Shape invariant model approach for functional data analysis in uncertainty and sensitivity studies

Ekaterina Sergienko, Fabrice Gamboa, Daniel Busby

Shape invariant model approach for functional data analysis in uncertainty and sensitivity studies

Ekaterina Sergienko[a,b], Fabrice Gamboa[a], Daniel Busby[b]

[a] *Université Paul Sabatier, IMT-EPS, 118, Route de Narbonne, 31062, Toulouse, France*
[b] *IFP Energies Nouvelles, 1-4 avenue de Bois-Préau, 92582, Rueil-Malmaison, France*

Abstract

Dynamic simulators model systems evolving over time. Often, it operates iteratively over fixed number of time-steps. The output of such simulator can be considered as time series or discrete functional outputs. Metamodeling is an effective method to approximate demanding computer codes. Numerous metamodeling techniques are developed for simulators with a single output. Standard approach to model a dynamic simulator uses the same method also for multi-time series outputs: the metamodel is evaluated independently at every time step. This can be computationally demanding in case of large number of time steps. In some cases, simulator outputs for different combinations of input parameters have quite similar behaviour. In this paper, we propose an application of shape invariant model approach to model dynamic simulators. This model assumes a common pattern shape curve and curve-specific differences in amplitude and timing are modelled with linear transformations. We provide an efficient algorithm of transformation parameters estimation and subsequent prediction algorithm. The method was tested with a CO_2 storage reservoir case using an industrial commercial simulator and compared with a standard single step approach. The method provides satisfactory predictivity and it does not depend on the number of involved time steps.

Keywords: Kriging, Functional data analysis, Semiparametric model

1. Introduction

Simulation models are used nowadays in many industrial applications to predict and analyze the behaviour of complex systems. A simulator is a complex computer code embedding the physical laws governing the physical system under investigation. The input of such simulators can be adjustable or uncontrollable parameters which are only partially known and thus are affected by uncertainty. Uncertainty analysis of numerical experiments is used to assess the confidence of the model outcomes which are then used to make decisions [1]. Here, we focus on a particular type of dynamic simulators that are used to make predictions in the future. These simulators are based typically on finite element/finite difference codes used for instance for the simulation of flows and transfers in porous media. Industrial applications using this type of simulators are for instance hydrocarbons reservoir forecasting and carbon dioxide (CO_2) underground storage [2, 3]. Such applications involve very complex numerical codes with a large number of inputs and with high uncertainty derived from an incomplete knowledge of subsurface formation [4]. The uncertainty on the simulator output is usually assumed to be mostly due to the propagation of the uncertainty on the input. Nevertheless, modelling errors can also play a role.

The simulator models a multi-phase 3-D fluid flow in heterogeneous porous media, operating over fixed number of time-steps. The typical output of such simulators consists of a sequence of outputs at different time-steps. Therefore, it represents time series related, for instance, to a recovery rate for a given well or a group of wells. It can be also a spatial output such as a pressure map or a saturation map also evolving with time. Here we focus on 1D time series output which can be typically measured in a well.

Let us consider the output of a simulator as a deterministic function $\mathrm{Y}(t) = \mathrm{F}(\bar{x}, t)$, where $\bar{x} \in \Omega \subset \mathbb{R}^d$ is a configuration of preselected input parameters and $0 < t < T$ refers to a time step. $\mathrm{Y}(t)$ is a time dependent output, e.g. oil production rate or reservoir pressure.

The function $F : \mathbb{R}^d \times [0, T] \to \mathbb{R}$ models the relationship between the input and the output of the numerical simulator. Methods such as Monte Carlo ones can be used to propagate uncertainty. However, each model evaluation being

generally very time-consuming this approach can be unpractical in industrial applications. Moreover, the probability distribution of the input may vary as new data comes in or within a process of dynamic data assimilation. Therefore, a response surface approach (also referred to a metamodel approach) can be more useful in such applications [3, 5, 6].

The advantage of a metamodel is that is fast to evaluate and it is designed to approximate the complex computer code based on a limited number of simulator evaluations. These evaluations of the simulator are taken at some well chosen input configurations also called the training set or the experimental design. Numerous experimental designs have been proposed and many of them are quite sophisticated. Here, we use Latin Hypercube designs [7] for their good space filling properties. Usually, it can be coupled with other criteria such as the maximin design (maximum minimum distance) [8, 9]. In this work, we focus on Gaussian process (GP) based metamodels also known as *kriging* [10, 8, 9].

The aim of this work is to propose a new method to address multi-time series outputs. For such dynamic simulator a standard approach assumes a single step GP model for each time step. This basic approach can be computationally intensive for a high number of time steps and when the construction of a single GP model is time consuming (i.e. when the size of the training set and the number of variables is large). Therefore, the procedure can become unpractical in some industrial applications.

The problem of metamodeling for dynamic simulators was recently investigated by different authors and several principal approaches can be distinguished. A first possible approach is GP metamodelling considering the time steps as the model additional input parameter [3, 11, 12]. This approach is easy to implement, however if we want to take into account all the information from an experimental design at every time steps, the size of new experimental design is multiplied by the number of time steps. It results to matrices of high dimensions that can lead to some numerical problems in case of large size of original experimental design or in case of high density of the steps in the time scale. Conti et al. (2009) [13] developed an iterative approach to build a model of dynamic computer code, assuming that the model output at a given time step depends only on the output at the previous time step. To reduce the dimensionality of the problem, we can also represent the given observations as truncated functions by some selected basis functions with the following GP modelling for the coefficient of the selected representation. Bayarri et al. (2007) [14] introduced wavelet decomposition. Campbell et al. (2006) [15], Higdon et al. (2008) [16] and Lamboni et al. (2009) [17] suggested application of principal component decomposition. Auder et al. (2010) [18] extended this approach by preliminary classification.

In this work, we propose a new functional approach involving a combination of Shape Invariant Model (SIM) approach and Gaussian Process modelling. Considering J time steps and a training set $\mathbf{X}^n = \{\bar{x}_1, .., \bar{x}_n\}$ the simulator output consist then in a set of curves:

$$\mathbf{Y}^n = \left\{ Y_{i,j} = F(\bar{x}_i, t_j), 1 \leq i \leq n, 1 \leq j \leq J \right\}.$$

In our practical example (the CO_2 storage case study) we have observed that usually these curves have quite similar behaviour. So that, we assume that there is a mean pattern from which all the curves can be deduced by a proper parametrical transformation. The shape invariant model assumes that the curves have a common shape, which is modelled nonparametrically. Then curve-specific differences in amplitude and timing are modelled with the linear transformations such as shift and scale [19, 20, 21]. Hence, we consider the following model:

$$Y_{i,j} = \alpha_i^* \ \ f(t_j - \theta_i^*) + v_i^* + \sigma_i^* \varepsilon_{ij}, \ \ 1 \leq i \leq n, 1 \leq j \leq J \tag{1}$$

where t_j stands for observation times and $\boldsymbol{\theta}^* = \{\theta_i, 1 \leq i \leq n\}$, $\boldsymbol{v}^* = \{v_i, 1 \leq i \leq n\}$, $\boldsymbol{\alpha}^* = \{\alpha_i, 1 \leq i \leq n\} \in \mathbb{R}^n$ are vectors of transformation parameters corresponding to horizontal and vertical shifts and scales of the unknown function f.

Without loss of generality we assume that the simulator provides the data at some constant time intervals. Time t is then equispaced in [0, T] with J time steps. The unknown errors $\sigma_i^* \varepsilon_{i,j}$ are independent zero-mean random variables with variance σ_i^{*2}. Here, we can also assume that $\sigma_i^{*2} = 1$ without loss of generality.

The approach proposed in this work for the functional outputs modeling combines an efficient estimation of the transformation parameters ($\boldsymbol{\alpha^*, \theta^*, v^*}$) and a subsequent GP modeling for these parameters that can be used to predict new curves without running the simulator.

The paper is organized as follows. Section 2 presents the basics of GP modeling and the model validation criteria. Section 3 describes the method of efficient estimation of transformation parameters in the shape invariant model. The

method is illustrated with an example on an analytical function. In Section 4 we present the forecast algorithm for dynamic simulators. Section 5 provides the practical application of the algorithm with a CO_2 storage reservoir case.

2. GP based metamodeling

The method proposed in this paper is based on a combination of a shape invariant model and a Gaussian process metamodel. In this section, we recall the basics of GP modeling or *kriging*.

The idea of modeling an unknown function by a stochastic process was introduced in the field of geostatistics by Krige in the 1950's [22] and formalized in 1960's by Matheron (1963) [10]. Later Sacks et al. (1989) [8] proposed the use of kriging for prediction and design of experiments. The theory and the algorithms are formalized in [8], [23] and [9].

Consider the output of a simulator as an unknown deterministic function $\mathrm{F}(\bar{x}) \in \mathbb{R}$, where $\bar{x} \in \Omega \subset \mathbb{R}^d$ is a specified set of selected input parameters. The function F is only known in predetermined design points: $\mathbf{X}^n = \{(\bar{x}_k, \mathrm{F}_k = \mathrm{F}(\bar{x}_k)), 1 \leq k \leq n\}$. The objective is to predict the function $\mathrm{F}_0 = \mathrm{F}(\bar{x}_0)$ for some new arbitrary input $\bar{x}_0$. The function is modeled as a sample path of a stochastic process of the form:

$$\widetilde{\mathrm{F}}(\bar{x}) = \sum_{j=1}^{m} h_j(\bar{x}) \cdot \beta_j + Z(\bar{x}) = \boldsymbol{\beta}^\top \mathbf{h}(\bar{x}) + Z(\bar{x}) \tag{2}$$

where:

- $\boldsymbol{\beta}^\top \mathbf{h}(\bar{x})$ is the mean of the process and corresponds to a linear regression model with preselected given real-valued functions $\mathbf{h} = \{h_i, 1 \leq i \leq m\}$. Here, we only consider the case $\mathbf{h} = \mathbf{1}$.
- $Z(\bar{x})$ is a centered Gaussian stationary random process. It is defined by its covariance function: $\mathrm{C}(\bar{x}, \bar{y}) = \mathrm{E}\left[Z(\bar{x})Z(\bar{y})\right] = \sigma^2 \mathrm{R}(\bar{x}, \bar{y})$. $\mathrm{R}(\bar{x}, \bar{y})$ is the correlation function and $\sigma^2 = \mathrm{E}[Z(\bar{x})^2]$ denotes the process variance. Stationarity condition assumes: $\mathrm{R}(\bar{x}, \bar{y}) = \mathrm{R}(|\bar{x} - \bar{y}|)$, where $|\bar{x} - \bar{y}|$ denotes the distance between $\bar{x} \in \Omega$ and $\bar{y} \in \Omega$.

Numerous families of correlation functions have been proposed in the literature. We use here Gaussian correlation function, the special case of the power exponential family. The power exponential correlation function is of the following form:

$$\mathrm{R}(\bar{x}, \bar{y}) = \exp\left(-\sum_{j=1}^{d} \frac{\left(x_j - y_j\right)^{p_j}}{\theta_j}\right) = \exp\left(-\sum_{j=1}^{d} \frac{d_j^{p_j}}{\theta_j}\right) \tag{3}$$

where $d_j = |x_j - y_j|$, $0 < p_j \leq 2$ and $\theta_j > 0$. The hyperparameters $(\theta_1, .., \theta_d)$ stands for correlation lengths which affect how far a sample point's influence extends. A high θ_i means that all points will have a high correlation ($\mathrm{F}(x_i)$ being similar across our sample), while a low θ_i means that there are significant difference between the $\mathrm{F}(x_i)$'s [24]. The parameters p_j corresponds to the smoothness parameters. These parameters determine mean-square differentiablity of the random process $Z(x)$. For $p_j = 2$ the process is infinitely mean-square differentiable and the correlation function is called Gaussian correlation function. Hence, Gaussian correlation function is infinitely mean-square differentiable and it leads to a stationary and anisotropic process $Z(x)$ [9, 23]. Regardless the choice of a correlation function, the estimation of hyperparameters $(\theta_1, .., \theta_d)$ is crucial for reliable prediction. We are using maximum likelihood estimation algorithm [9] that we will discuss later.

The experimental design points are selected in order to retrieve most information on the function at the lowest computational cost. The number of design points for a reliable response surface model depends on the number of inputs and on the complexity of the response to analyze [7, 9]. Latin Hypercube Designs (LHD) provides a uniform coverage of the input domain. If we wish to generate a sample of size $\mathbf{n}$, first, we partition the domain of each variable in $\mathbf{n}$ intervals of equal probability. Then, we randomly sample $\bar{x}_1$ according to the distribution of each of the $\mathbf{n}$ intervals. Further, for each of the $\mathbf{n}$ values for $\bar{x}_1$, we randomly select one interval to sample for $\bar{x}_2$, so that only one sample of $\bar{x}_2$ is taken in each interval. We continue the process of a random sampling without replacement until all the variables have been sampled. As a result we generate a sample where each of d inputs is sampled only once in each

of **N** intervals. Latin hypercube designs have been applied in many computer experiments since they were proposed by Mckay et al., (1979) [7].

In this work, we use modified version of LHD - maximin LHD. It is based on maximizing a measure of closeness of the points in a design $\mathbf{D}^n$:

$$\max_{\text{design } \mathbf{D}^n} \min_{\bar{x}_1,\bar{x}_2 \in \mathbf{D}^n} d(\bar{x}_1, \bar{x}_2)$$

It can guarantee that any two points in the design are not "too close". Hence, the design points are uniformly spread over the input domain.

Consequently, when we have the experimental design $\mathbf{X}^n = (\bar{x}_1, .., \bar{x}_n)$ and the observation data $\mathbf{Y}^n = (F(\bar{x}_1), .., F(\bar{x}_n))$ the multivariate distribution according to the model (2) for the Gaussian correlation function can be expressed as:

$$\begin{pmatrix} Y_0 \\ \mathbf{Y}^n \end{pmatrix} \sim \mathcal{N}_{1+n}\left[\begin{pmatrix} \mathbf{h}^\top(\bar{x}_0) \\ \mathbf{H} \end{pmatrix}\boldsymbol{\beta},\ \boldsymbol{\sigma}^2\begin{pmatrix} 1 & r^\top(\bar{x}_0) \\ r(\bar{x}_0) & \mathbf{R} \end{pmatrix}\right],$$

where $\mathbf{R} = (\mathrm{R}(\bar{x}_i, \bar{x}_j))_{1\le i,j\le n} \in \mathbb{R}^{n\times n}$ is the correlation matrix among the observations; $\mathbf{r}(\bar{x}_0) = (\mathrm{R}(\bar{x}_1, \bar{x}_0), .., \mathrm{R}(\bar{x}_n, \bar{x}_0))^\top \in \mathbb{R}^n$ is the correlation vector between the observations and the prediction point; $\mathbf{h}^\top(\bar{x}_0) = \left(h_j(\bar{x}_0)\right)_{1\le j\le m} \in \mathbb{R}^m$ is the vector of regression function at the prediction point $\bar{x}_0$ and $\mathbf{H} = \left(h_j(\bar{x}_i)\right)_{1\le i\le n, 1\le j\le m} \in \mathbb{R}^{n\times m}$ is the matrix of regression functions at the experimental design. The parameters $\boldsymbol{\beta}$ and $\boldsymbol{\sigma}$ are unknown.

Considering the unbiasedness constraint, the parameter $\boldsymbol{\beta}$ is replaced by the generalized least squares estimate $\widehat{\boldsymbol{\beta}}$ in (2). Here, $\widehat{\boldsymbol{\beta}}$ is of the following form: $\widehat{\boldsymbol{\beta}} = \left(\mathbf{H}^\top\hat{\mathbf{R}}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^\top\hat{\mathbf{R}}^{-1}\mathbf{Y}^n$. Assuming that the correlation function is the Gaussian correlation function, the prediction is therefore given by:

$$\widehat{F(\bar{x}_0)} = \mathbf{h}^\top(\bar{x}_0)\cdot\hat{\boldsymbol{\beta}} + \hat{\mathbf{r}}(\bar{x}_0)\ \ \hat{\mathbf{R}}^{-1}\left(\mathbf{F}^n - \mathbf{H}\cdot\hat{\boldsymbol{\beta}}\right) \tag{4}$$

The hyperparameters $\boldsymbol{\theta} = (\theta_1, .., \theta_m)$ and the process variance $\boldsymbol{\sigma}^2$ are estimated by Maximum Likelihood (MLE). Using the multivariate normal assumption, the MLE for $\boldsymbol{\sigma}^2$ is:

$$\hat{\boldsymbol{\sigma}}^2 = \frac{1}{m}\left(\mathbf{Y}^n - \mathbf{H}\,\hat{\boldsymbol{\beta}}\right)\ \mathbf{R}^{-1}\left(\mathbf{Y}^n - \mathbf{H}\,\hat{\boldsymbol{\beta}}\right) \tag{5}$$

Knowing the estimations for $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\sigma}}^2$, the coefficients $\boldsymbol{\theta}$ are estimated by maximizing the log likelihood:

$$l\left(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\sigma}}^2, \boldsymbol{\theta}\right) = -\frac{1}{2}\left[m\log\hat{\boldsymbol{\sigma}}^2(\boldsymbol{\theta}) + \log(det(\mathbf{R}(\boldsymbol{\theta})) + m\right] \tag{6}$$

The function (6) depends only on $\boldsymbol{\theta}$.

After having estimated all the model parameters, we now need to validate the model. In this work for estimation of prediction accuracy of the model, we use the predictivity index, Q_2, and root mean squared error, $RMSE$. The predictivity index is calculated basing on cross validation and it has the following form:

$$RMSE := \sqrt{\sum_{i=1}^{n}\frac{(\hat{S}_{X/x_i} - F(x_i))^2}{n}} \quad Q_2 := 1 - \frac{\sum_{i=1}^{n}\left(\hat{S}_{X/x_i} - F(x_i)\right)^2}{\sum_{i=1}^{n}\left(F(x_i) - \tilde{F}\right)^2} \tag{7}$$

$\hat{S}_{X/x_i}$ is the kriging model computed using all the design points $\mathbf{X}^n$ excepting $\bar{x}_i$ and $\tilde{F}$ is the mean of $F(\bar{x}_i)$.

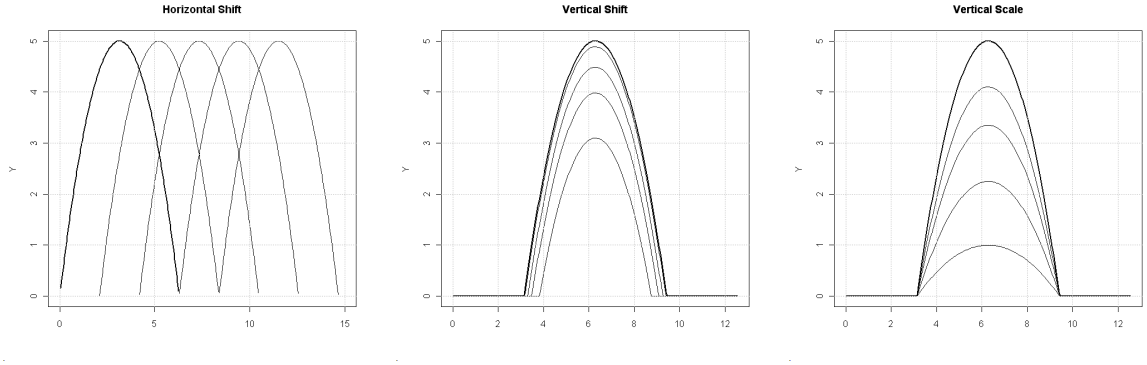
The closer Q_2 is to 1 or $RMSE$ is to 0, the higher is the model predictivity. These criteria can be also calculated on a separate validation test data by performing additional simulations. It provides higher accuracy measure though it requires additional time costs.

3. Shape Invariant Model

In this section, we discuss the shape invariant model representation and the procedure for efficient parameters estimation.

The shape invariant model was introduced by Lawton et al. (1972) [19]. The model assumes that we are working with a set of curves that have a common shape function that is modeled nonparametrically. The deformation of this function is modeled parametrically by choosing a proper parametrical transformation. We consider a class of linear transformations only. These parameters can be normally interpreted as shift in timing or scale in amplitude. For this reason, shape invariant model is widely applied to study periodic data such as temperature annual cycle [25] or traffic data analysis [26]. Indeed, in these cases there is always some variability in time cycles or amplitude. The model has also been used to study biological data , where the departure from the pattern can be caused by a different environmental conditions [21, 20]. In this work, we use the model to propagate uncertainty on a reservoir simulator output data. For example, we consider a reservoir pressure during CO_2 injection, then shift in time is caused by different moment of stopping injection. So that, by applying shape invariant model, we can study the influence of the model input parameters on the overall shapes of the selected output.

In figure (1) we display three possible transformations that we consider in our work: horizontal shift 1(a), vertical shift 1(b) and vertical scaling 1(c). The bold line represent original pattern shape.



(a) Horizontal Shift (b) Vertical Shift (c) Vertical Scale

Figure 1: Parametrical transformation examples

We are interested in the common shape as well as in the efficient estimation of transformation parameters. So that, we can reproduce any curve and align it to the pattern. Moreover, we can make a prediction of a possible new curve for an input configuration $\bar{x}_0$ by modeling the transformation parameters.

As already mentioned, for an experimental design $\mathbf{X}^n = \{\bar{x}_1, .., \bar{x}_n\}$ we have a set of observations:

$\mathbf{Y}^n = \left\{Y_{i,j} = \mathrm{F}(\bar{x}_i, t_j), 1 \leq i \leq n, \;\; 1 \leq j \leq J\right\}$, where $\bar{x}_i, \;\; 1 \leq i \leq n$ is the set of preselected input parameters and $t_j, \;\; 1 \leq j \leq J$ refers to the time sample. So that, $Y_{i,j}$ is the j^{th} observation on the i^{th} experimental design unit, with $1 \leq i \leq n$ and $1 \leq j \leq J$. Thereby, the idea is to find a general pattern curve that makes possible subsequent transformation of any curve to this selected pattern with properly adjusted transformation parameters.

We focus here on linear transformations. Thus, the model structure may be written as:

$$Y_{i,j} = \alpha_i^* \cdot f(t_j - \theta_i^*) + v_i^* + \sigma^{*2} \cdot \varepsilon_{ij} \tag{8}$$

where $t_j, \;\; 1 \leq j \leq J$ are observation times which are assumed to be known and equispaced in the interval $[0, T]$. The vector of parameters: $(\boldsymbol{\alpha^*, \theta^*, v^*}) = (\alpha_1, .., \alpha_n, \theta_1, .., \theta_n, v_1, .., v_n)$ is unknown as well as the pattern function f. The errors ε_{ij} are i.i.d. with a normal distribution, $(i, j) \in \{1, .., n\} \times \{1, .., J\}$. It characterizes observation noise. Without loss of generality we may assume that $\sigma^{*2} = 1$. The variance does not affect the parameters estimation procedure and the method still works for a variance σ^{*2}. The function f is assumed to be 2π-periodic [25, 26].

In this section we will provide the algorithm for efficient estimation of transformation parameters under unknown function pattern f. Since the functional pattern f is unknown, the pattern is replaced by its estimate. So that, it seems natural to study the problem (8) in a semi-parametric framework: the transformation shifts and scales are the parameters to be estimated, while the pattern stands for an unknown nuisance functional parameter. We use an M-estimator built on the Fourier series of the data. Under identifiability assumptions it is possible to provide a consistent

algorithm to estimate $(\boldsymbol{\alpha^*, \theta^*, \nu^*})$ when f is unknown. The algorithm is described in details in the following subsection with an illustration on an analytical function example.

3.1. Model Assumptions

Consider $\mathbf{Y}^n = F(\mathbf{X}^n, \boldsymbol{t}) = \{F(\bar{x}_i, t_j), \quad 1 \leq i \leq n, 1 \leq j \leq J\}$ is $(n \times J)$ matrix of observations. We model these observations in the following way:

$$Y_{i,j} = \alpha_i^* \cdot f(t_j - \theta_i^*) + v_i^* + \varepsilon_{ij}, \quad 1 \leq i \leq n, 1 \leq j \leq J \tag{9}$$

where $f : \mathbb{R} \to \mathbb{R}$ is an unknown 2π-periodic continuous function, $\boldsymbol{\theta}^* = (\theta_1^*, ..., \theta_n^*)$, $\boldsymbol{\alpha}^* = (\alpha_1^*, ..., \alpha_n^*)$, $\boldsymbol{\nu}^* = (\nu_1, ..., \nu_n) \in \mathbb{R}^n$ are unknown parameters, ε_{ij} is a Gaussian white noise with variance equal to 1. The time period is translated in such a way that: $[0, \mathrm{T}[\to [0, 2\pi[$, therefore $t_i = \frac{j-1}{J} 2\pi, \quad j = 1, .., J$ are equispaced in $[0, 2\pi[$.

The objective is to estimate the horizontal shift $\boldsymbol{\theta}^* = (\theta_1^*, ..., \theta_n^*)$, the vertical shift $\boldsymbol{\nu}^* = (\nu_1, ..., \nu_n)$ and the scale parameter $\boldsymbol{\alpha}^* = (\alpha_1^*, ..., \alpha_n^*)$ without knowing of the pattern f. Fourier analysis is well suited for the selected structure of the model. Indeed, this transformation is linear and shift invariant. Therefore, by applying a discrete Fourier transform to (9) and supposing J is odd, the model becomes:

$$d_{kl} = \begin{cases} \alpha_k^* e^{-il\theta_k^*} c_l(f) + w_{kl}, & 1 \leq k \leq n, \quad 0 < |l| \leq (J-1)/2 \\ \alpha_k^* c_0(f) + \nu_k^* + w_{k0}, & 1 \leq k \leq n, \quad l = 0 \end{cases} \tag{10}$$

where $c_l(f) = \frac{1}{J} \sum_{m=1}^{J} f(t_m) e^{-2i\pi \frac{ml}{J}}$, $\quad (|l| \leq (J-1)/2)$ are the discrete Fourier coefficients and w_{kl}, $\quad (1 \leq k \leq n, \ |l| \leq (J-1)/2)$ is a complex white Gaussian noise with independent real and imaginary parts.

We also notice that in order to ensure the identifiability of the model (10) we are working in the parameter space: $\mathbf{A} = \left\{(\boldsymbol{\alpha^*, \theta^*, \nu^*}) \in [-\pi, \pi[^{3\times n} \colon \alpha_1 = 1, \theta_1 = 0, \nu_1 = 0\right\}$.

To summarize, in this section we estimate the transformation parameters $(\boldsymbol{\alpha^*, \theta^*, \nu^*})$ without prior knowledge of the function f. The estimation depends on the unknown functional parameter $(c_l(f))_{|l| \leq (J-1)/2}$, the Fourier coefficients of the unknown function f. So that, we consider a semi-parametrical method based on an M-estimation procedure. M-estimation theory enables to build an estimator defined as a minimiser of a well-tailored empirical criterion that is given in the following subsection.

3.2. Parameters estimation procedure

The goal is to estimate the vector of parameters $(\boldsymbol{\alpha^*, \theta^*, \nu^*})$ that depends on the Fourier coefficients of the unknown function f. We consider a semi-parametric method based on an M-estimation procedure [26].

To construct an M-estimator, we define the rephased (untranslated and rescaled) coefficients for any vector $(\boldsymbol{\theta, \alpha, \nu}) \in \mathbf{A}$:

$$\tilde{c}_{kl}(\alpha, \theta, \nu) = \begin{cases} \frac{1}{\alpha_k} e^{il\theta_k} d_{kl}, & 1 \leq k \leq n, \quad 0 < |l| \leq (J-1)/2 \\ \frac{1}{\alpha_k} (d_{kl} - \nu_k), & 1 \leq k \leq n, \quad l = 0 \end{cases}$$

and the mean of these coefficients:

$$\hat{c}_l(\alpha, \theta, \nu) = \frac{1}{n} \sum_{k=1}^{n} \tilde{c}_{kl}(\alpha, \theta, \nu), \quad |l| \leq (J-1)/2$$

Therefore, for $(\boldsymbol{\alpha^*, \theta^*, \nu^*})$ we obtain:

$$\tilde{c}_{kl}(\boldsymbol{\alpha^*, \theta^*, \nu^*}) = c_l(f) + \frac{1}{\alpha_k^*} \ e^{il\theta_k^*} w_{kl} \ \ 1 \leq k \leq n$$

$$\hat{c}_l(\boldsymbol{\alpha^*, \theta^*, \nu^*}) = c_l(f) + \frac{1}{n} \sum_{k=1}^{n} \frac{e^{il\theta_k^*} \cdot w_{kl}}{\alpha_k^*}$$

.

Hence, $|\tilde{c}_{kl}(\alpha, \theta, \nu) - \hat{c}_l(\alpha, \theta, \nu)|^2$ should be small when $(\boldsymbol{\alpha, \theta, \nu})$ is closed to $(\boldsymbol{\alpha^*, \theta^*, \nu^*})$.

Now, consider a bounded positive measure μ on $[0, T]$ and set

$$\delta_l := \int_0^T \exp\left(\frac{2i\pi l}{T}\omega\right) d\mu(\omega), \quad (l \in \mathbb{Z}) \tag{11}$$

Obviously, the sequence (δ_l) is bounded. Without loss of generality we will assume that $\delta_0 = 0$ and $|\delta_l| > 0, l \neq 0$. Assume further that $\sum_l |\delta_l|^2 |c_l(f)|^2 < +\infty$. So that $f * \mu$ is a well defined square integrable function:

$$f * \mu(x) = \int f(x - y) d\mu(y)$$

We construct the following empirical contrast function (12):

$$M_n(\alpha, \theta, v) = \frac{1}{n} \cdot \sum_{k=1}^{n} \sum_{l=-\frac{J-1}{2}}^{\frac{J-1}{2}} |\delta_l|^2 \, |\tilde{c}_{kl}(\alpha, \theta, v) - \hat{c}_l(\alpha, \theta, v)|^2 \tag{12}$$

The random function M_n is non negative. Furthermore, its minimum value is reached closely to the true parameters $(\boldsymbol{\alpha^*, \theta^*, v^*})$. We define the estimator by:

$$(\widehat{\alpha}, \widehat{\theta}, \widehat{v})_n = \arg \min_{\alpha, \theta, v \in \mathbf{A}} M_n(\alpha, \theta, v)$$

The proof of convergence $(\widehat{\alpha}, \widehat{\theta}, \widehat{v})_n \xrightarrow[n \to +\infty]{\mathbf{P}} (\boldsymbol{\alpha^*, \theta^*, v^*})$ and asymptotic normality of the estimators can be found in [26, 25]. The weights δ_l in (11) are chosen to guarantee the convergence of the contrast function to a deterministic function M_n and to provide the asymptotic normality of the estimators. Moreover, the convergence can be speeded up by proper selection of weights. The analysis of covergence at different weights is presented in [26]. In this work we used the weight $\delta_l = 1/|l|^\beta$ with $\beta = 1.5$.

The computation of the estimators is very fast since only a Fast Fourier algorithm and a minimization algorithm of a quadratic functional are needed. The procedure is summarized in Algorithm (1)

Algorithm 1 Parameters estimation procedure

Input: Input set of curves from experimental design $\mathbf{Y}^n = \{Y_{ij}, i = 1, .., n; \quad j = 1, .., J\}$

Output: Transformation parameters estimation $(\boldsymbol{\alpha^*, \theta^*, v^*})$

Define the identifiability condition: $\mathbf{A} = \left\{(\boldsymbol{\alpha^*, \theta^*, v^*}) \in [-\pi, \pi[^{3 \times n} \colon \alpha_1 = 1, \theta_1 = 0, v_1 = 0\right\}$

Compute the matrix of discrete Fourier coefficients $D = \{d_{kl}, \quad k = 1, .., n; \quad |l| \leq (J-1)/2\}$

Compute the matrix of rephased Fourier coefficient $\tilde{C} = \{\tilde{c}_{kl}, \quad k = 1, .., n; \; |l| \leq (J-1)/2\}$

Compute the vector of mean of rephased coefficients $\widehat{C} = \{\widehat{c_l}, \quad |l| \leq (J-1)/2\}$

Choose the weight sequence δ_l

Define $M_n(\alpha, \theta, v) = \frac{1}{n} \cdot \sum_{k=1}^{n} \sum_{l=-\frac{J-1}{2}}^{\frac{J-1}{2}} |\delta_l|^2 \, |\tilde{c}_{kl}(\alpha, \theta, v) - \hat{c}_l(\alpha, \theta, v)|^2$

Compute $(\boldsymbol{\widehat{\alpha}, \widehat{\theta}, \widehat{v}}) = \arg\min_{\alpha, \theta, v \in \mathbf{A}} M_n(\alpha, \theta, v) \in \mathbb{R}^{3 \times (n-1)}$

Return: $(\boldsymbol{\widehat{\alpha}, \widehat{\theta}, \widehat{v}}) \in \mathbb{R}^{3 \times (n-1)}$

3.3. Analytical Function Example

In this section the shape invariant model and the efficient parameters estimation are presented on an analytical function. The minimization algorithm used in the estimation procedure is a Newton-type algorithm.

Let us consider the following function:

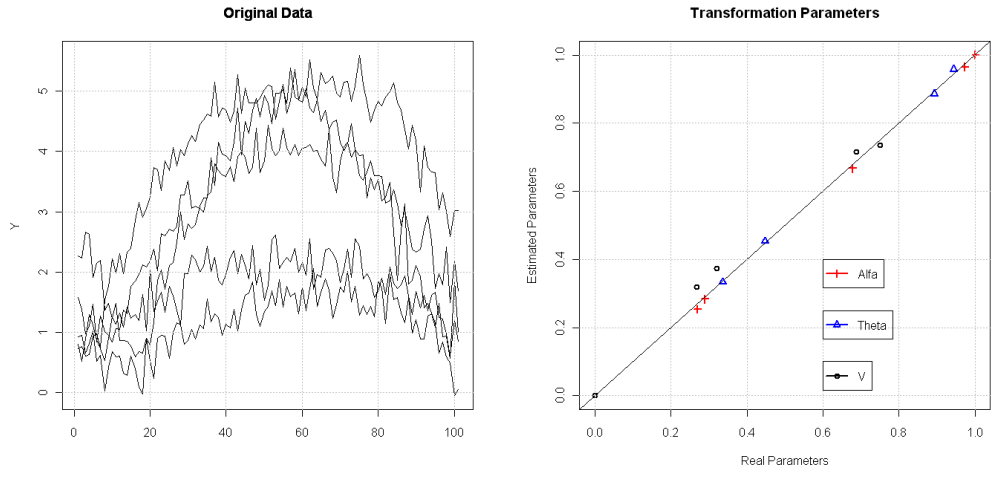
$$f(x) = 20 \cdot (1 - x/(2\pi)) \cdot x/(2\pi)$$

Simulated data are generated as follows:

$$Y_{ij} = \alpha_i^* \cdot f(t_j - \theta_i^*) + v_i^* + \varepsilon_{ij}, \quad 1 \le i \le n, 1 \le j \le J$$

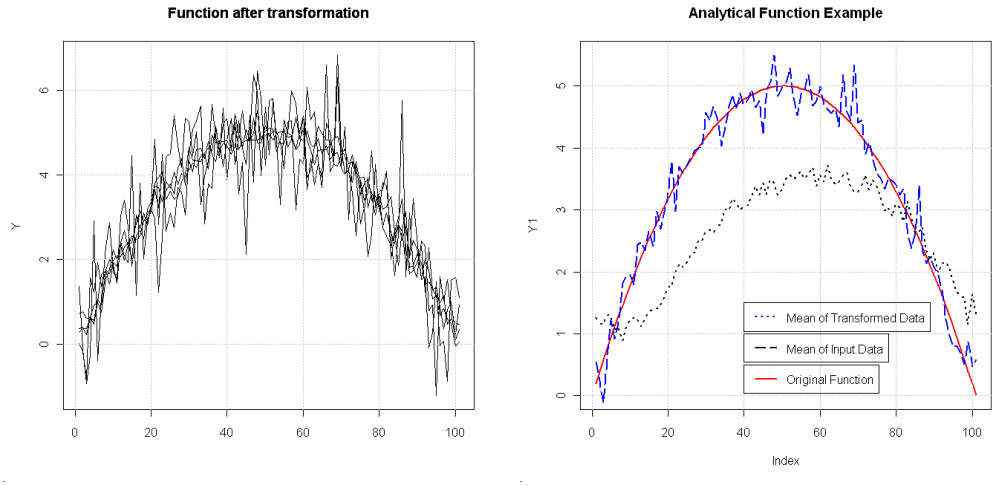
with the following choice of parameters: $J = 5$, $N = 101$, $t_j = \frac{j-1}{J} 2\pi$, $1 \le j \le J$ are equally spaced points on $[0, 2\pi[$. Transformation parameters $(\boldsymbol{\theta^*}, \boldsymbol{\alpha^*}, \boldsymbol{v^*})$ are uniformly distributed on $]0, 1]$, where $\theta_1^* = 0$, $v_1^* = 0$, $\alpha_1^* = 1$; the noise ε_{ji}, $j = 1, .., J, i = 1, .., N$ are simulated with a Gaussian law with mean 0 and variance 0.5.

Results are displayed in Figure 2. The function f is plotted by a solid red line in Figure 2(d). Figure 2(a) shows the original simulated noisy data $Y_{i,j}$. The cross-sectional mean function of these data is presented in Figure 2(d) by black dotted line. Figure 2(b) plots estimated transformation parameter versus the originally simulated parameters. As it can be seen, the estimations are very close to the original parameters. The inverse transformation using the estimated parameters is displayed in Figure 2(c) and the mean function of restored curves is displayed in Figure 2(d) by blue dashed line. Figure 2(d) compares the cross-sectional mean of inversely transformed data and the cross-sectional mean of originally simulated data. Despite the noise, it is noticeable that the data after inverse transformation are much more closer to the original function f than the original cross-sectional mean function.



(a) Original noisy data

(b) Calculated parameters



(c) Inverse Transformation

(d) Comparison

Figure 2: Analytical Example

This analytical example shows that the method is effective in estimating the transformation parameters of the shape invariant model. In the next section, we will explain how this model can be applied in reservoir engineering forecast problems.

4. Functional data approximation

To apply the shape invariant model approach to approximate functional data from a dynamic simulator, firstly we have to modify the parameters estimation procedure for large number of curves. When we are working with uncertainty modeling, we always start from an experimental design $\mathbf{X}^n$ and a set of observation $\mathbf{Y}^n$. As we have mentioned, the number of design points depends on the number of inputs and on the complexity of the response. So that, some optimization problems could arise when we compute the contrast function with a large number of curves. It can be time consuming and the results can be inaccurate. Therefore, we propose the following modification to the Algorithm (1): the original observation data is split into blocks and the optimization procedure is then performed on each of the blocks. Following the identifiability condition, we are working on a compact set $\mathbf{A} = \left\{(\boldsymbol{\alpha^*}, \boldsymbol{\theta^*}, \boldsymbol{\nu^*}) \in [-\pi, \pi[^{3\times n}\colon \boldsymbol{\alpha}_1 = 1, \boldsymbol{\theta}_1 = 0, \boldsymbol{\nu}_1 = 0\right\}$. This condition should be satisfied on every block optimization by adding the first reference curve to the block.

Algorithm 2 Parameters estimation procedure for large n

Input: Input set of curves from experimental design $\mathbf{Y}^n = \{Y_{ij}, i = 1, .., n; \quad j = 1, .., J\}$

Output: Transformation parameters estimation $(\boldsymbol{\alpha^*}, \boldsymbol{\theta^*}, \boldsymbol{\nu^*})$

Split the observation data into N_b blocks of $(K + 1)$ curves

for $m = 1, .., N_b$ **do**

Define block curves $\mathbf{Y}^{K+1} = \{Y_1, Y_{(m-1)(K+1)\ldots, Y_{mK}}\} = \{Y_{ij}, \quad i = 1, .., K + 1, j = 1, .., J\}$

Perform Algorithm (1)

Compute $(\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\nu}}) = \arg \min_{\alpha,\theta,\nu\in\mathbf{A}} M_n(\alpha, \theta, \nu)$, where $(\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\nu}}) \in \mathbb{R}^{3\times K}$

end for

Return: $(\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\nu}}) \in \mathbb{R}^{3\times(n-1)}$

With this procedure, we do not have limitations on experimental design of any size. As soon as we have estimated the parameters for every curve from observation data set, we can formulate the prediction algorithm. Instead of reproducing the simulator output for a prediction point $\bar{x}_0$ at every time step, we model the whole output curve with the transformation parameter. This curve will provide the approximation of the output for the selected input configuration $\bar{x}_0$ at each of considered time steps $\{t_j, \quad j = 1, .., J\}$. The transformation parameters for the input $\bar{x}_0$ are evaluated with the Gaussian process response surface modeling. The model is based on the experimental design and the set of evaluated transformation parameters calculated for the observation data curves. The prediction framework for an arbitrary input configuration $\bar{x}_0$ is presented by the following Algorithm (3).

Summing up, with this proposed algorithm the problem of response surface modeling for dynamic simulators is reduced from single step GP modeling for each of J time steps to an optimization problem and a GP modeling for the transformation parameters. However, it is worth to mention that before performing the algorithm it is important to analyze the curves behaviour for the observation data set. Studying the curves characterization, probably we may fix for example vertical shifts $\boldsymbol{\nu}$ or horizontal shifts $\boldsymbol{\theta}$ at zero. Also, if the curves have significantly different behaviour at different time intervals we can split the observation data in time as well in order to achieve higher prediction accuracy.

The next section presents an application with a dynamic reservoir simulator case.

5. CO_2 storage reservoir case

Carbon Capture and Storage technology stands for the collection of CO_2 from industrial sources and its injection underground. Carbon dioxide is stored in a deep geological formation that is sealed on a top by a low permeability cap rock formation. Subsurface storage of CO_2 is always associated with an excess pressure in the reservoir and one

Algorithm 3 Prediction algorithm for dynamic simulator

Input: Dynamic simulator $Y = F(\bar{x}, t)$ with $t \in \{t_j, \quad j = 1, .., J\}$ and prediction point $\bar{x}_0$

Output: Prediction $Y^0 = F(\bar{x}_0, t_j)$ for all $j = 1, .., J$

Generate an experimental design $\mathbf{X}^n = (\bar{x}_1, .., \bar{x}_n)$ to span the space of interest

Evaluate $\mathbf{Y}^n = \mathrm{F}(\mathbf{X}^n, t_j)$ at every time step $t_j, \quad 1 \leq j \leq J$

Generate a set of discrete curves $\{Y_{i,j}\}, \quad i = 1, .., n; j = 1, .., J$

Estimate the $(\boldsymbol{\alpha}, \boldsymbol{\theta}, \boldsymbol{\nu}) \in (\mathbb{R}^n)^3$ with Algorithm 2

Construct new experimental design for the function of parameters: $(\mathbf{X}^n, \boldsymbol{\theta}(\mathbf{X}^n)), \quad (\mathbf{X}^n, \boldsymbol{\alpha}(\mathbf{X}^n))$ and $(\mathbf{X}^n, \boldsymbol{\nu}(\mathbf{X}^n))$

Estimation of hyperparameters for GP models of transformation parameters

$\boldsymbol{\alpha}(\bar{x}_0), \quad \boldsymbol{\theta}(\bar{x}_0)$ and $\boldsymbol{\nu}(\bar{x}_0)$ are approximated with corresponding GP models

Reproduce: $F(\bar{x}_0, t_j) = \boldsymbol{\alpha}(\bar{x}_0) f(t_j - \boldsymbol{\theta}(\bar{x}_0)) + \boldsymbol{\nu}(\bar{x}_0)$ for all $\{t_j, \quad j = 1, .., J\}$

Return: Discrete time series $Y^0 = F(\bar{x}_0, t)$ with $t \in \{t_j, \quad j = 1, .., J\}$

of the primary environmental risks is a pressure-driven leakage of CO_2 from the storage formation. In order to assess the risk of CO_2 leakage through the cap rock we consider a synthetic reservoir model. The model is made up of three zones (3):

- a reservoir made of 10 layers
- a cap-rock made up of 9 layers
- a zone-to-surface composed of 1 layer

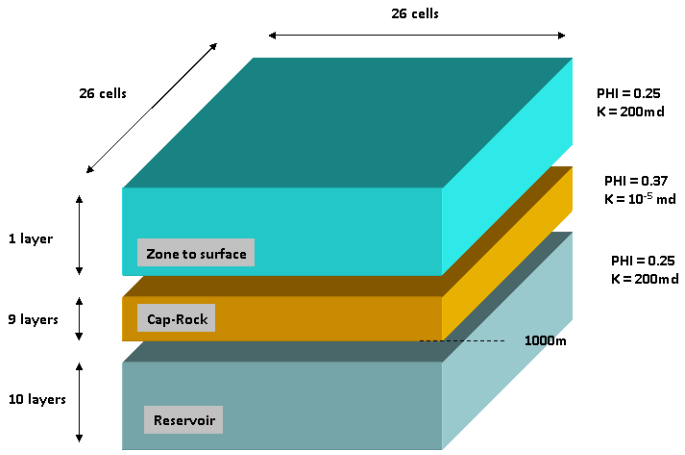


Figure 3: Reservoir Model

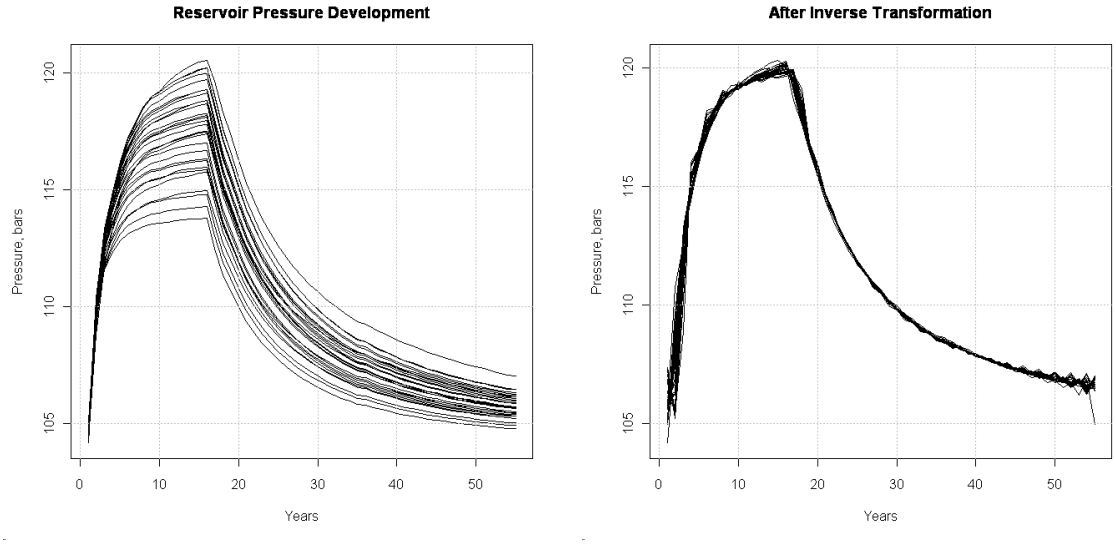
The XY size of the grid is set at 10 km total length. Each layer is 5m thick, including the cell above the cap-rock. The total number of cells is 13520 (26x26x20 model grid). The structure of the reservoir is reduced to its simplest expression. The zone above the cap-rock (up to the surface) is currently set to 1 layer. The salinity of the water is 35gm/l. The temperature of the reservoir is set to 60C and the initial pressure is hydrostatic. The injection bottom rate is set to 10E+06 tons/year. The fracture pressure is estimated by geomechanical experts to 158 bars. Exceeding this value of reservoir pressure can lead to a leakage. The simulation period is 55 years that include an injection period of 15 years followed by 40 years of storage. In this study we analyze the possibility of leakage through a cap rock. Therefore, we consider pressure in the storage reservoir as an objective function to be approximated.

The uncertain parameters selected for this study characterize the reservoir and the fluid properties. It implies different CO_2 flowing possibilities between the reservoir layers. The distribution law for the parameters is uniform. Table (1) represents the parameters description with their range of minimum and maximum values.

Name	Description	Min	Max
PORO	Reservoir Porosity	0.15	0.35
KSAND	Reservoir Permeability	10	300
KRSAND	Water relative permeability end-point	0.5	1.0

Table 1: Uncertain Parameters

We start from the observation data $\mathbf{Y}^n$ - $[30; 55]$ matrix of simulator outputs. By means of Algorithm (2) we provide the transformations parameters estimations. Figure (4(a)) provides the original set of curves and Figure (4(b)) represents the same set after inverse transformation. The pattern curve is differentiable after the inverse transformation with the estimated parameters.



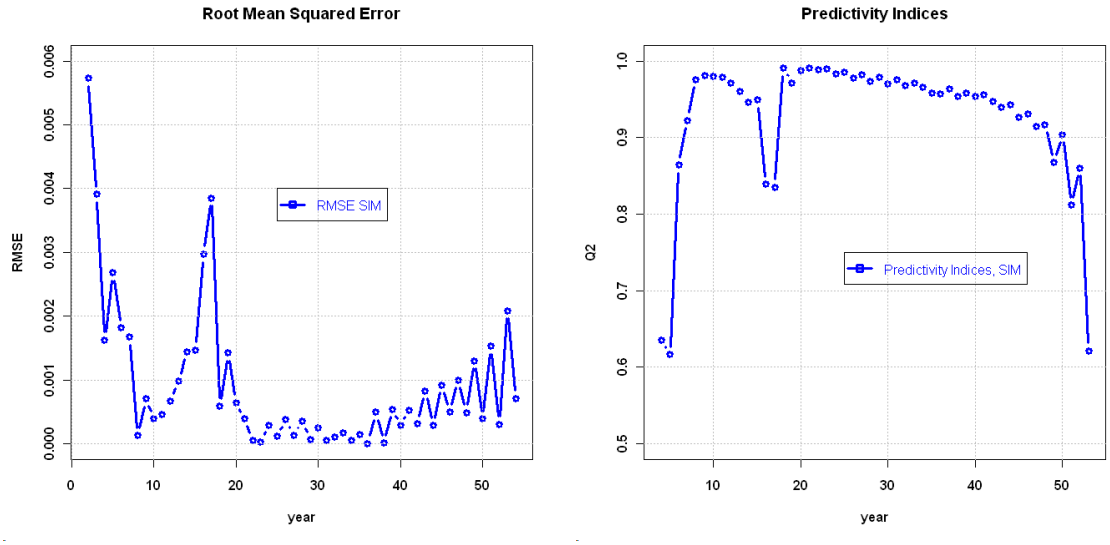
(a) Observation data

(b) Restored data

Figure 4: Original observation data and data after inverse transformation

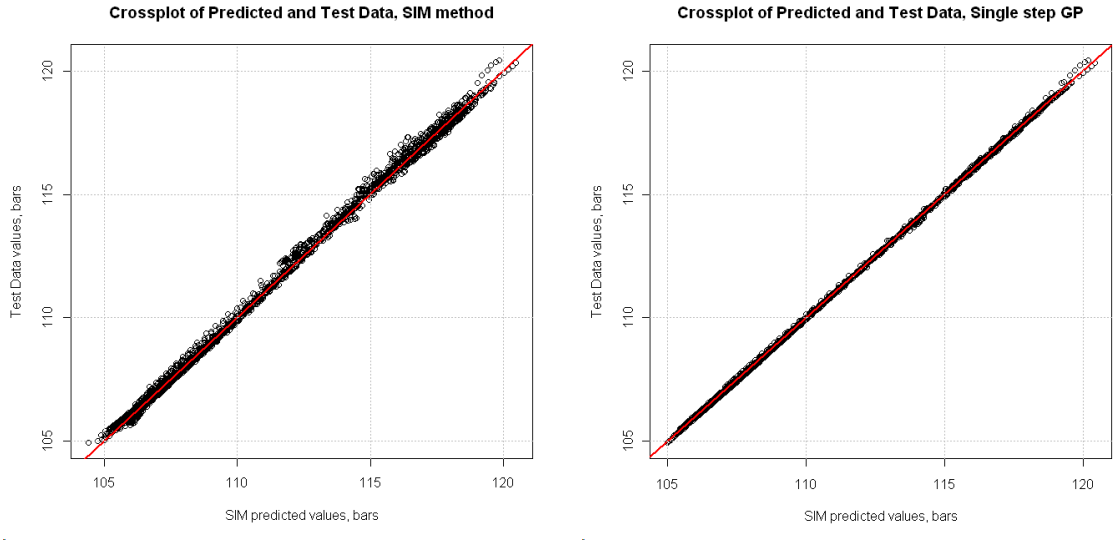
As we are sure here that the model parameters are efficiently estimated, we can proceed with the next step of prediction algorithm (3). The next step is to build Gaussian process response surface models for the transformation parameters basing on the estimations.

In figure (5) we display the model validation criteria: Root Mean Square Error (RMSE) and predictivity indices (Q2), calculated separately for every year. The criteria were computed with the help of additional confirmation test data. The low predictivity in the first and last years is caused by low variance of data in that period. In general, the method provides reliable level of predictivity. It is also reflected by crossplots of test and predicted data. Figure (6(a)) is based on new proposed approach and Figure (6(b)) corresponds to single step GP modelling. Both methods provide a good level of approximation.



(a) RMSE (b) Predicitivity Indices

Figure 5: Predictivity Indices and RMSE



(a) SIM method (b) Single Step GP

Figure 6: Crossplot comparison

Table (2) compares the simulation CPU time for both methods. SIM method provides suffucuently accurate results (although less accurate than the single step GP approach), but with a CPU reduction of a factor five.

	SIM method			Single step GP
	Optimization	Parameters modeling	Total	
CPU time	00:00:35	00:00:15	00:00:50	00:04:28

Table 2: CPU time comparison

It is worth to mention, that in this study we consider a simple model with only 3 uncertain parameters. So that, to estimate the function with GP model at every single step takes approximately 10 seconds. For more complex functions and more input uncertain parameters involved a single step model evaluation can take up to 10-20 minutes. So that, for the same simulation period of 55 years CPU time can increase to 10-20 hours. Whereas SIM approach does not depend on number of time steps and the method always conclude only a single optimization problem and as maximum 3 GP models for the transformation parameters.

6. Conclusion

This paper focuses on two general problems. First, we introduce a shape invariant model approach and we provide an efficient algorithm for estimation of transformation parameters of the model with the method specification for large sets of curves. Second, we suggest the application of this approach to model the time series outputs from a dynamic simulator. The proposed method reduces the problem of functional outputs modeling to one optimization problem and three GP response surface models. We have tested the method with a CO_2 storage reservoir case. We have also compared the method with the standard single-step approach. Presented numerical results show that the method provides satisfactory and comparable predictivity at lesser CPU time. The method also does not depend on the number of the involved time-steps. It can be very advantageous when we are working with a model involving large number of times steps such as CO_2 storage when the reservoir model simulation period can include up to hundreds or thousands time steps. However, if the set of output curves have significantly different behaviour, preliminary curves classification may be required.

7. Acknowledgements

This work has been partially supported by the French National Research Agency (ANR) through COSINUS program (project COSTA-BRAVA n ANR-09-COSI-015). I also would like to thanks Nicolas Mourand and Dan BOSSIE CODREANU for the presented CO_2 reservoir model.

References

[1] E. de Rocquigny, N. Devictor, S. Tarantola, Uncertainty in industrial practice: a guide to quantitative uncertainty management, Wiley, 2008.
[2] E. Sergienko, D. Busby, Optimal well placement for risk mitigation in co2 storage, in: 1st Sustainable Earth Sciences Conference, 2011.
[3] D. Busby, E. Sergienko, Combining probabilistic inversion and multi-objective optimization for production development under uncertainty, in: 12th European Conference in the Mathematics in Oil Recovery, 2010.
[4] S. Subbey, M. Christie, M. Sambridge, Prediction under uncertainty in reservoir modeling, Journal of Petroleum Science and Engineering 44 (1-2) (2004) 143–153.
[5] D. Busby, M. Feraille, Adaptive design of experiments for bayesian inversion an application to uncertainty quantification of a mature oil field, Journal of Physics: Conference Series 135.
[6] D. Busby, C. L. Farmer, A. Iske, Hierarchical nonlinear approximation for experimental design and statistical data fitting, SIAM J. Sci. Comput 29 (1) (2007) 49–69.
[7] M. D. McKay, R. J. Beckman, W. J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code,, Technometrics 21 (1979) 239–245.
[8] J. Sacks, W. Welch, T. Mitchell, H. Wynn, The design and analysis of computer experiments, Statistical Science 4.4 (1989) 409–435.
[9] T. Santner, B. Williams, W. Notz, The Design and Analysis of Computer Experiments, Springer Series in Statistics, New York, 2003.
[10] G. Matheron, Principles of geostatistics, Economic Geology 58 (1963) 1246–1266.
[11] S. Conti, A. OHagan, Bayesian emulation of complex multi-output and dynamic computer models, Journal of statistical planning and inference 140 (3) (2010) 640–651.
[12] P. Z. Qian, H. Wu, C. J. Wu, Gaussian process models for computer experiments with qualitative and quantitative factors, Technometrics 50 (3).
[13] S. Conti, J. P. Gosling, J. Oakley, A. O'Hagan, Gaussian process emulation of dynamic computer codes, Biometrika 96 (3) (2009) 663–676.
[14] M. Bayarri, J. Berger, J. Cafeo, G. Garcia-Donato, F. Liu, J. Palomo, R. Parthasarathy, R. Paulo, J. Sacks, D. Walsh, Computer model validation with functional output, The Annals of Statistics (2007) 1874–1906.
[15] K. Campbell, M. D. McKay, B. J. Williams, Sensitivity analysis when model outputs are functions, Reliability Engineering & System Safety 91 (10) (2006) 1468–1472.
[16] D. Higdon, J. Gattiker, B. Williams, M. Rightley, Computer model calibration using high-dimensional output, Journal of the American Statistical Association 103 (482).
[17] M. Lamboni, D. Makowski, S. Lehuger, B. Gabrielle, H. Monod, Multivariate global sensitivity analysis for dynamic crop models, Field Crops Research 113 (3) (2009) 312–320.

[18] B. Auder, A. De Crecy, B. Iooss, M. Marques, Screening and metamodeling of computer experiments with functional outputs. application to thermal–hydraulic computations, Reliability Engineering & System Safety 107 (2012) 122–131.
[19] W. Lawton, E. Sylvestre, M. Maggio, Self modeling nonlinear regression, Technometrics 14 (3) (1972) 513–532.
[20] L. C. Brumback, M. J. Lindstrom, Self modeling with flexible, random time transformations, Tech. Rep. 2 (2004).
[21] R. Izem, J. Marron, Analysis of nonlinear modes of variation for functional data, Electronic Journal of Statistics 1 (2007) 641–676.
[22] D. G. Krige, A statistical approach to some basic mine valuation problems on the Witwatersrand, Journal of the Chemical, Metallurgical and Mining Society of South Africa 52 (6) (1951) 119139.
[23] W. J. Welch, R. J. Buck, J. Sacks, H. P. Wynn, T. J. Mitchell, M. D. Morris, Screening, predicting, and computer experiments, Technometrics 34 (1) (1992) 15–25.
[24] A. I. Forrester, A. J. Keane, Recent advances in surrogate-based optimization, Progress in Aerospace Sciences 45 (1) (2009) 50–79.
[25] M. Vimond, Efficient estimation for a subclass of shape invariant models, The Annals of Statistics 38 (3) (2010) 1885–1912.
[26] F. Gamboa, J.-M. Loubes, E. Maza, Semi-parametric estimation of shifts, Electronic Journal of Statistics 1 (2007) 616–640.