Fast model updating coupling Bayesian inference and PGD model reduction

Paul-Baptiste Rubio, François Louf, Ludovic Chamoin

Fast model updating coupling Bayesian inference and PGD model reduction

Paul-Baptiste Rubio* François Louf* Ludovic Chamoin*

email: {prubio, louf, chamoin}@ens-paris-saclay.fr

Abstract

The paper focuses on a coupled Bayesian-Proper Generalized Decomposition (PGD) approach for the real-time identification and updating of numerical models. The purpose is to use the most general case of Bayesian inference theory in order to address inverse problems and to deal with different sources of uncertainties (measurement and model errors, stochastic parameters). In order to do so with a reasonable CPU cost, the idea is to replace the direct model called for Monte-Carlo sampling by a PGD reduced model, and in some cases directly compute the probability density functions from the obtained analytical formulation. This procedure is first applied to a welding control example with the updating of a deterministic parameter. In the second application, the identification of a stochastic parameter is studied through a glued assembly example.

keywords:Model updating, Reduced order model, Bayesian inference, PGD.

1 Introduction

Due to recent advances in model order reduction methods, the topic of model updating by Bayesian inference appears to have a growing interest in several industrial domains using systems driven by simulation tools (manufacturing, medical etc.). This concept called DDDAS (Dynamic Data Driven Application System) has for main goal to introduce a continuous exchange between simulation tools and experimental measurements in order to have a retroactive control loop. The aim is to predict the behavior of complex systems [26]. Since the system environment is not entirely known, an effective and fast model updating method is required to post-process experimental measurements and find the unknown parameters in real time (data assimilation). Once the model is fully determined, it can be used to predict the action to perform in order to control the system efficiently. However, data assimilation in real time presents two scientific challenges.

The first one consists in solving an inverse problem to identify the parameters of the model from measurements.

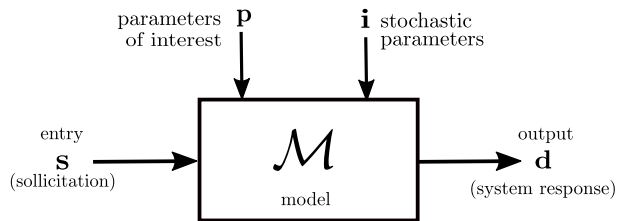Figure 1: Model environment

Figure 1 represents the model environment defined for studied systems. Each system is governed by a mathematical model $\mathcal{M}$. This model is the result of some physical assumptions and, in the most common case, leads to a partial differential equations system.The variables $\mathbf{s} \in \mathcal{S}$ are the entries of the system. The values assigned to those

entries are supposed to be known (in a deterministic way) and provides for outputs $\mathbf{d} \in \mathcal{D}$ that can be compared with the observed data. For example, $\mathbf{s}$ may correspond to loadings or boundary conditions, whereas outputs $\mathbf{d}$ may be local displacements, temperatures or stresses. In our model we additionally consider two kinds of model parameters. Firstly, the parameters $\mathbf{p} \in \mathcal{P}$ which are the parameters of interest. From the observations, the purpose of the study is to update the knowledge on those parameters; second, the stochastic parameters $\mathbf{i} \in \mathcal{I}$ which are parameters considered as random variables with known probability density function. These latter parameters can be identified from another stochastic inference process or given as data with a tolerance range.

With those definitions, the forward direct problem where the output is searched from the input and depending on all other parameters reads:

$$\text{Find } \mathbf{d} \in \mathcal{D}/\mathbf{d} = \mathcal{M}(\mathbf{s}, \mathbf{p}, \mathbf{i}) \ , \ (\mathbf{s} \in \mathcal{S}, \mathbf{p} \in \mathcal{P}, \mathbf{i} \in \mathcal{I}) \tag{1}$$

Conversely, the inverse problem where the parameters of interest are searched from the input, the other parameters and the observed data $\mathbf{d}^{\text{obs}}$, reads:

$$\text{Find } \mathbf{p} \in \mathcal{P}/\mathbf{d}^{\text{obs}} = \mathcal{M}(\mathbf{s}, \mathbf{p}, \mathbf{i}) \ , \ (\mathbf{s} \in \mathcal{S}, \mathbf{d}^{\text{obs}} \in \mathcal{D}, \mathbf{i} \in \mathcal{I}) \tag{2}$$

In the most general case, the inverse problem is ill-posed according to the Hadamard definition and the problem can be difficult to solve [1]. The resolution may be formulated in the determinist case in terms of a functional minimization including regularization terms (least squares, weighted least squares, Constitutive Relation Error [27], modified Constitutive Relation Error [28]). Another approach to deal with this difficulty is to use a stochastic approach in the Bayesian framework [1] [2]. This probabilistic approach leads directly to a regularized inverse problem and allows a natural treatment of uncertainties due to stochastic parameters, model and measurement errors. This method is quite popular and used in many fields such as identification of material properties [34], DDDAS [35], or monitoring and control of structures [30] [31] [32] [33]. We use this latter approach in this paper.

The second challenge is tied up with the constraint of a real-time simulation. This constraint implies to use lighter models, especially if the Bayesian inference framework is used in the general case where probability density functions are found with Monte-Carlo sampling [7]. In this case, the model is called a large amount of times which is very costly. Consequently, coupling model order reduction methods with the Bayesian inference is a relevant concept. In this case, posterior methods are often used like: Proper Orthogonal Decomposition (POD), Reduced Basis (RB) [15] [16] [4], Response Surface Approximation (RSA) [7] or Polynomial Chaos Expansion (PCE) [12].

Here, we propose a coupled Bayesian-PGD strategy to solve the identification problem on the most general case without assumption on probability density forms. We consider also models including stochastic parameters. The very interesting model reduction framework given by the Proper Generalized Decomposition (PGD) consists in finding a lighter surrogate model by separation of variables on a reasonable amount of modes [8] [19]. It is an *a priori* reduction method which provides an explicit solution with respect to parameters.

The aim of the paper is to use the multiparametric PGD model in order to speed up the system response computations needed in Bayesian inference. PGD modes are computed in an *offline* phase. Then, in the *online* phase, the computation cost is reduced to a simple summation of those modes. This study focuses on the improvement of the Bayesian inference in low dimension (parameters to identify, stochastic parameters) by the PGD model order reduction method.

The paper is organized as follows. In Section 2, we expose the general methodology of the Bayesian inference for model updating. In Section 3, we develop the PGD model order reduction applied to the identification problem with Bayesian inference. In Section 4, we illustrate the approach with two examples: a welding control quality problem with a deterministic parameter estimation and a glued assembly problem where the estimation of a stochastic parameter through hyper-parameters is made. Eventually, conclusions and prospects to this work are drawn in Section 5.

2 Bayesian updating

2.1 Bayes theorem

The Bayesian identification is based on the Bayes theorem (1763) known in probability theory. The theorem gives the probability of an event A knowing the event B:

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \tag{3}$$

The term $P(A)$ is the probability of the event A, it is the prior probability without any information on the event B. The term $P(A|B)$ is named the posterior probability of A knowing B. This probability directly depends on B which makes it posterior to the information of B. The probability $P(B|A)$ is named the likelihood function: it is the probability (likelihood) of having B knowing A.

If continuous probabilities are considered the Bayes theorem reads in term of probability density functions (pdf):

$$\pi(a|b) = \frac{1}{C}\pi(b|a).\pi(a) \tag{4}$$

where C is a normalization constant verifying:

$$\int_{-\infty}^{+\infty} \pi(a|b)da = 1 \tag{5}$$

2.2 Bayesian formulation for identification problems

Using the formulation postulated in the introduction, the general identification problem is defined as follows:

$$\text{Find } \mathbf{p} \in \mathcal{P} \text{ such that, } \mathbf{d}^{\text{obs}} = \mathcal{M}(\mathbf{s}, \mathbf{i}, \mathbf{p}) \text{ , } (\mathbf{s} \in \mathcal{S}, \mathbf{i} \in \mathcal{I}) \tag{6}$$

With:

- Parameters of interest (to identify): $\mathbf{p}$
- Model: $\mathcal{M}$
- Observations: $\mathbf{d}^{\text{obs}}$
- Model entries: $\mathbf{s}$
- Stochastic parameters: $\mathbf{i}$

For identification problems in solid mechanics the Bayes theorem is formulated in order to find the probability of having the parameters knowing the measurements. The parameters to identify take unique but unknown values represented in a random variable $\mathbf{p}$. The probability density function of the random variable $\mathbf{p}$ without any experimental knowledge is denoted $\pi(\mathbf{p})$. This density is postulated *a priori* with the information available (range of variation, most likely values) and can be obtained thanks to the maximum entropy principle, otherwise a non-informative density is taken (uniform density).

The aim of the Bayesian approach is to improve the knowledge on the parameter $\mathbf{p}$ with the update of the probability density function $\pi(\mathbf{p})$ from the comparison of the experimental data $\mathbf{d}^{obs}$ to the model output $\mathbf{d}$.

Then, the posterior probability density function represents the probability of having $\mathbf{p}$ such as the measurements coincide with the model output. Those definitions and the Bayes theorem ensure that the posterior density is given by the product of the likelihood function by the prior probability density function of the parameters:

$$\pi(\mathbf{p}|\mathbf{d}^{\text{obs}}) = \frac{1}{C}.\pi(\mathbf{d}^{\text{obs}}|\mathbf{p}).\pi(\mathbf{p}) \tag{7}$$

As a result, Bayesian identification is a stochastic approach of inverse problems; the result of this approach provides a probability density function. In comparison with deterministic methods used in identification theory, the Bayesian inference is a natural regularizing approach which provides a probability weight to each possible solution. In the specific case where all uncertainties, prior and errors densities are Gaussian it can be proved [1] that the Bayesian inference is equivalent to a weighted least square minimization with a Tikhonov regularization.

2.3 Building the likelihood function

In this study, where all forms of uncertainties are considered, the purpose is to propagate uncertainties through the model to obtain the probability density functions needed to build the likelihood function $\pi(\mathbf{d}^{\mathrm{obs}}|\mathbf{p})$. This function represents the probability to have the model output equal to the measurements knowing the parameters $\mathbf{p}$. Measurements being known, the likelihood function is seen as a function of $\mathbf{p}$. This probability is directly linked with uncertainties $\mathbf{i}$ considered in the model environment (see figure 1), model and measurements errors. Here, few parameters are considered so that the likelihood function will be built by sampling the discretized parametric space. Other alternative will be discussed in Section 5.

2.3.1 Measurement error

Considering an additive (and independent of $\mathbf{p}$ and $\mathbf{i}$) measurement noise $\mathbf{e}$, the output of the model is given by:

$$\mathbf{d}^{\mathrm{obs}} = \mathbf{d} + \mathbf{e} \text{ , with } \mathbf{d} = \mathcal{M}(\mathbf{p}, \mathbf{i}) \tag{8}$$

Considering $\pi_{err}(\mathbf{e})$ the probability density function of the measurement error, by a convolution product the likelihood function reads:

$$\pi(\mathbf{d}^{\mathrm{obs}}|\mathbf{p}) = \int \pi(\mathbf{d}|\mathbf{p}).\pi_{\mathrm{err}}(\mathbf{d}^{\mathrm{obs}} - \mathbf{d})\mathrm{d}\mathbf{d} \tag{9}$$

The probability density function $\pi(\mathbf{d}|\mathbf{p})$ represents the probability of having an output $\mathbf{d}$ for a given value of $\mathbf{p}$. This probability is directly linked to the model. If there is no ucertainty in the model (no stochastic parameter $\mathbf{i}$ and no model error) the probability density function reads:

$$\pi(\mathbf{d}|\mathbf{p}) = \delta(\mathbf{d} - \mathcal{M}(\mathbf{p})) \tag{10}$$

with δ the Dirac delta function. Then the computation of the integral (9) is explicit and the likelihood function is given by:

$$\pi(\mathbf{d}^{\mathrm{obs}}|\mathbf{p}) = \pi_{\mathrm{err}}(\mathbf{d}^{\mathrm{obs}} - \mathcal{M}(\mathbf{p})) \tag{11}$$

2.3.2 Uncertain parameters/model error

When uncertainties are considered in the model, the output becomes a random variable and the probability density function $\pi(\mathbf{d}|\mathbf{p})$ needs to be computed. Then, the probability density function of this random variable is obtained propagating those uncertainties in the model.

To determine this probability density function, a Monte-Carlo sampling can be performed using samples made with deterministic computations. The Monte-Carlo samples $\mathbf{d}_{\mathrm{mc}}$ are evaluated in the model with values of the uncertainties $\mathbf{i}_{\mathrm{mc}}$ drawn according to the probability density function assumed in the modeling:

$$\mathbf{d}_{\mathrm{mc}} = \mathcal{M}(\mathbf{p}, \mathbf{i}_{\mathrm{mc}}) \tag{12}$$

With the samples $\mathbf{d}_{\mathrm{mc}}$, the probability density function $\pi(\mathbf{d}|\mathbf{p})$ is built thanks to a kernel density estimation defined as follows:

$$\pi(\mathbf{d}|\mathbf{p}) \approx \frac{1}{N_{\mathrm{mc}}h} \sum_{\mathrm{mc}=1}^{N_{\mathrm{mc}}} K\left(\frac{\mathbf{d} - \mathbf{d}_{\mathrm{mc}}}{h}\right) \tag{13}$$

K is a kernel function and h is a smoothing parameter called the bandwidth. A common choice is to take K as a Gaussian function of a normal centered probability density function.

The output can also become a random variable if model error is considered. In this case $\pi(\mathbf{d}|\mathbf{p}) = \pi_{\mathrm{mod}}(\mathbf{d})$, with $\pi_{\mathrm{mod}}(\mathbf{d})$ the model error probability density function. Those computations are made for each value of the discretized space of $\mathbf{p}$ in order to sample the likelihood function.

2.4 Post processing

Once the likelihood function is computed, the product of the likelihood function by the prior probability density function gives the posterior probability density function. The Bayesian framework gives a substantial information on the parameter identified as a probability density function. The post-processing of this density can be done by different estimators:

- Maximum *a posteriori*: $p_{MAP} = \arg\max_p \pi(p|\mathbf{d}^{obs})$

- Mean *a posteriori*: $p_M = \int p.\pi(p|\mathbf{d}^{obs})\mathrm{d}p$

- Standard deviation: $\sigma = \sqrt{\int p^2.\pi(p|\mathbf{d}^{obs})\mathrm{d}p - p_M}$

- ...

When the parameter space dimension is $m > 1$ ($\mathbf{p} = (p_1,...,p_m)$) the Bayesian identification result is a joint probability density function $\pi_{joint}(p_1,...,p_m)$. To obtain the marginal probability density function of a parameter p_k, $k \in \{1,...,m\}$ the following quantity has to be evaluated:

$$\pi_k(p_k|\mathbf{d}^{\mathrm{obs}}) = \int_{\substack{m \\ \underset{l=1}{\otimes} I_l \\ l \neq k}} \pi_{\mathrm{joint}}\left((p_l)_{\substack{l \in \{1,..,m\} \\ l \neq k}}|\mathbf{d}^{\mathrm{obs}}\right) d\left((p_l)_{\substack{l \in \{1,..,m\} \\ l \neq k}}\right) \tag{14}$$

When a model updating process is considered from successive measurements (e.g. at some time steps), the Bayesian inference naturally allows to take successive informations into account and keeps an history of the previous identifications steps by taking the prior probability density function at the current step as the posterior probability density function of the previous step.

Eventually, the global algorithm of the general Bayesian framework for a given model $\mathcal{M}$ is summed up in the algorithm 1.

Algorithm 1 Bayesian identification - Monte-Carlo

Require: Measurements: $\mathbf{d}^{\mathrm{obs}} = \{\mathbf{d}_j^{\mathrm{obs}}\}_{j \in \{1,..,N\}}$, model $\mathcal{M}$, stochastic parameter pdf $\pi(\mathbf{i})$, prior density $\pi_1^{\mathrm{prior}}(\mathbf{p})$.

1: **for** j=1 to N **do**
2:    **for** $\mathbf{p} \in I^h$ **do**   ▷ Discretized parameter space
3:      Monte-Carlo sampling :
4:      **for** mc = 1 to N_{mc} **do**
5:        $\mathbf{i}_{\mathrm{mc}} \sim \pi(\mathbf{i})$
6:        $\mathbf{d}_{\mathrm{mc}} = \mathcal{M}(\mathbf{i}_{\mathrm{mc}}, \mathbf{p})$
7:      **end**
8:
9:      Uncertainty propagation with kernel density estimation :
10:

$$\pi(\mathbf{d}|\mathbf{p}) \approx \frac{1}{N_{\mathrm{mc}}h} \sum_{\mathrm{mc}=1}^{N_{\mathrm{mc}}} K\left(\frac{\mathbf{d} - \mathbf{d}_{\mathrm{mc}}}{h}\right)$$

11:
12:      Computation of the current likelihood function point $\mathbf{p}$:

$$\pi(\mathbf{d}_j^{\mathrm{obs}}|\mathbf{p}) = \int \pi(\mathbf{d}|\mathbf{p}).\pi_{\mathrm{err}}(\mathbf{d}^{\mathrm{obs}} - \mathbf{d})\mathrm{d}\mathbf{d}$$

13:    **end**
14:    $\pi(\mathbf{p}|\mathbf{d}^{\mathrm{obs}}) = \pi(\mathbf{d}_j^{\mathrm{obs}}|\mathbf{p}).\pi_j^{\mathrm{prior}}(\mathbf{p})$
15:    $\pi_{j+1}^{\mathrm{prior}}(\mathbf{p}) = \pi(\mathbf{p}|\mathbf{d}^{\mathrm{obs}})$
16: **end**
17: Post-processing for the multi-parametric case:
18: **for** k=1 to m **do**

$$\pi_k(p_k|\mathbf{d}^{\mathrm{obs}}) = \int_{\substack{m \\ \underset{l=1}{\otimes} I_l \\ l \neq k}} \pi_{\mathrm{joint}}\left((p_l)_{\substack{l \in \{1,..,m\} \\ l \neq k}}|\mathbf{d}^{\mathrm{obs}}\right) d\left((p_l)_{\substack{l \in \{1,..,m\} \\ l \neq k}}\right)$$

19: **end**

The Bayesian procedure applied to model updating coupled with Monte-Carlo sampling can handle with all forms of uncertainties. However, it leads to a huge amount of calls to the model. Indeed, in the algorithm 1 the model $\mathcal{M}$ needs to be evaluated $N \times \dim(I^h) \times N_{\mathrm{mc}}$ times. Furthermore, once the posterior probability density function is built, a post-processing needs to be done and add another computation cost. In basic cases the model of the system can be analytical but in the DDDAS framework complex systems are considered. If the system is represented by a finite element model, in a purpose of identification in real time, a direct solving cannot be done for all values of the parameters. That is why a surrogate lighter model is needed to perform the Bayesian inference.

3   The coupled Bayesian-PGD inference

An attractive model order reduction technique is given by the Proper Generalized Decomposition (PGD). The PGD method was introduced in [29] as "radial approximation" to solve nonlinear problems in structural mechanics. Since [19], this method was used in many fields: model verification and validation [18], virtual charts for the engineering [20] [21] etc. PGD is also used for identification problems in a deterministic framework [25] [10] [3] and the great possible number of parameters types to use in those models [8] seems to be well suited for Bayesian inference. A first PGD-Bayesian inference approach is given in [6] where a PGD model is used in a Monte Carlo Markov Chain framework. In our paper, we consider stochastic parameters which lead to additionnal sources of uncertainties. Those uncertainties are considered with the PGD model.

3.1   Progressive Galerkin PGD

The general case is considered where the problem is modelled by a system of partial differential equations. This problem is supposed to be formulated with its equivalent global weak formulation:

$$\text{Find } u \in \mathcal{U} / \forall v \in \mathcal{V} : a(u,v) = l(v) \tag{15}$$

Where a is a bilinear form, l a linear form, u is the unknown field and v the test field. d variables $p_1, ..., p_l$ are considered. Those variables can be space variables, time, load parameters, material properties etc. The following separated solution of the problem (15) is searched (tensorized representation with canonical format):

$$u(p_1, ...p_l) \approx u_{PGD}(p_1, ...p_l) = \sum_{n=1}^{m} \prod_{k=1}^{l} \lambda_{kn}(p_k) \tag{16}$$

m is the number of modes and $u \in \mathcal{U} = \otimes_{k=1}^{k=l} U_k$. The computation of modes is performed incrementally:

$$u_{PGD}(p_1, ...p_l) = \sum_{n=1}^{m-1} \prod_{k=1}^{l} \lambda_{kn}(p_k) + \prod_{k=1}^{l} \lambda_{km}(p_k) \tag{17}$$

The unknowns are then: $\lambda_{km} \in U_k$, $k \in \{1, ..., l\}$. The test field is taken in the separated form:

$$v = \sum_{k=1}^{l} \lambda_{km}^{*} \prod_{\substack{j=1 \\ j \neq k}}^{l} \lambda_{jm} \; , \; \lambda_{km}^{*} \in U_k \; , \; k \in \{1, ..., l\} \tag{18}$$

Verifying (15) thus leads to n decoupled equations:

$$\lambda_{km} = \mathcal{S}_k((\lambda_{jm})_{\substack{j \in \{1, ..., l\} \\ j \neq k}}) \; , \; k \in \{1, ..., l\} \tag{19}$$

This system of equations is solved by a fixed-point algorithm. This way, PGD is an *a priori* method where modes are generated while the problem is solved in the *offline* phase. Once the separated solution is computed in the *offline* phase, it can be reused in the *online* phase with a low computation cost since the solution is explicit regarding all coordinates p_k $k \in \{1, ..., l\}$.

3.2 PGD model reduction for Monte-Carlo sampling

The PGD model order reduction method allows to have the response of a system modelled with partial differential equations for all values of parameters. The aim is to build a PGD model to speed up the Bayesian process described in the algorithm 1. According to the model environment set in previous sections (figure 1) the following PGD representation is searched:

$$\mathcal{M}(\mathbf{p}, \mathbf{x}, \mathbf{i}) = \mathbf{d} \approx \sum_{n=1}^{m} \Lambda_n(\mathbf{x}) \alpha_{1n}(\mathbf{p}) \alpha_{2n}(\mathbf{i}) \tag{20}$$

The model is built by separation of each parameter/variable in the loop of algorithm 1. Thereby the online computation cost is reduced to the summation of modes (array) for each loop instead of a full resolution at each iteration. The smooth PGD framework for the separation of distinct type of parameters is well suited to generate the Monte-Carlo sampling then to build the likelihood function point by point as presented before.

Finally, the computation cost of the Bayesian identification process is focused on the evaluation of the PGD model (which is explicit) for each Monte-Carlo draw, each iteration on the parameter to identify and for each integral computation (for model/measurement errors, post-processing).

3.3 PGD for analytical uncertainties propagation

The PGD model reduction is also a very interesting approach as it provides an analytical form of the solution for all the parameters. In this part, the purpose is to use this analytical form to propagate the uncertainties directly without Monte-Carlo sampling. Here the study is restricted when the dimension of stochastic parameters $\mathbf{i}$ is one-dimensional.

In this case, the uncertainty propagation in the model can be found by a change of variable. A probability density function as any integrand verifies the variable change theorem.

Let $\pi_X(x)$ be the probability density function of the real random variable X and $Y = f(X)$ the variable change. If the function f is strictly monotonic and differentiable with a never null derivative, then the probability density function $\pi_Y(y)$ of the variable change is given by:

$$\pi_Y(y) = \left| \frac{1}{f'(f^{-1}(y))} \right| . \pi_X(f^{-1}(y)) \tag{21}$$

The previous case is generalized for the non-monotonic case as:

$$\pi_Y(y) = \sum_{k=1}^{N_y} \left| \frac{1}{f'(f_k^{-1}(y))} \right| . \pi_X(f_k^{-1}(y)) \tag{22}$$

where N_y is the number of the $x_k = f_k^{-1}(y)$ solutions of the equation $f(x) = y$. f has to verify the inverse function theorem: the set where the derivative is zero has to have a measure of zero.

This theorem can be applied with the PGD function:

$$f : i \to \sum_{n=1}^{m} \Lambda_n(\mathbf{x}) \alpha_{1n}(\mathbf{p}) \alpha_{2n}(i) \tag{23}$$

The probability density functions estimated by Monte-Carlo sampling and kernel density estimation can be replaced with the change of variable:

$$\pi(\mathbf{d}|\mathbf{p}) = \pi_{f(\mathbf{p}, \mathbf{x}, i)}(\mathbf{d}) \tag{24}$$

However this change of variable is presented for the one dimension case on the parameter i. It seems to be more difficult to study the multidimensional function in order to build the Jacobian matrix and the inverse. Eventually, the Bayesian inference method with analytical-PGD uncertainty propagation is summed up in algorithm 2.

Avoiding the Monte-Carlo sampling seems to be very promising because it is the main source of computation cost once the full order model is replaced by a reduced order model. In the next sections the goal is to apply this methodology of coupled Bayesian-PGD inference in model updating examples.

Algorithm 2 Bayesian identification - Analytical PGD

Require: Measurements: $\mathbf{d}^{\mathrm{obs}} = \{\mathbf{d}_j^{\mathrm{obs}}\}_{j \in \{1,..,N\}}$, PGD modes $\{\Lambda_n, \alpha_{1n}, \alpha_{2n}\}$, stochastic parameter pdf $\pi(\mathbf{i})$, prior
    density $\pi_1^{\mathrm{prior}}(\mathbf{p})$.

1: **for** j=1 to N **do**
2:     **for** $\mathbf{p} \in I^h$ **do**   ▷ Discretized parameter space
3:         Change of variable function:

$$f(i) = \sum_{n=1}^{m} \Lambda_n(\mathbf{x}) \alpha_{1n}(\mathbf{p}) \alpha_{2n}(i)$$

4:         Computation of the derivative f' to define the intervals N_y where the function is monotonic and com-
    putation of corresponding function inverses f_k^{-1}.
5:         Uncertainty propagation by analytical change of variable:

$$\pi(\mathbf{d}|\mathbf{p}) = \sum_{k=1}^{N_y} \left| \frac{1}{f'(f_k^{-1}(\mathbf{d}))} \right| . \pi(f_k^{-1}(\mathbf{d}))$$

6:         Computation of the current likelihood function point $\mathbf{p}$:

$$\pi(\mathbf{d}_j^{\mathrm{obs}}|\mathbf{p}) = \int \pi(\mathbf{d}|\mathbf{p}) . \pi_{\mathrm{err}}(\mathbf{d}^{\mathrm{obs}} - \mathbf{d}) \mathrm{d}\mathbf{d}$$

7:     **end**
8:     $\pi(\mathbf{p}|\mathbf{d}^{\mathrm{obs}}) = \pi(\mathbf{d}_j^{\mathrm{obs}}|\mathbf{p}).\pi_j^{\mathrm{prior}}(\mathbf{p})$
9:     $\pi_{j+1}^{\mathrm{prior}}(\mathbf{p}) = \pi(\mathbf{p}|\mathbf{d}^{\mathrm{obs}})$
10: **end**
11: Post-processing for the multi-parametric case:
12: **for** k=1 to m **do**

$$\pi_k(p_k|\mathbf{d}^{\mathrm{obs}}) = \int_{\substack{m \\ \otimes \, I_l \\ l=1 \\ l \neq k}} \pi_{\mathrm{joint}} \left((p_l)_{\substack{l \in \{1,..,m\} \\ l \neq k}} | \mathbf{d}^{\mathrm{obs}} \right) d\left((p_l)_{\substack{l \in \{1,..,m\} \\ l \neq k}} \right)$$

13: **end**
---

4 Numerical examples

4.1 A welding control quality example

In the industrial domain, the welding process control can be problematic and a process controlled by a numerical model through DDDAS framework can be an interesting approach. The aim of this example is to test the coupled Bayesian-PGD framework in the situation of a welding process controlled in real time. The studied problem is an academic example from [9]. Even if the industrial problem has been simplified, it is still sufficiently complex to evaluate the performance of the presented methodology.

4.1.1 Definition of the problem

The welding control quality problem is described in figure 2. Two metal plates are welded by a heat source which the center is moving along the geometry. The welding quality is directly linked to the welding depth d_w and the welding process is assumed to have a sufficient quality when the welded depth is above 0.5.
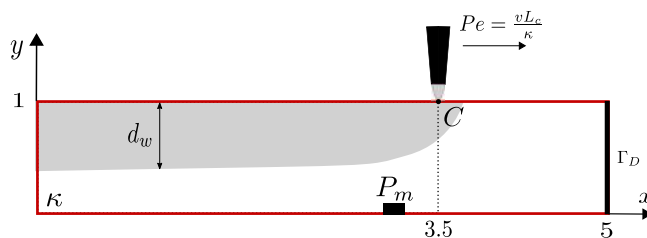

Figure 2: Joint-section view of the welding process

We assume the welding depth cannot be measured directly. Thus, we propose to estimate it using a numerical model. Since some parameters of the model are not well known, an updating process based on *in situ* measurements needs to be performed. Once the numerical model is fully determined the welding depth can be computed.

A minimum welding depth specifies the good quality of the process. If the computed welding depth does not verify this minimum, command parameters (intensity, heat source speed etc.) can be determined numerically to improve the quality of the process.

This study focuses only on the model updating part where unknown parameters of the model are identified from the measurements.

- A 2D unsteady convection-diffusion problem is considered with the geometry described in figure 2. A homogeneous isotropic material is assumed.

  The following non-dimensionalization of the temperature field is made:

  $$T(x, y; t) \equiv \frac{\overline{T}(x, y; t) - \overline{T}_\infty}{\overline{T}_f - \overline{T}_\infty} \tag{25}$$

  with: $\overline{T}_\infty$ the ambient temperature and $\overline{T}_f$ the melting temperature of the material. The welding depth is then defined by the isotherm $T = 1$.

- The torch velocity is v. The coordinate system is moving at the same speed as the torch. Thereby, a convective term is added to compensate the referential change. Then the non-dimensional unsteady convection-diffusion equation is obtained:

  $$\frac{\partial T}{\partial t} + \underline{v}.\underline{\mathrm{grad}}T - \kappa \Delta T = s \tag{26}$$

  with s the volume heat input, $\underline{v} = [Pe; 0]$ and $Pe = \frac{v.L_c}{\kappa}$ being the Peclet number. L_c is the characteristic length of the problem and κ is the thermal diffusivity of the material.

- The other radiation and diffusive phenomena are neglected.

- The spatial distribution of the heat input is assumed to be Gaussian centered at the point $C = (x_c; y_c) = (3.5; 1)$:

$$s(x, y; \sigma) = \frac{u}{2\pi\sigma^2} \exp\left(-\frac{(x - x_c)^2 + (y - y_c)^2}{2\sigma^2}\right) \tag{27}$$

  u is the non-dimensional heat input that is supposed to be known and σ is the distribution parameter that is supposed to be unknown and needs to be identified. To that end we compare the model output to the temperature measurements taken at the measurement point $P_m = (x_1, y_1)$ (figure 2). Then, the model output is defined as $T_1 = T(x_1, y_1)$.

- Boundaries conditions: on Γ_D the temperature is assumed to be equal to the ambient temperature and the remaining boundaries are assumed insulated. Finally the initial condition is: $T(x, y; t = 0) = 0$.

4.1.2 Finite element solution

In this part the purpose is to set up the finite element solution. The results will be compared to the PGD results and the solution given by other models will be compared to this reference.

The weak form is formulated from (26). The kinematic admissible temperature field space is defined as follows:

$$\mathcal{T} = \{T \in H^1(\Omega =]0; 5[\times]0; 1[), T = 0 \text{ on } \Gamma_D\} \tag{28}$$

The equation (26) is equivalent to finding $T \in \mathcal{T}$ such that $\forall T^* \in \mathcal{T}$:

$$a(T, T^*) = l(T^*) \tag{29}$$

with:

$$a(T, T^*) = \int_\Omega \left(\frac{\partial T}{\partial t} + \underline{v}.\text{grad}T\right).T^* + \kappa.\underline{\text{grad}}T.\underline{\text{grad}}T^* d\Omega \tag{30}$$

$$l(T^*) = \int_\Omega s.T^* d\Omega \tag{31}$$

This formulation is discretized by the triangulation of the domain Ω, the semi-discretized finite element problem reads:

Find $\{T\} \in \mathcal{T}_h$ such that:

$$[M]\{\dot{T}\} + [C_H]\{T\} = \{S\} \tag{32}$$

The temperature field is then interpolated at the nodes by the matrix $[N]$ which contains the finite element shapes functions: $T = [N]\{T\}$.

The finite elements matrices read:

$$[M] = \int_{\Omega_h} [N]^T[N] d\Omega \tag{33}$$

$$[H] = \int_{\Omega_h} \{V\}^T[dN]^T[N] d\Omega \tag{34}$$

$$[C] = \kappa.\int_{\Omega_h} [dN]^T[dN] d\Omega \tag{35}$$

$$[C_H] = [C] + [H] \tag{36}$$

The equation (32) is solved owing to a time discretization coupled with an implicit scheme.

The previous solution scheme is implemented in a MATLAB code. The temperature field obtained at different time steps is shown in figure 3.

With the current set of parameters, the stationarity is reached at $t = 0.3$ s. The isotherms of the solution are shifted to the left from the torch position because of the convective term.
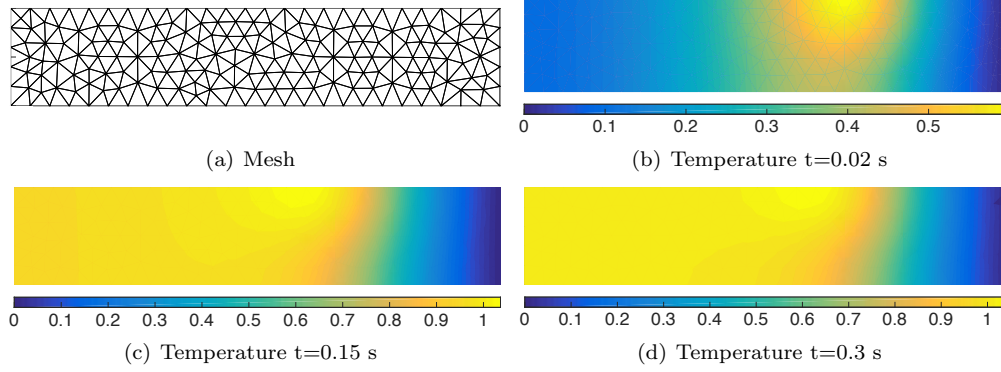
(a) Mesh

(b) Temperature t=0.02 s

(c) Temperature t=0.15 s

(d) Temperature t=0.3 s

Figure 3: Finite element solution

4.1.3 PGD solution

In the purpose of realizing the Bayesian identification described in the previous sections, the following part presents how the PGD reduced order model is built in order to reduce the computation cost of the likelihood function. As the parameter to identify is the width of the Gaussian heat source, the decomposition will be a variable separation of space/time/parameter σ:

$$T(x,y,t;\sigma) \approx T_m(x,y,t;\sigma) = \sum_{n=1}^{m} \Lambda_n(x,y)\lambda_n(t)\alpha_n(\sigma) \tag{37}$$

In the Bayesian-DDDAS framework the model needs to be evaluated at each time step, for all values of the parameter σ as explained in the algorithm 1. Instead of solving a finite element problem for each of those different values, a more general problem is solved by the PGD problem for all values of the different variables and parameters.

The PGD method presented in Section 3.1 is applied to the weak form (29). The PGD modes are given by the fixed-point algorithm 3. In Appendix the details of the PGD solution are explained.

Algorithm 3 Fixed point algorithm for PGD solution

Require: $I = [0, T_f]$, $\Sigma = [\sigma_{\min}, \sigma_{\max}]$.

1: **for** $p = 1$ to m **do**   ▷ Number of modes
2:      $\lambda = \frac{1}{\sqrt{T_f}}.\{1\}_I$
3:      $\alpha = \frac{1}{\sqrt{\sigma_{\max}-\sigma_{\min}}}.\{1\}_\Sigma$
4:      **for** $k = 1$ to $k_{\max}$ **do**
5:          $\Lambda = S_m(\lambda, \alpha)$
6:          $\lambda = T_m(\Lambda, \alpha)$
7:          $\alpha = P_m(\Lambda, \lambda)$
8:      **end**
9:      $\Lambda_p = \Lambda$
10:     $\alpha_p = \frac{\alpha}{\sqrt{\int_\Sigma \alpha^2 d\sigma}}$ ▷ Normalization of parametric modes
11:     $\lambda_p = \frac{\lambda}{\sqrt{\int_I \lambda^2 d\lambda}}$ ▷ Normalization of time modes
12: **end**

In the algorithm 3 the stopping criterion is $k_{\max}$. In practice, a coarse criterion ($k_{\max} = 4$) is sufficient to obtain a good convergence. In the Progressive Galerkin approach next PGD modes correct previous PGD modes regularizing the convergence. Parametric and time modes are normalized to improve the numerical stability of the solution scheme.

After a finite element discretization, a linear system is solved to obtain spatial and parametric modes at each step of the fixed-point algorithm. Time modes, which are solution of an ordinary differential equation, are computed using a Runge Kutta 4-5 time integration scheme.

11

The first four spatial modes are represented in figure 4. In comparison with the finite simulation in figure 3, the first mode is representative of the steady-state and the other modes are contributing to the transient state. The second mode is very similar to the finite element solution in the beginning of the transient state. The other modes improve the solution at the geometry edges and around the heat input.
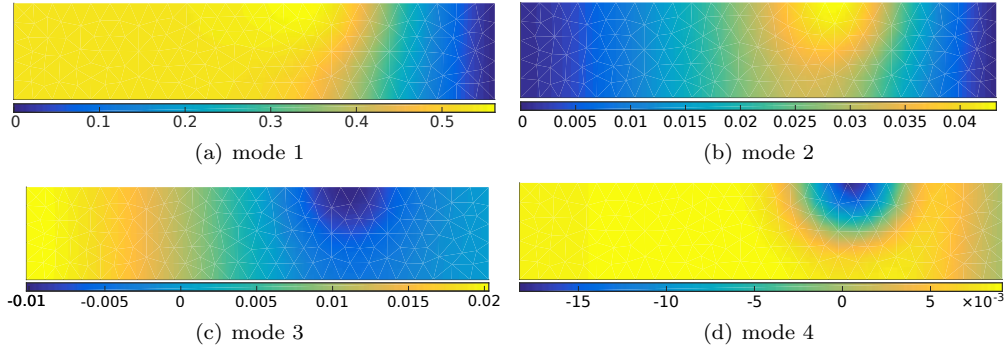

(a) mode 1

(b) mode 2

(c) mode 3

(d) mode 4

Figure 4: The four first spatial modes

As the time modes and parametric modes are normalized, the energetic norm of spatial modes gives the contribution of each mode to the global solution. This quantity is represented in figure 5. The six first modes are contributing for the most to the solution.
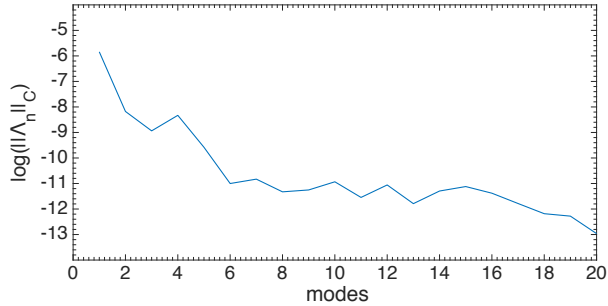

Figure 5: Energetic norm of the spatial modes

The normalized time modes are plotted in figure 6. The modes have a strong gradient at the beginning which requires a fine time discretization.
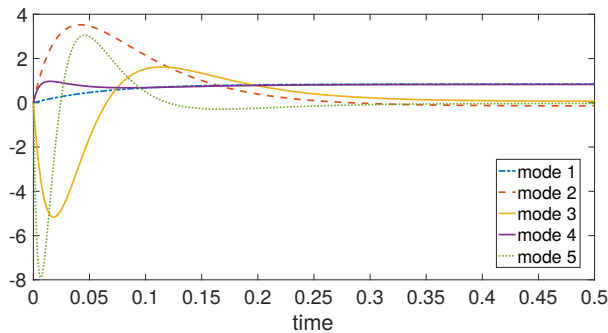

Figure 6: Time modes $\lambda_i(t)$, $i \in \{1, ..., 5\}$

Eventually, in figure 7 normalized parametric modes are represented. Figure 7 shows that the 4-th mode has a higher dependence on σ than the others. This result seems conform to the corresponding spatial mode 4(d) which exhibits a strong gradient near the torch center.

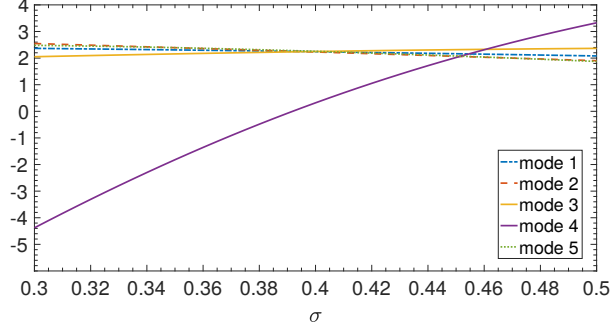Figure 7: Parametric modes $\alpha_i(\sigma)$, $i \in \{1, ..., 5\}$

4.1.4 Error quantification

The quality of the PGD model is studied and error quantification is done in the following section in order to use it for the Bayesian identification.

The functions T_{FE} and T_{PGD} are the temperature fields computed with the direct finite element model and the PGD model respectively. The direct problem is solved for each value of the parameter σ in order to obtain the function $T_{\mathrm{FE}}(x, y, t, \sigma)$. The T_1 functions are the temperatures at the measurement point.

The following errors are defined :

- Local error integrated in space:

$$\epsilon(t, \sigma) = \frac{\|T_{\mathrm{PGD}}(x, y, t, \sigma) - T_{\mathrm{FE}}(x, y, t, \sigma)\|_{L^2(\Omega)}}{\|T_{\mathrm{FE}}(x, y, t, \sigma)\|_{L^2(\Omega)}} \tag{38}$$

- Local error on the output T_1:

$$\epsilon_1(t, \sigma) = \frac{|T_{\mathrm{1PGD}}(t, \sigma) - T_{\mathrm{1FE}}(t, \sigma)|}{|T_{\mathrm{1FE}}(t, \sigma)|} \tag{39}$$

- Global error on the output T_1:

$$\bar{\epsilon} = \frac{\|T_{\mathrm{1PGD}}(t, \sigma) - T_{\mathrm{1FE}}(t, \sigma)\|_{L^2(\Sigma, I)}}{\|T_{\mathrm{1FE}}(t, \sigma)\|_{L^2(\Sigma, I)}} \tag{40}$$

Figure 8 shows the error $\bar{\epsilon}$ depending on the number of modes of the PGD solution.
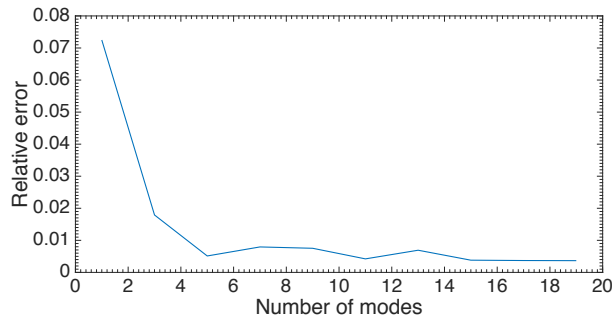

Figure 8: Error $\bar{\epsilon}$ depending on the number of modes

The finite element and PGD problems are discretized the same way so that a convergence of the PGD solution to the finite element solution is expected with a few number of modes. For this problem the minimum of the error is reached with a few number of modes (around 11 modes). Figure 9 shows the local errors $\epsilon(t, \sigma)$ and $\epsilon_1(t, \sigma)$ on a discretized map of parametric space and time. The error is high for the first time steps. Then, it strongly decreases
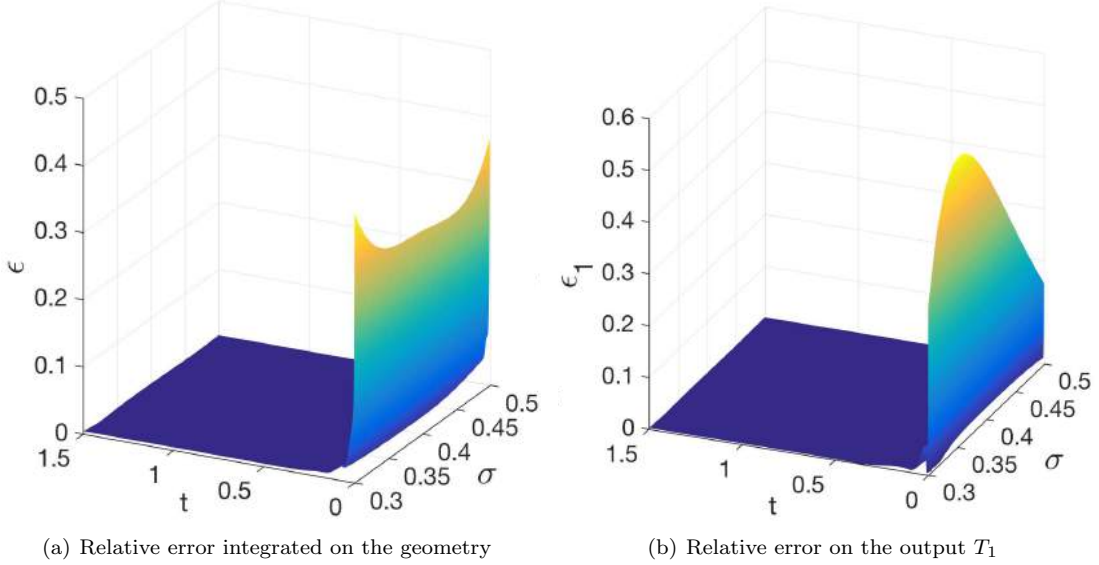
13

(a) Relative error integrated on the geometry   (b) Relative error on the output T_1

Figure 9: Relative local errors ϵ and ϵ_1 on $\Sigma \times I$

during the transient phase up to a very low value during the steady-state phase.

Those error quantifications can be partially justified with [13] [14] and [8] in which it is explained that the progressive Galerkin PGD method can encounters some difficulties to solve transient problems with convection. Indeed, to address problems where the bilinear form of the weak formulation is not symmetric it would be better to use a Petrov-Galerkin formulation. Furthermore, in this type of problem instabilities can be also encountered for some values of the Peclet number. The studied case is far from those instabilities but some other formulations (SUPGD [8] [11]) can be used to improve the stability of the solution on the opposite case.

In the following example the model error is included in the Bayesian identification. Here the discretization error made by the finite element method itself is not considered.

4.1.5   Bayesian inference with measurement error

In this section the model is supposed to be fully deterministic (no stochastic parameters **i** are considered). The only uncertainty source which is considered is a measurement error on the output. This output becomes randomized by adding a white noise.

The probability density function of the measurement noise $\pi_{\mathrm{err}}(e)$ is then modelled by a normal distribution with a zero mean and a 2% standard deviation (equal to the standard deviation used to simulate noise in the measurements).

According to the Bayesian identification process presented in Section 2 for a given value of the parameter σ the corresponding likelihood function value is given by (41):

$$\pi(T_1^{\mathrm{obs}}|\sigma) = \int \pi(T_1|\sigma).\pi_{\mathrm{err}}(T_1^{\mathrm{obs}} - T_1)\mathrm{d}T_1 \tag{41}$$

As the model is deterministic, we get:

$$\pi(T_1|\sigma) = \delta(T_1 - \mathcal{M}(x_1, y_1, t; \sigma)) \tag{42}$$

with δ the Dirac delta function and $\mathcal{M}$ the considered model (Finite Element or PGD).

In this case, the computation of the integral (41) is:

$$\mathcal{L}^{\mathrm{meas}}(\sigma) \equiv \int \pi(T_1|\sigma).\pi_{\mathrm{meas}}(T_1^{\mathrm{obs}} - T_1)\mathrm{d}T_1 \tag{43}$$

$$= \pi_{\mathrm{err}}(T_1^{\mathrm{obs}} - \mathcal{M}(x_1, y_1, t; \sigma)) \tag{44}$$

Figure 10 shows the likelihood functions obtained with the finite element model ($\mathcal{L}_{FE}^{\mathrm{meas}}$) and the PGD model ($\mathcal{L}_{PGD}^{\mathrm{meas}}$). The PGD model used is built with 20 modes. The computation is made at two different time steps for comparison.

(a) Measurement at $t = 0.032$ s
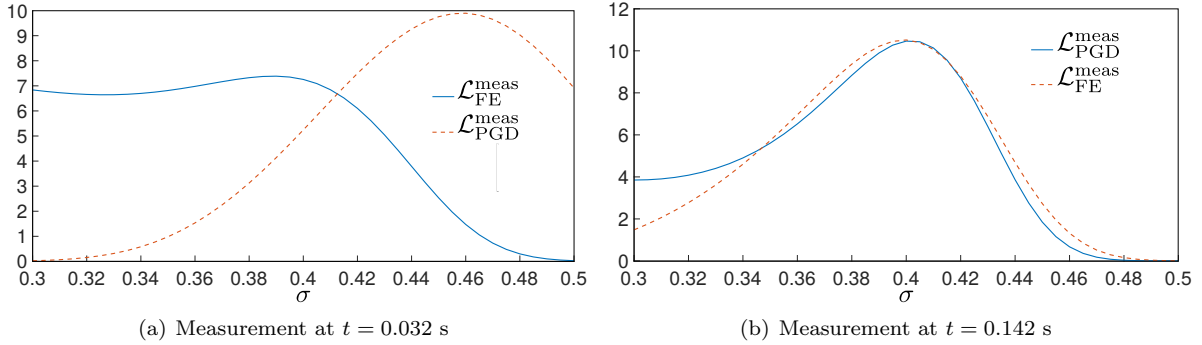
(b) Measurement at $t = 0.142$ s

Figure 10: Likelihood functions $\mathcal{L}^{\mathrm{meas}}$ for a single measurement at two different time steps. Comparison of the functions built with PGD and Finite Element models

The likelihood maximum should be obtained for $\sigma = 0.4$ (value used to simulate the measurements from the finite element model). For both models this maximum is shifted due to the measurement error. At $t = 0.142$ s the likelihood functions are very similar but quite different at $t = 0.032$ s. These results are directly linked to the error quantification analyzed before: the error is higher in the first time steps and nearly zero in the steady-state.

In the case of a model updating process in real time, a new measurement is available at each time step. According to Section 2, the posterior probability function at a given time step is improved considering the previous posterior as the current prior. In figure 11 a 25 measurements set process is considered. At each iteration the likelihood function

(a) Inference made with PGD model

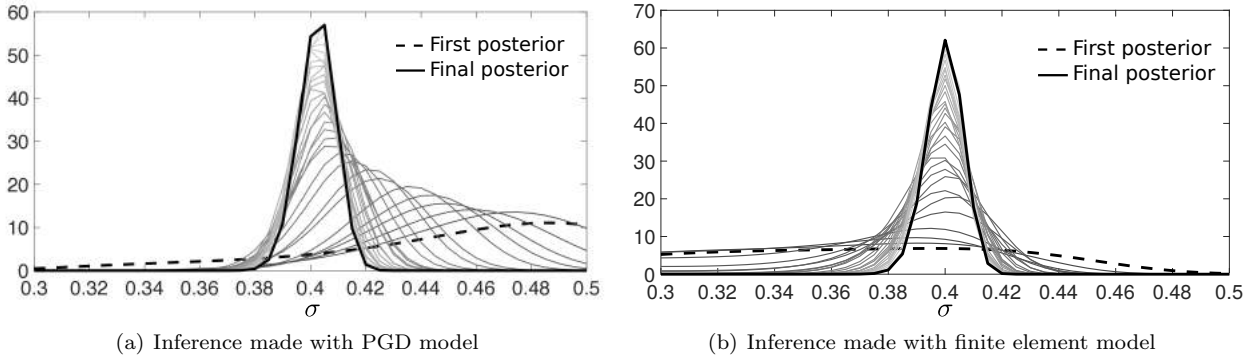(b) Inference made with finite element model

Figure 11: Posterior probability density functions after each measurement. The first posterior is represented with a thick discontinuous line and the final posterior with a thick continuous line. The intermediate posterior densities appear lighter during the iterations.

$\mathcal{L}^{\mathrm{meas}}$ is multiplied by the prior probability density function taken as the posterior of the previous iteration. A uniform density is considered as first prior. The posterior probability density function given by the first measurement is plotted with a thick discontinuous line and the final posterior with a thick continuous line. The intermediate posteriors appear lighter during the iterations. The final posterior density functions given by the finite element and PGD models are nearly equal with a maximum *a posteriori* for $\sigma = 0.4$. Furthermore, those densities have a lower variance giving more trust to the maximum value. Thus, in comparison to a single measurement, successive measurements can reduce the influence of both measurement and model errors.

4.1.6 Bayesian inference with model error

As explained in Section 2 the Bayesian approach is well-suited to take model error into account. The aim of this part is to study the influence of considering the model error to improve the Bayesian inference with the current

15

PGD model but also to study how the process can be implemented with a few number of PGD modes to reduce the computation cost.

The model error is considered owing to the quantification done in Section 4.1.4. For each value in the discretized parametric space, the maximum of the error ϵ_1 (equation (39)) over the time is considered. This gives the error function $\text{ErrMax}(\sigma)$ given in figure 12. Then, the model error is modeled by a uniform probability density function defined on the domain $\mathcal{D}_{\text{mod}} = [T_{\text{1PGD}} - \text{ErrMax}(\sigma).T_{\text{1PGD}}, T_{\text{1PGD}} + \text{ErrMax}((\sigma)).T_{\text{1PGD}}]$ and zero elsewhere. T_{1PGD} is the output value given by the PGD model.
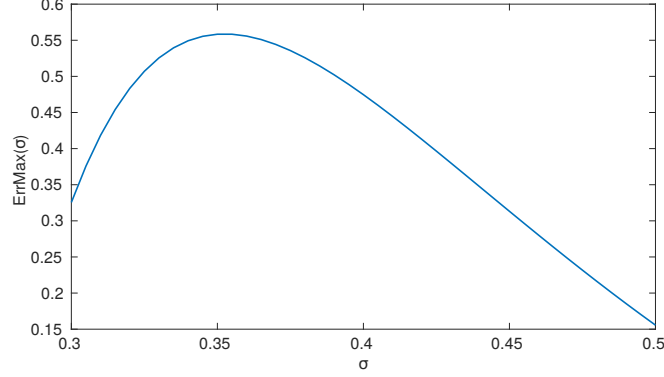

Figure 12: Maximum of the error ϵ_1 over the time interval as a function of the parameter σ

As in the previous part the likelihood function is given by:

$$\mathcal{L}^{\text{meas-mod}}(\sigma) \equiv \int \pi(T_1|\sigma).\pi_{\text{err}}(T_1^{\text{obs}} - T_1)\mathrm{d}T_1 \tag{45}$$

with:

$$\pi(T_1|\sigma) = \frac{1}{2.\text{ErrMax}(\sigma).T_{\text{1PGD}}}.\mathbf{1}_{\mathcal{D}_{\text{mod}}} \tag{46}$$

and $\mathbf{1}_{\mathcal{D}_{\text{mod}}}$ the indicator function on $\mathcal{D}_{\text{mod}}$.

The integral needs to be numerically computed. However, the PGD model gives an analytical probability density function so that the integration is performed numerically with a trapezoidal quadrature.

In figure 13 the influence of the model error is shown. $\mathcal{L}^{\text{meas-mod}}$ denotes the likelihood function built when the PGD model error is considered. $\mathcal{L}^{\text{meas}}_{\text{FE}}$ and $\mathcal{L}^{\text{meas}}_{\text{PGD}}$ denote likelihood functions built with Finite Element and PGD model respectively (without model error). In figure 13(a) the likelihood functions are computed for a time
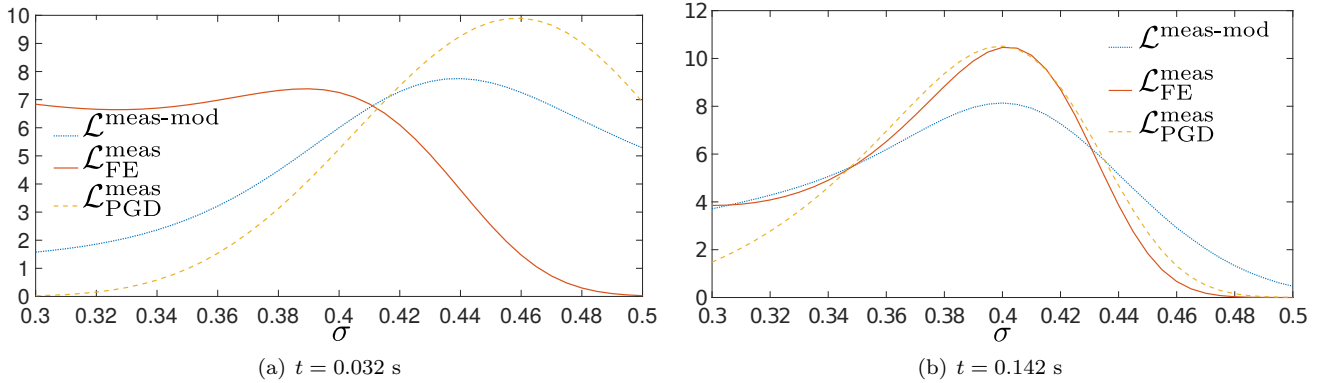

(a) $t = 0.032$ s

(b) $t = 0.142$ s

Figure 13: Likelihood functions with model and measurement errors for a single measurement

step of the transient state where the PGD model error is quite high. The likelihood function $\mathcal{L}^{\text{meas-mod}}$ is wider

than the likelihood function $\mathcal{L}_{\text{PGD}}^{\text{meas}}$ which gives less trust on a false estimation of the parameter. The likelihood maximum is also closer to the maximum given by the finite element model. The introduction of the model error in the identification process clearly improves the accuracy of the likelihood function. In figure 13(b) the time step considered provides a lower model error. Consequently, the likelihood functions $\mathcal{L}^{\text{meas}}$ given by finite element and PGD model are closer. However, the likelihood function $\mathcal{L}^{\text{meas-mod}}$ is much wider because of the error model. Indeed, as the maximum of the model error is considered for each value of the parameter σ, when the true model error is lower the resulting probability density function is less accurate.

In figure 14 the results of the likelihood functions $\mathcal{L}^{\text{meas-mod}}$ and $\mathcal{L}_{\text{PGD}}^{\text{meas}}$ built with only two PGD modes and the corresponding model errors are compared to $\mathcal{L}_{\text{FE}}^{\text{meas}}$. With this problem, when only two modes are considered the
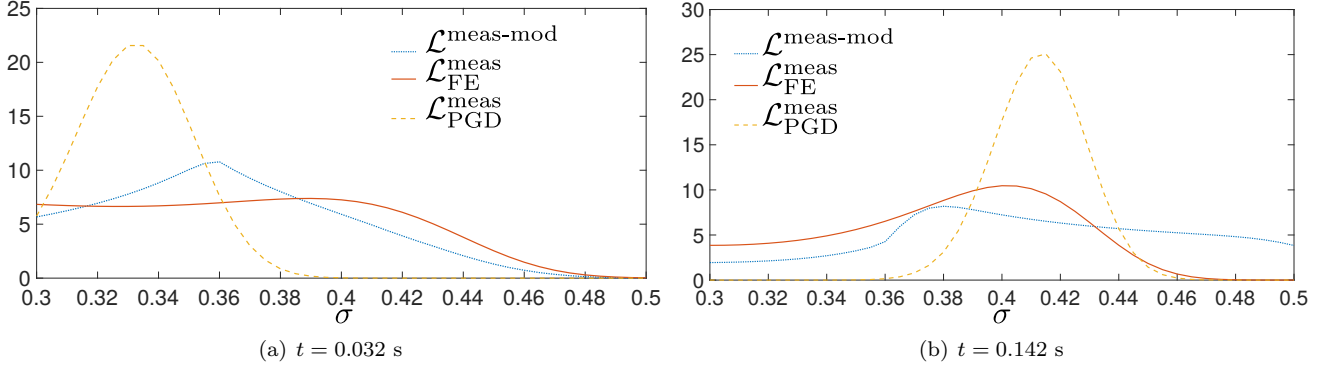

(a) $t = 0.032$ s

(b) $t = 0.142$ s

Figure 14: Likehood functions with two modes considering measurement and model error

error between PGD and FE models is high and the likelihood functions built are very wide giving lower information on parameter estimation. However, the results of the posterior probability density functions given for a set of measurements improve the estimation. In fact, in a real model updating process a compromise needs to be found between having many measurements or many modes to optimize the computation cost.

The computation of the model error has a crucial importance. Here, the error is quantified by comparing the finite element and PGD models for all values of parameter and time. In more complex cases, the error will be impossible to quantify and use online. Here, the Finite Model solution was the reference but it would be more accurate to have the error regarding the true solution. To that end, classic error estimation and reduction error estimation [17] [23] would be interesting to implement.

4.1.7 Uncertain parameter

In the previous test cases the interest of using a PGD model was to avoid solving a finite element problem for each value of the parameter σ during the iterations of the Bayesian identification process (and having an analytical form of this solution). However, considering no uncertain parameter a pre-computed finite element meta-model was used (to compare with the PGD results). Now, a source of uncertainty is considered in order to apply the methods of uncertainty propagation (algorithm 1 and 2). In this case, the response needs to be evaluated for all values of the parameter σ and the uncertain parameter.

First, the amplitude of the heat input u (equation (27)) is considered as an uncertain parameter (parameter **i** in figure 1). The amplitude u is then a random variable which is supposed to have a normal probability density function centered on the value used to simulate the measurements with a 3 % standard deviation.

Knowing the probability density function of the uncertain parameter i (here $i = u$) the change of variable can be done as shown in Section 3.3. As the parameter u has a proportional influence on the output, the change of variable function f is linear:

$$f(i) = i.T_{1u} \tag{47}$$

with T_{1u} the temperature value at the measurement point given by the PGD model for a unit amplitude. As the uncertain parameter is obviously decoupled from the other parameters and variables, the same PGD decomposition is kept:

$$T_m(x, y, t, \sigma, u) = u * \sum_{n=1}^{m} \Lambda_n(x, y)\lambda_n(t)\alpha_n(\sigma) \tag{48}$$

17

As the change of variable is strictly monotonic the change of variable theorem (21) is applied.

Figure 15 shows the probability density $\pi(T_1|\sigma = 0.345)$ built with the Monte-Carlo sampling (40000 samples) and the analytical change of variable. The two methods lead to the same density, providing at the end the same likelihood functions and posterior distributions.
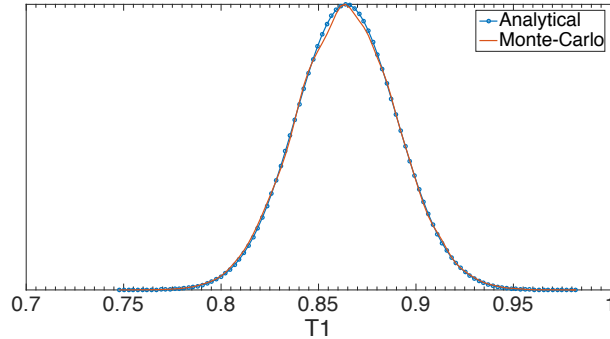

Figure 15: Probability density function $\pi(T_1|\sigma = 0.345)$

The probability density functions are known analytically so that the integration (45) to compute one point of the likelihood function with measurement error is still done numerically.

In a second time, we consider the Peclet number (Pe) as an uncertain parameter. The influence of the Peclet number on the output being no longer proportional a new PGD model (49) is done:

$$T_m(\mathbf{x}, t, \sigma, Pe) = \sum_{n=1}^{m} \Lambda_n(\mathbf{x})\lambda_n(t)\alpha_{1n}(\sigma)\alpha_{2n}(Pe) \tag{49}$$

The approach to build this new PGD model is the same as in the algorithm 3. The modes of this new PGD model are nearly the same as the previous model in terms of time, space and parameter σ. In figure 16 the modes associated with the parameter Pe are plotted.
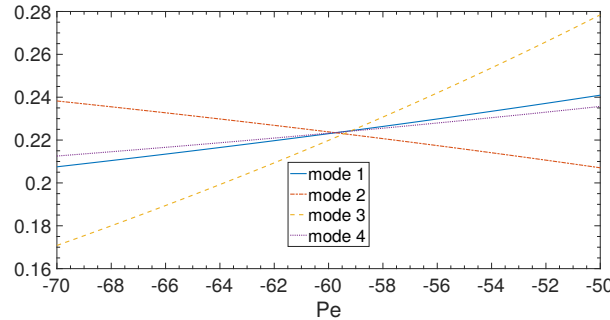

Figure 16: Parametric modes related to the Peclet number

To perform the Bayesian identification the uncertainty parameter Pe is assumed to have a normal distribution centered on the value used to simulate measurements with a 3 % standard deviation.

Considering this new model, the change of variable f is:

$$f : i \rightarrow \sum_{n=1}^{m} \Lambda_n(x_1, y_1)\lambda_n(t)\alpha_{1n}(\sigma)\alpha_{2n}(i) \tag{50}$$

However, this function is not necessarily monotonic. In figure 17 the f functions are plotted at the same time step $t = 0.034$, with the parameter values $\sigma = 0.3$ (figure 17(a)) and $\sigma = 0.35$ (figure 17(b)).

Therefore, to compute the likelihood function for a given time step, the function f has to be studied in order to determine in which part it is monotonic and to verify that it is not piecewise constant. To do so, the sign of $\frac{df}{di}$
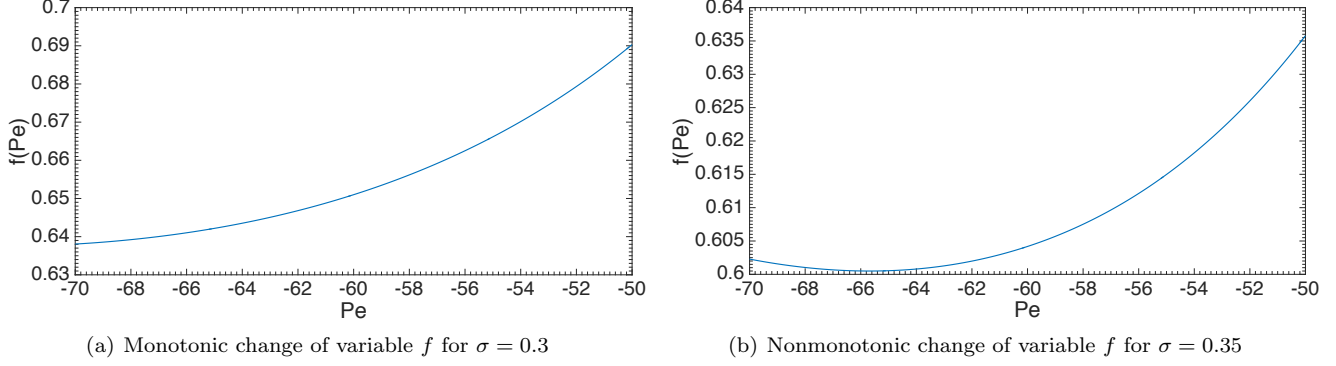
(a) Monotonic change of variable f for $\sigma = 0.3$

(b) Nonmonotonic change of variable f for $\sigma = 0.35$

Figure 17: Change of variable function f given by two values of parameter σ at $t = 0.034$ s

is studied for each value of σ. In each interval where $\frac{df}{di}$ keeps a constant sign, the inverses $f_k^{-1}(y)$ are computed. Then, the generalized form (22) is used to compute the probability density function given by the propagation of the uncertainty i through the model.

Figure 18 shows the probability density function $\pi(T_1|\sigma)$ computed with the same set of parameters as in figure 17(b). As seen before, the densities given by both analytical and Monte-Carlo uncertainty propagation are very similar leading to the same likelihood functions.
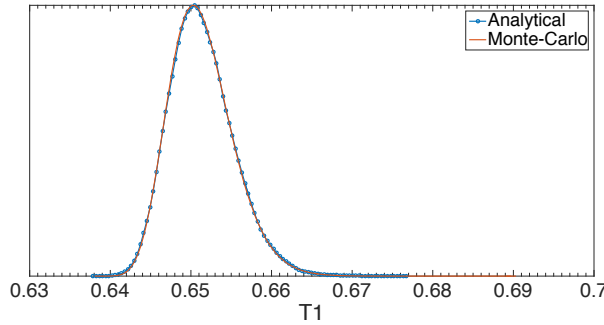

Figure 18: Probability density function $\pi(T_1|\sigma = 0.35)$ at $t = 0.034$ s

Finally, in figure 19 the posterior density functions obtained with both methods are plotted for a given time step $t = 0.034$ s. The prior is still taken uniform and the both methods give similar results.
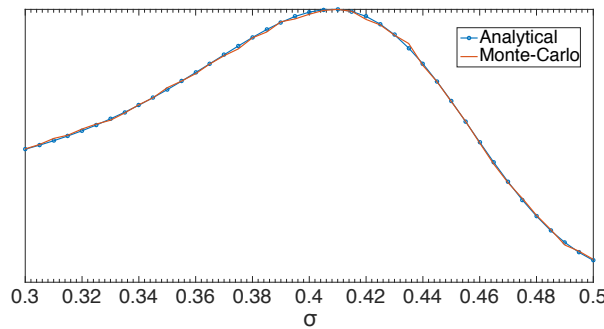
Figure 19: Final posterior built with analytical and Monte-Carlo methods with a single measurement at $t = 0.034$ s

With the change of variable method, studying the function f is costly especially if the function is not monotonic. Table 1 shows the results of the computation time required to compute one point of the likelihood function with: Finite Element model/Monte-Carlo method (40000 samples), PGD model/Monte-Carlo method (40000 samples)

and PGD model/analytical method. All computations are made on a MATLAB code on an Intel Xeon E5-1660 3.3 GHz × 6 PC.

Table 1: Comparison of the computational cost to build one point of the likelihood function

	Mean computation time	
Analytical PGD	Monte-Carlo PGD	Monte-Carlo FE
0.2678 s	2.5579 s	~ 4 h

For the sampling of one point of the likelihood function, the analytical approach leads to a 10 factor speed-up. Furthermore, this gain is multiplied by the number of the likelihood function points to compute, and the number of measurements. As a comparison, the Monte-Carlo method coupled with the direct Finite Element model gives a computation time of upper than 4 hours. In the context of DDDAS, using the direct model is impossible and those results show the potential of the PGD model order reduction in the Bayesian framework.

4.2 Glued assembly example: application to the estimation of hyper-parameters

4.2.1 Definition of the problem

The considered problem is a glued assembly adapted from [3]. The three structures Ω_1, Ω_2 and Ω_3 are glued by means of two elastic joints J_{12} and J_{23}. A force density $p = -50$ MPa is applied on the top of Ω_3 (figure 20).
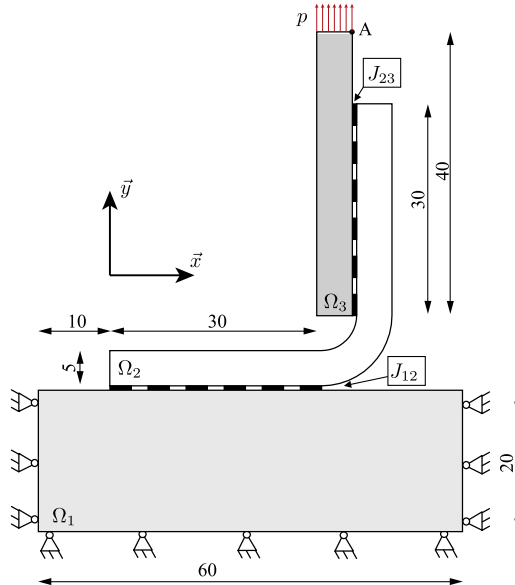

Figure 20: The geometry of the glued assembly

The purpose of this study is to identify the variability of the assembly process leading to the structure presented in figure 20. We assume that the glue joints can have random imperfections which lead to a variability of their stiffnesses.

Then, the Young modulus of the joints is defined as $E = e.E_0$ where e is a dimensionless random variable with a probability density function $\pi(e)$ and E_0 is supposed to be known. For each structure made by the assembly process, a simple test is done: a load p is applied at the top of the structure and the vertical displacement of the point A is measured. Based tests on different assemblies, the purpose is to identify the probability density function of the random variable e.

Unlike the previous example, the parameter to identify is no longer deterministic. However, the probability density function of the random variable e is assumed to be a normal distribution with the mean μ_e and the

standard deviation σ_e as unknown parameters:

$$\pi(e) = \frac{1}{\sigma_e\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{e-\mu_e}{\sigma_e}\right)^2\right) \tag{51}$$

Hence, the identification of the stochastic parameter leads to the identification of two deterministic parameters (μ_e, σ_e) called "hyper-parameters".

4.2.2 PGD solution

First, the PGD model of the problem is built in order to perform the Bayesian inference detailed in algorithm 1. The identification can be formulated as follows:

- Parameters to identify: σ_e and μ_e (defining the stochastic parameter e)

- System output: U, the vertical displacement of point A.

- Uncertain parameter: e.

The main difference of this example with the previous one is that the parameter is that the parameter e to identify leads directly to uncertainties in the model.

In order to proceed the Bayesian inference, the PGD model is searched as:

$$U_m(x_A, y_A, e) = \sum_{n=1}^{m} \Lambda_n(x_A, y_A).\alpha_n(e) \tag{52}$$

where (x_A, y_A) are coordinates of the point A. The reduced model is found thanks to the fixed-point algorithm 3. In figure 21, the first four parametric modes are plotted.
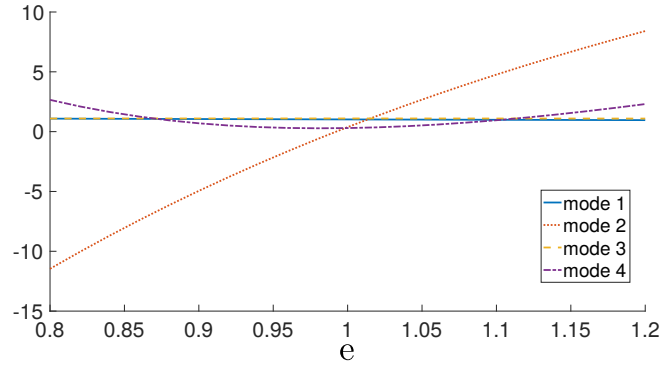

Figure 21: Parametric modes

The PGD model is compared to a direct finite element solution for a given value of the Young modulus amplitude $e = 0.8$ (figure 22). With one mode the relative error is $10^{-3.75}$ and the minimum error is $10^{-5.5}$. The PGD model for this static problem is very efficient.

4.2.3 Bayesian inference

In order to estimate the probability density function of the random variable e, the Bayesian inference method described in algorithm 3 is applied to the deterministic parameters to identify μ_e and σ_e. The Bayes formulation reads:

$$\pi(\mu_e, \sigma_e|u^{\text{obs}}) = \frac{1}{C}.\pi(u^{\text{obs}}|\mu_e, \sigma_e).\pi(\mu_e, \sigma_e) \tag{53}$$

In a first time, to compute the likelihood function, at each point $(\mu_e, \sigma_e) \in [0.9, 1.1] \times [0.01, 0.1]$ of the parameter domains, the uncertainties are propagated to the model assuming that the parameter e has a normal probability
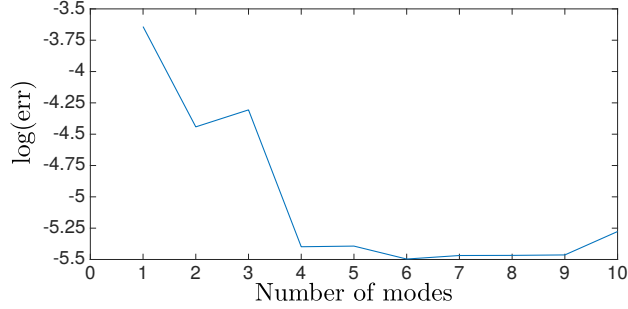
Figure 22: Relative error with respect to the number of modes

density function with a μ_e mean and a σ_e standard deviation. As in the previous example, to propagate the uncertainties through the PGD model the Monte-Carlo and analytical methods are leading to the same densities $\pi(u|\mu_e, \sigma_e)$. The corresponding point of the likelihood function is given by this probability density function evaluated at the measurement point u^{obs}. The measurements are simulated taking a sample of e in the probability density function with a mean $\mu_e = 1$ and a standard deviation $\sigma_e = 0.05$. An example of a 2D likelihood function given for a single measurement is plotted in figure 23.
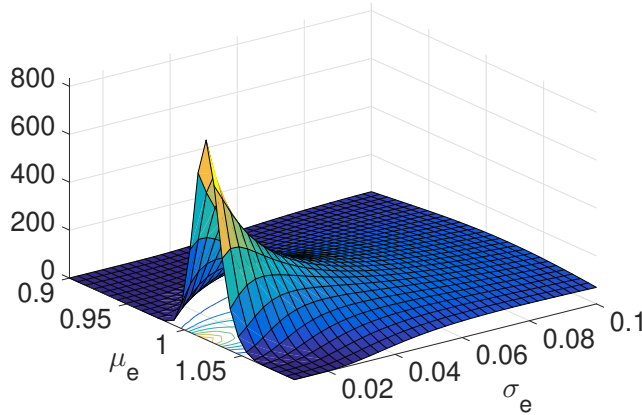

Figure 23: Likelihood function $\pi(u^{\text{obs}}|\mu_e, \sigma_e)$ for a single measurement

By multiplying this likelihood function by a prior probability density function (a uniform density function is chosen) the posterior probability density function is obtained. To obtain a separated posterior estimation of parameters, the marginal densities have to be computed:

$$\pi(\mu_e|u^{\text{obs}}) = \int_{0.01}^{0.1} \pi(\mu_e, \sigma_e|u^{\text{obs}})d\sigma_e \tag{54}$$

$$\pi(\sigma_e|u^{\text{obs}}) = \int_{0.9}^{1.1} \pi(\mu_e, \sigma_e|u^{\text{obs}})d\mu_e \tag{55}$$

The marginal densities given by the Bayesian inference from one measurement are shown in figure 24. The marginal density for the mean parameter is maximum for $\mu_e = 1$ which is the value used to simulate the measurements. The marginal density for the standard deviation is much wider and the maximum is obtained in the interval $[0.01, 0.025]$.

As in the welding example, the process can be improved doing the inference consecutively with several measurements and taking at each iteration the posterior of the previous step as the prior of the current step. In figure 25 the resulting posterior densities with consecutive simulated measurements are plotted. The first marginal density

22

(a) Marginal density of the mean parameter.

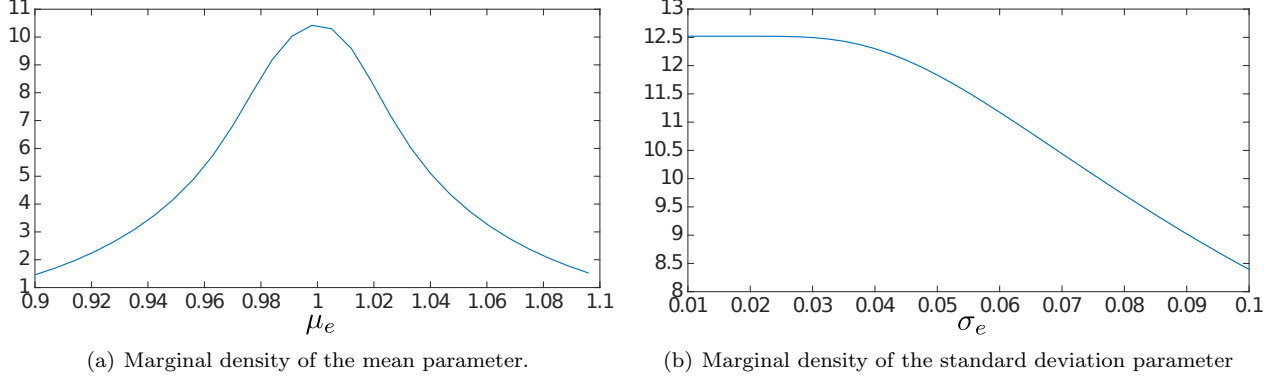(b) Marginal density of the standard deviation parameter

Figure 24: Marginal densities given by a single measurement

is plotted with a thick discontinuous line, the final posterior (after 20 measurements) with a thick continuous line and the intermediate densities appear lighter during the iterations.

(a) Marginal densities of the mean parameter.

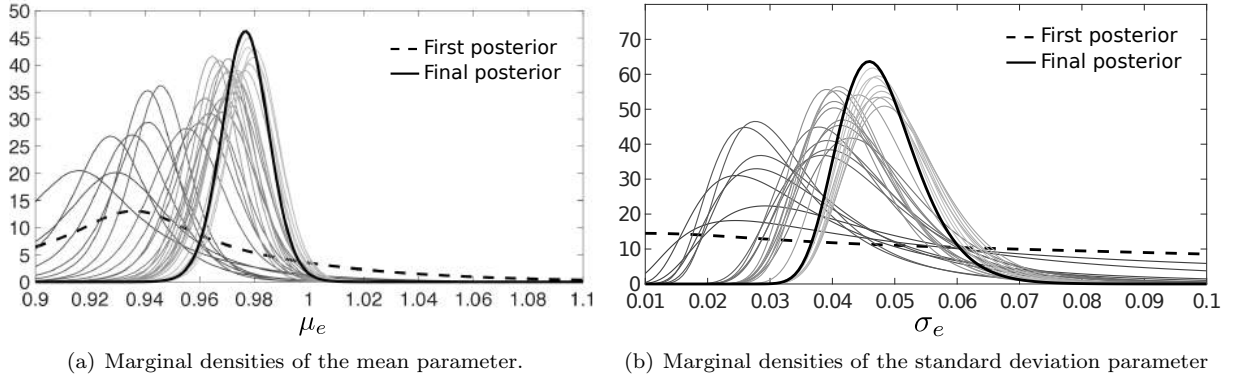(b) Marginal densities of the standard deviation parameter

Figure 25: Successive posterior marginal densities of a 20 successive measurements set

The marginal posterior probability density functions are more accurate during the iterations and the posterior marginal for the mean parameter seems to be converging much faster than the standard deviation parameter. The final result after 20 iterations gives an *a posteriori* maximum of 0.98 for the mean parameter and 0.045 for the standard deviation parameter (the values used to simulate measurements were $(\mu_e, \sigma_e) = (1, 0.05)$). In this case, the parameters estimation is very close to the true value. However, contrary to the estimation of a true deterministic parameter this estimation highly depends on the measurement sampled in the probability density function. Indeed, here only 20 measurements are considered (i.e. 20 draws on the normal distribution of e) which cannot be representative of the random variable e. Hence, the posterior marginals can be very different according to the sets of 20 successive measurements.

Figure 26 shows the variability of the identification regarding the measurements (with 20 successive measurements). Figures 26(a) and 26(b) represent respectively 25 marginal posterior densities of the mean and the standard deviation parameters with 20 successive measurements. Those figures show the distributions of the final posterior marginal densities after 20 successive measurements (only 25 posteriors are represented). To have a more quantitative information in those distributions, in figures 26(c) and 26(d) are plotted the densities of the Maximum A Posteriori (MAP) of 200 posterior marginal densities (always with sets of 20 measurements). Those densities are representing the variance of the identification process with the MAP estimator.

$$\mu_e^{\mathrm{MAP}} = \max_{\mu_e} \pi(\mu_e | u^{\mathrm{obs}}) \tag{56}$$

$$\sigma_e^{\mathrm{MAP}} = \max_{\sigma_e} \pi(\sigma_e | u^{\mathrm{obs}}) \tag{57}$$

23

(a) 25 posterior marginal densities of the mean parameter

(b) 25 posterior marginal densities of the standard deviation parameter

(c) Density of the estimator Maximum A Posteriori on 200 posterior marginal densities of the mean parameter

(d) Density of the estimator Maximum A Posteriori on 200 posterior marginal densities of the standard deviation parameter.
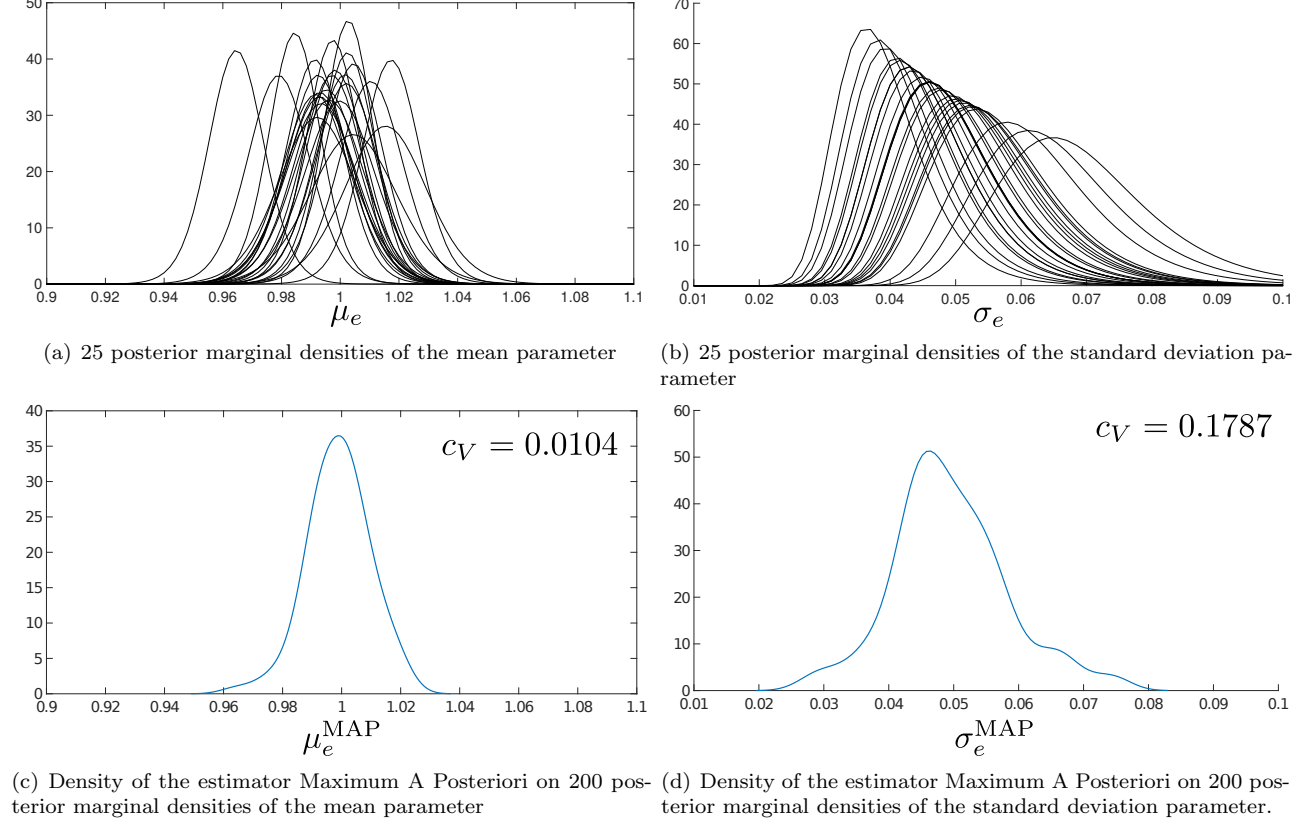
Figure 26: Results of the identification process with 20 successive measurements

The coefficient of variation for a probability density function $\pi(x)$ is also defined as:

$$c_V = \frac{\text{stdd}(\pi(x))}{\text{mean}(\pi(x))} \tag{58}$$

where stdd represents the standard deviation of the density.

the most likely estimation values with MAP estimator are $\mu_e^{\text{MAP}} = 1$ with $c_V = 1.0 \times 10^{-2}$ and $\sigma_e^{\text{MAP}} = 0.046$ with $c_V = 1.7 \times 10^{-1}$. The estimations of the parameter with the MAP estimator are close to the values used to simulate the measurements. However, according to the coefficients of variation the estimation of the parameter σ_e is much less accurate than the estimation of the parameter μ_e.

In order to show the influence of the size of the measurements sets, the same densities are plotted in figure 27 with sets of 60 successive measurements.

The results are significantly improved with coefficient of variation of 6.7×10^{-3} for the estimation of μ_e and 8.8×10^{-2} for the estimation of σ_e. Once again the estimation of the mean parameter offers less variability than the estimation of the standard deviation parameter.

Consequently, the estimation of hyper-parameters from a stochastic parameter is very difficult and the result highly depends on the quantity of information (measurements) available. However, this is the huge advantage of Bayesian identification: with a low amount of information (60 measurements) a large amount of information (a stochastic parameter) can be estimated correctly. A compromise has to be found between the knowledge needed on the parameter and the cost of additional measurements.

Similar to the welding example the model reduction error is studied. In figure 28 the relative error between the PGD models defined (52) with 10 modes and 1 mode is plotted:

$$\text{err} = \frac{|U_{10} - U_1|}{|U_1|} \tag{59}$$

24

(a) 25 posterior marginal densities of the mean parameter

(b) 25 posterior marginal densities of the standard deviation parameter

(c) Density of the estimator Maximum A Posteriori on 200 posterior marginal densities of the mean parameter

(d) Density of the estimator Maximum A Posteriori on 200 posterior marginal densities of the standard deviation parameter
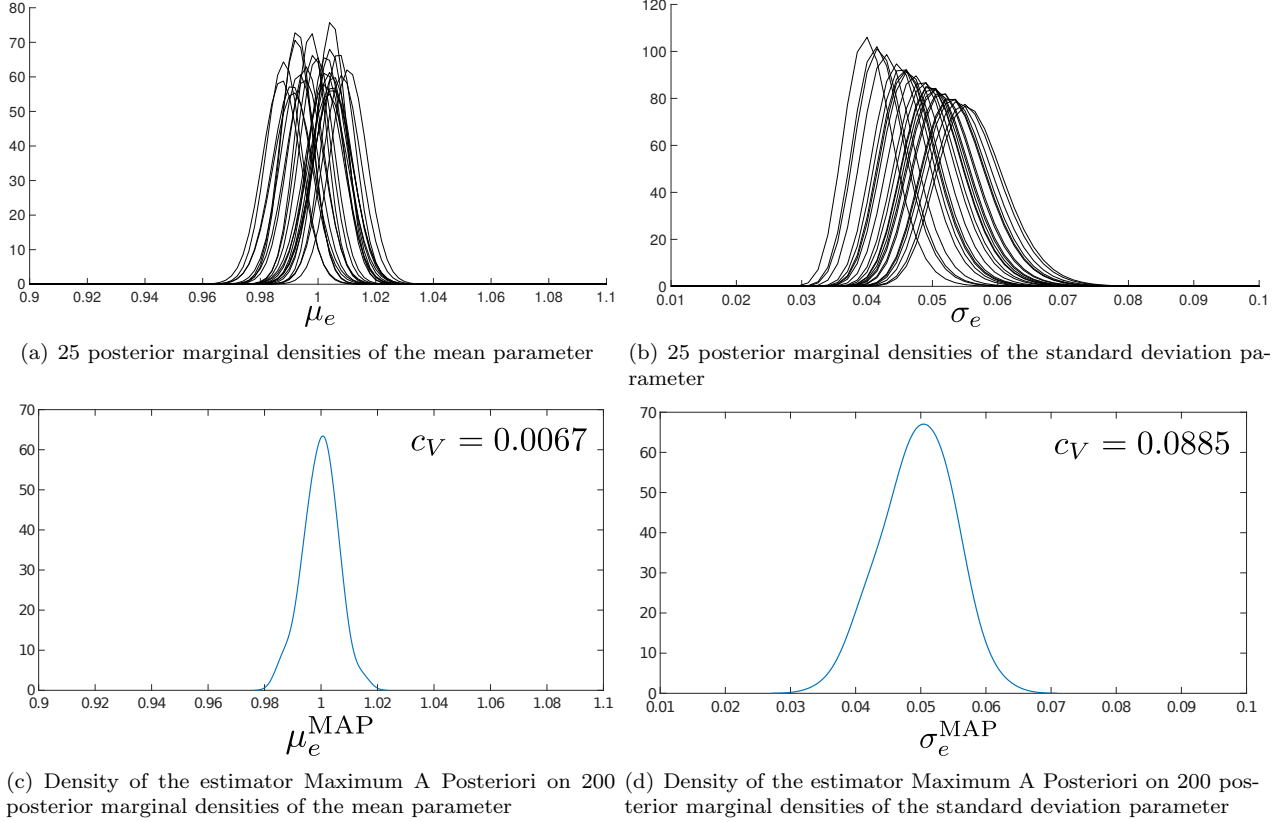
Figure 27: Results of the identification process with 60 successive measurements

The relative error between the two reduced order models is very low and this error seems to be even more regularized after the Bayesian inference process done. Indeed in figure 29 the relative error between the posterior marginals found with 10 modes and 1 mode model is presented. In this example the PGD reduction is so efficient that the Bayesian process can be done with only one mode.

From now on the measurements are simulated from the direct model with a Young modulus amplitude sampled under a normal distribution with a mean $\mu_e = 1$ and a standard deviation of $\sigma_e = 0.05$. Here, a Gaussian centered noise is added on the output with a $\sigma_{\mathrm{err}} = 0.003$ standard deviation.

The same computation as before is done except that the probability to have the measurements is no more direct but evaluated with the integration:

$$\pi(u^{\mathrm{obs}}|\mu_e, \sigma_e) = \int \pi(u|\mu_e, \sigma_e).\pi_{\mathrm{err}}(u^{\mathrm{obs}} - u)\mathrm{d}u \tag{60}$$

with $\pi_{\mathrm{err}}(e)$ the Gaussian probability density function with a standard deviation of σ_{err} and a zero mean. In figure 30, three likelihood functions are plotted. First, in figure 30(a) the likelihood function is built from a noisy measurement without considering the measurement error. In figure 30(b) the likelihood function built from the simulated measurement without the noise is represented. Finally, in figure 30(c) the likelihood function is built from the same noised measurement but taking into account the measurement error.

Figure 31 shows the results of the same computations in terms of marginal densities.

The ideal likelihood function which could be obtained from the given measurement is represented in figure 30(b) (respectively the "meas" graph in figure 31). The comparison between this likelihood and the likelihood function in figure 30(a) (respectively the "meas+noise" graph in figure 31) represents the error made if the measurement noise is not modeled. In this case, the likelihood maximum is shifted and as the density is very "sharp" the probability of having the true values of parameters is very low.

Figure 30(c) (respectively the "meas+noise+meas err" graph in figure 31) represents the likelihood function built from the noisy measurement but considering the measurement error. The modeling of measurement error
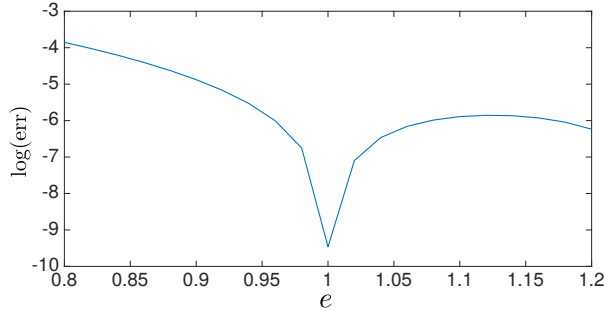
Figure 28: Relative error between PGD model with 1 and 10 modes

(a) Relative error for mean marginal density

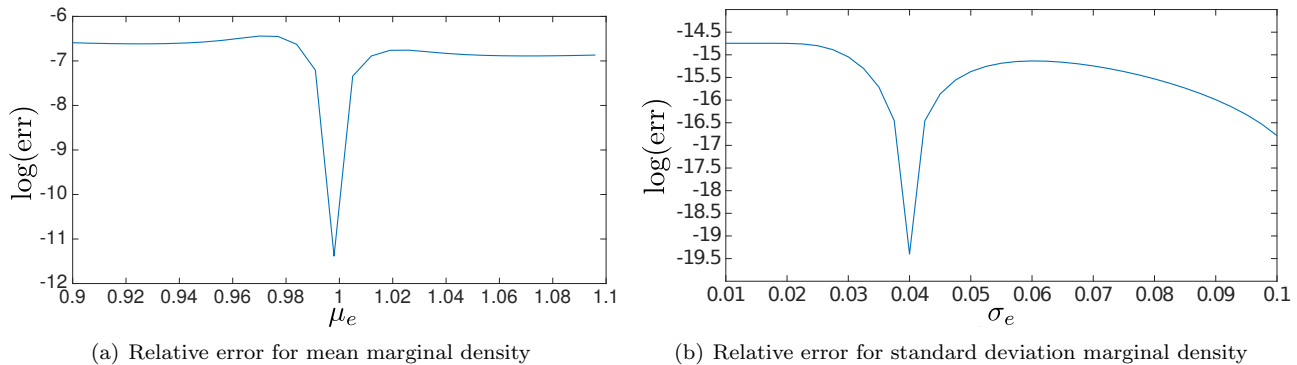(b) Relative error for standard deviation marginal density

Figure 29: Relative errors for posterior marginal densities

allows to have a much wider density. So, less trust is given to the likelihood maximum. Futhermore, this maximum is closer to the likelihood maximum obtained with the true measurement.

5   Conclusion

Through the examples studied, this paper shows the benefits of using a PGD model reduction formulation of the system studied for Bayesian inference in the context of fast model updating. The Bayesian inference has the advantage to always lead to well posed problems. However, the most difficult point is to build the likelihood function in order to have the posterior probability density function of the parameters to identify. In the general case without any assumption on the form of uncertainties, the likelihood function needs to be sampled with Monte-Carlo method. In considered problems, with a few number of parameters (1-2) the idea was to propagate the uncertainties through the model for all discretized values of parameters. In that way the likelihood function is built "point by point". In order to reduce the uncertainty propagation cost a PGD reduced model is used to obtain explicitly the problem solution for all values of Monte-Carlo draws. However, a more efficient method consists in using the PGD analytical expression to directly compute the probability density functions after uncertainty propagation. This method shows a greater speed-up and seems to be well-suited for low dimensional model updating problems (with only one stochastic parameter). The explicit formulation given by the PGD shows great benefits for Bayesian updating leading to analytical computation of integrals, uncertainty propagation at a very low cost (summation of modes).

Futhermore, when the dimension of the parameters $\mathbf{p}$ is high, the integrations over the parametric space, which are needed to compute estimator and marginalization, will lead to a high computation cost. To that end samples from the posterior are needed to perform Monte-Carlo integration. A good framework to sample the posterior density in the Bayesian framework is to use Markov Chain Monte Carlo (MCMC) methods [2][4][5]. For comparison the MCMC method (with Metropolis-Hastings algorithm) is applied to the previous numerical examples. The length of the Markov Chain was set to 50000 which is a lower bound of the values found in the literature [4][6]. The

(a) Likelihood function built from measurement with noise without considering measurement error

(b) Likelihood function built from true measurement (without noise)

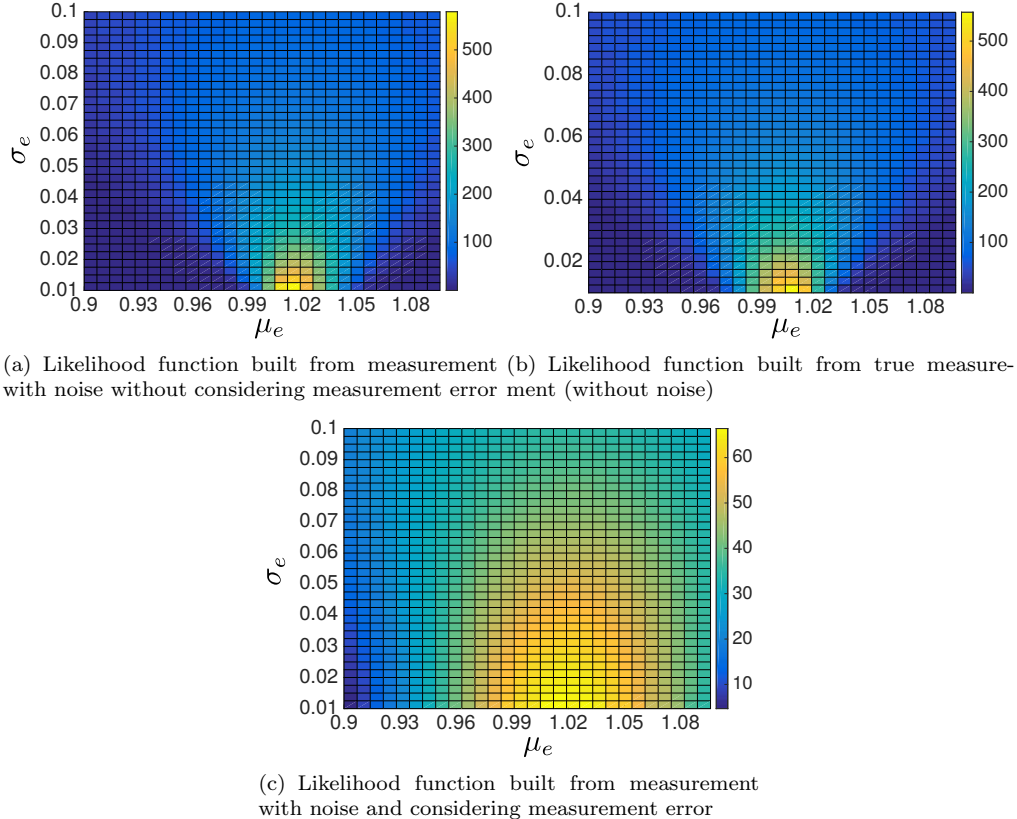(c) Likelihood function built from measurement with noise and considering measurement error

Figure 30: Likelihood functions regarding measurement error

computation times to build posterior marginal densities are compared between the "grid" and the MCMC methods in Table 2. For each method the PGD model is used with the "PGD-analytic" method presented in Section 3 to propagate uncertainties. For the grid method the number of elements in the grid is specified. All computations are made on a MATLAB code on an Intel Xeon E5-1660 3.3 GHz × 6 PC. In the studied cases (low dimension) the MCMC method is obviously costlier than the grid method. Despite the fact that additional numerical integration is required in the grid method, the number of points in the grid (i.e. number of model evaluations) is lower than 50000 (length of the chain). The welding example is more costly than the glued assembly example because of the number of modes, the form of uncertainties and the change of variable function study.

Table 2: Comparison of the computational cost to build marginal posterior densities

Mean computation time			
Welding		Glued assembly	
Grid (41)	MCMC	Grid (37×29)	MCMC
10.56 s	~ 12000 s	6.63 s	~ 1300 s

The Bayesian framework allows an easy treatment of model and measurement errors. However, in this paper the model error is used thanks to error quantization between Finite Element and PGD models. In future work the aim will be to use error estimation methods [22] [23] to model the error committed regarding the true solution. Eventually, another idea will be to use the coupled Bayesian-PGD formulation in the context of data assimilation as an less restrictive alternative of Kalman filter [24][25].

(a) Marginal densities for mean parameter

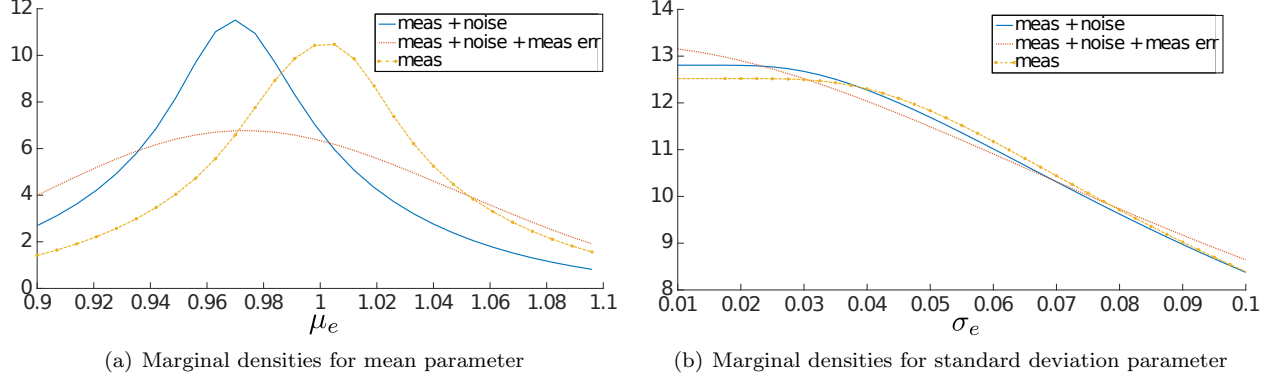(b) Marginal densities for standard deviation parameter

Figure 31: Likelihood marginal densities regarding measurement error. "meas" represents the marginal densities corresponding to the likelihood function in figure 30(b), "meas + noise" to the figure 30(a) and "meas+noise+meas err" to the figure 30(c).

6 Appendix

In this section the details of a PGD solving is illustrated on the welding example from Section 4.

6.1 Problem

The same problem as in Section 4 is considered with the convection-diffusion equation:

$$\frac{\partial T}{\partial t} + \underline{v}.\underline{\text{grad}}T - \Delta T = s \tag{61}$$

With:

$$s(x, y; \sigma) = \frac{u}{2\pi\sigma^2}\exp\left(-\frac{(x - x_c)^2 + (y - y_c)^2}{2\sigma^2}\right) \tag{62}$$

The purpose is to build a multiparametric reduced order model with separation of space, time and parameter σ.

6.2 Progressive Galerkin PGD

As presented in Section 3 the PGD modes are built recursively thanks to the Galerkin orthogonality. The spaces of variation of each parameter is defined as follows: $I = [0, T_f]$ the time interval and $\Sigma = [\sigma_{\min}, \sigma_{\max}]$, the space of variation of σ. The admissible field spaces are defined:

$$\mathcal{T} = \{T \in H^1(\Omega =]0; 5[\times]0; 1[), T = 0 \text{ on } \Gamma_D\} \tag{63}$$

$$\mathcal{I} = \{T, \int_I \|T(x, y, .; \sigma)\|_{H^1(\Omega)}^2 < \infty, \forall (x, y, \sigma) \in \Omega \times \Sigma\} \tag{64}$$

$$\mathcal{E} = \{T, \int_\Sigma \|T(x, y, t; .)\|_{H^1(\Omega)}^2 < \infty, \forall (x, y, t) \in \Omega \times I\} \tag{65}$$

The weak formulation of equation (61) on each space reads:
Find $T \in \mathcal{T} \otimes \mathcal{I} \otimes \mathcal{E}$, such as $\forall T^* \in \mathcal{T} \otimes \mathcal{I} \otimes \mathcal{E}$:

$$a(T, T^*) = l(T^*) \tag{66}$$

with:

$$a(T, T^*) = \int_{I \times \Sigma \times \Omega} \frac{\partial T}{\partial t}.T^* + \underline{v}.\underline{\text{grad}}T.T^* + \underline{\text{grad}}T.\underline{\text{grad}}T^* \, dt d\sigma d\Omega \tag{67}$$

$$l(T^*) = \int_{I \times \Sigma \times \Omega} s.T^* \, dt d\sigma d\Omega \tag{68}$$

28

The solution is searched in the separated form:

$$T_m(x,y,t;\sigma) = \sum_{n=1}^{m} \Lambda_n(x,y)\lambda_n(t)\alpha_n(\sigma) \tag{69}$$

The $m-1$ first modes are supposed to be known and the m-th mode is searched. Then the solution reads:

$$\{T_m(x,y,t;\sigma)\} = \sum_{n=1}^{m-1} \lambda_n(t)\alpha_n(\sigma)\Lambda_n(x,y) + \lambda(t)\alpha(\sigma)\Lambda(x,y) \tag{70}$$

The unknowns are the functions: Λ, λ et α.
The test field $T^* \in \mathcal{T} \otimes \mathcal{I} \otimes \mathcal{E}$ is taken in the separated form:

$$T^* = \lambda^*\alpha\Lambda + \lambda\alpha^*\Lambda + \lambda\alpha\Lambda^* \tag{71}$$

Using this form, the variational formulation (66) leads to decoupled problems with the applications S_m, T_m, P_m such as:

$$\Lambda = S_m(\lambda, \alpha) \tag{72}$$
$$\lambda = T_m(\alpha, \Lambda) \tag{73}$$
$$\alpha = P_m(\lambda, \Lambda) \tag{74}$$

6.2.1 Spatial application S_m

The decoupled weak formulation for space problem reads:

$$a\left(T_{m-1} + \lambda\alpha\Lambda, \lambda\alpha\Lambda^*\right) = l(\lambda\alpha\Lambda^*) \tag{75}$$

with:

$$a\left(\lambda\alpha\Lambda, \lambda\alpha\Lambda^*\right) = \int_{I\times\Sigma\times\Omega} \lambda\dot{\lambda}\alpha^2\Lambda\Lambda^* + \alpha^2\lambda^2\underline{v}.\mathrm{grad}\Lambda\Lambda^* + \alpha^2\lambda^2\underline{\mathrm{grad}}\Lambda\underline{\mathrm{grad}}\Lambda^* d\Omega d\sigma dt \tag{76}$$

The use of a P1 discretization on all fields reads:

$$\Lambda = [N_x]\{\Lambda\} \tag{77}$$
$$\lambda = [N_t]\{\lambda\} \tag{78}$$
$$\alpha = [N_\sigma]\{\sigma\} \tag{79}$$

where $[N_\bullet]$ represents the shape functions matrix and $\{\bullet\}$ the nodal values of the fields.
Then:

$$a\left(\lambda\alpha\Lambda, \lambda\alpha\Lambda^*\right) = \{\Lambda^*\}^T[A_\Lambda]\{\Lambda\} \tag{80}$$

with:

$$[A_\Lambda] = \left(\{\alpha\}^T[M_\sigma]\{\alpha\}\right).\left[\left(\{\lambda\}^T[H_t]\{\lambda\}\right)[M] + \left(\{\lambda\}^T[M_t]\{\lambda\}\right)[C_H]\right] \tag{81}$$

Where the following matrices are defined:

$$[M_\bullet] = \int_\bullet [N_\bullet]^T[N_\bullet]d\bullet \tag{82}$$

$$[H_\bullet] = \int_\bullet [dN_\bullet]^T[N_\bullet]d\bullet \tag{83}$$

$$[C_\bullet] = \int_\bullet [dN_\bullet]^T[dN_\bullet]d\bullet \tag{84}$$

$$[C_H] = \kappa.[C_x] + [H_x] \tag{85}$$

Likewise:

$$a\left(T_{m-1}, \lambda\alpha\Lambda^*\right) = \{\Lambda^*\}^T \sum_{n=1}^{m-1} [A_{\Lambda_n}]\{\Lambda_n\} \tag{86}$$

with:

$$[A_{\Lambda_n}] = \left(\{\alpha\}^T [M_\sigma]\{\alpha_n\}\right) . \left[\left(\{\lambda\}^T [H_t]\{\lambda_n\}\right) [M] + \left(\{\lambda\}^T [M_t]\{\lambda_n\}\right) [C_H]\right] \tag{87}$$

$$l(\{\lambda\alpha\Lambda^*\}) = \int_{I\times\Sigma\times\Omega} \lambda(t)\alpha(\sigma)s(x,y;\sigma).\Lambda^*(x,y)dt d\sigma d\Omega \tag{88}$$

The right-hand side of the variational formulation can be written as:

$$l(\lambda\alpha\Lambda^*) = \{\Lambda^*\}^T\{Q_\Lambda\} \tag{89}$$

However the integrand is not represented in a separated form. In order to do so, an asymptotic expansion at the center σ_0 of the P1 element at the d order is used.

The volumic load s is approximated as:

$$s(x,y;\sigma_0 + \delta\sigma) \approx \sum_{i=0}^{d} \frac{\partial^i s}{\partial\sigma^i}(x,y;\sigma_0)\frac{\delta\sigma^i}{i!} \tag{90}$$

which leads to:

$$\int_{\sigma_{\min}}^{\sigma_{\max}} s(x,y;\sigma)\alpha(\sigma)d\sigma = \sum_{k=1}^{N} \int_{\sigma_k}^{\sigma_{k+1}} s(x,y;\sigma)\alpha(\sigma)d\sigma \tag{91}$$

$$= \sum_{k=1}^{N} \int_{\sigma_k - \sigma_{0k}}^{\sigma_{k+1} - \sigma_{0k}} s(x,y;\sigma_{0k} + \tilde{\sigma})\alpha(\sigma_{0k} + \tilde{\sigma})d\tilde{\sigma} \tag{92}$$

$$= \sum_{k=1}^{N}\sum_{i=0}^{d} \int_{\sigma_k - \sigma_{0k}}^{\sigma_{k+1} - \sigma_{0k}} \frac{\tilde{\sigma}^i}{i!}\alpha(\sigma_{0k} + \tilde{\sigma})d\tilde{\sigma}\frac{\partial^i s}{\partial\sigma^i}(x,y;\sigma_{0k}) \tag{93}$$

$$= \sum_{k=1}^{N}\sum_{i=0}^{d} \int_{\sigma_k}^{\sigma_{k+1}} \frac{(\sigma - \sigma_{0k})^i}{i!}\alpha(\sigma)d\sigma\frac{\partial^i s}{\partial\sigma^i}(x,y;\sigma_{0k}) \tag{94}$$

In practice $d = 1$ is sufficient to have a good approximation with the P1 discretization of the interval Σ.

The right-hand side reads now:

$$l(\lambda\alpha\Lambda^*) = \{1_t\}^T [M_t]\{\lambda\}.\sum_{k=1}^{N}\sum_{i=0}^{d} \int_{\sigma_k}^{\sigma_{k+1}} \frac{(\sigma - \sigma_{0k})^i}{i!}\alpha(\sigma)d\sigma\{\Lambda^*\}^T\{S_{ik}\} \tag{95}$$

Finally the application S_m leads to a linear system at each iteration of the fixed-point algorithm with the unknown $\{\Lambda\}$.

6.2.2 Application temporelle T_m

For the time application a Runge-Kutta algorithm with automatic adjustment of the time step is used to solve the encountered ordinary differential equation:

$$a.\dot{\lambda}(t) + b.\lambda(t) = c - \sum_{n}^{m-1} a_n.\dot{\lambda}_n(t) + b_n.\lambda_n(t) \tag{96}$$

with:

$$a = \int_{\Sigma\times\Omega} \alpha^2\Lambda^2 d\sigma d\Omega = \left(\{\alpha\}^T [M_\sigma]\{\alpha\}\right)\left(\{\Lambda\}^T [M]\{\Lambda\}\right) \tag{97}$$

$$b = \int_{\Sigma\times\Omega} \alpha^2 \left(\underline{v}.\underline{\text{grad}}\Lambda\Lambda + \underline{\text{grad}}\Lambda\underline{\text{grad}}\Lambda\right) d\sigma d\Omega = \left(\{\alpha\}^T [M_\sigma]\{\alpha\}\right)\left(\{\Lambda\}^T [C_H]\{\Lambda\}\right) \tag{98}$$

$$c = \int_{\Sigma \times \Omega} \alpha(\sigma) s(x,y;\sigma).\Lambda(x,y) dt d\sigma d\Omega = \sum_{k=1}^{N} \sum_{i=0}^{d} \int_{\sigma_k}^{\sigma_{k+1}} \frac{(\sigma - \sigma_{0k})^i}{i!} \alpha(\sigma) d\sigma \{\Lambda\}^T \{S_{ik}\} \tag{99}$$

$$a_n = \int_{\Sigma \times \Omega} \alpha \alpha_n \Lambda \Lambda_n d\sigma d\Omega = \left(\{\alpha\}^T [M_\sigma]\{\alpha_n\}\right) \left(\{\Lambda\}^T [M]\{\Lambda_n\}\right) \tag{100}$$

$$b_n = \int_{\Sigma \times \Omega} \alpha \alpha_n \left(\underline{v}.\underline{\mathrm{grad}} \Lambda \Lambda_n + \underline{\mathrm{grad}} \Lambda \underline{\mathrm{grad}} \Lambda_n\right) d\sigma d\Omega = \left(\{\alpha\}^T [M_\sigma]\{\alpha_n\}\right) \left(\{\Lambda\}^T [C_H]\{\Lambda_n\}\right) \tag{101}$$

At each iteration of the fixed-point algorithm an ordinary differential equation on the unknown λ is solved.

6.2.3 Parametric application P_m

Here, the same approach used in the spatial application is used. The decoupled weak form for the parametric problem reads:

$$a\left(T_{m-1} + \lambda \alpha \Lambda, \lambda \alpha^* \Lambda\right) = l(\lambda \alpha^* \Lambda) \tag{102}$$

$$a\left(\lambda \alpha \Lambda, \lambda \alpha^* \Lambda\right) = \int_{I \times \Sigma \times \Omega} \lambda \dot{\lambda} \alpha \alpha^* \Lambda^2 + \alpha \alpha^* \lambda^2 \underline{v}.\underline{\mathrm{grad}} \Lambda \Lambda + \alpha \alpha^* \lambda^2 \underline{\mathrm{grad}} \Lambda \underline{\mathrm{grad}} \Lambda dt d\sigma d\Omega \tag{103}$$

A P1 discretization leads to:

$$a\left(\lambda \alpha \Lambda, \lambda \alpha^* \Lambda\right) = \{\alpha^*\}^T [A_\alpha]\{\alpha\} \tag{104}$$

with:

$$[A_\alpha] = [\left(\{\Lambda\}^T [M]\{\Lambda\}\right)\left(\{\lambda\}^T [H_t]\{\lambda\}\right) + \left(\{\Lambda\}^T [C_H]\{\Lambda\}\right)\left(\{\lambda\}^T [M_t]\{\lambda\}\right)][M_\sigma] \tag{105}$$

The contribution of the previous modes reads:

$$a\left(T_{m-1}, \lambda \alpha^* \Lambda\right) = \{\alpha^*\}^T \sum_{n=1}^{m-1} [A_{\alpha_n}]\{\alpha_n\} \tag{106}$$

with:

$$[A_{\alpha_n}] = [\left(\{\Lambda\}^T [M]\{\Lambda_n\}\right)\left(\{\lambda\}^T [H_t]\{\lambda_n\}\right) + \left(\{\Lambda\}^T [C_H]\{\Lambda_n\}\right)\left(\{\lambda\}^T [M_t]\{\lambda_n\}\right)][M_\sigma] \tag{107}$$

Finally the right-hand side reads:

$$l(\lambda \alpha^* \Lambda) = \{1_t\}^T [M_t]\{\lambda\}.\sum_{k=1}^{N} \sum_{i=0}^{d} \int_{\sigma_k}^{\sigma_{k+1}} \frac{(\sigma - \sigma_{0k})^i}{i!} \alpha^*(\sigma) d\sigma \{\Lambda\}^T \{S_{ik}\} \tag{108}$$

At each iteration of the fixed-point algorithm a linear system is solved with the unknown $\{\alpha\}$.

References

[1] Tarantola A. Inverse Problem Theory. *SIAM* 2005.

[2] Kaipo J, Somersalo E. Statistical and computational inverse problems. *Springer* 2005.

[3] Louf F, Champaney L. Fast validation of stochastic structural models using a pgd reduction scheme. *Elsevier*, 2013.

[4] Cui T, Marzouk Y, Willcox K. Data-driven model reduction for the Bayesian solution of inverse problems. *SIAM Review*, 2014.

[5] Green PL, Worden K. Bayesian and Markov chain Monte Carlo methods for identifying nonlinear systems in the presence of uncertainty *Phil.Trans.R.*, 2015.

[6] Berger J, Orlande H. R. B, Mendes N. Proper Generalized Decomposition model reduction in the Bayesian framework for solving inverse heat transfer problems.

Inverse Problems in Science and Engineering, 2016.

[7] Gogu C. Facilitating bayesian identification of elastic constants through dimensionality reduction and response surface methodology. *PhD thesis*, École Nationale Supérieure des Mines de Saint-Étienne, 2009.

[8] Chinesta F, Keunings R, Leygue A. The Proper Generalized Decomposition for Advanced Numerical Simulations. *Springer*, 2014.

[9] Grepl M. Reduced-Basis Approximation and A Posteriori Error Estimation. *PhD Thesis*, 2005.

[10] Bouclier R, Louf F, Chamoin L. Real-time validation of mechanical models coupling PGD and constitutive relation error. *Computational Mechanics* 2013; **52**(4):861–883.

[11] Maday Y, Manzoni A, Quarteroni A. An online intrinsic stabilization strategy for the reduced basis approximation of parametrized advection-dominated problems. *Springer*, 2014.

[12] Marzouk Y, Najm H. Dimensionality reduction and polynomial chaos acceleration of Bayesian inference problems. *Journal of Computational Physics* 2009; **228**(6):1862–1902.

[13] Nouy A. A priori model reduction through Proper Generalized Decomposition for solving time-dependent partial differential equations. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**(23-24):1603–1626.

[14] Allier P E, Chamoin L, Ladevèze P. Proper generalized decomposition computational methods on a benchmark problem: introducing a new strategy based on constitutive relation error minimization. *Springer*, 2015.

[15] Huynh D. B. P., Nguyen N. C., Rozza G., Patera A. T. Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized PDEs. 3(January), 2007.

[16] Peherstorfer B, Willcox K. Dynamic data-driven reduced-order models. *Computer Methods in Applied Mechanics and Engineering*, 291:21–41, 2015.

[17] Ammar A, Chinesta F, Díez P, Huerta A. An error estimator for separated representations of highly multidimensional models. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**(25-28):1872–1880.

[18] Chamoin L, Allier P E, Marchand B. Synergies between the Constitutive Relation Error concept and PGD model reduction for simplified V& V procedures. *Advanced Modeling and Simulation in Engineering* 2016; accepted.

[19] Chinesta F, Ladevèze P, Cueto E. A short review on model order reduction based on proper generalized decomposition. *Archives of Computational Methods in Engineering* 2011; **18**:395–404.

[20] Vitse M, Néron D, Boucard P.A. Virtual charts of solutions for parametrized nonlinear equations. *Computational Mechanics*. Vol 54. Num 6. Pages 1529-1539. 2014

[21] Courard A, Néron D, Ladevèze P, Ballère L. Integration of PGD-virtual charts into an engineering design process. *Computational Mechanics* 2016; **57**(4):637–651.

[22] Ladevèze P, Chamoin L. On the verification of model reduction methods based on the proper generalized decomposition. *Computer Methods in Applied Mechanics and Engineering* 2011; **200**(23-24):2032–2047.

[23] Ladevèze P, Chamoin L. Toward guaranteed PGD-reduced models. *Bytes and Science, CIMNE, Barcelona* 2012.

[24] Kalman RE. A new approach to linear ltering and prediction problems. *Journal of Basic Engineering* 1960; 82(1): 3545

[25] Marchand B., Chamoin L., Rey C. Real-time updating of structural mechanics models using Kalman filtering, modified Constitutive Relation Error, and Proper Generalized Decomposition. *International Journal for Numerical Methods in Engineering*. Vol 107. Num 9. Pages 786-810. 2016

[26] Darema F. Dynamic Data Driven Applications Systems: A new paradigm for applications simulations and measurements. *Computational Science-ICCS 2004: 4th International Conference, Springer* 2004; 662-669

[27] Ladevèze P, Nedjar D, Reynier M. Updating of finite element models using vibrations tests. *AIAA Journal* 1994; **32**(7), pp1485-1491.

[28] Allix O, Feissel P, Nguyen H. Identification strategy in the presence of corrupted measurements. *Engineering Computations* 2005; 487-504.

[29] Ladevèze P. The large time increment method for the analysis of structures with non-linear behavior described by internal variables. *Comptes rendus de l'académie des sciences, Serie II* 1989; **309**(11):10951099.

[30] Beck JL. Bayesian system identification based on probability logic. *Structural Control and Health Monitoring* 2010; **17**(7):825–847

[31] Jensen HA, Vergara C, Papadimitrou C, Millas E. The use of updated robust reliability measures in stochastic dynamical systems. *Computer Methods in Applied Mechanics and Engineering* 2010; **267**:825–847

[32] Yan WJ, Katafygiotis LS. A novel Bayesian approach for structural model updating utilizing statistical modal information from multiple setups. *Structural Safety* 2015; **52**:260-271.

[33] Papadimitriou C, Papadioti DC. Component mode synthesis techniques for finite element model updating. *Computers and Structures* 2013; **126**:15-28

[34] Gogu C, Yin W, Haftka RT, Ifju P, Molimard J, Le Riche R, Vautrin, A. Bayesian Identification of Elastic Constants in Multi-Directional Laminate from Moiré Interferometry Displacement Fields *Experimental Mechanics* 2013; **53**(4):635-648.

[35] Allaire D, Chambers J, Cowlagi R, Kordonowy D, Lecerf M, Mainini L, Ulker F, Willcox K. An Offline/Online DDDAS Capability for Self-Aware Aerospace Vehicles *Procedia Computer Science* 2013; **18**: 19591968.