Sequential Design Strategy for Kriging and Cokriging-Based Machine Learning in the Context of Reservoir History-Matching

A. Thenon, V. Gervais, M. Le Ravalec

Sequential design strategy for kriging and cokriging-based machine learning in the context of reservoir history-matching

A. Thenon*, V. Gervais, M. Le Ravalec

IFP Energies nouvelles, 1-4 avenue de Bois-Préau, 92852 Rueil-Malmaison Cedex – France
e-mail: veronique.gervais@ifpen.fr
**now with Modis*

Abstract

Numerical models representing geological reservoirs can be used to forecast production and help engineers to design optimal development plans. These models should be as representative as possible of the true dynamic behavior and reproduce available static and dynamic data. However, identifying models constrained to production data can be very challenging and time consuming. Machine learning techniques can be considered to mimic and replace the fluid flow simulator in the process. However, the benefit of these approaches strongly depends on the simulation time required to train reliable predictors. Previous studies highlighted the potential of the multi-fidelity approach rooted in cokriging to efficiently provide accurate estimations of fluid flow simulator outputs. This technique consists in combining simulation results obtained on several levels of resolution for the reservoir model to predict the output properties on the finest level (the most accurate one). The degraded levels can correspond for instance to a coarser discretization in space or time, or to less complex physics. The idea behind is to take advantage of the coarse level low-cost information to limit the total simulation time required to train the meta-models. In this paper, we propose a new sequential design strategy for iteratively and automatically training (kriging and) cokriging based meta-models. As highlighted on two synthetic cases, this approach makes it possible to identify training sets leading to accurate estimations for the error between measured and simulated production data (objective function) while requiring limited simulation times.

Keywords meta-modeling; multi-fidelity; machine learning; kriging; cokriging; sequential design; history-matching

1. Introduction

Numerical models representing underground geological reservoirs are used to forecast production for various exploitation schemes. They help engineers to design optimal development plans. A reservoir model consists of a grid representing the structure of the reservoir; it is populated with facies and

petrophysical properties such as porosity and permeability. These characteristics are generally distributed in the grid using geostatistical simulation methods. These techniques generate equiprobable realizations with given statistical properties. They can be constrained to available static data such as values of the target properties collected at wells (e.g. derived from logs) or secondary information inferred from 3D seismic data. Then, fluid flow simulators are used to compute the pressure evolution and fluid dynamics in the grid during production.

The reservoir models used for production forecast should be as representative as possible of the true dynamic behavior. They are thus chosen as the ones that reproduce not only the static data, but also the dynamic ones, namely production and 4D-seismic related data acquired on the field. However, the process of identifying such calibrated models can be very challenging and time consuming due to various factors, of which the potentially large number of model input parameters that need to be adjusted (petrophysical property in each grid block but also reservoir properties such as fluid characteristics, well properties …), the need to preserve the model geological consistency, the complexity of the non-linear physical processes involved, and the computation time required for a single fluid-flow simulation.

Different tools have been developed to assist in the resolution of this calibration – or history-matching – process. They aim at reducing the computational overburden required to solve the problem, and in particular the number of simulations to be performed [5,17]. Optimization algorithms can be applied to drive the perturbation of the model input parameters in order to minimize a cost (or objective) function that quantifies the mismatch between the dynamic data and the corresponding simulated values. Preliminary sensitivity analyses can also be conducted to identify the model parameters that influence the most this cost function. Considering the other parameters as constant in the calibration process can then simplify the problem and reduce computation times.

Simultaneously, machine learning techniques can be considered to mimic the simulator and provide fast and accurate estimations of its outputs for any parameter values. More precisely, the idea is to build meta-models that approximate the relation between the simulator input parameters and output properties. Such meta-models are trained on the basis of a sample of values, also called training set: they correspond to points from the parameter space and their associated simulated responses. Different methods exist to build meta-models (see for instance [7] for a review): polynomial regression, polynomial chaos expansion, Gaussian processes (rooted in kriging), neural networks, support vector regression ... Several of them have already been used in the reservoir engineering context. Reviews and examples can be found in [31,32,5] for example.

The benefit of using machine learning techniques in the calibration workflow strongly depends on the simulation time required to train relevant and accurate meta-models. The training set should be sufficiently informative on the relationship between the input and output properties, but also not too large to limit the total simulation time. If selecting a small ensemble, the meta-model may lack accuracy in some parts of the parameter space. Considering larger sets should improve the predictivity of the meta-model, but at the cost of longer simulation times. In practice, no predefined rules exist for the definition of a training set optimal in terms of simulation time and prediction accuracy, and the definition and simulation of this ensemble is a potential source of time wasting in the workflow. As a

result, sequential strategies have been proposed in the literature to overcome this limitation. They consist in generating first a small sample of the parameter space, and then in complementing iteratively this ensemble according to some criteria that depend on the objective of the study. Some approaches favor optimization purposes and aim at identifying new points close to optimum values [8]. Others are dedicated to the building of predictors that are accurate in the whole parameter space. We refer to these second approaches in what follows.

Kriging-based meta-models not only provide a prediction, but also a measure of the associated uncertainty through the kriging variance that represents the model mean square error. This variance can serve as a basis to identify regions of the parameter space that may not be well approximated by the current predictor. This was considered for instance in [27]: the new point added to the training set is the one that maximizes the kriging variance. In addition, as this variance only depends on the distance between the sampling points and not on their target output values, the approach can easily be extended to identify several new points at a time as described in [11]. This property of the kriging variance is also used in [27,25] to estimate the reduction of the integrated variance over the whole parameter space (Integrated Mean Square Error or IMSE) that would be obtained for given additional points. The training set is then augmented with the point corresponding to the maximum reduction of the IMSE. To better integrate the predictor accuracy in the sequential process, some authors refer to the cross-validation errors to complement the training set (e.g. [1,11]). These errors are used in [1] to identify regions of the parameter space in which performing additional simulations, and in [11] to weight the kriging variance with regards to Voronoi polygons around the training set. The additional points are the ones that maximize this weighted variance.

Another degree of freedom to reduce simulation time is the level of resolution of the numerical reservoir models: discretization in time and space, physics complexity… Coarser levels of resolution are generally faster to simulate, and even if their output values are less accurate, they can capture trends or features also representative of finer scales. This scale dependency can be integrated in the meta-model definition to save simulation time. For instance, multi-scale meta-models rooted in cokriging were proposed in [10, 13]. They provide estimations of the property of interest on the fine level while being trained with values simulated on different resolution levels (both fine and coarse). More precisely, the values simulated on the coarse level are considered as secondary information for the fine level, providing a trend that is adjusted according to the fine level training set. Multi-fidelity cokriging based meta-models proved to be efficient in reservoir engineering in [15] to compute quality maps and in [29] in the context of history-matching for two levels of resolution. Less simulation time can be required to build accurate predictors compared to meta-models rooted in kriging, provided that the coarse level is sufficiently informative on the fine level. In these reservoir engineering studies [15,29], the potential of the multi-fidelity approach was assessed on fixed training sets. The objective of the present paper is to complement the multi-scale workflow developed in [29] for history-matching with a sequential design strategy to make the process fully efficient and suitable for the operational context. Compared to the single level case, this strategy should not only identify the parameter values for the additional points, but also the level of resolution on which performing the corresponding simulations to get a good trade-off between simulation cost and information inferred for the process. Multi-scale design approaches were already proposed in [30] and

[11] for instance. The strategy introduced in [30] consists in enlarging an initial sample on both levels to generate largest nested samples. In [11], two steps are considered. First, the simulation point to be added is identified as the one that maximizes the weighted cokriging variance of the fine level predictor. Then, the level of fidelity on which performing the simulation for this point is chosen based on IMSE reduction and simulation time ratio between the levels. Here, we propose a new sequential approach derived from existing strategies for the building of both kriging and cokriging based estimators. It is designed to efficiently provide estimations of scalar outputs of interest that are accurate in the whole parameter space, but also to limit computational requirements for identifying new points. The potential of this approach is assessed for the meta-modeling of the objective function in the context of history-matching. The resulting estimations can then be considered for sensitivity analysis to identify the most relevant parameters to be adjusted in the minimization process. The developed strategy may also be envisioned for the exploration step in probabilistic approaches, such as the one proposed in [5]. Note however that the proposed sequential approach is generic and can be considered for the prediction of any scalar output whatever the simulation context.

The paper outline is as follows. The definitions of kriging and cokriging based meta-modeling are recapped in section 2 in a generic context. Section 3 is dedicated to the proposed sequential design strategy. It is first introduced in the kriging framework and then extended to the multi-fidelity context. Finally, its application for the prediction of the history-matching error is presented in section 4, considering two synthetic test cases.

2. Kriging and cokriging based meta-modeling

In this section, we recap the basics of kriging and cokriging based meta-models for scalar functions (sections 2.1 and 2.2, respectively), and how they extend to functional properties such as time-dependent ones (section 2.4). The approaches are presented in a generic context. The indices computed to quantify the quality of the meta-models are introduced in section 2.3.

2.1 Kriging based meta-modeling

Let $\boldsymbol{y}(\boldsymbol{x})$ be the target scalar output that needs to be approximated and $\boldsymbol{x} = (\boldsymbol{x_1}, \dots, \boldsymbol{x_d}) \in \mathbb{R}^{\boldsymbol{d}}$ the vector of input parameters. In reservoir engineering, $\boldsymbol{y}$ can refer to the objective function or to any production property at a given time such as the oil flow rate at a producer. This output is considered here to be a realization of a Gaussian process $\boldsymbol{Y}(\boldsymbol{x})$ defined by:

$$Y(\boldsymbol{x}) = m(\boldsymbol{x}) + Z(\boldsymbol{x}) \tag{1}$$

$m(\boldsymbol{x})$ represents the mean of the Gaussian process and $Z(\boldsymbol{x})$ the fluctuations around the mean. m is generally modeled by $\boldsymbol{h^T}(\boldsymbol{x})\boldsymbol{\beta}$ where $\boldsymbol{h}(\boldsymbol{x}) = (h_j(\boldsymbol{x}))_{j=1\dots p}$ are some regression functions and $\boldsymbol{\beta} = (\beta_j)_{j=1\dots p}$ the corresponding regression coefficients. $Z(\boldsymbol{x})$ is a stationary Gaussian process with zero mean and covariance K. In practice, K is generally defined with a parametric form. In what follows, we consider the Matérn 5/2 covariance function [22]:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 R(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \prod_{j=1}^{d} k_j(x_j, x_j')$$

with $k_j(x_j, x_j') = \left(1 + \frac{\sqrt{5}|x_j - x_j'|}{\lambda_j} + \frac{5(x_j - x_j')^2}{3\lambda_j^2}\right) exp\left(-\frac{\sqrt{5}|x_j - x_j'|}{\lambda_j}\right)$. σ^2 denotes the process variance and $(\lambda_j)_{j=1\ldots d}$ the parameters of the correlation function R.

Let us now assume that the value of the target output $\boldsymbol{y}$ is known for a sample of the parameter space $\boldsymbol{D} = \{\boldsymbol{x^1}, \ldots, \boldsymbol{x^n}\}, \boldsymbol{x^i} \in \mathbb{R}^{\boldsymbol{d}}$, also called design of experiments. This ensemble of $\boldsymbol{n}$ values $(\boldsymbol{y}(\boldsymbol{x^1}), \ldots, \boldsymbol{y}(\boldsymbol{x^n}))$ is denoted $\boldsymbol{y^D}$ in what follows. Then, the best linear predictor of $\boldsymbol{y}$ at a new point $\boldsymbol{x}^*$ given $\boldsymbol{y^D}$ is:

$$\hat{y}(\boldsymbol{x}^*) = \boldsymbol{h^T}(\boldsymbol{x}^*)\boldsymbol{\beta} + \boldsymbol{r}^T(\boldsymbol{x}^*)\boldsymbol{R^{-1}}\left(\boldsymbol{y^D} - \boldsymbol{H\beta}\right) \quad (2)$$

where $\boldsymbol{R} \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{n}}$, $\boldsymbol{H} \in \mathbb{R}^{\boldsymbol{p}\times\boldsymbol{p}}$ and $\boldsymbol{r}(\boldsymbol{x}^*) \in \mathbb{R}^{\boldsymbol{n}}$ are defined as: $\boldsymbol{R_{ij}} = \boldsymbol{R}(\boldsymbol{x^i}, \boldsymbol{x^j})$, $\boldsymbol{H_{ij}} = \boldsymbol{h_j}(\boldsymbol{x^i})$ and $\boldsymbol{r_j}(\boldsymbol{x}^*) = \boldsymbol{R}(\boldsymbol{x^j}, \boldsymbol{x}^*)$. The variance of predictor $\hat{\boldsymbol{y}}$, also called kriging variance, is given by:

$$\hat{s}^2(\boldsymbol{x}^*) = \sigma^2\left(1 - (\boldsymbol{h^T}(\boldsymbol{x}^*) \quad \boldsymbol{r^T}(\boldsymbol{x}^*))\begin{pmatrix} 0 & \boldsymbol{H^T} \\ \boldsymbol{H} & \boldsymbol{R} \end{pmatrix}^{-1}\begin{pmatrix} \boldsymbol{h}(\boldsymbol{x}^*) \\ \boldsymbol{r}(\boldsymbol{x}^*) \end{pmatrix}\right) \quad (3)$$

It represents the model mean square error. Let us note that, for a given covariance function K, this variance only depends on the location of the sample in the parameter space, and not on the actual value of the target output for the sample. It can thus be easily computed for any sample $\boldsymbol{D}$ without the corresponding estimation of $y(\boldsymbol{D})$. This property will be used to identify a batch of points in sequential design strategies (see [11] for instance).

The last step to complete the definition of predictor $\hat{y}$ consists in estimating the covariance function characteristics σ^2 and $(\lambda_j)_{j=1\ldots d}$, as well as the $\boldsymbol{\beta}$ regression coefficients. This is usually done by maximizing the logarithm of the likelihood function [20]. Assuming the Gaussian vector $Y(\boldsymbol{D})$ follows a multivariate normal distribution and ignoring constant terms, the logarithm of the L likelihood function associated to the hyper-parameters is defined as [28]:

$$\log\left(L\left(\boldsymbol{y^D}|\boldsymbol{\beta}, \sigma^2, \boldsymbol{\lambda}\right)\right) = -\frac{1}{2}\left(n\log\sigma^2 + \log\left(\det(\boldsymbol{R})\right) + \frac{\left(\boldsymbol{y^D} - \boldsymbol{H\beta}\right)^T\boldsymbol{R^{-1}}\left(\boldsymbol{y^D} - \boldsymbol{H\beta}\right)}{\boldsymbol{\sigma^2}}\right) \quad (4)$$

The maximum likelihood estimates for $\boldsymbol{\beta}$ et $\sigma^{\boldsymbol{2}}$ given $\boldsymbol{\lambda}$ are then:

$$\widehat{\boldsymbol{\beta}}(\boldsymbol{\lambda}) = \left(\boldsymbol{H^T R^{-1} H}\right)^{-1}\boldsymbol{H^T R^{-1} y^D} \quad (5)$$

and

$$\widehat{\sigma^2}(\boldsymbol{\lambda}) = \frac{1}{n}\left(\boldsymbol{y^D} - \boldsymbol{H}\widehat{\boldsymbol{\beta}}\right)^T\boldsymbol{R^{-1}}\left(\boldsymbol{y^D} - \boldsymbol{H}\widehat{\boldsymbol{\beta}}\right) \quad (6)$$

Substituting these estimators in equation (4), the log likelihood rewrites:

$$\log\left(L\left(\boldsymbol{y^D}|\boldsymbol{\lambda}\right)\right) = -\frac{1}{2}\left(n\log\widehat{\sigma^2}(\boldsymbol{\lambda}) + \log\left(\det(\boldsymbol{R})\right) + n\right) \quad (7)$$

This expression depends on $\boldsymbol{\lambda}$ alone. The maximum likelihood estimate for $\boldsymbol{\lambda}$ can then be obtained by minimizing $n \log \widehat{\sigma^2} + \mathbf{l}\log\big(\det(\boldsymbol{R})\big)$. In this work, the kriging based meta-models are computed using the DiceKriging R package [26] and a constant mean m.

2.2 Cokriging based meta-modeling

Let us now consider the multi-fidelity framework. We assume that different levels of resolution are available for the simulations and that they form a hierarchy of codes with decreasing levels of accuracy. They can be obtained for instance with discretizations coarsening in time or space. The objective of the multi-fidelity meta-modeling is then to approximate the output of interest on the finest level using the values simulated for this property on all levels. To do so, we refer to the approach rooted in cokriging that has been introduced in [10] and further improved in [13]. For simplicity, it is recalled here with two levels of resolution. They are referred to as the fine and the coarse levels in what follows, the coarse one being the less accurate. However, the method is valid for any number of levels.

Let us denote $y_f(\boldsymbol{x})$ the output property of interest on the fine level and $y_c(\boldsymbol{x})$ the corresponding output on the coarse level, $\boldsymbol{x} \in \mathbb{R}^d$ being the vector of input parameters. As previously, these outputs are assumed to be realizations of Gaussian processes denoted by $Y_f(\boldsymbol{x})$ and $Y_c(\boldsymbol{x})$, respectively. The model proposed in [10] is based on the following Markov-type property assumption:

$$Cov\left(Y_f(\boldsymbol{x}), Y_c(\boldsymbol{x}')\middle| Y_c(\boldsymbol{x})\right) = 0 \quad \forall\, \boldsymbol{x} \neq \boldsymbol{x}' \tag{8}$$

In other words, we consider that, if $Y_c(\boldsymbol{x})$ is known, the values simulated on the coarse level for any other input parameter values $\boldsymbol{x}' \neq \boldsymbol{x}$ will not provide additional information on $Y_f(\boldsymbol{x})$. As demonstrated in [19], this assumption guarantees the existence of $\rho(\boldsymbol{x})$ such that $Y_f(\boldsymbol{x}) - \rho(\boldsymbol{x})Y_c(\boldsymbol{x})$ and $Y_c(\boldsymbol{x})$ are independent and leads to the following representation of the Gaussian process Y_f:

$$Y_f(\boldsymbol{x}) = \rho(\boldsymbol{x})Y_c(\boldsymbol{x}) + Y_d(\boldsymbol{x}) \tag{9}$$

$Y_d(\boldsymbol{x})$ denotes a Gaussian process independent of $Y_c(\boldsymbol{x})$ that represents the difference $y_d(\boldsymbol{x})$ between $y_f(\boldsymbol{x})$ and $\rho(\boldsymbol{x})y_c(\boldsymbol{x})$. For simplicity, the scale factor ρ is taken constant in what follows. Formulations associated to a parametric form can be found in [13].

Referring to equation (1) and to the notations introduced in section 2.1, we denote by $\boldsymbol{h_c^T}(\boldsymbol{x})\boldsymbol{\beta_c}$ and $\boldsymbol{h_d^T}(\boldsymbol{x})\boldsymbol{\beta_d}$ the mean of Gaussian processes $Y_c(\boldsymbol{x})$ and $Y_d(\boldsymbol{x})$, respectively, and by $K_c(\boldsymbol{x},\boldsymbol{x}') = \sigma_c^2 R_c(\boldsymbol{x},\boldsymbol{x}')$ and $K_d(\boldsymbol{x},\boldsymbol{x}') = \sigma_d^2 R_d(\boldsymbol{x},\boldsymbol{x}')$ their covariance functions. Let us also assume known $\boldsymbol{y_f^D} = y_f(\boldsymbol{D_f}) \in \mathbb{R}^{n_f}$, the values of the target output on the fine level for a design of experiments $\boldsymbol{D_f} = \left\{\boldsymbol{x_f^1}, \dots, \boldsymbol{x_f^{n_f}}\right\}, \boldsymbol{x_f^i} \in \mathbb{R}^d$, as well as $\boldsymbol{y_c^D} = y_c(\boldsymbol{D_c}) \in \mathbb{R}^{n_c}$ the values of the target output on the coarse level for another sample of the parameter space $\boldsymbol{D_c} = \left\{\boldsymbol{x_c^1}, \dots, \boldsymbol{x_c^{n_c}}\right\}, \boldsymbol{x_c^i} \in \mathbb{R}^d$. We assume here that the two designs $\boldsymbol{D_f}$ and $\boldsymbol{D_c}$ are nested in the sense that $\boldsymbol{D_f} \subseteq \boldsymbol{D_c}$. This assumption is not mandatory but simplifies the model parameter estimation as explained in [13,12]. Following equation (2), the best linear predictor of y_c at a new point $\boldsymbol{x}^*$ given $\boldsymbol{y_c^D}$ is:

$$\hat{y}_c(\boldsymbol{x}^*) = \boldsymbol{h_c^T}(\boldsymbol{x}^*)\boldsymbol{\beta_c} + \boldsymbol{r_c^T}(\boldsymbol{x}^*)\boldsymbol{R_c^{-1}}\left(\boldsymbol{y_c^D} - \boldsymbol{H_c}\boldsymbol{\beta_c}\right) \quad (10)$$

with associated variance

$$\hat{s}_c^2(\boldsymbol{x}^*) = \sigma_c^2\left(1 - (\boldsymbol{h_c^T}(\boldsymbol{x}^*) \quad \boldsymbol{r_c^T}(\boldsymbol{x}^*))\begin{pmatrix} 0 & \boldsymbol{H_c^T} \\ \boldsymbol{H_c} & \boldsymbol{R_c} \end{pmatrix}^{-1}\begin{pmatrix} \boldsymbol{h_c}(\boldsymbol{x}^*) \\ \boldsymbol{r_c}(\boldsymbol{x}^*) \end{pmatrix}\right). \quad (11)$$

Then, considering the recursive formulation introduced in [13,12], the multi-fidelity estimator for $\boldsymbol{y_f}$ is given by:

$$\hat{y}_f(\boldsymbol{x}^*) = \rho\hat{y}_c(\boldsymbol{x}^*) + \boldsymbol{h_d^T}(\boldsymbol{x}^*)\boldsymbol{\beta_d} + \boldsymbol{r_d^T}(\boldsymbol{x}^*)\boldsymbol{R_d^{-1}}\left(\boldsymbol{y_f^D} - \boldsymbol{H_d}\boldsymbol{\beta}_d\right),\ \boldsymbol{x}^* \in \mathbb{R}^d \quad (12)$$

with associated variance

$$\hat{s}_f^2(\boldsymbol{x}^*) = \rho^2\hat{s}_c^2(\boldsymbol{x}^*) + {\sigma_d}^2\left(1 - \left(\boldsymbol{h_d^T}(\boldsymbol{x}^*) \quad \boldsymbol{r_d^T}(\boldsymbol{x}^*)\right)\begin{pmatrix} 0 & \boldsymbol{H_d^T} \\ \boldsymbol{H_d} & \boldsymbol{R_d} \end{pmatrix}^{-1}\begin{pmatrix} \boldsymbol{h_d}(\boldsymbol{x}^*) \\ \boldsymbol{r_d}(\boldsymbol{x}^*) \end{pmatrix}\right) \quad (13)$$

In these formula, $(\boldsymbol{R_d})_{ij} = R_d(\boldsymbol{x_f^i}, \boldsymbol{x_f^j})$, $\boldsymbol{H_d} = \boldsymbol{h_d^T}(\boldsymbol{D_f})$ and $\boldsymbol{r_d^T}(\boldsymbol{x}^*) = R_d(\boldsymbol{D_f}, x^*)$. This recursive formulation for the multi-fidelity model is equivalent to the one initially proposed in [10] but reduces the computational cost in terms of time and memory [13]. Indeed, the building of a cokriging estimator with N levels of resolution becomes equivalent to the building of N consecutive kriging-based estimators, reducing the size of the matrices to be inverted and stored.

Thanks to the independence condition, parameters related to the coarse level ($\boldsymbol{\beta_c}, \sigma_c^2, \boldsymbol{\lambda_c}$) can be estimated independently from the ones characterizing the fine level ($\rho, \boldsymbol{\beta}_d, \sigma_d^2, \boldsymbol{\lambda_d}$). Likelihood maximization can be considered to estimate parameters ($\boldsymbol{\beta_c}, \sigma_c^2, \boldsymbol{\lambda_c}$) characterizing the coarse level kriging estimator $\hat{y}_c$ (equations (4) to (7)). For the fine level, closed-form expressions for the estimates of the scale factor ρ and regression coefficients $\boldsymbol{\beta}_d$ given $\boldsymbol{\lambda_d}$ are proposed in [12,13]. They are obtained from a joint Bayesian estimation of ρ and $\boldsymbol{\beta}_d$, leading to the following estimates:

$$\begin{pmatrix} \hat{\rho} \\ \widehat{\boldsymbol{\beta}_d} \end{pmatrix} = \left({\boldsymbol{F}_d}^T\boldsymbol{R_d^{-1}}\boldsymbol{F}_d\right)^{-1}{\boldsymbol{F}_d}^T\boldsymbol{R_d^{-1}}\boldsymbol{y_f^D}$$

with $\boldsymbol{F}_d = (\boldsymbol{y_c}(\boldsymbol{D_f}) \quad \boldsymbol{H_d})$. The estimate for σ_d^2 corresponds to the restricted maximum likelihood estimate: $\widehat{\sigma_d^2} = \frac{1}{n_f - p - 1}\left(\boldsymbol{y_f^D} - \boldsymbol{F}_d\begin{pmatrix} \hat{\rho} \\ \widehat{\boldsymbol{\beta}_d} \end{pmatrix}\right)^T\boldsymbol{R_d^{-1}}\left(\boldsymbol{y_f^D} - \boldsymbol{F}_d\begin{pmatrix} \hat{\rho} \\ \widehat{\boldsymbol{\beta}_d} \end{pmatrix}\right)$. Finally, hyperparameters $\boldsymbol{\lambda_d}$ are obtained by maximizing the concentrated log-likelihood : $(n_f - p - 1)\log\widehat{\sigma_d^2} + \log\left(\det(\boldsymbol{R_d})\right)$. The interested reader can refer to [12,13] for more detailed computations. This formulation is the one used in the present work through its implementation in the MuFiCokriging [14] R package. Note that constant values are considered here for the mean and scale factor.

As for kriging, given ρ and covariance functions K_c, K_d, the variance of the cokriging predictor $\hat{s}_f^2$ only depends on the location of the nested sample in the parameter space, and not on the actual value of the target output for this sample. It can thus be easily computed for any sample without the corresponding estimations of the target output.

2.3 Meta-model quality assessment

Different indices can be computed to evaluate the accuracy or quality of the meta-models, *i.e.* their ability to provide estimations close to the true simulated values. They refer to the computation of the error between some simulations and their corresponding estimations.

First, these simulations can be the training set $\boldsymbol{D}$ itself, referring to cross-validation. We consider more specifically the leave-one-out cross-validation technique (LOO-CV). It consists in computing additional meta-models using $n-1$ points of the training set, and in considering the prediction error on the left-out points to assess the meta-model predictivity. More precisely, denoting $\hat{y}_{-i}$ and $\hat{s}_{-i}^2$ the kriging estimator and variance computed from the training set $\boldsymbol{D}_{-i} = \left(\boldsymbol{x^j}\right)_{1\leq j\neq i\leq n}$, the cross-validation squared error at point $\boldsymbol{x^i}$ is equal to $(e_{LOO-CV}^i(\boldsymbol{x^i}))^2 = (y(\boldsymbol{x^i}) - \hat{y}_{-i}(\boldsymbol{x^i}))^2$. It can also be normalized by the kriging variance as

$$\eta(\boldsymbol{x^i}) = \frac{(e_{LOO-CV}^i(\boldsymbol{x^i}))^2}{\hat{s}_{-i}^2(\boldsymbol{x^i})}. \tag{14}$$

Then, the overall predictivity can be assessed through coefficient $Q2$ as:

$$Q2 = 1 - \frac{\sum_{j=1}^{n}(e_{LOO-CV}^i(\boldsymbol{x^i}))^2}{\sum_{j=1}^{n}(y(\boldsymbol{x^j}) - \bar{y})^2} \tag{15}$$

where $\bar{y}$ represents the mean of $\boldsymbol{y^D} = y(\boldsymbol{D})$. The $Q2$ coefficient gets closer to 1 when cross-validation errors decrease. In practice, some characteristics of $\hat{y}$ such as the covariance function can be used for meta-models $\hat{y}_{-i}$ to limit the additional computation cost. More details can be found for instance in [4] for kriging and in [12] for cokriging.

If the user can afford it, an additional sample of the parameter space, independent of the training set $\boldsymbol{D}$ and referred to as the test set $\boldsymbol{D_{test}} = \{\tilde{\boldsymbol{x}}^{\boldsymbol{1}}, \dots, \tilde{\boldsymbol{x}}^{\boldsymbol{n_{test}}}\}, \tilde{\boldsymbol{x}}^{\boldsymbol{i}} \in \mathbb{R}^d$, can also be considered to assess the meta-model quality. Once the output of interest y has been simulated for this sample, the overall predictivity of the meta-model $\hat{y}$ can be assessed through the $R2$ coefficient as:

$$R2 = 1 - \frac{\sum_{j=1}^{n_{test}}(y(\tilde{\boldsymbol{x}}^{\boldsymbol{j}}) - \hat{y}(\tilde{\boldsymbol{x}}^{\boldsymbol{j}}))^2}{\sum_{j=1}^{n_{test}}(y(\tilde{\boldsymbol{x}}^{\boldsymbol{j}}) - \bar{y})^2} \tag{16}$$

$\bar{y}$ represents the mean of $\boldsymbol{y^{test}} = y(\boldsymbol{D_{test}})$. The R2 coefficient gets closer to 1 when prediction errors decrease.

The $R2$ and $Q2$ indices can be computed in both simple and multi-fidelity frameworks, for the coarse and fine levels. For cross-validation in the multi-fidelity context, both observations $y_f(\boldsymbol{x^i})$ and $y_c(\boldsymbol{x^i})$ are considered unknown when building $\hat{y}_{f,-i}$.

2.4 Functional outputs

The meta-models introduced above apply to scalar properties. Let us now assume that the property to be estimated also depends on time. In reservoir engineering, this can refer for instance to the evolution of pressure or production rate through time at a given well. To build meta-models approximating these properties, a first possibility consists in considering the property at each time independently and to apply the meta-modeling approaches described above. Such functional outputs can also be approximated with less meta-models using a reduced-basis decomposition as proposed in [21,3] and extended to the multi-fidelity context in [29]. More precisely, let us denote $y_f(\boldsymbol{x}, \boldsymbol{t})$ the functional target output on the fine level, where $\boldsymbol{x} \in \mathbb{R}^d$ refers to the vector of input parameters and $\boldsymbol{t}$ refers to time. We assume known the values of the target output for a sample of the parameter space $\boldsymbol{D_f} = \left\{\boldsymbol{x_f^1}, \dots, \boldsymbol{x_f^{n_f}}\right\}, \boldsymbol{x_f^i} \in \mathbb{R}^d$ and a set of time values $\{\boldsymbol{t_1}, \dots, \boldsymbol{t_T}\}$. They are denoted by $\boldsymbol{y_f^D} \in \mathbb{R}^{T \times n_f}$ with $\left(\boldsymbol{y_f^D}\right)_{ij} = y_f(\boldsymbol{x_f^j}, \boldsymbol{t_i})$. Then the estimator of y_f at times $\boldsymbol{t_i}, i \in \{1 \dots T\}$ for a new point $\boldsymbol{x^*} \in \mathbb{R}^{\boldsymbol{d}}$ is given by:

$$\hat{y}_f(\boldsymbol{x^*}, \boldsymbol{t_i}) = \bar{y}_f(\boldsymbol{t_i}) + \sum_{l=1}^{L} \hat{\alpha}_k(\boldsymbol{x^*}) \phi_k(\boldsymbol{t_i}) \tag{17}$$

$\bar{y}_f(\boldsymbol{t_i})$ is the average value of $y_f(\boldsymbol{D_f}, \boldsymbol{t_i})$. The basis vectors ϕ_k are obtained by Proper Orthogonal Decomposition (POD) of the data set $\boldsymbol{y_f^D} \in \mathbb{R}^{T \times n_f}$ [18]. They are sorted by decreasing order of explained variance. To limit computation times, only the L first vectors of the basis that correspond to a sufficiently large percentage of explained variance (e.g. 95%) are retained. Meta-model $\hat{\alpha}_k$ approximates the projection coefficient of y_f on ϕ_k – the component α_k -, using the corresponding projection coefficients computed for the training set $\boldsymbol{y_f^D}$. In this approach, L meta-models are thus built independently from each other, one per component.

Let us now assume known $\boldsymbol{y_c^D} \in \mathbb{R}^{T \times n_c}$, the values of the coarse target output y_c at times $\{\boldsymbol{t_1}, \dots, \boldsymbol{t_T}\}$ for a sample $\boldsymbol{D_c}$ of the parameter space, $\boldsymbol{D_f} \subseteq \boldsymbol{D_c}$. Then we can estimate $y_f(., \boldsymbol{t_i}), i = 1 \dots T$, by computing multi-fidelity meta-models for each component $\alpha_k, k = 1 \dots L$, using the projection of data sets $\boldsymbol{y_f^D}$ and $\boldsymbol{y_c^D}$ on the basis vectors ϕ_k.

The approach described here also applies to outputs that vary with the location in the spatial domain [21]. In this case, $\boldsymbol{t}$ refers to spatial location and the discretization over $\boldsymbol{t}$ to grid blocks.

3. Sequential design strategy

Previous works [15,29] highlight the strong potential of the multi-fidelity approach in reservoir engineering. However, these studies do not provide clear guidelines to define an optimal training set and avoid simulation time wasting. Here, we propose to develop a sequential strategy to make the building

of cokriging based meta-models for the objective function fully automatic. To do so, we first introduce a new generic sequential design approach for the kriging and cokriging based estimation of scalar outputs. It is presented in section 3.1 for kriging-based meta-models, and extended to the multi-fidelity context in section 3.2. The way it is applied to estimate the objective function in the history-matching process is discussed later (section 4.1).

3.1 Single level of resolution

The sequential method proposed in this work for kriging based meta-models follows two steps: first, identify a region of the parameter space where the estimator accuracy is low, and then identify the point in this region that maximizes the kriging variance. This approach is derived from the one proposed in [11]. More precisely, we introduce as in [11] a partitioning of the parameter space defined by the Voronoi polygons around the design of experiments $\boldsymbol{D}$. However, it is only used here to narrow the search space. In addition, the next point to be simulated is identified among a large sample of the area (using here a Sobol' sequence) rather than with an optimization approach. The objective of these changes is twofold: give more weight to the cross-validation error; and reduce the computation times through a limited search space and the possibility to estimate the kriging variance at once for the whole sample using matrix computations.

The proposed algorithm is described below for the identification of one point at a time. Its extension to a batch version (several points added per iteration) is given in a second step. The Voronoi partitioning associated to the sample $\boldsymbol{D} = \{\boldsymbol{x^1}, \dots., \boldsymbol{x^n}\}, \boldsymbol{x^i} \in \mathbb{R}^{\boldsymbol{d}}$ is denoted by $(\boldsymbol{V_i})_{\boldsymbol{i=1\dots n}}$. The target scalar output is denoted by $\boldsymbol{y}$.

Algorithm 1

1) Generate an initial sample of the parameter space $\boldsymbol{D}$ and compute the corresponding values of the target output $\boldsymbol{y^D} = \boldsymbol{y}(\boldsymbol{D}) \in \mathbb{R}^{\boldsymbol{n}}$
2) Build a kriging-based estimator $\hat{\boldsymbol{y}}$ for the target output using the training set $\boldsymbol{y^D}$
3) While the stopping criterion is not fulfilled
    a. Identify the point $\boldsymbol{x^M}$ in $\boldsymbol{D}$ for which the normalized cross-validation squared error $\boldsymbol{\eta}$ defined in equation (13) is the largest:
    $$\boldsymbol{x^M} = \underset{\boldsymbol{x^i \epsilon D}}{\mathbf{argmax}}\, \boldsymbol{\eta}(\boldsymbol{x^i})$$
    b. Generate a large sample $\boldsymbol{S}$ of the Voronoi polygon $\boldsymbol{V_M}$ (using for instance a Sobol' sequence) and compute the kriging variance for the points of this sample
    c. Identify the point $\boldsymbol{x_{new}}$ in $\boldsymbol{S}$ with the largest kriging variance and add it to the design $\boldsymbol{D} \leftarrow \boldsymbol{D} \cup \boldsymbol{x_{new}}$
    d. Perform simulation for $\boldsymbol{x_{new}}$ and complement the training set $\boldsymbol{y^D}$: $\boldsymbol{y^D} \leftarrow \boldsymbol{y^D} \cup \boldsymbol{y}(\boldsymbol{x_{new}})$
    e. Build a kriging-based estimator $\hat{\boldsymbol{y}}$ for the target output $\boldsymbol{y}$ with the augmented training set $\boldsymbol{y^D}$

In this algorithm, the sampling of the Voronoi polygons is performed in two steps: first, a Sobol' sequence is generated in the parameter space; then the points of the sample that are within the Voronoi cell of interest are identified. This is repeated until a large enough number of points in the cell has been obtained (e.g. several thousands). The stopping criterion at step 3 can be defined as a target value for the predictor accuracy, defined on the cross-validation $Q2$ coefficient (15) or on the $R2$ coefficient

computed on a test sample (16). In addition, a maximum size for the design can be considered. It corresponds to the maximum number of simulations the user can afford.

In practice, the user may have the possibility to run several simulations simultaneously, for instance on a computer cluster. It appears interesting in this case to add several new points in the design at once and run the corresponding simulations simultaneously. To do so, we propose to select the q points to be added in the q Voronoi polygons corresponding to the largest q cross-validation errors computed on the design of experiments. More precisely, the first point is selected in the Voronoi polygon associated to the largest cross-validation error, and so forth until q points have been identified. Note that the kriging variance is updated each time a new point is selected. As mentioned previously, this can be done without running the corresponding simulation for unchanged hyper-parameters. The resulting version of Algorithm 1 for this batch configuration is given below.

Algorithm 1 – batch version

1) Generate an initial sample of the parameter space $\boldsymbol{D} = \{\boldsymbol{x^1}, \dots., \boldsymbol{x^n}\}, \boldsymbol{x^i} \in \mathbb{R}^{\boldsymbol{d}}$ and compute the corresponding values of the target output $\boldsymbol{y^D} = \boldsymbol{y}(\boldsymbol{D}) \in \mathbb{R}^{\boldsymbol{n}}$
2) Build a kriging-based estimator $\widehat{\boldsymbol{y}}$ for the target output using the training set $\boldsymbol{y^D}$. Let $\hat{\boldsymbol{s}}^{\boldsymbol{2}}$ denotes the predictor variance (equation (3))
3) While the stopping criterion is not fulfilled
    a. Let $\boldsymbol{D'} = \boldsymbol{D}$. For $\boldsymbol{j} = \boldsymbol{1} \dots \boldsymbol{q}$, do
        i. Identify the point $\boldsymbol{x^M}$ in $\boldsymbol{D'}$ for which the normalized cross-validation squared error is the largest:
        $$\boldsymbol{x^M} = \underset{\boldsymbol{x^i \epsilon D'}}{\mathbf{argmax}}\, \boldsymbol{\eta}(\boldsymbol{x^i})$$
        ii. Remove $\boldsymbol{x^M}$ from $\boldsymbol{D'}$: $\boldsymbol{D'} \leftarrow \boldsymbol{D'} \backslash \{\boldsymbol{x^M}\}$
        iii. Generate a large sample $\boldsymbol{S}$ of $\boldsymbol{V_M}$ and compute the kriging variance for the points of this sample applying equation (3) with design $\boldsymbol{D}$ and $\widehat{\boldsymbol{y}}$ hyper-parameters
        iv. Identify the point $\boldsymbol{x^j_{new}}$ in $\boldsymbol{S}$ with the largest kriging variance and add it to the design $\boldsymbol{D} \leftarrow \boldsymbol{D} \cup \boldsymbol{x^j_{new}}$
    b. Perform the simulations for $\boldsymbol{x^1_{new}}, \dots, \boldsymbol{x^q_{new}}$ and complement the training set $\boldsymbol{y^D}$: $\boldsymbol{y^D} \leftarrow \boldsymbol{y^D} \cup \{\boldsymbol{y}(\boldsymbol{x^1_{new}}), \dots, \boldsymbol{y}\,(\boldsymbol{x^q_{new}})\}$
    c. Build a kriging-based estimator $\widehat{\boldsymbol{y}}$ for the target output with the enlarged training set $\boldsymbol{y^D}$

3.2 Multi-fidelity context

To extend the sequential strategy presented above to the multi-fidelity context, two questions need to be addressed: which points should be added to the sample at each iteration, and on which level of resolution should these points be simulated. The results obtained in previous work [29] tend to show that, for a sufficient linear correlation between the coarse and fine values of the property, complementing a set of fine-level simulated values with coarse ones induces a rapid increase of the prediction accuracy for this property before a stagnation (plateau). Based on this, we propose here to add points on the coarse level only in a first step to retrieve as much low-cost information as possible before to complement the training set with points simulated on all levels. More precisely, we give priority to the accuracy of the coarse level estimator to drive the sequential approach: new points are identified first and foremost to get and then maintain a satisfying accuracy for this meta-model, using

the sequential approach proposed in section 3.1. The idea behind is that, if the two levels are correlated enough, a good accuracy of the coarse level kriging model should improve at low cost the multi-fidelity predictor accuracy on the fine level. In the second step of the process, the sequential approach described in section 3.1 is considered on the fine level for the identification of new points aiming at improving the cokriging based estimator. These points are systematically simulated here on the two levels of resolution. However, this may be improved in future work, considering for instance simulation time ratios and IMSE reduction as in [11] to choose the proper level of simulation. Finally, the selection of any number q of points per iteration follows the same principle as in the simple fidelity context, with one point selected per Voronoi cell of interest.

The resulting algorithm is described below. Index $\boldsymbol{c}$ refers to the training set on the coarse level and to the associated kriging based meta-model of the target output. Index $\boldsymbol{f}$ refers to the training set on the fine level and to the cokriging based meta-model of the target output. The Voronoi partitioning associated to samples $\boldsymbol{D_f}$ and $\boldsymbol{D_c}$ are denoted by $(\boldsymbol{V_i^f})_{\boldsymbol{i=1\dots n_f}}$ and $(\boldsymbol{V_i^c})_{\boldsymbol{i=1\dots n_c}}$ respectively.

Algorithm 2 – batch version

1) Generate an initial nested sample of the parameter space $\boldsymbol{D_f} = \left\{\boldsymbol{x_f^1}, \dots, \boldsymbol{x_f^{n_f}}\right\}$, $\boldsymbol{D_c} = \left\{\boldsymbol{x_c^1}, \dots., \boldsymbol{x_c^{n_c}}\right\}$, $\boldsymbol{D_f} \subseteq \boldsymbol{D_c}$, and compute the corresponding values of the target output $\boldsymbol{y_f^D} \in \mathbb{R}^{\boldsymbol{n_f}}$, $\boldsymbol{y_c^D} \in \mathbb{R}^{\boldsymbol{n_c}}$
2) Build a cokriging based estimator $\widehat{\boldsymbol{y}}_{\boldsymbol{f}}$ for the target output on the fine level using the nested training set $\boldsymbol{y_f^D}, \boldsymbol{y_c^D}$. It includes the kriging-based estimator of the target output on the coarse level $\widehat{\boldsymbol{y}}_{\boldsymbol{c}}$.
3) While the global stopping criterion is not fulfilled
    a. <u>Case 1:</u> The quality criterion on the coarse level is not fulfilled.
        i. Let $\boldsymbol{D_c'} = \boldsymbol{D_c}$. For $\boldsymbol{j} = \boldsymbol{1} \dots \boldsymbol{q}$, do
            1. Identify the point $\boldsymbol{x_c^M}$ in $\boldsymbol{D_c'}$ for which the normalized cross-validation squared error $\boldsymbol{\eta_c}$ defined in equation (14) is the largest:
$$\boldsymbol{x_c^M} = \underset{\boldsymbol{x_c^i \epsilon D_c'}}{\mathbf{argmax}}\, \boldsymbol{\eta_c}(\boldsymbol{x_c^i})$$
            2. Remove $\boldsymbol{x_c^M}$ from $\boldsymbol{D_c'}$: $\boldsymbol{D_c'} \leftarrow \boldsymbol{D_c'} \backslash \{\boldsymbol{x_c^M}\}$
            3. Generate a large sample $\boldsymbol{S}$ of $\boldsymbol{V_M^c}$ and compute the kriging variance for the points of this sample applying equation (11) with design $\boldsymbol{D_c}$ and $\widehat{\boldsymbol{y}}_{\boldsymbol{c}}$ hyper-parameters
            4. Identify the point $\boldsymbol{x_{new}^j}$ in $\boldsymbol{S}$ with the largest kriging variance and add it to the design $\boldsymbol{D_c} \leftarrow \boldsymbol{D_c} \cup \boldsymbol{x_{new}^j}$
        ii. Perform coarse level simulations for $\boldsymbol{x_{new}^1}$,…,$\boldsymbol{x_{new}^q}$ and complement the training set $\boldsymbol{y_c^D}$: $\boldsymbol{y_c^D} \leftarrow \boldsymbol{y_c^D} \cup \{\boldsymbol{y_c}(\boldsymbol{x_{new}^1}),\dots, \boldsymbol{y_c}(\boldsymbol{x_{new}^q})\}$
    b. <u>Case 2:</u> The quality criterion on the coarse level is fulfilled.
        i. Let $\boldsymbol{D_f'} = \boldsymbol{D_f}$. For $\boldsymbol{j} = \boldsymbol{1} \dots \boldsymbol{q}$, do
            1. Identify the point $\boldsymbol{x^M}$ in $\boldsymbol{D_f'}$ for which the normalized cross-validation squared error $\boldsymbol{\eta_f}$ defined in equation (14) is the largest:
$$\boldsymbol{x_f^M} = \underset{\boldsymbol{x_f^i \epsilon D_f'}}{\mathbf{argmax}}\, \boldsymbol{\eta_f}(\boldsymbol{x_f^i})$$
            2. Remove $\boldsymbol{x_f^M}$ from $\boldsymbol{D_f'}$: $\boldsymbol{D_f'} \leftarrow \boldsymbol{D_f'} \backslash \{\boldsymbol{x_f^M}\}$

3. Generate a large sample $\boldsymbol{S}$ of $\boldsymbol{V_M^f}$ and compute the cokriging variance for the points of this sample applying equation (13) with designs $\boldsymbol{D_f}$, $\boldsymbol{D_c}$ and $\widehat{\boldsymbol{y}}_{\boldsymbol{f}}$ hyper-parameters
4. Identify the point $\boldsymbol{x_{new}^j}$ in $\boldsymbol{S}$ with the largest cokriging variance and add it to the designs associated to both levels: $\boldsymbol{D_f} \leftarrow \boldsymbol{D_f} \cup \boldsymbol{x_{new}^j}$ and $\boldsymbol{D_c} \leftarrow \boldsymbol{D_c} \cup \boldsymbol{x_{new}^j}$

ii. Perform a coarse and a fine level simulation for $\boldsymbol{x_{new}^1}$,… $\boldsymbol{x_{new}^q}$ and complement the training sets $\boldsymbol{y_f^D}$ and $\boldsymbol{y_c^D}$: $\boldsymbol{y_f^D} \leftarrow \boldsymbol{y_f^D} \cup \{\boldsymbol{y_f}(\boldsymbol{x_{new}^1}),\ldots,\boldsymbol{y_f}(\boldsymbol{x_{new}^q})\}$ and $\boldsymbol{y_c^D} \leftarrow \boldsymbol{y_c^D} \cup \{\boldsymbol{y_c}(\boldsymbol{x_{new}^1}),\ldots,\boldsymbol{y_c}(\boldsymbol{x_{new}^q})\}$

c. Build a cokriging based estimator $\widehat{\boldsymbol{y}}_{\boldsymbol{f}}$ for the target output on the fine level with the augmented nested training set $\boldsymbol{y_f^D}$, $\boldsymbol{y_c^D}$

The sampling of the Voronoi polygons is performed as previously in two steps: first, a Sobol' sequence is generated in the parameter space; then the points of this sample that are within the Voronoi cell of interest are identified.

Note that a different number of points q could be identified in cases 1 and 2.

Global stopping criterion for the process

The global stopping criterion aims at identifying the optimal iteration at which ending the sequential process, in terms of both fine level prediction accuracy and total simulation time. Several aspects can be embedded in this criterion, such as a target value for the $R2$ coefficient computed on the cokriging estimator (equation (16)) or a maximum simulation budget.

Quality criterion on the coarse level

The quality criterion on the coarse level aims at identifying a good tradeoff between the coarse level meta-model accuracy and the simulation time. It should thus identify robustly either a satisfying accuracy of the predictions, or a potential stagnation of this accuracy despite the addition of new simulations. To achieve this, we introduce a moving average for the cross-validation index $Q2_c$ through iterations defined by:

$$\gamma(k) = \frac{1}{p}\sum_{i=k-p+1}^{k} Q2_c(i) - \frac{1}{p}\sum_{i=k-2p+1}^{k-p} Q2_c\,(i)$$

k represents the current iteration index, $Q2_c(i)$ the cross-validation coefficient at iteration i (equation (15)) and p the size of the moving window. For instance, p can be taken equal to half the number of coarse level simulations equivalent in time to a fine level simulation. The idea is to evaluate the gain in accuracy within a total simulation time equal to a fine level one. The quality criterion at iteration k is then defined by:

$$(Q2_c(k) \geq 0.95 \text{ and } \gamma(k) < 0.005) \text{ or } (Q2_c(k) < 0.95 \text{ and } 0 < \gamma(k) < 0.005)$$

In other words, we keep on adding points on the coarse level as long as the accuracy improves sufficiently ($\gamma(k) \geq 0.005$), or if it shows a degradation combined to an unsatisfying cross-validation

index ($\gamma(k) < 0$ and $Q2_c(k) < 0.95$). In addition, for robustness purposes, we require that this quality criterion is met a certain number of times more than its complementary to stop the process.

4. Application to reservoir engineering

Let us now assess the potential of the proposed sequential design strategies in the context of history-matching for the prediction of the error between production and simulated data (objective function). The formulation used to quantify this error and the conducted experiments are specified in section 4.1. The two synthetic cases considered for validation are described in section 4.2, and the results presented and analyzed in section 4.3.

4.1 Experiments

For the reservoir under consideration, we assume available an ensemble of production data at wells over a period of time. They can be for instance gas-oil ratio, pressure, or fluid rate measurements. The calibration or history-matching process then consists in identifying numerical models of the considered reservoir that reproduce these data. In practice, this is achieved by adjusting the value of some model input parameters within realistic uncertainty ranges. These parameters characterize for instance the petrophysical property distributions, fluids, wells … Optimization algorithms can also be considered to automatically drive the perturbation of these parameters in order to minimize the objective function (OF) that quantifies the error between the production data and the corresponding simulated values. This function is usually defined using a weighted least-square formulation as

$$OF(\boldsymbol{x}) = \frac{1}{2}\sum_{p=1}^{P}\omega_p \sum_{k=1}^{T(p)}\left(\frac{y_p(\boldsymbol{x},t_k) - y_p^{ref}(t_k)}{\sigma_p(t_k)}\right)^2 = \frac{1}{2}\sum_{p=1}^{P}\omega_p OF_p(\boldsymbol{x}) \qquad (18)$$

The production data to be reproduced are denoted by $y_p^{ref}(t)$ where t represents time. Index p refers to the P available data series, each series corresponding to the set of $T(p)$ available measurements through time of a given dynamic property at a given well (gas-oil ratio, water cut, bottomhole pressure…). $\sigma_p(t)$ represents the measurement uncertainty for data series p at time t. The vector of the d uncertain input parameters to be adjusted is denoted by $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{d}}$. For a given reservoir model characterized by $\boldsymbol{x}$, the value computed by the fluid flow simulator for data series p at time t is denoted $y_p(\boldsymbol{x},t)$. Finally, weights ω_p are constant through time for each data series. They aim at balancing the contribution of the data in the objective function.

Meta-modeling of the objective function

The objective of the study is to efficiently build predictors for the objective function (18) that are accurate for any value of the uncertain parameters within their uncertainty range. These predictors can then be considered instead of the simulator to quantify the impact of the uncertain input parameters on the objective function and identify the most influential ones. Discarding the other parameters in the calibration process can then simplify the problem and make it easier to solve.

We assume known the values simulated for each production data y_p on a given sample $\boldsymbol{D}$ of the parameter space and the set of acquisition times $\{t_1, \dots, t_{T(p)}\}$. We can derive the values $OF^{\boldsymbol{D}} = OF(\boldsymbol{D})$ of the objective function for this sample using equation (18). Then, the kriging based meta-modeling technique described in section 2.1 can be considered to build predictors $\widehat{OF}$ for the objective function based on the training set $OF^{\boldsymbol{D}}$. In this case, the target output y corresponds to the objective function and parameters $\boldsymbol{x}$ to the uncertain reservoir characteristics to be adjusted in the calibration process. This way of building estimations for the objective function based on sample $OF^{\boldsymbol{D}}$ will be referred to as the direct approach in what follows. Its extension to the multi-fidelity context is straightforward as soon as additional simulations of properties y_p are available on a coarser level of resolution for a sample $\boldsymbol{D_c} \supset \boldsymbol{D}$**.** However, results presented in [29] in a non-sequential setting show that cokriging based metamodels of the objective function computed following this direct approach can be disappointing: they do not necessarily provide accurate predictions within less simulation time than kriging based metamodels. As detailed in [29], these results are probably due to the fact that the values of the objective function on the two levels of resolution are poorly correlated for the considered case study. To overcome this limitation, a less straightforward meta-modeling strategy was thus introduced in [29]. It consists of two steps:

1. Using the approach detailed in section 2.4 for functional outputs (equation (17)), compute meta-models $\hat{y}_p(\boldsymbol{x}, .)$ that approximate at times $\{t_1, \dots, t_{T(p)}\}$ each time-dependent simulated production output $y_p(\boldsymbol{x}, .)$ considered in the objective function definition (18). These meta-models can be built using kriging or cokriging depending on the available information.
2. Use the resulting meta-models $\hat{y}_p(\boldsymbol{x}, .)$ to predict the objective function value as

$$\widehat{OF}(\boldsymbol{x}^*) = \frac{1}{2}\sum_{p=1}^{P} \omega_p \sum_{k=1}^{T(p)} \left(\frac{\hat{y}_p(\boldsymbol{x}^*, t_k) - y_p^{ref}(t_k)}{\sigma_p(t_k)} \right)^2 \quad (19)$$

$$\text{with } \hat{y}_p(\boldsymbol{x}^*, t_k) = \bar{y}_p(t_k) + \sum_{l=1}^{L(p)} \hat{\alpha}_{pl}(\boldsymbol{x}^*)\phi_{pl}(t_k) \text{ and } \boldsymbol{x}^* \in \mathbb{R}^d$$

This strategy led to better performances in [29] for both kriging and cokriging and will be referred to as "vectorial" hereafter. Note that the quality of the objective function estimations is assessed similarly in the direct and vectorial approaches using the error between $\widehat{OF}(\boldsymbol{x})$ and the corresponding true value.

Let us now discuss the application of the proposed sequential approaches for the meta-modeling of the objective function (18) in the direct and vectorial cases. Algorithms 1 and 2 proposed in section 3 can be directly applied to generate a sequential sampling aiming at improving the objective function predictors computed with the direct approach. However, the random variable representing this function in the vectorial approach (19) is not necessarily a Gaussian process. As a first approximation, we propose nevertheless to apply algorithms 1 and 2 to its variance to complement the training set. Several hypotheses are made to estimate this variance. First, we assume that the processes representing $\alpha_{pl}(\boldsymbol{x})$ are independent for $l \in \{1, \dots, L(p)\}$. The variance of the predictor for the objective function term $OF_p(\boldsymbol{x})$ related to data series y_p^{ref} is then equal to:

$$Var_p(\boldsymbol{x}) = 2\sum_{i=1}^{T(p)}\sum_{j=1}^{T(p)}\frac{\sum_{l=1\ldots L(p)}\phi_{pl}(t_i)\phi_{pl}(t_j)\hat{s}_{pl}^2(\boldsymbol{x})}{\sigma_p^2(t_i)\sigma_p^2(t_j)}\left(\sum_{l=1}^{L(p)}\phi_{pl}(t_i)\phi_{pl}(t_j)\hat{s}_{pl}^2(\boldsymbol{x}) + 2\mu_p(\boldsymbol{x},t_i)\mu_p(\boldsymbol{x},t_j)\right) \quad (20)$$

where $\mu_p(\boldsymbol{x},t) = \bar{y}_p(t) + \sum_{l=1}^{L(p)}\hat{\alpha}_{pl}(\boldsymbol{x})\phi_{pl}(t) - y_p^{ref}(t)$. Then, we make the additional assumption that the random variables representing $OF_p(\boldsymbol{x})$ are independent for all p. This may however not be the case in practice. It results in the following estimation for the variance associated to the random variable representing the objective function:

$$Var(\boldsymbol{x}) = \frac{1}{4}\sum_p \omega_p^2 Var_p(\boldsymbol{x}) \quad (21)$$

with $Var_p(\boldsymbol{x})$ defined by (20). These equations still hold with the cokriging based meta-model for the fine level.

Experiment settings

For the validation experiments, the proposed sequential strategies are initiated from a small sample of the parameter space generated using Latin Hypercube Sampling (LHS) [27]. This technique is chosen here as it makes it possible to control the size of the sample. In addition, it is designed to provide a good coverage of the parameter space for the chosen size of the sample, with points homogeneously distributed in the range of variation of each parameter. Fine and coarse level simulations are then performed for all the models of the initial sample and used as a starting point for the kriging and cokriging based sequential strategies. Here, sequential processes are performed considering one and three points added per iteration for both the vectorial and direct modeling of the objective function. These four experiments are repeated from other initial samples of the same size to obtain more robust results. Finally, the sequential processes are stopped after a given number of simulations as the study aims to compare the efficiency of the different approaches.

The sequential experiments are also complemented with non-sequential ones that serve as a basis for comparison. More precisely, samples of various sizes are generated using the LHS approach and used to build kriging based meta-models of the objective function. In addition, the fine level LHS samples $\boldsymbol{D_f}$ used to initiate the multi-fidelity sequential processes are combined to larger independent LHS $\boldsymbol{D_c}$ to obtain nested samples. To do so, we proceed as proposed in [13]: for each $\boldsymbol{x_f^i} \in \boldsymbol{D_f}$, the point of $\boldsymbol{D_c}$ the closest to $\boldsymbol{x_f^i}$ (in the sense of the Euclidian distance) is removed from $\boldsymbol{D_c}$ and replaced by $\boldsymbol{x_f^i}$. The resulting nested samples are used to build cokriging based meta-models. Again, several nested designs of the same size are generated.

4.2 Test cases

The experiments described above are conducted on two synthetic cases derived from the Punq-S3 [6] and Brugge [23] benchmarks.

In both cases, the lower level of resolution corresponds to a coarser discretization of the reservoir grid (less grid blocks). Geostatistical methods are used to populate the fine level geological model with porosity and horizontal permeability. The FFT-MA method [16] with conditioning at wells is applied for porosity. Horizontal permeability follows a lognormal distribution. Its values are assigned to the reservoir grid by cosimulation with porosity using the SGS method [9]. The values associated to the coarse model are then derived from the resulting fine-level distributions using upscaling techniques: arithmetic averaging for porosity and a method combining arithmetic and harmonic averaging for permeability [2]. The vertical permeability is taken equal to 10% of the horizontal one.

The Punq case reservoir consists of a simple dome structure and is produced by depletion from six wells with an aquifer support. The model deals with 3-phase flow simulations and is characterized by outputs at wells with irregular behaviors, due in particular to gas production and threshold constraints on pressure at wells. The Brugge case study includes only two phases but presents additional complexity compared to the Punq case and aims at consolidating the results. The reservoir includes an internal fault and is discretized into more grid blocks, leading to larger simulation times (40mn on average for Brugge and only 3mn for Punq on the finest levels of resolution). It is produced by water injection and contains 30 wells. This induces more data series to consider in the calibration process, and thus more meta-models to compute for the vectorial based estimation of the objective function. The number of parameters is also larger, and some of them affect the petrophysical property distributions. Finally, the production data are the ones provided during the benchmark and were not generated with one model of the chosen uncertainty space contrary to the Punq based case. More details are given below.

Punq case

The first test case is the one already used in [29] to assess the potential of the multi-fidelity approach. It is derived from the Punq-S3 case, defined for benchmark purposes within the framework of the PUNQ European project [6].

The Punq case represents a dome-shaped reservoir with an horizontal extension of about 3.4 km X 5 km, bounded by a sealing fault to the east and south and by an aquifer otherwise (Figure 1). Its upper part is initially filled with gas. It is produced by depletion from 6 producers located around the gas/oil contact (Figure 1). The production scheme is defined over 16 years with a first year of well testing followed by 3 years of shut-in before production starts for a period of 12 years. During this 12-year production period, wells are closed two weeks every year for well testing. Otherwise, production is driven by oil flow rate targets at wells, with a minimum accepted bottomhole pressure of 120 bars.

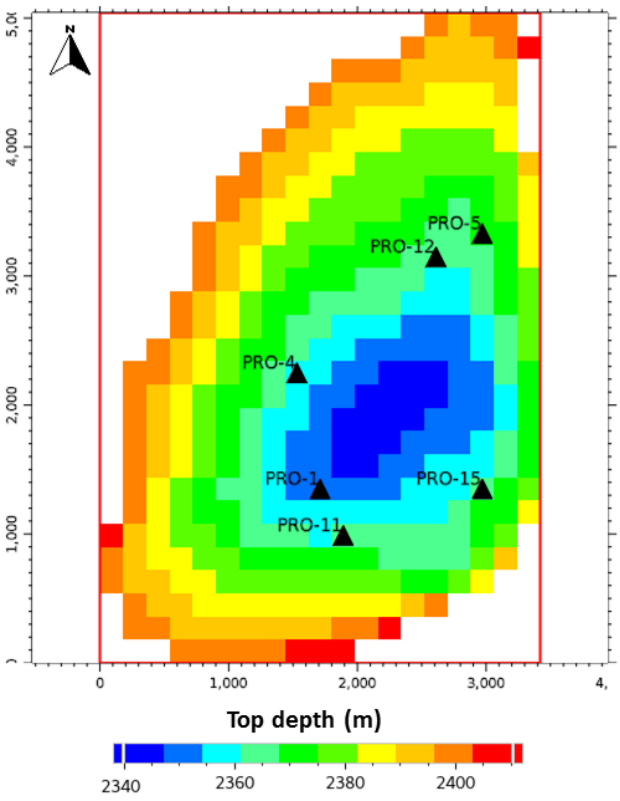

Figure 1 – Punq case: top reservoir depth and well location

The structure of the reservoir is discretized into 5 layers of about 5m thick. Two horizontal discretizations of these layers are then considered, which characterize the fine and coarse levels of resolution for the multi-fidelity context:

- The coarse model is defined on the same grid as the Punq-S3 model. It is composed of 19 X 28 X 5 grid blocks, among which 1761 are active (see Figure 1)
- The discretization grid used for the fine model is obtained by splitting horizontally each grid block of the coarse model into 3 X 3 grid blocks of equal size. This results in 57 X 84 grid blocks in each of the five layers. 15849 in total are active.

Geostatistical simulation methods are used in each layer independently to populate the fine model with petrophysical properties as detailed above. A 3-phase flow simulation is then performed on the resulting fine geological model, considered in what follows as the "true" one. This provides production data at the 6 wells used for calibration, namely gas-oil ratio, water cut and bottomhole pressure.

We now assume that this reference model is known, except for the 7 characteristics given in *Table 1*, related to horizontal permeability multipliers (constant per layer), aquifer properties and Kr-Pc curves. The calibration problem then consists in perturbing these 7 parameters in order to identify models reproducing the reference data. As a result, our objective here is to build meta-models that approximate the objective function (18) as a function of $\boldsymbol{x} \in \mathbb{R}^7$ within the uncertainty ranges defined in *Table 1*. This objective function is computed for the P=18 production data series (gas-oil ratio, water cut and

bottomhole pressure at the 6 producers) with values available at T=241 times. The uncertainty on the measurements is taken equal to $\sigma_p(t) = \max(0.1 \times y_p^{ref}(t), 0.005)$. The weights ω_p are taken equal at all wells for a given dynamic property and chosen to balance the average contribution of each property. They are thus different for water cut, gas oil ratio and bottomhole pressure. For the meta-modeling of the data series in the vectorial approach, the number L of basis vectors retained in the POD is fixed to 3 for all properties and training sets. Finally, as the perturbation of the considered 7 parameters does not imply an update of the petrophysical property distributions, the difference in simulation time between the two levels is only related to the fluid flow computations. The equivalent total simulation time for a nested design of size (n_f, n_c) is here taken equal to $n_{eq} = n_f + 0.054 n_c$. This formula uses an average simulation time ratio between the levels estimated from a set of models.

Name	Reference value	Min.	Max.	Unit	Description
SWIR	0.2	0.2	0.3	-	Critical water saturation
SORW	0.2	0.15	0.25	-	Critical oil-water saturation
SORG	0.1	0.1	0.2	-	Critical oil-gas saturation
MPH3	1	0.8	1.2	-	Horizontal permeability multiplier for layer 3
MPH4	1	0.8	1.2	-	Horizontal permeability multiplier for layer 4
MPH5	1	0.8	1.2	-	Horizontal permeability multiplier for layer 5
AQ1K	137.5	100	200	mD	Permeability of aquifer 1

Table 1 - Uncertain parameters for the Punq case

Brugge case

The second test case considered in this study is derived from the Brugge synthetic case, defined by TNO for benchmark purposes [23,24]. Based on North Sea Brent-type reservoir characteristics, this field has an elongated half dome structure with an internal fault as shown in Figure 2. Its horizontal dimensions are about 10 X 3 km^2. The reservoir spans vertically over 4 geological formations, namely Schelde, Maas, Waal and Schie. High-permeable layers thus alternate with low-permeable ones.

The field is operated during 10 years from 20 producing wells located in the upper part of the reservoir, denoted by P-j, j=1...20, and from 10 water injection wells surrounding the producers and located near the oil/water contact. Injectors are denoted by I-j, j=1…10. The field is initially produced by depletion: an additional producer is opened every month, with a target production rate of 2000 bbl/day and a minimum bottomhole pressure of 725 psi. Water injection starts once the producers are all opened, after 20 months of production: an additional injector is then opened every month, with a target injection rate of 4000 bbl/day and a minimum accepted bottomhole pressure of 2611 psi.

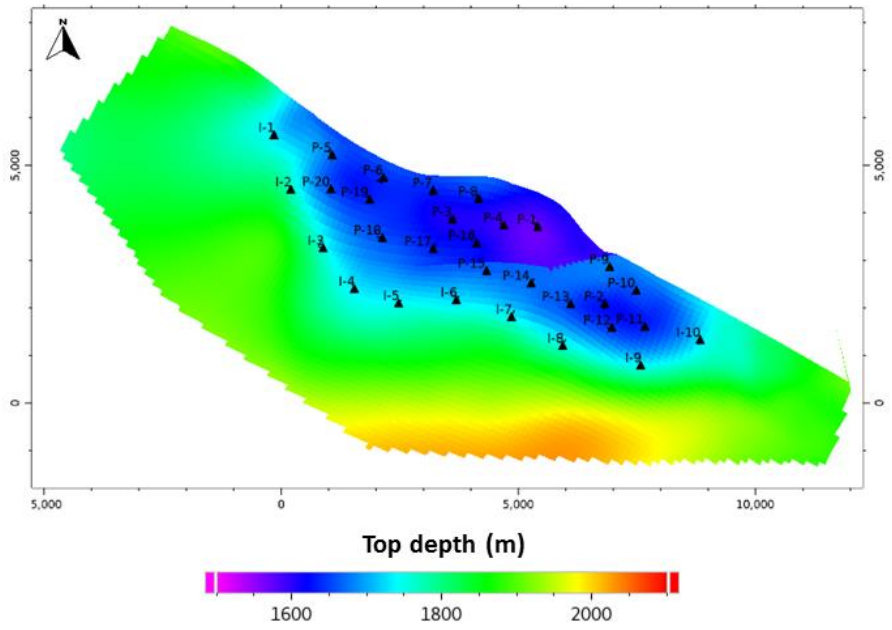

Figure 2 – Brugge case: top reservoir depth and well location

The structure of the reservoir is discretized vertically into 9 layers. Two horizontal discretizations of these layers are considered to characterize the fine and coarse levels of resolution for the multi-fidelity context:

- The coarse model is defined on the same grid as the initial Brugge model. It is composed of 139 X 48 X 9 grid blocks, among which 44550 are active (see Figure 2)
- The discretization grid used for the fine model is obtained by splitting each coarse grid block into 2 X 2 grid blocks of equal size. This results in 278 X 96 grid blocks in each of the nine layers. 534600 in total are active.

As described above, geostatistical simulation methods are used to populate the fine model with petrophysical properties, each formation being considered independently from the others. The statistical properties are estimated from one of the benchmark realizations, and the realizations are conditioned at wells based on the benchmark well log data. For each level, rock types are attributed to grid blocks following thresholds on porosity values.

The reference production data considered for history-matching are the ones provided by TNO to the benchmark participants, considering 8 years of production. They consist of 70 data series: oil and water flow rates at the 20 producers and bottomhole pressure at all wells. The calibration problem considered here then consists in adjusting the 19 input parameters described in Table 2. They are related to the definition of reservoir properties (capillary pressure, relative permeability) and petrophysical properties (permeability multipliers, mean porosity). Perturbing these last characteristics makes it necessary to re-compute the petrophysical property distribution on the fine level whatever level of resolution considered for the flow simulation. The duration of a coarse level simulation is thus larger if the

equivalent fine level model has not already been simulated: it takes here on average 0.068 times a fine level simulation in this case, and 0.03 times a fine level simulation otherwise.

Table 2 – Uncertain parameters for the Brugge case

Name	Min.	Max.	Unit	Description
CRO	4.5	5.5	-	Corey exponent for oil
CRW	2.5	3.5	-	Corey exponent for water
KROW	0.3	0.5	-	Oil relative permeability endpoint
KRWM	0.5	0.7	-	Water relative permeability endpoint
SORW	0.1	0.2	-	Oil residual saturation
SWI	-0.05	+0.05	-	Increment on the irreducible water saturation
PCWM	1.6	2	psi	Maximum capillarity pressure
POR1	0.16	0.18	-	Mean porosity in Schelde on the fine level
POR2	0.14	0.16	-	Mean porosity in Maas on the fine level
POR3	0.22	0.24	-	Mean porosity in Waal on the fine level
POR4	0.13	0.15	-	Mean porosity in Schie on the fine level
MKHi*	0.8	1.2	-	Horizontal permeability multiplier per reservoir zone
MKVi*	0.5	2	-	Vertical permeability multiplier per reservoir zone

MKHi/MKVi: i from 1 to 4 refers to Schelde, Maas, Waal and Schie.

The proposed sequential strategies are considered to predict the objective function defined using (18) for the P=70 available data series given at T=102 times, considering the parameter space defined in Table 2. The uncertainty on the measurements $(\sigma_p(t))$ is taken equal to 10% of the data, with a lower limit to 50. The series weights are taken equal at all wells for a given dynamic property and chosen to balance the average contribution of bottomhole pressure with that of all flow rates. They are thus different for pressure and flow rates. For the meta-modeling of the data series in the vectorial approach, fixed numbers of basis vectors are considered for each series and chosen to obtain a good tradeoff between reasonable computation times and sufficient explained variance. They are given in Table 3.

	P-1	P-6	P-9	P-11	P-13	Other wells
Bottomhole pressure	2	2	5	3	3	2
Water flow rate	5	5	5	2	2	2
Oil flow rate	5	5	5	2	2	2

Table 3 – Brugge case: size of the reduced basis per well and data series. P-i denotes producer i.

4.3 Results

The sequential processes described in section 4.1 were performed from 10 initial samples of size 8 for the Punq case and from 5 initial samples of size 20 for the Brugge case. Similarly, for each non sequential configuration, 10 different samples were generated for the Punq case, and 5 for the Brugge case.

The results for the Punq case are summarized in Figure 3 for the direct estimation of the objective function and in Figure 4 for the vectorial approach. The ones obtained for the Brugge case are presented in Figure 5 and Figure 6. All these graphs show the evolution of the $R2$ coefficient on the OF estimations

averaged over the set of similar experiments as a function of the total simulation time expressed in terms of equivalent number of fine level simulations. The $R2$ coefficient is computed following equation (16) on an independent sample of size 200 for the Punq case and 101 for the Brugge case.

In all these figures, the black curves correspond to kriging-based meta-modeling and the red curves to cokriging based approaches. The results obtained with (nested) LHS of increasing size are given in dashed lines. The other curves correspond to the sequential processes, complementing the initial designs with one (plain lines) and three (lines with circles) points per iteration. The results obtained in the batch case (three points per iteration) are presented in two different ways: for the curves with filled circles, we assume that, within each iteration, the three simulations on a given level are simulated simultaneously, while for the curves with empty circles, we assume that these simulations are run sequentially. These second curves aim at highlighting the impact of updating or not hyper-parameters between the identification of two successive points. The simulations of the (nested) initial LHS models are assumed to be performed sequentially. In the multi-fidelity experiments, the iterations during which fine level simulations are performed depend on the initial sample, so that interpolated curves are considered to estimate the averaged $R2$.

The results obtained for the Punq case (Figure 3 and Figure 4) highlight the potential of the proposed sequential approach for kriging-based meta-models. Indeed, it provides an accuracy equivalent to the one obtained with LHS samples of the same size for the direct approach, and a better accuracy for the vectorial approach. In addition, no degradation can be observed in the batch case, even if the meta-model hyper-parameters are not updated within each iteration (only the kriging variance is). The proposed sequential approach thus provides very satisfying results in terms of simulation time and accuracy if the simulations can be performed simultaneously within each iteration.

As already pointed out in [29], the results obtained with the direct approach in the multi-fidelity context are quite disappointing: the technique clearly fails to provide accurate predictions within less simulation time than kriging. This is probably due to the fact that, as highlighted in Figure 7(a), the values of the objective function on the two levels of resolution are poorly correlated. This is due in particular to a shift between some coarse and fine level simulated production series. The interested reader can refer to [29] for a more thorough analysis. The sequential multi-level approach leads to results similar to the kriging-based sequential one. The results obtained in the vectorial case are however much more encouraging. Indeed, in the case of one point added per iteration, the $R2$ increase is faster in the multi-fidelity case than in the simple one and reaches the same plateau. In addition, only a limited degradation can be observed in the batch case if running the simulations sequentially. However, running these simulations simultaneously at each iteration strongly improves the results in both the simple and multi-fidelity processes and makes them become equivalent: the $R2$ increase is almost vertical with values getting very close to one and there is not much room for improvement at this point. The advantage of the multi-fidelity workflow compared to the simple fidelity one thus strongly depends on the computation capacity, in particular on the possibility to run several simulations simultaneously. In other words, the multi-fidelity approach appears more appropriate with lower computation capacity.

The results obtained for the Brugge case lead to similar conclusions (Figure 5 and Figure 6). For this case study, the vectorial approach outperforms the direct one for both kriging and cokriging. However, the gain appears less important than for the Punq case in the multi-fidelity context as the direct approach already performs quite well. This is probably due to the almost linear correlation that can be observed between the objective function values computed on the fine and coarse levels, as shown in Figure 7(b). Again, the kriging-based sequential approach provides better results than fixed LHS sample. Identifying a set of new points at each iteration does not seem to degrade the results. As a consequence, the batch sequential algorithm for kriging-based meta-modeling provides very satisfying results and makes it possible to define on the fly a good training set. The sequential multi-level strategy outperforms fixed nested LHS in terms of prediction accuracy in the vectorial case and provides equivalent results in the direct case. No degradation can be observed in the batch case. The sequential approach with 3 points added per iteration then provides the best results in terms of simulation time and accuracy. They do not leave much room for improvement, in particular in the vectorial case. However, the simple fidelity sequential approach may in the end reach the same performance if increasing the number of points added simultaneously at each iteration.

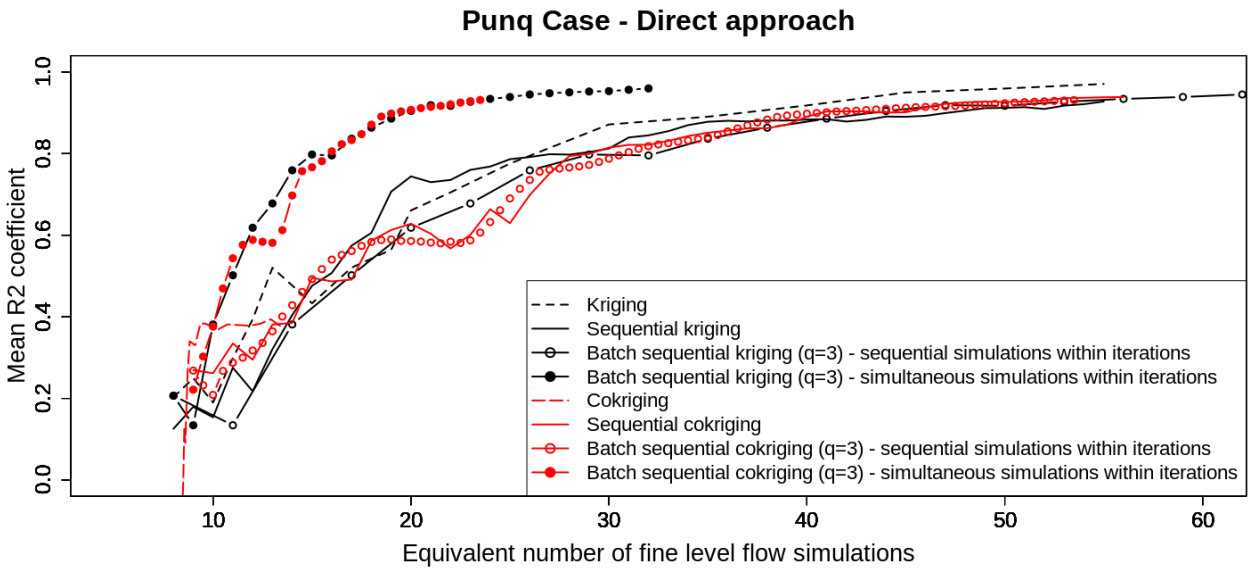

Figure 3 – Punq case : average R2 coefficient obtained with kriging-based (black) and cokriging-based (red) direct estimations of the objective function using fixed LHS (dotted line) and sequential design considering the identification of 1 point (plain line) or 3 points (lines with circles) per iteration. For the lines with filled (resp. empty) circles, we assume that, at each iteration, the 3 simulations on a given level are performed simultaneously (resp. sequentially).

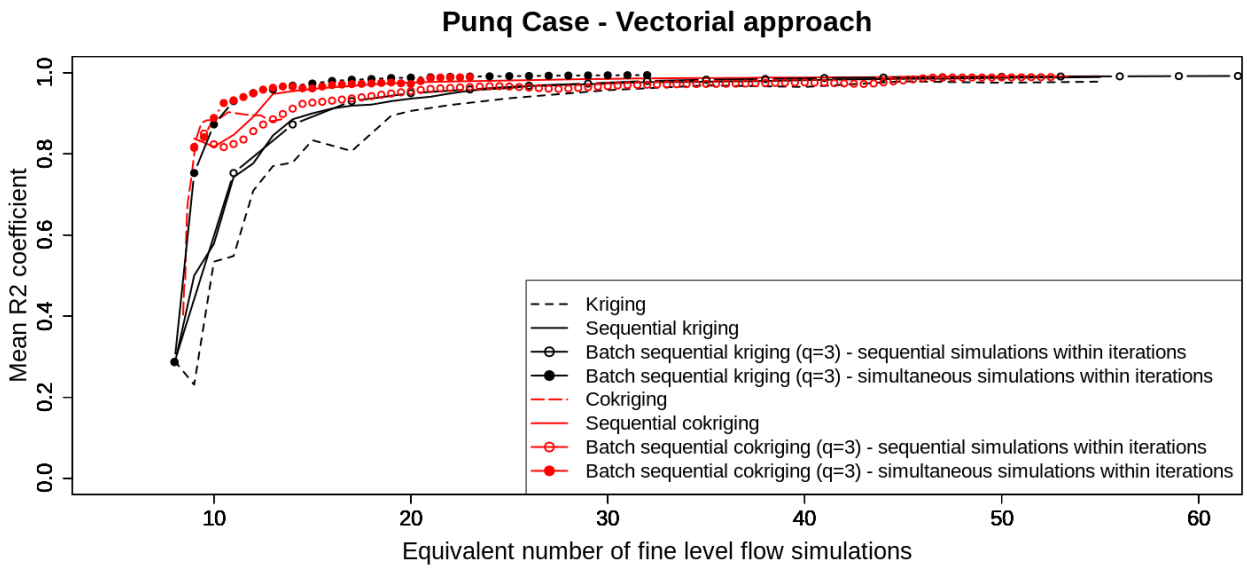

Figure 4 – Punq case : average R2 coefficient with kriging-based (black) and cokriging-based (red) vectorial estimations of the objective function using fixed LHS (dotted lines) and sequential design considering the identification of 1 point (plain lines) or 3 points (lines with circles) per iteration. For the lines with filled (resp. empty) circles, we assume that, at each iteration, the 3 simulations on a given level are performed simultaneously (resp. sequentially).

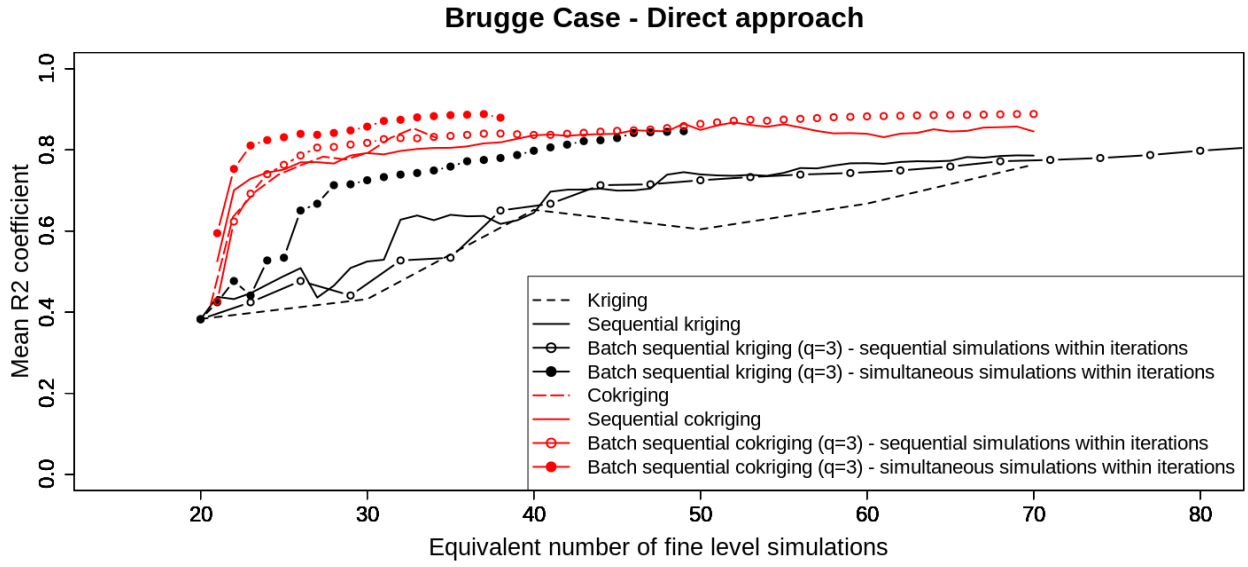

Figure 5 - Brugge case : average R2 coefficient with kriging-based (black) and cokriging-based (red) vectorial estimations of the objective function using fixed LHS (dotted lines) and sequential design considering the identification of 1 point (plain lines) or 3 points (lines with circles) per iteration. For the lines with filled (resp. empty) circles, we assume that, at each iteration, the 3 simulations on a given level are performed simultaneously (resp. sequentially).

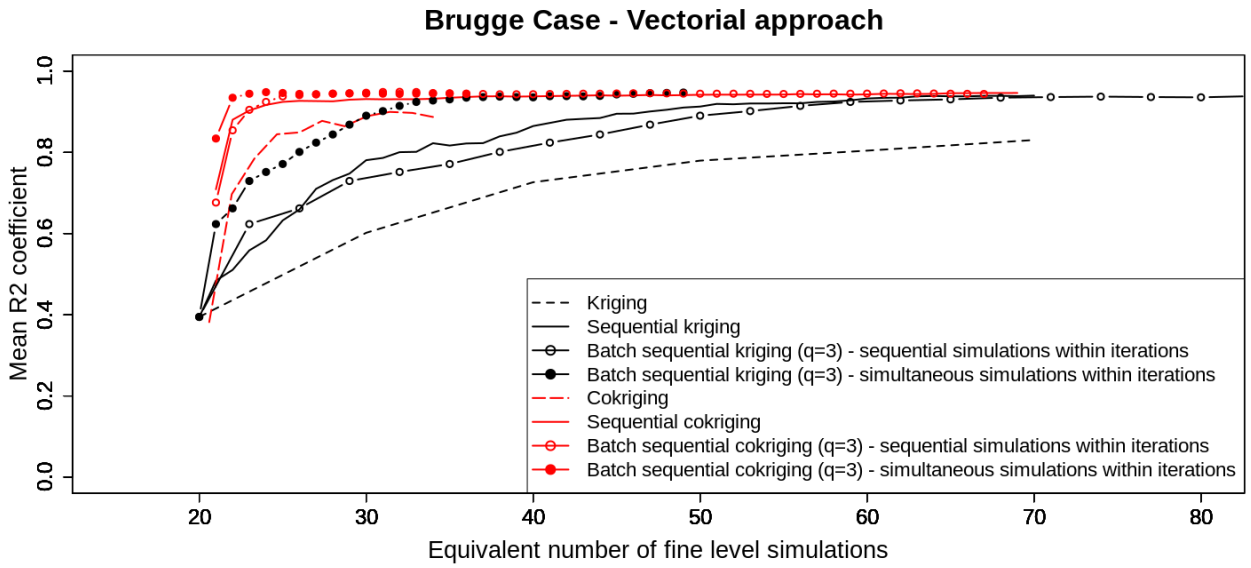

Figure 6 - Brugge case: average R2 coefficient with kriging-based (black) and cokriging-based (red) vectorial estimations of the objective function using fixed LHS (dotted lines) and sequential design considering the identification of 1 point (plain lines) or 3 points (lines with circles) per iteration. For the lines with filled (resp. empty) circles, we assume that, at each iteration, the 3 simulations on a given level are performed simultaneously (resp. sequentially).

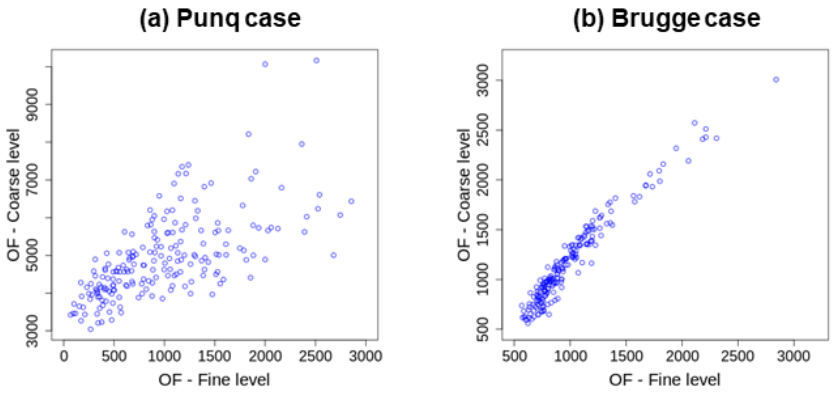

Figure 7 - Coarse versus fine objective function values obtained for the test sample for the Punq (a) and Brugge (b) test cases

5. Conclusions and perspectives

In this paper, we developed an automated sequential workflow dedicated to the building of kriging and cokriging based response surfaces for scalar outputs. The proposed approach provides very satisfying results for the estimation of the objective function in the context of history-matching. It makes it possible to identify on the fly a training set leading to accurate estimations of the objective function at least as efficiently in terms of simulation time as fixed LHS. In addition, considering coarse level simulations in the training set can improve the results obtained on a single level for an equivalent

computational power (*i.e.* for the same number of simulations performed simultaneously). However, for a sufficiently important computation power, the gain in using the multi-fidelity approach is limited.

These conclusions are based on the comparison of the prediction accuracy as a function of the direct simulation times. In practice, machine learning related computations (POD, building of the response surfaces, identification of the new points to be simulated) may also take a non-negligible time compared to direct simulations. This time increases with the number of meta-models to be built, the number of points in the training set. The sampling of the Voronoi cells may also be expensive. However, the proposed workflow is intended to reservoir models with long simulation times. In addition, as can be seen in Figure 4 and Figure 6, the multi-fidelity approach makes it possible to reach very good predictions in only a few iterations, thus limiting both simulation and computation times.

Different leads could nevertheless be considered to decrease the computation time required at each iteration of the sampling strategy. For instance, we could envision to not update the POD or the response surfaces at each new iteration, for example for the data series with low contributions in the objective function or if the main basis vectors do not change a lot. We may also consider parallelizing the computations or building the meta-models simultaneously on several processors. In addition, it could be interesting to reduce the computation time required to sample Voronoi cells.

Finally, the results obtained on the synthetic case studies considered here should be reinforced with application on more complex cases. In future work, it would also be interesting to refine the multi-scale sequential approach and not systematically perform the additional simulations on the two levels of resolution. This should be of interest for cases with lower correlations between the levels.

Acknowledgments
The authors thank TNO for providing the Brugge data set.

Data availability statement
The original PUNQ-S3 data set used to build the first test case considered in this paper is available online: http://www.imperial.ac.uk/earth-science/research/research-groups/perm/standard-models/eclipse-dataset. The original Brugge data set used to build the second test case was provided by TNO. It can be requested to TNO via www.isapp2.com.

Declarations
Conflict of interest: Authors declare they have no conflict of interest

6. References

1. Busby, D.: Hierarchical adaptive experimental design for Gaussian process emulators. Reliab. Eng. Syst. Safe. 94(7), 1183–1193 (2009). http://doi.org/10.1016/j.ress.2008.07.007

2. Cardwell Jr., W.T., Parsons, R.L.: Average permeabilities of heterogeneous oil sands. Society of Petroleum Engineers (1945). http://doi.org/10.2118/945034-G
3. Douarche, F., Da Veiga, S., Feraille, M., Enchéry, G., Touzani, S., & Barsalou, R.: Sensitivity Analysis and Optimization of Surfactant-Polymer Flooding under Uncertainties. Oil & Gas Science and Technology – Revue d'IFP Energies Nouvelles (2014) http://doi.org/10.2516/ogst/2013166.
4. Dubrule, O.: Cross validation of kriging in a unique neighborhood. Journal of the International Association for Mathematical Geology, 15(6), 687–699 (1983) https://doi.org/10.1007/BF01033232
5. Feraille, M., & Marrel, A.: Prediction under Uncertainty on a Mature Field. Oil Gas Sci. Technol. – Rev. IFP Energies Nouvelles, 67(2), 193–206 (2012) https://doi.org/10.2516/ogst/2011172
6. Floris, F.J.T., Bush, M.D., Cuypers, M., Roggero, F., Syversveen, A.-R.: Methods for quantifying the uncertainty of production forecasts: a comparative study. Petrol. Geosci., 7, S87–S96 (2001). doi:10.1144/petgeo.7.S.S87
7. Forrester, A.I.J., Keane, A.J.: Recent advances in surrogate-based optimization. Prog. Aerosp. Sci. 45, 50-79 (2009). doi:10.1016/j.paerosci.2008.11.001
8. Jones, D., Schonlau, M., & Welch, W.: Efficient Global Optimization of Expensive Black-Box Functions. J. Global Optim. **13**(4), 455–492 (1998). doi:10.1023/A:1008306431147
9. Journel, A.G., Deutsch, C.V.: GSLIB Geostatistical Software Library and Users Guide, 2nd edn. Oxford University Press, New York (1998)
10. Kennedy, M., O'Hagan, A.: Predicting the output from a complex computer code when fast approximations are available. Biometrika 87, 1–13 (2000). doi:10.1093/biomet/87.1.1
11. Le Gratiet, L., & Cannamela, C.: Cokriging-Based Sequential Design Strategies Using Fast Cross-Validation Techniques for Multi-Fidelity Computer Codes. Technometrics 57(3), 418–427 (2015). doi:10.1080/00401706
12. Le Gratiet, L., & Garnier, J. : Recursive Cokriging Model For Design Of Computer Experiments With Multiple Levels Of Fidelity. International Journal for Uncertainty Quantification, 4(5), 365–386 (2014) **DOI:** 10.1615/Int.J.UncertaintyQuantification.2014006914
13. Le Gratiet, L.: Bayesian Analysis of Hierarchical Multifidelity Codes. SIAM/ASA Journal on Uncertainty Quantification, 1(1), 244–269 (2013). http://doi.org/10.1137/120884122
14. Le Gratiet, L.: MuFiCokriging: multi-fidelity cokriging models. R package version 1.2 (2014)
15. Le Ravalec, M.: Optimizing well placement with quality maps derived from multi-fidelity meta-models. Paper SPE 154416 presented at the EAGE Annual Conference and Exhibition incorporating SPE Europec, Copenhagen, Denmark (2012) https://doi.org/10.2118/154416-MS
16. Le Ravalec, M., Noetinger, B., Hu, L.: The FFT moving average (FFT-MA) generator: an efficient numerical method for generating and conditioning gaussian simulations. Math. Geol. **32**, 701–723 (2000). doi:10.1023/A:1007542406333
17. Le Ravalec, M., Da Veiga, S., Derfoul, R., Enchéry, G., Gervais, V. & Roggero, F. : Integrating data of different types and different supports into reservoir models, Oil and Gas Science and Technology, 67(5), 823-840 (2012), DOI: 10.2516/ogst/2012024.

18. Loeve, M.: Probability Theory, vol. I–II. Springer, New York (1978)
19. O'Hagan A.: A Markov Property for Covariance structures. University of Nottingham (1998).
20. Mardia, K. V., & Marshall, R. J.: Maximum likelihood estimation of models for residual covariance in spatial regression. Biometrika 71(1), 135-146 (1984).
21. Marrel, A., Perot, N., & Mottet, C.: Development of a surrogate model and sensitivity analysis for spatio-temporal numerical simulators. Stochastic Environmental Research and Risk Assessment, 29(3), 959–974 (2015). http://doi.org/10.1007/s00477-014-0927-y
22. Matérn, B.: Spatial Variation. Springer, New York (1986)
23. Peters, L., Arts, R., Brouwer, G., Geel, C., Cullick, S., Lorentzen, R.J., Chen, Y., Dunlop, N., Vossepoel, F.C., Xu, R., et al.: Results of the brugge benchmark study for flooding optimization and history matching. SPE Reserv. Eval. Eng. **13**(03), 391–405 (2010) https://doi.org/10.2118/119094-PA
24. Peters, E., Chen, Y., Leeuwenburgh, O., and Oliver, D.: Extended Brugge benchmark case for history matching and water flooding optimization. Computational geosciences, 50, 16–24 (2013). https://doi.org/10.1016/j.cageo.2012.07.018
25. Picheny, V., Ginsbourger, D., Roustant, O., Haftka, R., Kim N-H: Adaptive designs of experiments for accurate approximation of a target region. *Journal of Mechanical Design*, 132(7), 071008-1 – 071008-9 (2010) https://doi.org/10.1115/1.4001873
26. Roustant, O., Ginsbourger, D., Deville, Y.: Dicekriging, DiceOptim: two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. J. Stat. Softw. **51**(1), 1–55 (2012). doi:10.18637/jss.v051.i01
27. Sacks, J., Welch, W. J., Mitchell, T. J., & Wynn, H. P.: Design and Analysis of Computer Experiments. Statist. Sci., (4), 409–423 (1989). http://doi.org/10.1214/ss/1177012413
28. Santner, T. J., Williams, B. J., Notz, W.: The design and analysis of computer experiments. Springer-Verlab, New York (2003)
29. Thenon, A., Gervais, V., & Ravalec, M. L.: Multi-fidelity meta-modeling for reservoir engineering - application to history matching. Computational Geosciences, 20, 1231–1250 (2016). http://doi.org/10.1007/s10596-016-9587-y
30. Xiong, S., Qian, P. Z. G., & Wu, C. F. J.: Sequential Design and Analysis of High-Accuracy and Low-Accuracy Computer Codes. Technometrics, 55(1), 37–46 (2012). http://doi.org/10.1080/00401706.2012.723572
31. Yeten, B., Castellini, A., Guyaguler, B., & Chen, W. H.: A Comparison Study on Experimental Design and Response Surface Methodologies. In SPE-93347-MS. SPE: Society of Petroleum Engineers (2005). https://doi.org/10.2118/93347-MS
32. Zubarev, D. I.: Pros and Cons of Applying Proxy-models as a Substitute for Full Reservoir Simulations. In SPE-124815-MS. SPE: Society of Petroleum Engineers (2009). https://doi.org/10.2118/124815-MS