Mukhoti. Inverse problem for a time-series valued computer simulator via scalarization. *Open Journal of Statistics*, 6:528–544, 2016.
- [23] Jonathan Rougier. Efficient emulators for multivariate deterministic functions. *Journal of Computational and Graphical Statistics*, 17(4):827–843, 2008.
- [24] R. T. Rubinstein and D.P. Kroese. *Simulation and the Monte Carlo method*. John Wiley and Sons, Inc., Hoboken, New Jersey, 2008.
- [25] J. Sacks, W. Welch, T. Mitchell, and H. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4:409–435, 1989.
- [26] T. J. Santner, B.J. Williams, and W.I. Notz. *The design and analysis of computer experiments*. Springer, New York, 2003.
- [27] B. Williams, D. Higdon, J. Gattiker, L. Moore, M. McKay, and S. Keller-MacNulty. Combining experimental data and computer simulations, with an application to flyer plate experiments. *Bayesian Analysis*, 1:765–792, 2006.