Space-time Gaussian processes for the approximation of partially converged simulations

Victor Picheny, David Ginsbourger

Space-time Gaussian processes for the approximation of partially converged simulations

Victor Picheny

CERFACS, Toulouse, France.

David Ginsbourger

University of Bern, Switzerland.

Abstract. In the context of expensive numerical experiments, a promising solution to alleviate the computational costs consists of using partially converged simulations instead of exact solutions. The gain in computational time is at a price of precision in the response. This work addresses the issue of fitting a Gaussian process metamodel to partially converged simulation data, for further use in prediction and optimization. The main challenge consists in the adequate approximation of the error due to partial convergence, which is correlated in both design variables and time directions. Here, we propose to fit a Gaussian process in the joint space of design parameters and computational time. The model is constructed by building a covariance function that reflects accurately the actual structure of the error. Practical solutions are proposed to solve the learning issues associated with the model. The method is applied to a CFD simulator test-case, and shows significant improvement in prediction compared to a classical kriging model.

1. Introduction

Using computer experiments and metamodels for facilitating optimization and statistical analysis of engineering systems has become commonplace (Sacks et al. (1989); Jones et al. (1998)). However, despite the continuous growth of computational capabilities, the complexity of the simulators still drastically limit the number of available experiments, which are often insufficient to build accurate metamodels.

An efficient solution to alleviate the computational cost consists of using degraded versions of the expensive simulator to provide faster but less accurate evaluations of the simulator output. Such approximations can be obtained by using coarser mesh (in Finite Element methods), a simpler partial differential equations problem, or geometry simplification for instance. The degraded simulator is often called low-fidelity (LF) model and the expensive version high-fidelity (HF) model. Using metamodels in this context has been addressed by many authors in the literature. For instance, Alexandrov et al. (2000) and Gano et al. (2006) used metamodels to approximate the difference between LF and HF models. Kennedy and O'Hagan (2000) proposed a so-called auto-regressive model to integrate data with various fidelities. All these approaches assume that (1) a discrete (small) number of fidelities is available, and (2) LF responses are smoother than HF responses.

A less explored but promising alternative is to use partially converged simulations as a low-fidelity model, by stopping artificially the convergence of the simulator solver at early stage. Such approach has many advantages, among which the use of a single simulator instead of a different simulator for each fidelity level, and the possibility of having as many levels of

Table 1. Design variable bounds

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
Lower bound	4	15	5	5	20	9	9
Upper bound	11	45	20	11	60	60	60

accuracy as desired.

Using metamodels with such data is an open and difficult question, that differs from the classical multifidelity framework since unconverged responses are likely to be a lot rougher than converged ones, and the number of fidelity levels can be very large. In the pioneer article of Forrester et al. (2006), it is observed that all simulations within the design space tend to converge in unison, so partially converged responses are corrected using a constant shifting value and fused with fully converged responses to build a classical metamodel. Although demonstrated to be quite efficient already, this approach hinders the potential of partial convergence, since it allows the use of only two fidelity levels, and using a constant shift requires simulations to achieve a relatively high level of convergence.

This work addresses the issue of fitting a metamodel to partially converged simulation data, when convergence level potentially varies from one design to another. To do so, we propose to use a Gaussian process model in the joint space of design parameters and computational time. The model is constructed by building a covariance function that reflects accurately the actual structure of the error.

In the next section, we describe a Computational Fluid Dynamics (CFD) simulator optimization problem, which response illustrates the important behaviors of partially converged simulations. Then, we present a Gaussian process model for the joint design-time space, followed by learning issues and solutions specific to this model. Finally, the model is applied to the analysis of the CFD problem.

2. An Illustrative example: S-shaped pipe flow

To motivate our approach and highlight the important properties of partial convergence, we consider the optimization problem of an S-shaped pipe, which form is defined parametrically. A two-dimensional CFD model is built using OpenFOAM and its solver *simpleFoam* (steady-state, incompressible, turbulent flow). A constant flow velocity is imposed at the pipe input, and a null pressure at the output. The pipe contour is defined with the help of seven parameters, as shown in figure 1. The parameter bounds are given in table 1. The objective is to maximize the uniformity of the flow velocity at the end of the S-section, so the objective function (referred to as f_{SD}) is taken as the velocity standard deviation between P9 and P10.

OpenFOAM allows us to monitor the velocity field for each solver step, so we can measure the convergence directly on the objective function. First, we generate 20 designs using Latin hypercube sampling (LHS), and for each solver step, we compute f_{SD}. Figure 2 shows the evolution of the 20 designs for all time steps. Although converging to different values, all the convergence curves have similar shapes, and it seems reasonable to assume that most of the information required for prediction or optimization can be obtained before full convergence.

Now, in order to represent the data in the joint design-time space, we fix all the parameters to their nominal value but x_2 (which is the most sensitive parameter), and 100 designs are generated for x_2 values uniformly distributed between its bounds. For all designs, 500

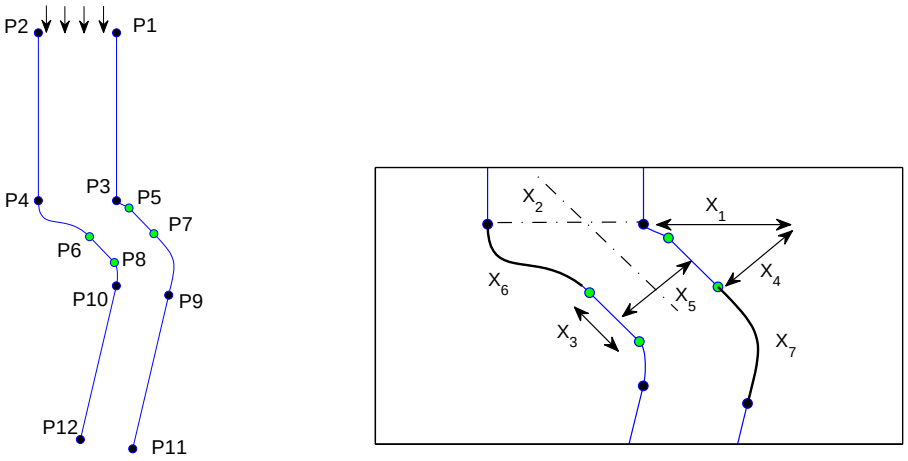

Figure 1. Contour and shape parameters of the 2D pipe model. x_2 is an angle, x_6 and x_7 define the curvatures of the Bezier curves (bold lines, right figure), x_1, x_3, x_4, x_5 are distances.

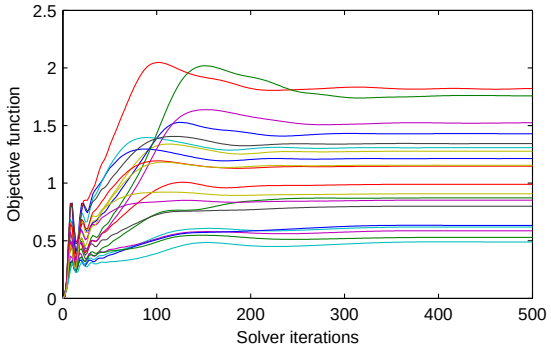

Figure 2. Response convergence for 20 designs.

solver iterations are used for convergence. Figure 3 shows three designs and their converged velocity fields, for minimum (left), mean (center) and maximum (right) values of x_2.

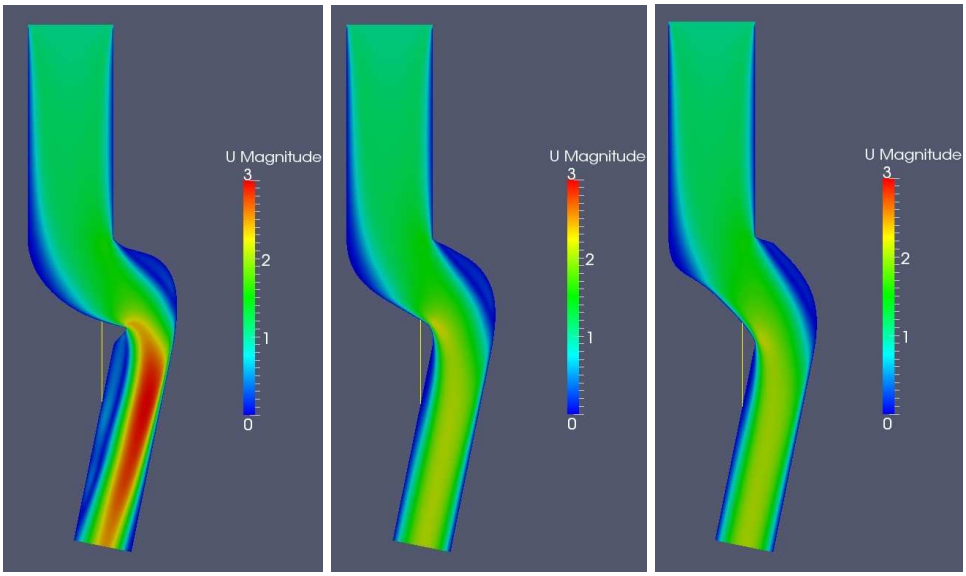

Figure 3. Three designs and velocity fields for x_2 taking its minimum (left), mean (center) and maximum values (right).

The objective function f_{SD} and the convergence error are then shown in the x_2 and time t plan (Figure 4). The convergence error is here taken as the current objective function value minus the value at step 500.

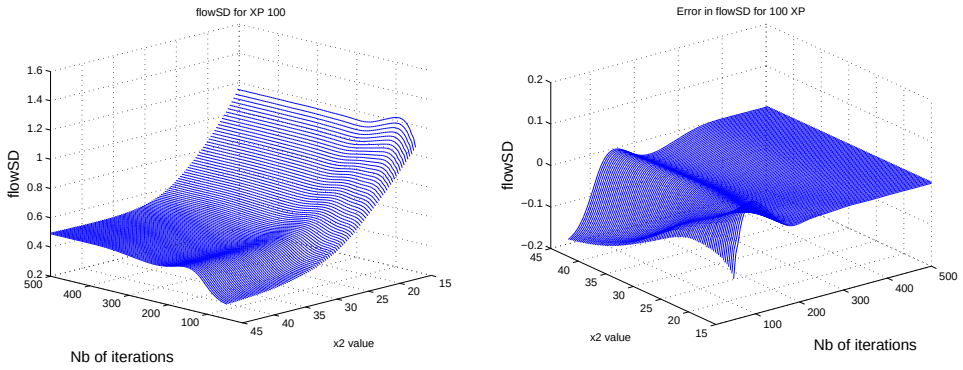

Figure 4. Evolution of objective function (left) and objective function error (right) as a function of x_2 and t. Time axis direction is inversed in the left figure to increase readability

First, we can observe that the response in smooth in both x_2 and t directions, which means that two close designs with the same number of convergence steps will have similar responses. Obviously, when t increases, the error decreases and tends towards zero, so the response becomes invariant with respect to t. One can also observe that the error

fluctuations have a higher frequency for small t than for high t. These are the three key characteristics that we want to include in our model, as we describe in the next sections.

3. A brief review of the ordinary kriging (OK) model

This section contains a brief review of the ordinary kriging model, which is used as a basis for our space-time model.

3.1. OK equations

We denote by y the response of a numerical simulator or function that is to be studied: $y : x \in D \subset \mathbb{R}^d \longrightarrow y(x) \in \mathbb{R}$. In the framework of ordinary kriging Matheron (1969), y is assumed to be a realization of a Gaussian process Y with unknown constant mean μ and stationary (location-invariant) covariance kernel. The kriging model amounts to conditioning Y on the observations $\mathbf{Y}$ evaluated at a set of input parameters $\mathbf{X} = \{\mathbf{x}^i, 1 \leq i \leq n\}$ called the design of experiments. The conditional mean and variance of Y define respectively the kriging best predictor m_{OK} and prediction variance s^2_{OK}, and are given by the following equations:

$$\begin{aligned} m_{OK}(\mathbf{x}) &= \mathbb{E}[Y(\mathbf{x})|Y(\mathbf{x}^i) = y_i, 1 \leq i \leq n] \\ &= \hat{\mu} + \mathbf{k}(\mathbf{x})^T \mathbf{K}^{-1} \left(\mathbf{Y} - \hat{\mu}\mathbf{1}\right), \end{aligned} \tag{1}$$

and

$$\begin{aligned} s^2_{OK}(\mathbf{x}) &= \mathrm{Var}[Y(\mathbf{x})|Y(\mathbf{x}^i) = y_i, 1 \leq i \leq n] \\ &= k(\mathbf{x},\mathbf{x}) - \mathbf{k}(\mathbf{x})^T \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}) + \frac{\left(1 - \mathbf{1}^T \mathbf{K}^{-1} \mathbf{k}(\mathbf{x})\right)^2}{\mathbf{1}^T \mathbf{K}^{-1} \mathbf{1}} \end{aligned} \tag{2}$$

where:

- $|$ means "conditional on",
- $\mathbf{Y} = (y_1, \ldots, y_n)^T$,
- $\mathbf{K} = \left(k(\mathbf{x}^i, \mathbf{x}^j)\right)_{1 \leq i,j \leq n}$,
- $\mathbf{k}(\mathbf{x}) = (k(\mathbf{x}, \mathbf{x}^1), \ldots, k(\mathbf{x}, \mathbf{x}^n))^T$,
- $\mathbf{1}$ is a $n \times 1$ vector of ones, and
- $\hat{\mu} = \frac{\mathbf{1}^T \mathbf{K}^{-1} \mathbf{Y}}{\mathbf{1}^T \mathbf{K}^{-1} \mathbf{1}}$ is the best linear unbiased estimate of μ.

Often times, it is assumed that the response is shifted by a linear trend instead of a constant; this is the framework of universal kriging, which is not presented here for the sake of conciseness but for which the method presented here applies without difficulty. Detailed calculations and statistical interpretation can be found in Matheron (1969), Cressie (1992) or Rasmussen and Williams (2006) for instance.

When response is observed in Gaussian, independent noise, i.e. observations are of the form $Y(\mathbf{x}^i) + \varepsilon^i$ and $cov(\varepsilon^i, \varepsilon^j) = 0, i \neq j$, equations remain valid except that a diagonal

matrix Δ has to be added to the covariance matrix $\mathbf{K}$ at every occurrence [Rasmussen and Williams (2006), pp.16-17], with terms $\Delta_{i,j} = cov(\varepsilon^i, \varepsilon^j) = \delta_{i,j} \times var(\varepsilon^i), 1 \leq i, j \leq n$. This model is often referred to as *Gaussian process regression* in machine learning. Note that this model can be easily generalized to the case where the ε^i's are correlated, Δ being non-diagonal.

3.2. Covariance function and parameter learning

In this work, the covariance function k used is the anisotropic Matern covariance with smoothness parameter $\nu = 5/2$:

$$k(\mathbf{x}, \mathbf{x}') = k_x(h_x) = \sigma^2 \left(1 + \sqrt{5}h_x + \frac{5}{3}h_x^2\right) \exp\left(-\sqrt{5}h_x\right) \tag{3}$$

where $h_x = \sqrt{\mathbf{x}^T \Sigma \mathbf{x}'}$ with $\Sigma = diag\left(\left[1/\theta_1^2, \ldots, 1/\theta_d^2\right]\right)$. The matrix Σ accounts for anisotropy in the x space.

The parameters σ^2 and $\theta_1, \ldots, \theta_d$ are often referred to as *process variance* and *ranges*, respectively. They are usually not known by the user and must be estimated based on a sample of observations. One of the most popular method to do so is the maximum likelihood estimation (MLE), which amounts to maximizing the probability density function of $\mathbf{Y}$ seen as a function of the covariance parameters:

$$L(\sigma^2, \theta_1, \ldots, \theta_d) = (2\pi)^{-\frac{n}{2}} \det(\mathbf{K})^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{Y} - \hat{\mu}\mathbf{1})^T \mathbf{K}^{-1} (\mathbf{Y} - \hat{\mu}\mathbf{1})\right) \tag{4}$$

Here, $\mathbf{K}$ can be factorized by σ^2: $\mathbf{K} = \sigma^2\mathbf{R}$ (with $\mathbf{R}$ independent of σ^2). Then, for fixed $\theta_1, \ldots, \theta_d$, the optimal σ^2 is given by:

$$\hat{\sigma}^2 = \frac{1}{n}(\mathbf{Y} - \hat{\mu}\mathbf{1})^T \mathbf{R}^{-1} (\mathbf{Y} - \hat{\mu}\mathbf{1}) \tag{5}$$

By injecting this quantity into equation 4 and applying a logarithmic transformation, the MLE problem simplifies to the minimization of the so-called *concentrated log-likelihood* with respect to the range parameters only:

$$\min_{\theta_1, \ldots, \theta_d} n \log\left(\frac{1}{n}(\mathbf{Y} - \hat{\mu}\mathbf{1})^T \mathbf{R}^{-1} (\mathbf{Y} - \hat{\mu}\mathbf{1})\right) + \log(\det(\mathbf{R})), \tag{6}$$

the MLE of σ^2 being computed afterwards using equation 5.

4. A Gaussian process surrogate for partially converged simulations

The Ordinary Kriging model presented in the previous section relies on a set of assumptions, in particular the stationarity of the response y, that are - approximately - met in many computer experiments situations. Here, the particular behavior of the response strongly violates some of these assumptions. This section presents a model based, like Ordinary Kriging, on Gaussian process conditioning, that fits adequately partially converged responses.

4.1. Desired properties

When partial convergence is considered, an observation y_i is defined by both input parameters $\mathbf{x} \in D$ and computational time $t \in R^{+*}$ (typically equal or proportional to the number of solver iterations). In order to predict such types of responses, Gaussian processes are particularly adapted since they allow the definition of models that can inherit the structure of the function to approximate.

Indeed, we consider that the observed function is a realization of a random process $Y(\mathbf{x}, t)$, which is the sum of a process F independent of t, and a process G that depends on both $\mathbf{x}$ and t:

$$Y(\mathbf{x}, t) = F(\mathbf{x}) + G(\mathbf{x}, t) \tag{7}$$

F is the response given by the simulator with complete convergence, and then can be modeled with the usual assumptions in computer experiments Sacks et al. (1989): stationarity, ergodicity, etc. (as for a kriging model in a classical framework). G is the error term due to partial convergence, and has a more complex structure. Under the hypothesis of independence between F and G, the kernel k_Y of Y writes simply as the sum of the kernels of F and G, so all the modeling difficulty lies in the characterization of the convergence error G.

In the $\mathbf{x}$ space, it can be observed (Figure 4) that two runs with close sets of input parameters converge in a similar fashion, hence their convergence errors are correlated. In the t direction, except for the first few iterations that often show large oscillations, the convergence is smooth so the responses evaluated at successive time steps are also correlated. In addition, the convergence error tends to zero when the computational time increases. It can be assumed reasonably that the error variance decreases monotonically with computational time, which makes G instationary in the t direction. The speed of convergence may differ slightly from one design to another, but assuming this speed constant seems reasonable here. Finally, one can observe that the oscillation frequency of the error tends to decrease with time, which is another instationary behavior in the t direction.

4.2. Modifying usual covariance functions

Most usual covariance functions in the kriging framework are stationary (i.e. $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}')$), hence are not suitable for our problem. However, lots of possibilities exist to modify usual kernels to make new ones with the desirable properties, see Rasmussen and Williams (2006) (chapter 4, pp.94-95) for a detailed discussion. In particular, we use here the three following properties:

- given two kernels $k_1(\mathbf{x}, \mathbf{x}')$ and $k_2(\mathbf{x}, \mathbf{x}')$, their sum and their product are a kernel:

$$k_3(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}'), \qquad k_4(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') \times k_2(\mathbf{x}, \mathbf{x}')$$

- given any function $a : D \to D$, then its composition with the kernel is a kernel:

$$k_5(\mathbf{x}, \mathbf{x}') = k\left(a(\mathbf{x}), a(\mathbf{x}')\right)$$

Proofs are direct by verifying that the following property is met:

A kernel k on $D \times D$ is positive definite if and only if it is symmetric ($k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$ for all $\mathbf{x}, \mathbf{x}' \in D$) and for all $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in D$ ($n \in \mathbb{N}$) and all $\{a_1, \ldots, a_n\} \in \mathbb{R}$:

$$\sum_{i,j=1}^{n} a_i a_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \tag{8}$$

4.3. A covariance function for partial convergence

Recall that G is a process autocorrelated in $\mathbf{x}$ and t with decreasing amplitude and increasing smoothness when t increases. To account for the decreasing amplitude, we propose to use a covariance of the form:

$$k_G(\mathbf{u}, \mathbf{u}') = \sigma^2(t, t') r_G(\mathbf{x}, \mathbf{x}', t, t') \tag{9}$$

where $\mathbf{u} = (\mathbf{x}, t)$, r_G is a correlation function and $\sigma^2(t, t')$ is a decreasing function of t and t'. To ensure that k_G is a positive definite kernel, it is sufficient to choose $\sigma^2(t, t')$ as a covariance function. Since G tends to zero, its variance $\sigma^2(t, t')$ should be null when $t \to +\infty$. Here we choose a decreasing exponential form for the variance:

$$\sigma^2(t, t') = \sigma_G^2 \exp(-\alpha \frac{t + t'}{2}), \tag{10}$$

with $\alpha \in \mathbb{R}_+^*$ a parameter that accounts for the convergence speed. Another choice, among many, could be: $\sigma^2(t, t') = \frac{\sigma_G^2}{(t+t')^\alpha}$.

Although not necessary, it is convenient to choose a separable function for r_G:

$$r_G(\mathbf{u}, \mathbf{u}') = r_{Gx}(\mathbf{x}, \mathbf{x}') \times r_{Gt}(t, t'), \tag{11}$$

which allows to handle different regularities in $\mathbf{x}$ and t directions.

The correlation r_{Gx} can be taken as stationary, i.e. $r_{Gx}(\mathbf{x}, \mathbf{x}') = r_{Gx}(|\mathbf{x} - \mathbf{x}'|)$, for instance, the Matern 5/2 function of equation 3.

The correlation r_{Gt} has to account for the increasing smoothness of the error (high oscillations for the first steps, then smooth convergence). To do so, we propose to use a classical covariance (for instance the Matern 5/2 function) and define for a distance depending on time, for instance:

$$h_t = \frac{|t - t'|}{\theta(t, t')} = \frac{|t - t'|}{\theta_0 + \frac{\Delta_\theta}{2}(t + t')}, \tag{12}$$

with $\theta_0, \Delta_\theta \in R^+$.

Finally, the kernel of the process Y is the sum of the kernels of F and G, assuming that they are independent of each other:

$$k_Y(\mathbf{u}, \mathbf{u}') = k_F(\mathbf{x}, \mathbf{x}') + k_G(\mathbf{u}, \mathbf{u}') \tag{13}$$

Using this kernel, we are able to perform simulation, conditional simulation, hence learning with Gaussian processes.

Let $\mathbf{Y}_n = [y_1, \ldots, y_n]^T$ be a set of observations, $\mathbf{X}$ the matrix of design parameters, $\mathbf{T}$ the vector of times and $\mathbf{U} = [\mathbf{X}, \mathbf{T}]$ the experimental matrix. In the fashion of Ordinary Kriging, the mean and variance of Y at $\mathbf{u}^* = (\mathbf{x}^*, t^*)$ conditional on the observations $\mathbf{Y}$ are given by:

$$m(\mathbf{u}^*) = \hat{\mu} + \mathbf{k}_Y(\mathbf{u}^*)^T \mathbf{K}_Y^{-1} (\mathbf{Y} - \hat{\mu}\mathbf{1}) \tag{14}$$

$$s^2(\mathbf{u}^*) = k_Y(\mathbf{u}^*, \mathbf{u}^*) - \mathbf{k}_Y(\mathbf{u}^*)^T \mathbf{K}_Y^{-1} \mathbf{k}_Y(\mathbf{u}^*) + \frac{\left(1 - \mathbf{1}^T \mathbf{K}_Y^{-1} \mathbf{k}_Y(\mathbf{x})\right)^2}{\mathbf{1}^T \mathbf{K}_Y^{-1} \mathbf{1}} \tag{15}$$

with: $\mathbf{K}_{Y\,i,j} = k_Y(\mathbf{u}_i, \mathbf{u}_j)$ and $\mathbf{k}_Y = [k_Y(\mathbf{u}^*, \mathbf{u}_1) \ldots k_Y(\mathbf{u}^*, \mathbf{u}_n)]$.

The functions $m(.)$ and $s^2(.)$ define the Gaussian process model, which provides a prediction mean and variance for any given design with convergence level. As for the Ordinary Kriging model, m is equal to the observations and s is equal to zero at the points of the DOE.

In most applications, in particular for optimization, the value of interest is the actual response, i.e. the asymptotic value for $t = +\infty$. From equation 13, the covariance $k_Y(\mathbf{u}, \mathbf{u}^*)$ is defined for $\mathbf{u}^* = (\mathbf{x}^*, +\infty)$ and is simply equal to $k_F(\mathbf{x}, \mathbf{x}^*)$ (indeed, $\sigma_G^2(+\infty, .) = 0$ which implies $k_G = 0$). Then, we can define an asymptotic prediction independent of t, equal to:

$$m_\infty(\mathbf{x}^*) = \hat{\mu} + \mathbf{k}_F(\mathbf{x}^*)^T \mathbf{K}_Y^{-1} (\mathbf{Y} - \hat{\mu}\mathbf{1}) \quad (16)$$

$$s_\infty^2(\mathbf{x}^*) = \sigma_F^2 - \mathbf{k}_F(\mathbf{x}^*)^T \mathbf{K}_Y^{-1} \mathbf{k}_F(\mathbf{x}^*) + \frac{\left(1 - \mathbf{1}^T \mathbf{K}_Y^{-1} \mathbf{k}_F(\mathbf{x})\right)^2}{\mathbf{1}^T \mathbf{K}_Y^{-1} \mathbf{1}} \quad (17)$$

One can notice that these equations take the form of an Ordinary Kriging with correlated residuals, since $\mathbf{K}_Y = \mathbf{K}_F + \mathbf{K}_G$, $\mathbf{K}_G$ playing the role of Δ in section 3.

4.4. Discussion

4.4.1. Comparison with co-kriging

One might prefer to limit the responses to two (or a few) convergence levels only, as in Forrester et al. (2006). In that case, the data is similar in form to a multi-fidelity framework, for which the co-kriging model Kennedy and O'Hagan (2000) has been proved to be an efficient tool for prediction and optimization.

When t levels of response are considered, the co-kriging model assumes that the more accurate response Z_t is equal to the less accurate response Z_{t-1} multiplied by a scaling factor ρ_{t-1} plus a stationary Gaussian process independent of Z_t and Z_{t-1}:

$$Z_t(\mathbf{x}) = \rho Z_{t-1}(\mathbf{x}) + \delta_t(\mathbf{x}) \quad (18)$$

In the framework of this paper, we have $Z_t(\mathbf{x}) = F(\mathbf{x}) + G(\mathbf{x}, t)$ and $Z_{t-1}(\mathbf{x}) = F(\mathbf{x}) + G(\mathbf{x}, t-1)$. Hence, the two models differ for two reasons. First, by the scaling factor ρ: this factor is intuitive in a multi-fidelity framework, since data may come from different simulators, so they are different in nature and may have different amplitudes. This behavior is not so clear with partial convergence.

The second difference is the co-kriging assumption of independence of the differences between two fidelity levels: $cov\left(\delta_{t1}(\mathbf{x}), \delta_{t2}(\mathbf{x})\right) = 0, t_1 \neq t_2$. This would imply that $G(\mathbf{x}, t_1)$ is independent of $G(\mathbf{x}, t_2)$, meaning that the convergence error is correlated in the x-direction but not in the time direction, which is obviously false from figure 2. Co-kriging might apply to partial convergence only if the convergence times $t_1, t_2, \ldots$ are sparse enough so the hypothesis of independence in the time direction holds.

4.4.2. Monte-Carlo convergence

The space-time model allows us to deal with a framework closely related to partial convergence that is typical of robust design for instance: an observation is computed by averaging an arbitrary number t_i of independent drawings (or repeated experiments):

$$\tilde{Y}_i = \frac{1}{t_i} \sum_{j=1}^{t_i} F(\mathbf{x}_i) + \varepsilon_{i,j}, \quad (19)$$

when $F(\mathbf{x})$ is the function of interest, observed with noise $\varepsilon_{i,j} \sim \mathcal{N}(0, \tau^2)$. We have then $\widetilde{Y}_i \sim \mathcal{N}\left(F(\mathbf{x}_i), \frac{\tau^2}{t_i}\right)$. F is observed exactly for $n_i \to +\infty$, and the process error G is equal to:

$$G(\mathbf{x}^i, t^i) = \frac{1}{t_i} \sum_{j=1}^{t_i} \varepsilon_{i,j}, \quad t_i \leq n_i \tag{20}$$

In a classical framework, one would only use the observation $\tilde{Y}_i$ and build a kriging with noisy observations by adding diagonal terms $\frac{\tau^2}{n_i}$ to the covariance matrix, as explained in section 3. In contrast, the space-time model presented here takes the whole trajectory of G into account, that is $\{G(\mathbf{x}^i, 1), \ldots, G(\mathbf{x}^i, t_i)\}$. One may wonder if this adds any helpful information for prediction. We show below that the two models are actually equivalent, due to the Markovian property of G here.

Indeed, since all $\varepsilon_{i,j}$ are uncorrelated, the covariance of G is null in the x direction:

$$cov\left(G(\mathbf{x}^i, t^i), G(\mathbf{x}^j, t^j)\right) = 0 \text{ for any } \mathbf{x}^i \neq \mathbf{x}^j$$

For a given trajectory (fixed $\mathbf{x}^i$, $t_i^{(1)}, t_i^{(2)} \leq t_i$), it is easy to find that we have:

$$cov\left(G(\mathbf{x}^i, t_i^{(1)}), G(\mathbf{x}^i, t_i^{(2)})\right) = \frac{\tau^2}{max(t_i^{(1)}, t_i^{(2)})} = \tau^2 \frac{min(t_i^{(1)}, t_i^{(2)})}{t_i^{(1)} t_i^{(2)}} \tag{21}$$

So the kernel of G is:

$$k_G((\mathbf{u}^i, \mathbf{u}^j)) = \frac{\tau^2}{t_i^{(p)} t_j^{(q)}} min(t_i^{(p)}, t_j^{(q)}) \delta_{\mathbf{x}^i = \mathbf{x}^j} \tag{22}$$

where $\mathbf{u}^i = \{\mathbf{x}^i, t_i^{(p)}\}$ and $\mathbf{u}^j = \{\mathbf{x}^j, t_j^{(q)}\}$, $1 \leq t_i^{(p)} \leq t_i$, $1 \leq t_j^{(q)} \leq t_j$.

With such kernel, we show that given a (space-time) model conditioned on the observations $\tilde{Y}_i$ (as defined in 19), adding any $Y(u)$ with $u = (\mathbf{x}^i, t_u)$ for $i \in \{1, \ldots, n\}$ and $t_u \leq t_i$ has no effect on the model. This property can be seen as a *screening effect* [Stein (2002)] in the time dimension. The proof is given in appendix. Note that this effect is true only when the covariance is markovian in the time direction and null in the x direction.

Hence, in this case the space-time model coincides with a kriging with noisy observations, so taking into account the convergence trajectories is useless. The use of space-time models makes sense only when the convergence path is not markovian or when the errors are correlated in the x direction.

5. Learning model parameters

In Ordinary Kriging, the covariance parameters are most of the time learned using an optimization process, for instance by maximizing the likelihood of the observations, or by minimizing the cross-validation error. This step is particularly critical for the accuracy of the kriging model, and is known to be difficult, in particular when the number of observations is small and the number of parameters large.

Our model requires the knowledge of the parameters of the covariance function of k_Y. Assuming anisotropy in the $\mathbf{x}$ space and Matern 5/2 shape for all covariances, we have:

- for the stationary covariance k_F: $d+1$ parameters, σ_F^2, $\theta_F^1, \ldots, \theta_F^d$,

- for the stationary correlation r_{Gx}: d parameters, $\theta_G^1, \ldots, \theta_G^d$,
- for the correlation r_{Gt}: two parameters, θ_0 and Δ_θ,
- for the process variance σ^2: two parameters, σ_G^2 and α.

Learning these $2d+5$ parameters in a single optimization loop seems unrealistic here, since the objective function is likely to be highly multimodal, and ensuring a good exploration may be too expensive computationally.
Besides, with partial convergence, the design of experiments takes a particular form, which can be used to simplify the learning process. Indeed, when an observation is made at $\mathbf{x}$ with time t, the response can be calculated without any computational effort for all the time steps smaller than t. In other words, one has access to the response convergence for the design x from one to t: $\{y(\mathbf{x},1), y(\mathbf{x},2), \ldots, y(\mathbf{x},t)\}$. In the following, we refer to a series of data for the same $\mathbf{x}$ and increasing t as *response* (or *error*) *trajectory*.
Then, we propose to decompose the kernel parameters learning in two steps: first, we learn the parameters related to time only, and then the parameters related to $\mathbf{x}$.

5.1. Learning time parameters

The process variance function accounts for the convergence speed of the simulator (the variance of the error due to partial convergence). This speed might differ from one design to another, especially if the design space is large, but it is reasonable to consider speed as uniform, and then learn it from a small number of simulations.
We assume here that the user has performed a small number K of fully converged simulations ($3 \leq K \leq 10$, typically), well spread in the design space. Let N be the number of steps required for full convergence, we have then an initial set of $K \times N$ observations:

$$\{y(\mathbf{x}_1, t_1), \ldots, y(\mathbf{x}_1, t_N), \ldots, y(\mathbf{x}_K, t_1), \ldots, y(\mathbf{x}_K, t_N)\}.$$

The error trajectories can be known exactly by subtracting the converged responses to the partially converged response trajectories: $g(\mathbf{x}_i, t_j) = y(\mathbf{x}_i, t_j) - y(\mathbf{x}_i, t_N)$. We have then realizations of the process G for K designs and N times:

$$g(\mathbf{x}_1, t_1), \ldots, g(\mathbf{x}_1, t_N), \ldots, g(\mathbf{x}_K, t_1), \ldots, g(\mathbf{x}_K, t_N).$$

We assume then that the correlation in $\mathbf{x}$ is null, which is reasonable considering that K is very small and the observations are away one from each other. In that case, the kernel of G is:

$$k_G((\mathbf{u}^i, \mathbf{u}^j)) = \sigma^2(t_i, t_j) r_{Gt}(t_i, t_j) \delta_{\mathbf{x}^i = \mathbf{x}^j} \tag{23}$$

The parameters σ_G^2, θ_0, Δ_θ and α can then be estimated by maximum likelihood, i.e. by solving:

$$\min_{\left\{\sigma_G^2, \theta_0, \Delta_\theta, \alpha\right\}} l = \log\det \mathbf{K}_G + \mathbf{g}^T \mathbf{K}_G^{-1} \mathbf{g} \tag{24}$$

As for Ordinary Kriging, the covariance matrix can be factorized by σ_G^2: $\mathbf{K}_G = \sigma_G^2 \mathbf{R}_G$, so the concentrated log-likelihood can be used:

$$\left\{\hat{\theta}_0, \hat{\Delta}_\theta, \hat{\alpha}\right\} = \arg\min \left[n \log\left(\frac{1}{n} \mathbf{g}^T \mathbf{R}_G^{-1} \mathbf{g} \right) + \log\left(\det\left(\mathbf{R}_G\right)\right) \right] \tag{25}$$

$$\hat{\sigma}_G^2 = \mathbf{g}^T \mathbf{R}_G^{-1} \mathbf{g} \tag{26}$$

This problem is only three-dimensional, which makes it easy to solve. Moreover, computing the concentrated log-likelihood is here facilitated by the fact that $\mathbf{R}_G$ is block-diagonal (see section 6).

5.2. Learning x-space parameters

Once the time-related parameters are estimated, the remaining unknown parameters are related to the covariance of F (σ_F^2, $\theta_F^1, \ldots, \theta_F^d$) and the correlation r_x of G ($\theta_G^1, \ldots, \theta_G^d$). The direct optimization of the log-likelihood may be overly challenging, especially if d is large. In order to reduce the problem dimension, we assume that F and G share the same anisotropy, i.e. the respective influence of the parameters will be the same for the actual process and the error. Thus, we set:

$$\theta_G^i = \rho\theta_F^i, \quad 1 \leq i \leq d \tag{27}$$

with ρ a factor of proportionality.

The number of parameters is then reduced to $d+2$, which makes it feasible to use MLE, hence solving the problem:

$$\left\{\hat{\sigma}_F^2, \hat{\theta}_F^1, \ldots, \hat{\theta}_F^d, \hat{\rho}\right\} = \arg\min \left(\log\det \mathbf{K}_Y + (\mathbf{Y} - \hat{\mu}\mathbf{1})^T \mathbf{K}_Y^{-1} (\mathbf{Y} - \hat{\mu}\mathbf{1})\right) \tag{28}$$

Note that here, the matrix $\mathbf{K}_Y$ cannot be factorized by σ_F^2, so concentrated log-likelihood cannot be used to estimate σ_F^2 separately.

6. Numerical issues

The major numerical issue with partial convergence comes from the huge amount of data available. The covariances matrices used either for parameter learning or prediction are of very large size, and their inversion can be at the same time computationally intensive and subject to numerical instabilities.

A first numerical trick to facilitate the inversion, well-known of kriging users, consists of adding a small diagonal matrix (nugget) to the covariance matrix, which amounts to relaxing the constraint of exactly interpolating the data. Here, since the diagonal of $\mathbf{K}_G$ is not constant and typically shows variations of several orders of magnitude, it is preferable to add a value proportional to the diagonal term, for instance $10^{-4} \times \sigma^2(t_i, t_i)$. Thus, the relaxation is similar for all the data points.

Another natural option to reduce the computational cost is to use only a subset of the available data. This solution is discussed separately in the parameter learning and prediction situations.

6.1. Data reduction for parameter estimation

The first step of parameter estimation is to use a small number K of error trajectories to estimate the time-related parameters. Since the trajectories are assumed to be independent of each other, the matrix $\mathbf{K}_G$ is block diagonal: $\mathbf{K}_G = diag\left(\mathbf{K}_G^1, \ldots, \mathbf{K}_G^N\right)$. Then, we have:

$$\begin{aligned} \log\left(\det\left(\mathbf{K}_G\right)\right) &= \prod_{k=1}^{K} \log\left(\det\left(\mathbf{K}_G^k\right)\right) \\ \mathbf{K}_G^{-1} &= diag\left((\mathbf{K}_G^1)^{-1}, \ldots, (\mathbf{K}_G^N)^{-1}\right) \end{aligned}$$

The matrices $\mathbf{K}_G^i$ are of size $N \times N$ (N being the number of steps required to achieve full convergence). N typically varies from hundreds (as for the application presented here) to thousands for complex simulations. If it is too large, one must use a subset of the data only. Regular subsets (one observation every p steps) may ensure a better inference of the error decrease rate (parameters α and σ_G), but this is at the price of the regularity information (local smoothness), which may impact the estimation of θ_0 and Δ_θ. Irregular sub-sampling may offer the best trade-off.

When estimating the parameters related to the $\mathbf{x}$ space, using a subset of the data seems particularly necessary since the inversion of $\mathbf{K}_Y$ is embedded in an optimization loop and is likely to be calculated numerous times. The question is then to choose the subset that will provide the most information about k_G and k_F. Choosing the last point of each trajectory seems obvious since these points provide the most information on F. In addition, the subset should favor data with equal times (i.e. alignments in the t direction), since they are the points with highest correlation value across trajectories.

6.2. Data reduction for prediction

It is well-known that the classical kriging predictor at a location $\mathbf{x}^*$ is mainly determined by the few observations nearest to the prediction point, so that a kriging based only on these neighbor observations provides the same predictor (and prediction variance) than the kriging with all the observations. This phenomenon is often called *screening effect* [Cressie (1992); Stein (2002)], and is used to compute fast predictions in the case of large data sets. Data selection is typically performed by building a hyper-rectangle (or ellipsoid) in the $\mathbf{x}$ space, centered on the prediction point.

The definition of neighborhood in our context is not straightforward for the asymptotic prediction, i.e. prediction of the actual response F. Indeed, with the convention $t = +\infty$ for asymptotic prediction, all the observations are equally far away from the prediction point in the time space.

A simple conservative approach consists of selecting all the data for which $k_F(\mathbf{x}^*, \mathbf{x}^i)$ is higher than a certain level, or equivalently, define the neighborhood of $\mathbf{x}^*$ as:

$$\Omega = \{\mathbf{x} \in D | \frac{1}{\sigma_F^2} k_F(\mathbf{x}^i, \mathbf{x}^*) > \beta\} \tag{29}$$

for some level $0 \leq \beta \leq 1$. This ensures (see equations 14 and 15) that all the influent observations are taken into account, but may select a lot more observations that what is actually necessary.

Indeed, as we noticed before, the last term of a trajectory (corresponding to the highest computational time) is the one that contains the most information for asymptotic prediction. However, since the trajectories are not Markovian, the other terms also have an influence on the prediction. In particular, the very last terms provide (seeing it as finite differences) the derivative information of G in the t direction.

Hence, we propose as a rule of thumb to choose the last three observations of each trajectory in Ω as our subset for asymptotic prediction.

7. Application to the pipe flow example

In this section, we illustrate the learning steps of the previous section applied to the data on the CFD example. For this analysis, two data sets are generated, one for learning and

the other for testing. Both are based on 200-point LHS designs with *maximin* criterion. Some combination of parameters lead to unfeasible configurations (detected at the meshing stage) and are removed from the data sets (14 points for the learning set and 18 for the test set).

7.1. Learning time parameters

Four points, randomly chosen in the first LHS, are used to generate fully converged runs (with 500 steps), from which we extract the corresponding error trajectories. For each trajectory, the first 50 steps are removed since the convergence behavior is non-smooth. By construction (see section 5.1), the last term of each error trajectory is zero, which is slightly incorrect (the actual error is of the order of the solver tolerance). To avoid bias, the last 20 steps are also removed. The corresponding data (1720 error values) is represented in Figure 6 (left).

Then, the MLE is performed using the full dataset on a $32 \times 32 \times 32$ grid. We found $\hat{\sigma}_G^2 = 0.57966$, $\hat{\alpha} = 0.0172$, $\hat{\theta_T} = 62$ and $\hat{\Delta}_\theta = 1/60$. Figure 5 shows the concentrated likelihood in the α-θ_T direction at optimal Δ_θ; the optimization problem is here unimodal and the optimal values are well-defined.

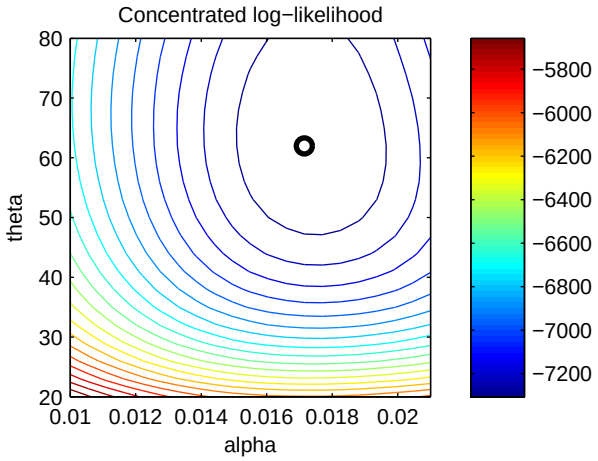

Figure 5. Square error and variance model.

To validate visually the values of $\hat{\sigma}_G^2$ and $\hat{\alpha}$, we draw in Figure 6 (right) the error trajectories divided by $\sigma^2(t,t)$. As we can see, the four trajectories can now be considered in first approximation as stationary with variance equal to one. The amplitude of the curves seems however non-constant, which indicates that the model might be improved by considering a process variance that also depends on $\mathbf{x}$. However, this may make the learning problem very difficult to solve.

In order to illustrate the model, we represent the actual error trajectory of a new design (randomly chosen), and the associated Gaussian process model based on 20 observations of this trajectory, uniformly chosen between $t = 0$ and $t = 500$. The trajectory and GP model (mean and 95% confidence interval) are shown in Figure 7. Note that such kind of data is not realistic, since in a real case the response would be known for all the intermediate time steps, but the shape of the GP mean and confidence interval reflects the accuracy of the

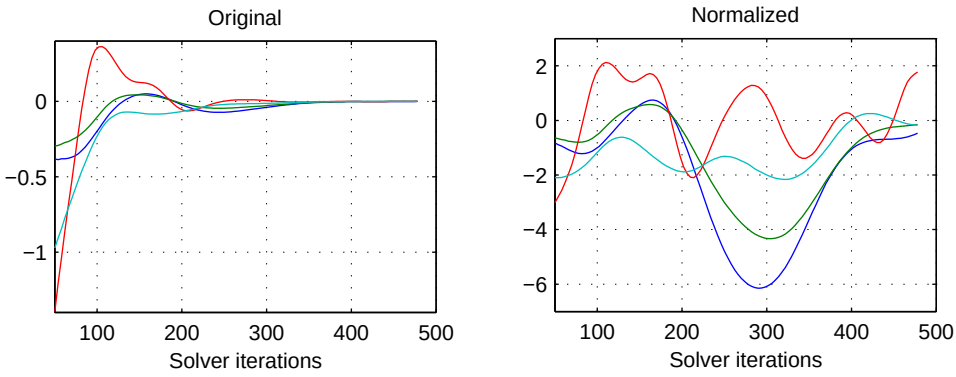

Figure 6. Original error trajectories and rescaled trajectories using estimated parameters.

model. Here, the smoothness of the model mean is similar to the one of the actual process, except for the very first time steps, where it shows very high variability. The confidence intervals are also quite realistic, and account for the fact that the process becomes flatter for large t.

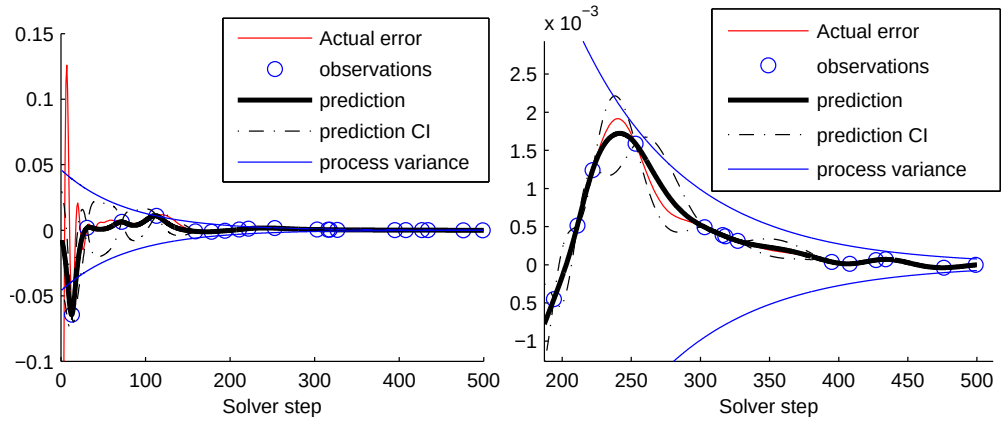

Figure 7. Example of error trajectory approximation using a GP model (left: complete trajectory, right: detail).

7.2. Learning design parameters

Now, for the remaining 182 designs of the learning DOE, partially converged simulations are run. 45 designs use the minimum convergence level (50 steps), another 45 use 60 steps, the other use random values between 50 and 500. With such setup, the DOE consists of four fully converged observations, one half of very inexpensive observations that ensures a good space filling, and the other half of heterogeneously converged observations. The total number of steps is equal to 18,500, which is the computational resource required to run 37 fully converged simulations.

The estimated parameters by maximum likelihood are:

- $\hat{\sigma}_F^2 = 0.51375$
- $\hat{\theta}_{Fx} = [2 \quad 0.97649 \quad 1.3783 \quad 1.4784 \quad 0.50908 \quad 2 \quad 1.2162]$
- $\hat{\rho} = 0.52272$ (meaning that F is smoother than G in the x direction)

7.3. 7D analysis and comparison to Ordinary Kriging

Here, we compare our model to two versions of Ordinary Kriging:

- Ordinary (interpolating) Kriging based on 37 fully converged simulations,
- Ordinary (regressing) Kriging based based on the 186 partially converged simulations

The first model corresponds to the standard situation (full convergence, interpolating model) and the number of observations is chosen so that the computational budget (i.e. total number of solver iterations) is equal to the budget of the partially converged DOE. The 37 points are chosen as a subset of the initial LHS using a maximin criterion to ensure a good space-filling.

The second model is also standard and corresponds to a simplified error model: all the errors are treated as gaussian, centered and independent of each other. The diagonal matrix that accounts for the error variances is taken as $\Delta = diag([\sigma_G^2(t_1, t_1), \ldots, \sigma_G^2(t_n, t_n)])$, that is, the error variances given by the space-time model. We can then measure by how much we gain in prediction by using a complex error model.

For all models, the same parameters $\hat{\sigma}_F^2$ and $\hat{\theta}_{Fx}$ are used, so the differences are only due to the model structures and the design of experiments. The question of parameter estimation is left appart here, since for the first model estimating an anisotropic model (eight parameters) is very challenging and would lead to a huge variability in the results.

The predicting performances are given in figure 8 and table 2. The histograms represent the differences between the model means and the actual converged values, from which is also computed the RMSE (root mean square error) statistic. In addition, the 95% confidence intervals are drawn in order to visualize if the model uncertainty reflects the reality. To assess the global uncertainty of each model, the average prediction variance at test points (referred to as integrated mean square error [IMSE], which is the classical terminology in computer experiments [Sacks et al. (1989)]) and the maximum prediction variance (maxMSE) are computed.

For the space-time model, ten actual values are outside the interval, which shows a very good calibration of the prediction variance (since 5% of the data is expected to be outside the interval). The Ordinary Kriging with partially converged data is on the contrary over-confident, since almost half of the data is outside the interval. Inversely, the Ordinary Kriging with fully converged data seems slightly over-conservative (5 data outside the intervals). The IMSE values confirm that the predicted uncertainty is a lot higher with 37 observations than with the space-time model.

The RMSE errors show that assuming that the errors are gaussian, centered and independent of each other leads to a very poor model. The very high RMSE value is due to a strong bias in the model, in particular the high values of the actual function are most of the time underestimated (central figure of figure 8. In comparison, using only fully converged simulations lead to a safer and more accurate model. The space-time model offers here the best results in terms of RMSE.

Table 2. Prediction statistics of the three models

Model	*RMSE*	*IMSE*	maxMSE
Space-Time	0.129	0.0161	0.0532
Ordinary Kriging with 186 observations	0.398	0.0206	0.0649
Ordinary Kriging with 37 observations	0.195	0.0492	0.1705

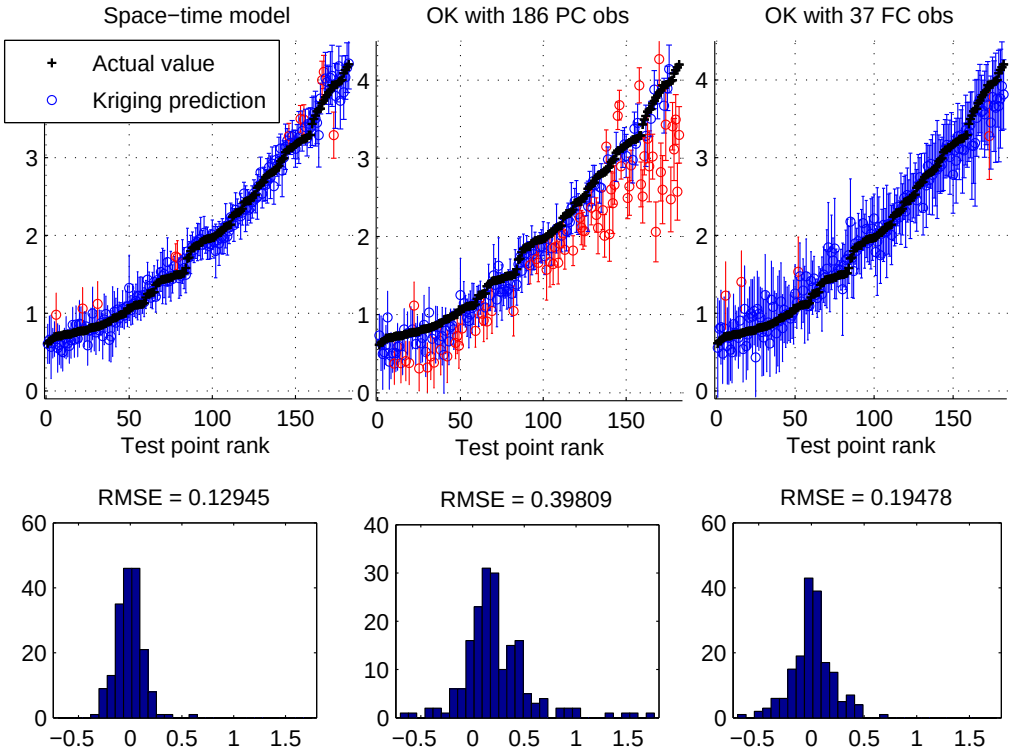

Figure 8. Comparison of the predicting capacity of the space-time model and two Ordinary Kriging models. The top figures show the actual responses values along with the predictions, represented by the mean (circle) and ± 1.96 times the standard deviation (errorbars). The 182 test points are ranked by their response value. Red errorbars indicate points where the actual value is outside the kriging 95% interval.

7.4. Optimal design of experiments for prediction

We have observed in the previous section that the average prediction variance was a lot smaller using partially converged simulations that using fully converged ones. In other words, the model was more accurate when spreading the budget into the 186 simulations instead of concentrating it on 37.

Finding the most efficient design of experiments for both learning parameters and prediction is already a challenging question with classical kriging models and seems an unreachable objective. However, it is possible to see if there exists an optimal trade-off between the number of observations and their precision, for a model with known parameters and given a fixed computational budget.

Here, we use the parameters values obtained previously, but we replace the existing DOE by a subset of the 186-point LHS with constant convergence level. The total budget is taken as $18,500$, so the number of observations varies between 37 (with 500 steps for each simulation) and 186 (with 100 steps for each). The RMSE, IMSE and maxMSE metrics are computed in each case. Note that contrarily to the RMSE, the IMSE and maxMSE do not depend on the observation values and can be computed off-line, so an optimal strategy for those criteria can be found before running any simulation (assuming that the parameters are known).

For each configuration, 20 subsets are taken randomly from the initial LHS. The results are presented in the form of boxplots in figure 9.

The boxplots clearly show that some sampling strategies are better than others. In terms of maximum prediction variance, using limited convergence and more simulations is more efficient, with optimal values for 124 or 136 steps (137, 147 simulations). The maxMSE is very sensitive to holes in the design space, so using a large number of observations allows a better coverage of the design space. However, when the number of observations becomes to high (here 186), the response uncertainty overcomes this advantage.

Similarly, the IMSE shows that there is an optimal trade-off, here situated at 160-172 steps and 107-117 observations. This trade-off is different from the one for the maxMSE criterion. For the error in the model mean, a trade-off again appears, but favors more accurate simulations. Those trade-off actually depend on the total budget: figure 10 show the IMSE values obtained for a budget of 9,250 steps: here, the optimal number of steps is 97, while (almost) full convergence is the worst option.

The difference between the IMSE and RMSE results indicate that theoretical criteria may not be the perfect tool to choose experiments, since they do not take into account any modeling error, which can be significant. Hence, a good alternative may be to choose a majority of simulations with optimal convergence level, and complete this design with a couple of simulations with heterogeneous convergence level.

8. Conclusion

In this paper, we explored the possibility of using partially converged simulations for learning and optimizing expensive-to-evaluate computer codes. We have proposed to use Gaussian processes to approximate the simulator response in the joint design-time space. The main idea was to build a covariance kernel that reflects accurately the actual structure of the response considered: the observed response was modeled as the sum of a stationary process depending on design parameters only and an error process which variance decreases towards zero when time tends to infinity.

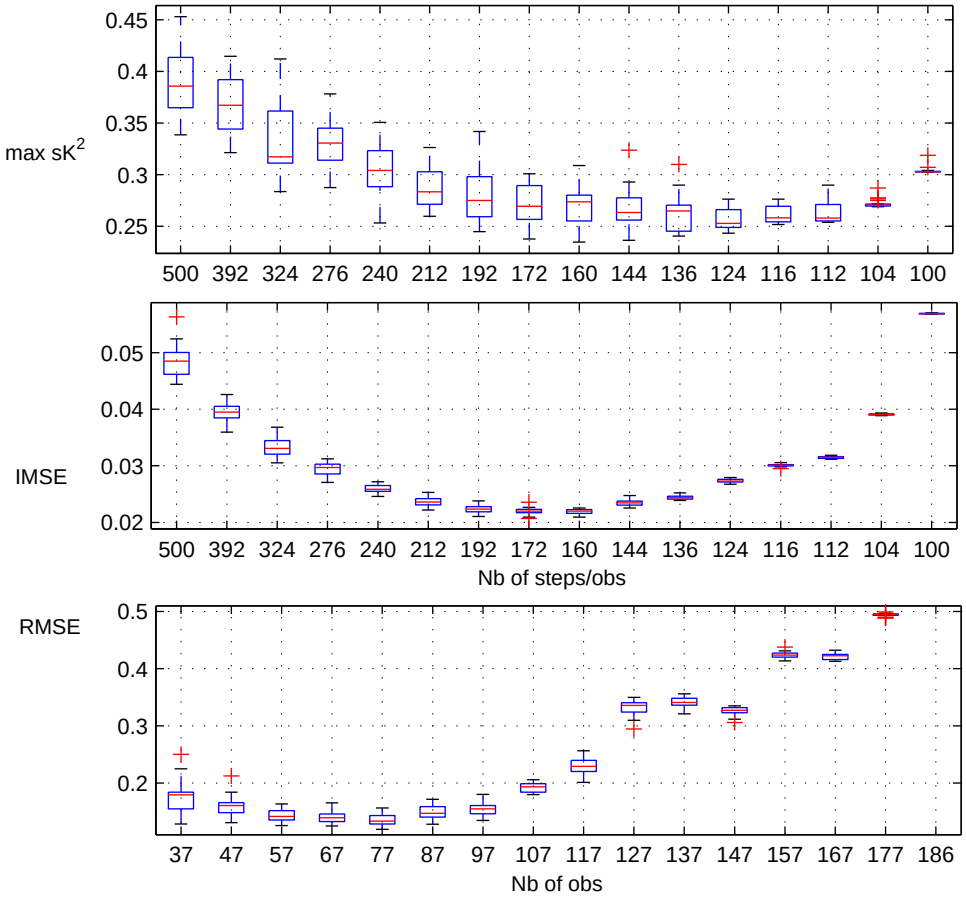

Figure 9. Boxplots of the RMSE, IMSE and maxMSE of the space-time model based on different DOE size for a constant computational budget of 18,500 steps. The x-axis is written either in terms of number of steps for one simulation or total number of simulations.

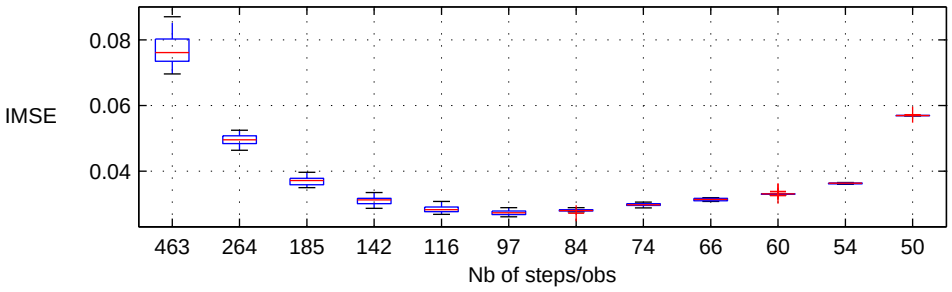

Figure 10. Boxplots of the IMSE of the space-time model based on different DOE size for a constant computational budget of 9,250 steps.

In addition, we proposed a procedure for the learning of the model parameters, by decomposing in into a series of simpler optimization problems, and discussed some numerical issues. Finally, we have applied our model to a real simulator, and showed some substantial improvement in learning compared to the classical framework.

Future research may include the application of this model to higher dimension problems, and optimization under partial convergence using EGO-like strategies.

Acknowledgements

This work was partially supported by French National Research Agency (ANR) through COSINUS program (project OMD2 ANR-08-COSI-007). The authors greatly acknowledge this support.

Appendix: Screening effect in the time dimension in the case of Monte-Carlo convergence

Property: *Once the $\tilde{Y}_i$ are taken into account in the model, adding any $Y(\mathbf{u})$ with $\mathbf{u} = (\mathbf{x}^i, t_u)$ for $i \in \{1, \ldots, n\}$ and $t_u \leq t_i$ has no effect on the model.*

Proof:
Let us denote by $\lambda_1, \ldots, \lambda_n$ the kriging weights corresponding to a prediction at an arbitrary point $\mathbf{x} \in D$ when $\tilde{Y}_1, \ldots, \tilde{Y}_n$ are known, the kriging mean being equal to $\sum_{k=1}^n \lambda_k \tilde{Y}_k$. By characterization of the kriging mean as projection of $Y(\mathbf{x})$ onto $Span\{\tilde{Y}_1, \ldots, \tilde{Y}_n\}$, we know that:

$$E\left[\left(Y(\mathbf{x}) - \sum_{k=1}^n \lambda_k \tilde{Y}_k\right)\tilde{Y}_i\right] = 0, \qquad \forall i \in \{1, \ldots, n\} \tag{30}$$

We will now show that $Y(\mathbf{x}) - \sum_{k=1}^n \lambda_k \tilde{Y}_k$ is also orthogonal to $Y(\mathbf{u})$, which is a sufficient condition for the conditional independence in question.

Indeed, denoting γ the scalar product between those two quantities, we have:

$$\begin{aligned}
\gamma &= E\left[\left(Y(\mathbf{x}) - \sum_{k=1}^n \lambda_k \tilde{Y}_k\right)\tilde{Y}(\mathbf{u})\right] && (31)\\
&= E\left[\left(Y(\mathbf{x}) - \sum_{k=1}^n \lambda_k \tilde{Y}_k\right)\left(F(x^i) + \frac{1}{t_u}\sum_{j=1}^{t_u}\varepsilon_{i,j}\right)\right] && (32)\\
&= E\left[\left(Y(\mathbf{x}) - \sum_{k=1}^n \lambda_k \tilde{Y}_k\right)\left(\tilde{Y}_i - \frac{1}{t_i}\sum_{j=1}^{t_i}\varepsilon_{i,j} + \frac{1}{t_u}\sum_{j=1}^{t_u}\varepsilon_{i,j}\right)\right] && (33)\\
&= E\left[\left(Y(\mathbf{x}) - \sum_{k=1}^n \lambda_k \tilde{Y}_k\right)\left(\frac{1}{t_u}\sum_{j=1}^{t_u}\varepsilon_{i,j} - \frac{1}{t_i}\sum_{j=1}^{t_i}\varepsilon_{i,j}\right)\right], && (34)
\end{aligned}$$

$\tilde{Y}_i$ being removed due to eq. 30.

Then, by hypothesis all the $\varepsilon_{i,j}$ are independent of each other and have a expectation equal to zero, so the expectation of the term on the right parenthesis has a null expectation.

Since $Y(\mathbf{x})$ and $\lambda_k \tilde{Y}_k$ are independent of $\varepsilon_{i,j}$ for $k \neq i$, eq. 34 reduces to:

$$\gamma = E\left[-\lambda_i \tilde{Y}_i \left(\frac{1}{t_u}\sum_{i=1}^{t_u} \varepsilon_{i,j} - \frac{1}{t_i}\sum_{i=1}^{t_i} \varepsilon_{i,j}\right)\right] \tag{35}$$

Then:

$$\begin{aligned}
\gamma &= E\left[-\lambda_i \left(F(\mathbf{x}^i) + \frac{1}{t_i}\sum_{j=1}^{t_i} \varepsilon_{i,j}\right)\left(\frac{1}{t_u}\sum_{j=1}^{t_u} \varepsilon_{i,j} - \frac{1}{t_i}\sum_{i=1}^{t_i} \varepsilon_{i,j}\right)\right] && (36)\\
&= -\lambda_i E\left[\left(\frac{1}{t_i}\sum_{j=1}^{t_i} \varepsilon_{i,j}\right)\left(\frac{1}{t_u}\sum_{j=1}^{t_u} \varepsilon_{i,j} - \frac{1}{t_i}\sum_{j=1}^{t_i} \varepsilon_{i,j}\right)\right] && (37)\\
&= -\lambda_i \left(\frac{1}{t_u t_i}\sum_{j=1}^{t_u}\sum_{k=1}^{t_i} E\left[\varepsilon_{i,j}\varepsilon_{i,k}\right] - \frac{1}{t_i^2}\sum_{j=1}^{t_u}\sum_{k=1}^{t_i} E\left[\varepsilon_{i,j}\varepsilon_{i,k}\right]\right) && (38)\\
&= -\lambda_i \left(\frac{1}{t_u t_i}\sum_{j=1}^{t_u}\sum_{k=1}^{t_i} \delta_{j,k} - \frac{1}{t_i^2}\sum_{j=1}^{t_u}\sum_{k=1}^{t_i} \delta_{j,k}\right) && (39)\\
&= -\lambda_i \left(\frac{1}{t_u t_i} t_u - \frac{1}{t_i^2} t_i\right) && (40)\\
&= 0 && (41)
\end{aligned}$$

References

Alexandrov, N., R. Lewis, C. Gumbert, L. Green, and P. Newman (2000). Optimization with variable-fidelity models applied to wing design. *AIAA paper 841*(2000), 254.

Cressie, N. (1992). Statistics for spatial data. *Terra Nova 4*(5), 613–617.

Forrester, A., N. Bressloff, and A. Keane (2006). Optimization using surrogate models and partially converged computational fluid dynamics simulations. *Proceedings of the Royal Society A 462*(2071), 2177.

Gano, S., J. Renaud, J. Martin, and T. Simpson (2006). Update strategies for kriging models used in variable fidelity optimization. *Structural and Multidisciplinary Optimization 32*(4), 287–298.

Jones, D., M. Schonlau, and W. Welch (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization 13*(4), 455–492.

Kennedy, M. and A. O'Hagan (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika 87*(1), 1.

Matheron, G. (1969). Le krigeage universel. *Cahiers du centre de morphologie mathématique 1*.

Rasmussen, C. and C. Williams (2006). *Gaussian processes for machine learning*. Springer.

Sacks, J., W. Welch, T. Mitchell, and H. Wynn (1989). Design and analysis of computer experiments. *Statistical science*, 409–423.

Stein, M. (2002). The screening effect in kriging. *Annals of statistics*, 298–323.