université PARIS-SACLAY

NNT : 2019SACLN055

Thèse de doctorat

Stratégies numériques innovantes pour l'assimilation de données par inférence bayésienne

Thèse de doctorat de l'Université Paris–Saclay
préparée à l'École Normale Supérieure Paris-Saclay

École doctorale n°579 : Sciences mécaniques et énergétiques, matériaux et géosciences
Spécialité de doctorat : Solides, structures et matériaux

Thèse présentée et soutenue à Cachan, le 15 octobre 2019, par

M. Paul-Baptiste Rubio

Composition du Jury :

M. Antonio Huerta Professeur, Universitat Politècnica de Catalunya	Rapporteur
M. Youssef Marzouk Associate professor, Massachusetts Institute of Technology	Rapporteur
M. Pierre Kerfriden Assistant Professeur HDR, Mines ParisTech Senior Lecturer, Cardiff University	Examinateur
M. Pierre-Étienne Charbonnel Ingénieur de recherche, CEA Paris-Saclay	Examinateur
M. Marc Bonnet Directeur de recherche, ENSTA ParisTech	Président
M. Stéphane Roux Directeur de recherche, École normale supérieure Paris-Saclay	Examinateur
M. Ludovic Chamoin Professeur, École normale supérieure Paris-Saclay	Directeur de thèse
M. François Louf Maître de conférence, École normale supérieure Paris-Saclay	Co-Directeur de thèse

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences mécaniques et énergétiques, matériaux et géosciences (SMEMAG)

Titre : Stratégies numériques innovantes pour l'assimilation de données par inférence bayésienne

Mots clés : simulation numérique, réduction de modèle, recalage de modèle, inférence bayésienne, quantification d'incertitudes, temps réel

Résumé : Ce travail se place dans le cadre de l'assimilation de données en mécanique des structures. Il vise à développer de nouveaux outils numériques pour l'assimilation de données robuste et en temps réel afin d'être utilisés dans diverses activités d'ingénierie. Une activité cible est la mise en œuvre d'applications DDDAS (Dynamic Data Driven Application System) dans lesquelles un échange continu entre les outils de simulation et les mesures expérimentales est requis dans le but de créer une boucle de contrôle rétroactive sur des systèmes mécaniques connectés. Dans ce contexte, et afin de prendre en compte les différentes sources d'incertitude (erreur de modélisation, bruit de mesure,...), une méthodologie stochastique puissante est considérée dans le cadre général de l'inférence bayésienne. Cependant, un inconvénient bien connu d'une telle approche est la complexité informatique qu'elle engendre et qui rend les simulations en temps réel et l'assimilation séquentielle des données difficiles. Le travail de thèse propose donc de coupler l'inférence bayésienne avec des techniques numériques attrayantes et avancées afin d'envisager l'assimilation stochastique de données de façon séquentielle et en temps réel. Premièrement, la réduction de modèle PGD est introduite pour faciliter le calcul de la fonction de vraisemblance, la propagation des incertitudes dans des modèles complexes et l'échantillonnage de la densité a posteriori. Ensuite, l'échantillonnage par la méthode des Transport Maps est étudiée comme un substitut aux procédures classiques MCMC pour l'échantillonnage de la densité a posteriori. Il est démontré que cette technique conduit à des calculs déterministes, avec des critères de convergence clairs, et qu'elle est particulièrement adaptée à l'assimilation séquentielle de données. Là encore, l'utilisation de la réduction de modèle PGD facilite grandement le processus en utilisant les informations des gradients et hessiens d'une manière simple. Enfin, et pour accroître la robustesse, la correction à la volée du biais du modèle est abordée à l'aide de termes d'enrichissement fondés sur les données. Aussi, la sélection des données les plus pertinentes pour l'objectif d'assimilation est abordée. Cette méthodologie globale est appliquée et illustrée sur plusieurs applications académiques et réelles, comprenant par exemple le recalage en temps réel de modèles pour le contrôle des procédés de soudage, ou l'étude d'essais mécaniques impliquant des structures endommageables en béton instrumentées par mesures de champs.

Université Paris–Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences mécaniques et énergétiques, matériaux et géosciences (SMEMAG)

Title: Development of innovative numerical strategies for Bayesian data assimilation

Keywords: numerical simulation, model reduction, Bayesian inference, uncertainty quantification, real-time simulation

Abstract: The work is placed into the framework of data assimilation in structural mechanics. It aims at developing new numerical tools in order to permit real-time and robust data assimilation that could then be used in various engineering activities. A specific targeted activity is the implementation of DDDAS (Dynamic Data Driven Application System) applications in which a continuous exchange between simulation tools and experimental measurements is envisioned to the end of creating retroactive control loops on mechanical systems. In this context, and in order to take various uncertainty sources (modeling error, measurement noise,..) into account, a powerful and general stochastic methodology with Bayesian inference is considered. However, a well-known drawback of such an approach is the computational complexity which makes real-time simulations and sequential assimilation some difficult tasks. The PhD work thus proposes to couple Bayesian inference with attractive and advanced numerical techniques so that real-time and sequential assimilation can be envisioned. First, PGD model reduction is introduced to facilitate the computation of the likelihood function, uncertainty propagation through complex models, and the sampling of the posterior density. Then, Transport Map sampling is investigated as a substitute to classical MCMC procedures for posterior sampling. It is shown that this technique leads to deterministic computations, with clear convergence criteria, and that it is particularly suited to sequential data assimilation. Here again, the use of PGD model reduction highly facilitates the process by recovering gradient and Hessian information in a straightforward manner. Eventually, and to increase robustness, on-the-fly correction of model bias is addressed using data-based enrichment terms. The overall cost-effective methodology is applied and illustrated on several academic and real-life test cases, including for instance the real-time updating of models for the control of welding processes, or that of mechanical tests involving damageable concrete structures with full-field measurements.

Université Paris–Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

Remerciements

Enfin la rédaction de cette page de remerciements. L'écriture de celle-ci signifie la fin de cette thèse mais aussi la fin de l'aventure cachanaise après sept années...

Tout d'abord merci à mon jury et particulièrement à Antonio Huerta et Youssef Marzouk qui ont eut la lourde tache de rapporter ces travaux. Merci aussi à Antonio Huerta pour sa bonne humeur, son enthousiasme et les questions posées durant la soutenance. Un grand merci aussi à Youssef Marzouk pour sa disponibilité et sa pédagogie, je suis très heureux et fier de pouvoir continuer à travailler sur ses thématiques de recherche avec lui au MIT. Ma reconnaissance va aussi aux autres membres de mon jury : Marc Bonnet qui a présidé ce jury et avec qui j'ai indirectement fais mes premiers pas dans le monde du calcul numérique grâce à son livre sur la méthodes des éléments finis, Pierre-Étienne Charbonnel et Pierre Kerfriden pour l'intérêt qu'ils ont porté à mes travaux et les questions soulevées lors de la soutenance, Stéphane Roux pour son sens critique et son expertise que j'avais déjà pu apprécier pendant mes trois années au LMT.

Cette thèse je la doit avant tout à mes deux encadrants : François Louf et Ludovic Chamoin. Merci à eux de m'avoir recueilli sur leur sujet ambitieux. Je tiens à les remercier pour m'avoir mis en confiance et d'avoir instaurer une ambiance de décontraction et de bonne humeur dès le début. Merci aussi à Ludo d'avoir tout au long de la thèse voulu valoriser nos travaux lors des conférences et différents articles. Ainsi en ayant le recul de ces trois ans j'ai eu beaucoup de chance d'avoir eu cette configuration encadrement/sujet et j'espère que l'on sera amener à retravailler ensemble.

Grâce à l'environnement offert par le LMT et des qualités humaines de ses membres je dois aussi une partie de mes travaux grâce à des collaborations moins formelles. Tout d'abord merci à François Soleilhet avec qui j'ai pu réaliser la campagne expérimentale *Odysseus*. Merci ensuite à Matthieu Vitse qui m'a fournit son code et son aide précieuse pour le calcul du modèle PGD d'endommagement. Merci également à Clément Jailin qui m'a beaucoup aidé durant la dernière année, tant d'un point de vue scientifique et technique que par sa vision globale de la recherche. D'un point de vue général merci à tous ceux avec qui j'ai pu discuter de mon sujet de thèse au CdC, en pause au bar, par sms etc. Je tiens aussi à remercier Farid, Amélie, Bruno, Charles, Marie, Gabriel, Xuyang, Yannis, Richard, Philippe et Bhimal d'avoir répondu présents lors de mes pré-soutenances.

D'un point de vue amical je me réjouit également de toutes les rencontres que j'ai pu faire grâce à la thèse: mes amis "matériaux" et l'équipe des filles GC avec qui j'ai pu partager de nombreuses pauses, les gens du CdC que j'ai pu déranger régulièrement

sans qu'ils m'en tiennent rigueur. Merci à Thomas, François, Camille, Maxime, Réjane, Richard, Clément, Manon, Marie, Stéphane, Charles, Boubou et Orestis avec qui j'ai pu partagé toutes sortes d'activité ludiques et sportives en dehors du labo. Ses liens dans le labo sont aidés et facilités grâce aux "piliers" du labo. Une grosse pensée pour notre Frisou qui était indéniablement l'un d'entre eux et qui nous a quitté trop tôt.

Je tiens aussi à remercier tout ceux qui m'on permis d'enchainer les quatre années à l'ENS en particulier Romain et Guillaume qui m'ont aider et supporter pendant ces années.

Parallèlement à toutes ces années je ne peux oublier le soutien de mes amis *Spignifochi di Veru* qui m'ont permis de réviser mes cours de prépa, les partiels de l'ENS et lire des articles de biblio pour la thèse durant nos gardes ensemble.

Enfin, merci à famille et mes proches de m'avoir soutenu pendant toutes ces années et être venus en nombre à Cachan depuis la Corse et Orléans pour ma soutenance.

Cursichella sfurtunata
mamma di tanti figlioli
sarai prestu abandunata
si qualchi rimediu ùn trovi

Pampasgiolu di l'Acquale -
Lamentu di Cursichella

Contents

Introduction

Context and applications of data assimilation

Due to the constant advances in the development of numerical models, they are increasingly used to study physical systems. These numerical models are based on physical principles that are often derived into systems of partial differential equations. The efficient computation tools provided through the years allow to use them to predict the behavior of physical systems. However, models always depend on parameters that can vary for each application case. Thus, they need to be calibrated with data collected from the studied physical system in order to take the most of their accuracy. This data assimilation has for objective to fit at best models with observed physics.

Data assimilation has many applications in mechanical engineering with different time constraints. The first application which is worth mentioning is the identification of mechanical material properties. This classical application aims to fit the material parameters of a postulated constitutive law (e.g. elastic isotropic behavior) and fit the material properties (e.g. Young Modulus and Poisson ratio) using classical tests and associated measurements (e.g. tensile test of Figure 1). In this case, there is no real-time constraint, data can be collected prior to the identification procedure and post-processed when the test is over. Another application of data assimilation is health monitoring, particularly for civil engineering structures [18, 25]. In this application, *in situ* measurements are collected in order to predict the evolution of damage and certify the good health of the structure. Figure 2 shows the monitoring of crack opening in a historical construction thanks to a linear variable differential transformer sensor. In this framework, measurements are treated on the fly during the lifetime of the structure. Thus, data assimilation has to be performed with quite large real-time constraints with respect to the time scale of the observations.

However, applications of data assimilation with the strongest real-time constraints are the ones involved in the Dynamic Data Driven Application Systems framework [39]. In this context, the numerical model is used to control a physical system, which can only be performed if the model precisely and constantly describes the evolving system. In order to do so, *in situ* measurements have to be sequentially assimilated in order to dynamically update parameters of the model in real-time (see Figure 3). This challenging framework with many applications (see http://www.dddas.org for examples) requires the elaboration of fast and robust data assimilation techniques.

These applications are only possible if useful data are available. In the industrial con-

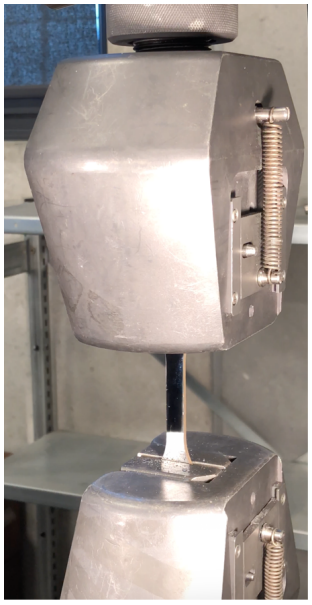


Figure 1: Tensile test on steel in order to characterize mechanical properties after heat treatment

Figure 2: Monitoring of crack opening at Basilica Papale di Santa Maria Maggiore (Photo credit: T. Heitz)

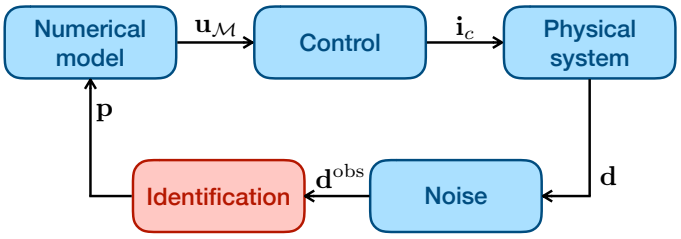


Figure 3: DDDAS principle

text, due to cost and accessibility constraints, data can be limited with a poor sensitivity with respect to the parameters of interest that makes data assimilation even more difficult. However, in mechanical engineering an appealing measurement method is given by Digital Image Correlation (DIC)[60, 141] and Digital Volume Correlation (DVC) [19]. These techniques allow one to determine the full-field displacement of a structure from successive pictures or 3D scans. Then, the measured field can be used for parameter estimation [6]. This remote measurement can also be used in the context of health monitoring [42, 96] and can even be envisioned in the DDDAS framework as in [64].

Data assimilation: an inverse problem

In the presented applications, the purpose is to use measurements in order to update the knowledge on model parameters. In most cases, these parameters cannot be measured directly their estimation from indirect measurements leads to the solution of a so-called inverse problem. Figure 4 represents the considered model environment for studied systems.

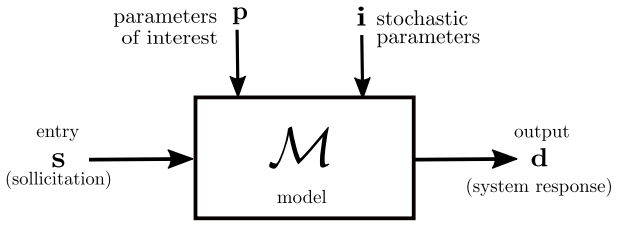


Figure 4: Model environment

The physical system is assumed to be 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 system of partial differential equations (PDEs). 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 provide 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. The model is assumed to have two kinds of parameter dependency.

- First, 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:

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

However, the determination of parameters from indirect noisy observations often leads to ill-posed inverse problems [143] in the Hadamard sense [6]. In other terms, the solution can not verify one or more of those properties:

1. a solution exists
2. the solution is unique
3. the solution changes continuously with problem inputs

The ill-posedness of the problem depends on the quality of the information that is given by the observed data. A high noise level and few discretized observations of the system output will lead to an ill-posed problem. Moreover, engineering applications are plagued with multiple sources of uncertainties (e.g. modeling error, unknown variation of environment parameters or corrupted sensors) that need to be accounted for in order to perform data assimilation.

Consequently, solving inverse problems is a challenging numerical task especially in the extremely demanding DDDAS framework. It entails the use of systems that deliver relevant data in (near) real-time to computational models of the evolution of physical phenomena of interest, so as to predict and control outputs and meet a set of objectives [40]. To that end, efficient numerical methods have to be developed in order to construct an advanced (fast, robust, effective) strategy for the numerical processing of a large collection of noisy data, obtained from continuous updates and being assimilated sequentially in time. This calls for real-time and robust computational strategies that will be addressed in this thesis.

Thesis contributions

The framework of this thesis deals with data assimilation in order to update numerical models. The purpose is to convert noisy and indirect observational data into useful characterizations of the unknown parameters of a numerical model, in order to be able to precisely describe the state of a physical system with the updated numerical model. Thus, degrees of freedom of the physical system state are assumed to be restricted to a few number of stationary parameters. The objective of this work is to provide a robust

and fast data assimilation framework based on Bayesian inference and advanced numerical techniques (model reduction, fast sampling, automatic correction of model bias,...) that is able to take into account all uncertainties which may be encountered. After this brief introduction, the manuscript is organized as follows:

- Chapter 1 is a state-of-the-art on inverse problems and model order reduction methods. A particular attention is paid and developments are made for the Bayesian formulation of inverse problems and the Proper Generalized Decomposition (PGD) method.

- In Chapter 2, a first coupling is performed between the PGD model reduction method and Bayesian inference. This study focuses on fast computation of the posterior density when different types of uncertainties occur such as measurement error, model error and uncertain parameters. Here, few parameters are considered and the posterior densities are characterized with sampling on a regular grid. Interests and performance of such a method are highlighted on several numerical examples.

- In Chapter 3, the context is the sequential inference of numerical models. The purpose is to improve the knowledge on model parameters in real-time. An algorithm coupling Transport Map and PGD model reduction methods is proposed in order to efficiently compute the solution of sequential Bayesian inference problems. Performance of the algorithm is shown in a numerical example in the context of real-time simulation of a welding process.

- In the first part of Chapter 4, the experimental campaign performed during this PhD is presented. This study aims to characterize the damage in a notched concrete beam during a three points bending test. In the second part of this chapter, the algorithm developed in Chapter 3 is applied in the context of the experimental campaign in order to sequentially update the parameters of a PGD damage model. Then, the updated model is used to predict the crack propagation in the beam.

- In Chapter 5, developments are made in order to learn model bias from data. A method to build and identify the model bias is proposed in order to improve accuracy of sequential Bayesian inference solutions. This method is applied to numerical examples for behavior and PGD truncation errors.

- The objective of Chapter 6 is to optimize the use of the Digital Image Correlation in Bayesian inference with PGD models. First, a method is presented in order to select only the most relevant information based on sensitivity with respect to the variation of parameters and with respect to noise. Second, a method to improve the sensitivity of the field measurement itself is proposed by optimization of the speckle pattern. The methods are applied to the measurements obtained during the bending tests on the concrete beams.

- Eventually, general conclusions and limits of this work are drawn. Prospects and future works with different complexities are also mentioned.

General computational tools settings This work aims to compare the computation cost of different methods. Unless otherwise specified, the computations are performed with the following settings:

- A medium effort has been done in order to optimize an in-house software developed in Matlab or Python.
- The seed of the random generator number is fixed for each code run.
- The computations are performed using a laptop computer with 2.7 GHz Intel Core i5 without parallelization.

Chapter 1

State-of-the-art

This chapter aims to present the classical methods to solve inverse problems and their couplings with reduced order modeling. Robustness and computational cost of such approaches will be discussed.

1.1 Classical methods to solve inverse problems

In this section, the focus is on the classical methods to regularize and solve inverse problems.

1.1.1 Deterministic methods

In the deterministic framework, inverse problems are formulated in terms of functional minimization. The solution of the inverse problem is a deterministic vector $\mathbf{p}^{\text{est}}$. This solution is found by the minimization of a functional $\mathcal{J}$:

$$\mathbf{p}^{\text{est}} = \underset{\mathbf{p}\in\mathcal{P}}{\text{argmin}}\ \mathcal{J}(\mathbf{p}) \tag{1.1}$$

Once the functional $\mathcal{J}$ is determined, classical minimization algorithms can be used to solve the inverse problem. Thus, in this framework, the main technicity of inverse problems is focused on the development of a functional that can produce in a robust way a good estimation of the parameters of interest. In the most common cases, the functional will be based on the discrepancy between the model output for a given value of $\mathbf{p}$ and the observed data.

1.1.1.1 Least-squares formulation

The most usual and easiest way to compute the discrepancy model prediction and the observed data is to use the euclidian norm (L^2 norm when data are functions instead of samples). In that case, the functional reads:

$$\mathcal{J}_{\text{LS}}(\mathbf{p}) = ||\mathcal{M}(\mathbf{p}) - \mathbf{d}^{\text{obs}}||_2^2 \tag{1.2}$$

Following the notations of the introduction, $\mathcal{M}$ is the model of interest, with $\mathbf{d} = \mathcal{M}(\mathbf{p})$ the output that can be compared to $\mathbf{d}^{\text{obs}}$. In that case, the parameter estimation is the value

of parameters which provides the minimal residual estimation in the least-squares sense. This objective function is the most common choice as it can be applied without considering the physics of the problem and can be used in any situation, for any kind of parameters. This method is also interesting for practical reasons as it only requires the evaluation of the model in a non-intrusive way. However, the general framework provided by this formulation has its counterparts. This method can have low performance depending on the problem nature. For example, this formulation is not robust when data are corrupted with a non-uniformly distributed noise [133].

In order to improve the lack of robustness of the classical least-squares formulation, regularizing terms can be added. First, when statistical information is available concerning the noise level, a weighted least-squares functional can be introduced under the form:

$$\mathcal{J}_{\text{WLS}}(\mathbf{p}) = \left(\mathcal{M}(\mathbf{p}) - \mathbf{d}^{\text{obs}}\right)^T \mathbb{C}^{-1} \left(\mathcal{M}(\mathbf{p}) - \mathbf{d}^{\text{obs}}\right) \tag{1.3}$$

where $\mathbb{C}$ is the covariance matrix of the measurement noise. Thus, in this formulation the contribution of each observation to the final result is weighted by its corresponding noise level. This approach can be determined by uncertainty quantification of the noise level before data assimilation. If no information about noise is known *a priori*, $\mathbb{C}$ is taken as the identity matrix leading to the classical formulation (1.2).

Another issue with this least-square formulation is that the ill-posedness of the inverse problem can be preserved. If the number of experimental data is not sufficient, the solution provided by the least-squares minimization can be sensitive to the initial conditions. The functional $\mathcal{J}_{\text{WLS}}$ can also have several local minima and the solution can be not unique. Thus, the minimization can be very difficult and costly. A method that can help the minimization of a least-squares functional is the introduction of a regularization term $\mathcal{R}(\mathbf{p})$ known as Tikhonov regularization [12]:

$$\mathcal{J}_{\text{WLS-R}}(\mathbf{p}) = \mathcal{J}_{\text{WLS}}(\mathbf{p}) + \alpha\mathcal{R}(\mathbf{p}) \tag{1.4}$$

where α is a weighting term associated to the regularization. The most common choice for the regularization term is $\mathcal{R}(\mathbf{p}) = ||\mathbf{p} - \mathbf{p}^0||_2^2$. This regularization imposes that the solution is close to the prior knowledge on parameter $\mathbf{p}^0$. Another choice of regularization term when dealing with parameter field estimation is $\mathcal{R}(\mathbf{p}) = ||\nabla\mathbf{p}||_2^2$. Here, the spatial smoothness of the parameter field is enforced. In order to find the best weight α between the prior knowledge and the least-squares functional, the L-curve method can be used (see [59] for details). The graphical interpretation is that in the plan $(J_{\text{WLS}}, \mathcal{R}(\mathbf{p}))$, the plot of the functional (1.4) with respect to α has a L-shape and the optimal choice of α is given by the corner of this shape.

1.1.1.2 Constitutive relation error

Adding prior knowledge linked to the specificity of the problem helps to regularize the least-squares formulation. In that sense, some functionals are directly based on the physics of the problem. It is the case of the approach based on the Constitutive Relation Error (CRE). This notion as been first introduced to certify the quality of finite element computations [78]. This method is based on the idea that among the equations of the problem, the constitutive relation is usually the one which is less reliable. The constitutive equation

is the relation that links the primal quantity $\mathbf{u}$ (displacement field in mechanics) and the dual quantity σ. In the linear case, the constitutive equation reads:

$$\sigma = \mathbf{C} : \epsilon(\mathbf{u}) \tag{1.5}$$

Then, the constitutive relation error e is defined as:

$$e = ||\sigma - \mathbf{C} : \epsilon(\mathbf{u})||_{\mathbf{C}^{-1}} \tag{1.6}$$

with $||X||_{\mathbf{C}^{-1}}$ the energy norm defined such that $||X||^2_{\mathbf{C}^{-1}} = \int_\Omega X : \mathbf{C}^{-1} : X \mathrm{d}\mathbf{x}$.

This error formulation was widely used for the verification of numerical simulations [81, 77, 76]. A first approach to deal with inverse problems is to directly use the measurements in formulation of the field admissibility [78]. Another approach is to add a measurement error term to the CRE [47] that leads to the definition of the modified constitutive relation error (mCRE):

$$\mathcal{E}^2(\mathbf{u}, \sigma, \mathbf{p}) = ||\sigma - \mathbf{C} : \epsilon(\mathbf{u})||^2_{\mathbf{C}^{-1},\Omega} + \gamma ||\Pi\mathbf{u} - \mathbf{d}^{\text{obs}}||^2_2 \tag{1.7}$$

where Π is an operator projection that allows to compare the system state $\mathbf{u}$ to the observations $\mathbf{d}^{\text{obs}}$. The scalar γ is a parameter that weights the influence of the measurement error term. Thus, in this two-step iterative approach, the unknown fields $(\sigma, \mathbf{u})$ is found by the minimization of (1.7) with fixed $\mathbf{p}$:

$$(\sigma_{\text{ad}}, \mathbf{u}_{\text{ad}}) = \underset{(\sigma,\mathbf{u})\in\mathcal{U}_{\text{ad}}\times\mathcal{S}_{\text{ad}}}{\text{argmin}} \quad \mathcal{E}^2(\mathbf{u}, \sigma, \mathbf{p}) \tag{1.8}$$

where $\mathcal{U}_{\text{ad}}$ and $\mathcal{S}_{\text{ad}}$ are the kinematically and statically admissible spaces, respectively defined by:

$$\mathcal{U}_{\text{ad}} = \{\mathbf{u}(\mathbf{x}) / u(\mathbf{x}) = \mathbf{u}_d \ \forall \mathbf{x} \in \partial_u\Omega\} \tag{1.9}$$

$$\mathcal{S}_{\text{ad}} = \{\sigma(\mathbf{x}) / \sigma(\mathbf{x}).\mathbf{n} = \mathbf{f}_d \ \forall \mathbf{x} \in \partial_f\Omega, \text{div}(\sigma(\mathbf{x})) = \mathbf{0} \ \forall \mathbf{x} \in \Omega\} \tag{1.10}$$

with $\partial\Omega = \partial_u\Omega \cup \partial_f\Omega$ and $\partial_u\Omega \cap \partial_f\Omega = \emptyset$.

Then, the modified constitutive relation error functional can be written as:

$$\mathcal{J}_{mCRE} = \mathcal{E}^2(\sigma_{\text{ad}}, \mathbf{u}_{\text{ad}}, \mathbf{p}) \tag{1.11}$$

and the identification problem comes down to the minimization of $\mathcal{J}_{mCRE}$ with respect to $\mathbf{p}$. Therefore, the inverse problem is solved using a double minimization: the first aims to find the optimal admissible fields of the problem, depending on data, and the second aims to find the best parameter fit in the sense of the modified constitutive equation.

This method was applied in a very wide range of applications involving inverse problems: in [9] for the identification of material properties, for nonlinear mechanics in [108, 41, 100], for dynamical problems in [16, 47, 80], with *in-situ* measurements in [24],etc. In those applications, the method has proven to be particularly robust with respect to highly corrupted data. However, this method can be very costly as it requires a double minimization to find the admissible fields and perform the parameters estimation.

1.1.1.3 Methods using full-field measurements

With the development of full-field measurements with the Digital Image Correlation (2D) and Digital Volume Correlation (3D), specific methods have been developed to use them in order to identify material properties. Those measurements give a very large amount of data that improve the regularity of the inverse problems. Among those methods, we can cite:

- The Finite Element Model Updating (FEMU) [70] and Integrated FEMU [85, 106] which use a weighted least-squares formulation to identify material properties from DIC measurements.
- The equilibrium gap method introduced in [32] that aims to find parameters that minimize the equilibrium residual.
- The reciprocity gap method used for crack detection [4, 48] that only requires data on the domain boundary.
- The virtual field method [55] where the inverse problem is solved in a straightforward manner thanks to the principle of virtual work and appropriate choice of virtual fields [6].

1.1.2 Methods based on the Bayesian inference

Another approach to solve inference problems is to use a stochastic framework within the Bayesian inference. This framework leads to an automatic regularization of inverse problems [143, 67, 140] and is a natural way to consider uncertainties as measurement errors, modeling errors or naturally stochastic parameters. This approach has been used in many engineering fields such as monitoring and control of structures [8, 66, 150], structural reliability analysis [139, 114], or identification of material properties [38, 51].

1.1.2.1 General formulation

The Bayesian framework is based on the Bayes theorem [7] which leans on the definition of conditional probabilities. Let A and B be two events. The relation between conditional probabilities reads:

$$P(A|B)P(B) = P(A \cap B) = P(B|A)P(A) \tag{1.12}$$

with $P(A \cap B)$ the joint probability that both events A and B occur. $P(A)$ and $P(B)$ are marginal probabilities on A and B, respectively, and $P(A|B)$ and $P(B|A)$ are the conditional probabilities. Dividing (1.12) by $P(B) \neq 0$ leads to the Bayes theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1.13}$$

Thus, if marginal probabilities are known, this relation enables one to invert the conditional knowledge between A and B. It is this property that is exploited to solve inverse problems.

In the Bayesian formulation of inverse problems, the random variables of interest are the observations $\mathbf{d}^{\text{obs}}$ and the parameters $\mathbf{p}$. That way, the Bayes formulation of inverse problems in terms of densities reads:

$$\pi(\mathbf{p}|\mathbf{d}^{\text{obs}}) = \frac{\pi(\mathbf{d}^{\text{obs}}|\mathbf{p}).\pi(\mathbf{p})}{\pi(\mathbf{d}^{\text{obs}})} \tag{1.14}$$

The posterior density $\pi(\mathbf{p}|\mathbf{d}^{\text{obs}})$ is the result of the Bayesian inversion; it gives de probability density function (pdf) on the parameters knowing the measurements. The prior density $\pi(\mathbf{p})$ is the prior knowledge on the parameter coming from another estimation, expertise or maximum entropy study. The term $\pi(\mathbf{d}^{\text{obs}})$, called model evidence, is a normalization constant that ensures that $\int \pi(\mathbf{p}|\mathbf{d}^{\text{obs}})\text{d}\mathbf{p} = 1$. The density $\pi(\mathbf{d}^{\text{obs}}|\mathbf{p})$ is the likelihood function; it is the probability that the model produces the observation $\mathbf{d}^{\text{obs}}$ for a given value of $\mathbf{p}$. Its computation requires some efforts that will be explained in the following developments.

The model governing the studied physical system, denoted $\mathcal{M}$, is assumed to be explicit, so that it provides the observation prediction $\mathbf{d}$ as:

$$\mathbf{d} = \mathcal{M}(\mathbf{p}) \tag{1.15}$$

The link between the model prediction and the observation can be made by taking into account a noise model $\mathcal{B}$:

$$\mathbf{d}^{\text{obs}} = \mathcal{B}\left(\mathcal{M}(\mathbf{p}), \mathbf{e}\right) \tag{1.16}$$

where $\mathbf{e}$ is the random variable of noise. When $\mathbf{p}$ and $\mathbf{e}$ are fixed, $\mathbf{d}^{\text{obs}}$ is completely specified, that is, $\mathbf{d}^{\text{obs}} = \mathcal{B}\left(\mathcal{M}(\mathbf{p}), \mathbf{e}\right)$, so:

$$\pi(\mathbf{d}^{\text{obs}}|\mathbf{p}, \mathbf{e}) = \delta\left(\mathbf{d}^{\text{obs}} - \mathcal{B}\left(\mathcal{M}(\mathbf{p}), \mathbf{e}\right)\right) \tag{1.17}$$

where δ is the Dirac function. Therefore, the likelihood function can be computed as:

$$\pi(\mathbf{d}^{\text{obs}}|\mathbf{p}) = \int \delta(\mathbf{d}^{\text{obs}} - \mathcal{B}\left(\mathcal{M}(\mathbf{p}), \mathbf{e}\right)).\pi_{\text{err}}(\mathbf{e}|\mathbf{p})\text{d}\mathbf{e} \tag{1.18}$$

where $\pi_{\text{err}}(\mathbf{e}|\mathbf{p})$ is the pdf of measurement error conditionally to parameters $\mathbf{p}$. The model of an additive noise is usually chosen such that the likelihood function reads:

$$\pi(\mathbf{d}^{\text{obs}}|\mathbf{p}) = \pi_{\text{err}}(\mathbf{d}^{\text{obs}} - \mathcal{M}(\mathbf{p})|\mathbf{p}) \tag{1.19}$$

If, in addition, the noise is independent of the parameters $\mathbf{p}$, the even more classical expression of the likelihood function reads:

$$\pi(\mathbf{d}^{\text{obs}}|\mathbf{p}) = \pi_{\text{err}}(\mathbf{d}^{\text{obs}} - \mathcal{M}(\mathbf{p})) \tag{1.20}$$

In this case, the likelihood function is the distance between the model prediction and the observation weighted by the measurement error. Then, the posterior density reads:

$$\pi(\mathbf{p}|\mathbf{d}^{\text{obs}}) \propto \pi_{\text{err}}(\mathbf{d}^{\text{obs}} - \mathcal{M}(\mathbf{p})).\pi(\mathbf{p}) \tag{1.21}$$

It is worth noticing that if the prior is taken as a Gaussian distribution with mean $\mathbf{p}^0$ and covariance matrix $\mathbb{C}_{\mathbf{p}}$, and the measurement noise is also Gaussian with zero mean and covariance matrix $\mathbb{C}_{\mathbf{d}}$, the posterior density reads:

$$\pi(\mathbf{p}|\mathbf{d}^{\text{obs}}) \propto \exp\left(-\frac{1}{2}\left[\left(\mathcal{M}(\mathbf{p})-\mathbf{d}^{\text{obs}}\right)^T \mathbb{C}_{\mathbf{d}}^{-1}\left(\mathcal{M}(\mathbf{p})-\mathbf{d}^{\text{obs}}\right)+\left(\mathbf{p}-\mathbf{p}^0\right)^T \mathbb{C}_{\mathbf{p}}^{-1}\left(\mathbf{p}-\mathbf{p}^0\right)\right]\right) \tag{1.22}$$

Finding the maximum of this density (maximum *a posteriori* (MAP)) gives an estimation of the parameters $\mathbf{p}^{\text{MAP}}$. Finding this maximum is equivalent to minimizing the functional:

$$\mathcal{J}_{\text{MAP}}(\mathbf{p}) = \left(\mathcal{M}(\mathbf{p})-\mathbf{d}^{\text{obs}}\right)^T \mathbb{C}_{\mathbf{d}}^{-1}\left(\mathcal{M}(\mathbf{p})-\mathbf{d}^{\text{obs}}\right)+\left(\mathbf{p}-\mathbf{p}^0\right)^T \mathbb{C}_{\mathbf{p}}^{-1}\left(\mathbf{p}-\mathbf{p}^0\right) \tag{1.23}$$

Thus, in this case the Bayesian solution of the inverse problem is equivalent to minimizing a weighted least-squares functional with the prior density acting as a regularization term.

1.1.2.2 Sequential inference

When the observations $\mathbf{d}^{\text{obs}}$ are not available simultaneously during a time-dependent problem, the formulation of the sequential Bayesian inference problem can be easily derived from the classical formulation. In such a case, for each new set of assimilated measurements, the prior density of the formulation (1.14) for the current assimilation step is taken as the posterior of the previous assimilation step. Let us consider a physical system to be studied through N_t assimilation time steps t_i, $i \in \{1, ..., N_t\}$. The model $\mathcal{M}(\mathbf{p}, t)$ governing the physical system is assumed to be time-dependent in addition to its parameters dependency.

Then, at the considered assimilation time step t_i, $i \in \{1, ..., N_t\}$, the posterior density function for having the parameters knowing the sets of measurements $\{\mathbf{d}_1^{\text{obs}}, ..., \mathbf{d}_i^{\text{obs}}\}$ reads:

$$\pi(\mathbf{p}|\mathbf{d}_1^{\text{obs}}, ..., \mathbf{d}_i^{\text{obs}}) \propto \prod_{j=1}^{i} \pi_{t_j}(\mathbf{d}_j^{\text{obs}}|\mathbf{p}).\pi(\mathbf{p}) \tag{1.24}$$

For a given set of measurements $\mathbf{d}_j^{\text{obs}}$, with independent additive measurement noise, the likelihood function $\pi_{t_j}(\mathbf{d}_j^{\text{obs}}|\mathbf{p})$ reads:

$$\pi_{t_j}(\mathbf{d}_j^{\text{obs}}|\mathbf{p}) = \pi_{\text{meas}}\left(\mathbf{d}_j^{\text{obs}} - \mathcal{M}\left(\mathbf{p}, t_j\right)\right) \tag{1.25}$$

1.1.2.3 Quantity of interest over the posterior density

In such classical Bayesian formulations of inverse problem with noisy measurements, the posterior density directly depends on the output of the forward model. However, knowing the probability density function of the unknown parameters is not directly useful to characterize those parameters. In parameter inference one wishes to compute estimators such as the maximum *a posteriori*, the mean, the variance etc. These estimators require multi-dimensional integration over the posterior density as well as the computation of the normalization constant itself. The inference problem thus becomes post-processing and integration numerical issues.

Integration over the posterior density can be difficult due to the dimension (number of parameters) and the fact that the support of the integrals is unknown. A way to avoid the

increasing computation cost of the integrals with respect to the dimension of the parametric space is to use Monte-Carlo integration. Let us consider the multi-dimensional probability density function $\pi(\mathbf{p})$ with $\mathbf{p} \in \mathbb{R}^d$, as well as the quantity of interest (1.26) involving $\pi(\mathbf{p})$ and a squared-integrable function $h(\mathbf{p})$.

$$\mathbb{E}[h] = \int_P h(\mathbf{p})\pi(\mathbf{p})\mathrm{d}\mathbf{p} \tag{1.26}$$

This scalar quantity is the mathematical expectation of h, but it could also correspond to any moment of another random variable. Considering the samples $(\mathbf{p}_i)_{i\in\{1,\dots,N_{\mathrm{mc}}\}}$ drawn according to the density π and using Monte-Carlo quadrature, the integral (1.26) can be approximated by the mean estimator:

$$\mathbb{E}[h] \approx \bar{h} = \frac{1}{N_{\mathrm{mc}}} \sum_{i=1}^{N_{\mathrm{mc}}} h(\mathbf{p}_i) \tag{1.27}$$

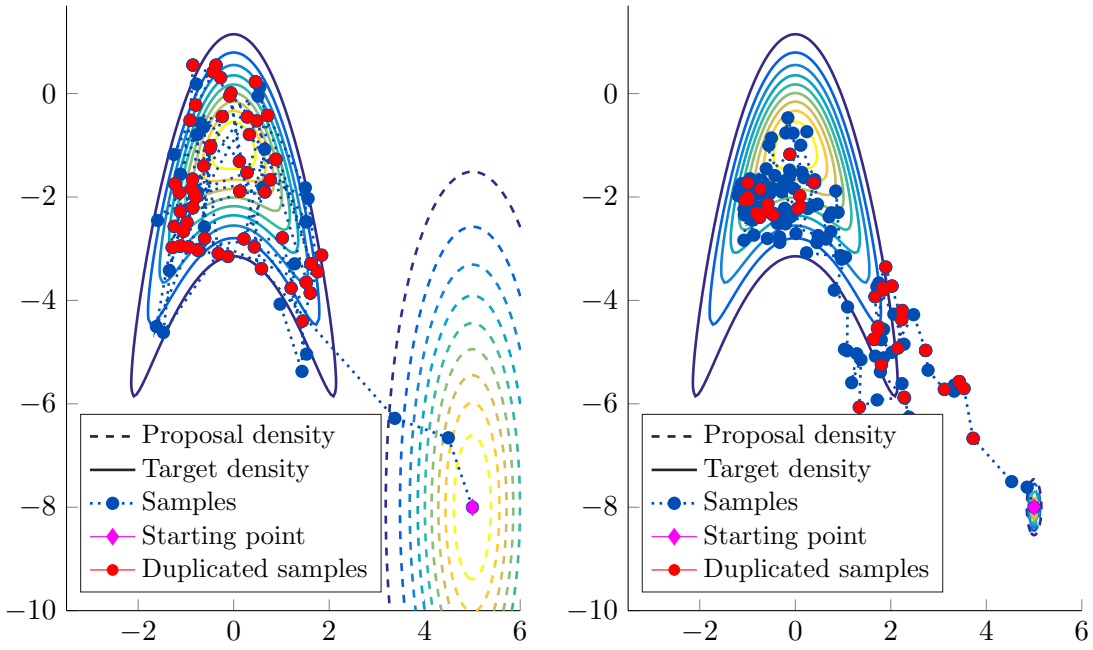
However, generating samples directly from π is very difficult in the general case where π does not belong to a classical family of distributions (normal, uniform,...). Consequently, in the context of Bayesian inference with non-linearity assumptions on the models and non-Gaussian densities, specific sampling methods are required to perform integration over the posterior density.

1.1.2.4 Markov Chain Monte-Carlo sampling

The idea of sampling methods is to use sample generators of reference densities (normal, uniform) to build samples distributed according to the target density. A classical sampling method consists in generating a Markov chain which has the target distribution as equilibrium distribution [121]. The most common algorithm to build Markov chains is the Metropolis-Hastings algorithm. This algorithm is well-suited to Bayesian inference as only the evaluation of the target density up to a multiplicative constant is necessary, and therefore the computation of the normalization constant is not necessary.

The idea of this algorithm is to generate samples from a classical density (proposal density q) then accept or reject each sample according to the comparison with the probability of the previous accepted sample. If the proposed sample has a higher probability than the previous sample of the chain, it is accepted. Otherwise, it is accepted with a probability depending on the probability ratio between its probability and the probability of the previous sample. If the proposed sample is rejected, the chain proceeds by duplicating the previous sample.

However, the proposal distribution has to be set *a priori* and can be difficult to calibrate with respect to the target density. Figure 1.1 shows the first 200 samples generated with the Metropolis-Hastings with "small" and "large" proposals. Here the target density is a two-dimensional banana-shaped density. It can be observed that a large proposal will lead to often rejected samples while a small proposal will lead to often accepted samples. Both extreme cases will lead to a poor diversity of the chain and correlated samples either by having lots of duplicated samples or samples stagnating in the same region for a long time [67]. Furthermore, the beginning of the chains corresponds to a "burn-in" phases when the samples are not yet distributed according to the target density. This phase can



(a) Markov chain with large proposal

(b) Markov chain with small proposal

Figure 1.1: First 200 samples generated by MCMC algorithm with small and large proposals

be very long depending on initialization and which proposal density is used. Samples of this phase must be discarded which slows down the convergence of the algorithm.

A way to set the proposal distribution automatically is to use the concept of adaptive MCMC [58]. The idea is to use theoretical results on optimal proposal distribution for high-dimensional normal distributions [122]. The optimal distribution for a normal distribution of dimension d with covariance matrix Σ is:

$$q(\mathbf{p}_k) = \mathcal{N}\left(\mathbf{p}_{k-1}, \frac{(2.38)^2}{d}\Sigma\right) \tag{1.28}$$

As the covariance matrix Σ of the target distribution is not known *a priori*, the idea of adaptive MCMC is to use the approximation of the chain covariance matrix Σ_k known at iteration k. The initial covariance matrix is given as a function of the target distribution dimension d. To avoid the covariance matrix collapsing to zero, a regularizing coefficient $0 < \beta < 1$ may be added [123] to compose the approximation of the optimal distribution with a fixed proposal distribution (see Algorithm 1).

A huge drawback when using MCMC methods is the difficulty to assess when the number of samples of the chain is sufficient. First, the burn-in phase has to be identified. In practice, an arbitrary large number of samples is discarded to only keep samples distributed according to the density. This phase slows down the convergence of the chain and can not be controlled on the fly during the MCMC algorithm. Besides that, setting a stopping criterion is also a difficult task [136]. In the literature, most used convergence criteria are based on convergence of averages [49]. However, such criteria can be difficult to set and often lead to a high number of samples to generate [121].

Algorithm 1 Adaptative Metropolis

Initialize: $\mathbf{p}_0$
$\Sigma_0 = \frac{(0.1)^2}{d}\mathbb{I}_d$
for $k = 1...N_{\text{mc}}$ **do**
  $\mathbf{p}_k^* \sim (1-\beta)\mathcal{N}(\mathbf{p}_{k-1}, \Sigma_{k-1}) + \beta\mathcal{N}(\mathbf{p}_{k-1}, \Sigma_0)$
  $\alpha = \min\left(1, \frac{\pi(\mathbf{p}_k^*)}{\pi(\mathbf{p}_{k-1})}\right)$
  $r \sim \mathcal{U}[0,1]$
  if $r \leq \alpha$ **then**
    $\mathbf{p}_k = \mathbf{p}_k^*$
  else
    $\mathbf{p}_k = \mathbf{p}_{k-1}$
  end if
  $\Sigma_k = \frac{(2.38)^2}{d}\text{cov}(\mathbf{p}_0, ..., \mathbf{p}_k)$
end for

1.1.2.5 Kalman and particle filters

In the case of sequential data assimilation, Kalman filters are widely used [68, 149, 34, 45]. The purpose of this method is to estimate and predict the states of a system with the sequential data assimilation of noisy data. This method allows to take into account uncertainties associated to measurements and model errors. This method can in fact be presented as a particularization of the sequential Bayesian inference when all uncertainties are considered as Gaussian, with independence assumptions and the posterior density is post-processed by computing the maximum *a posteriori* [99, 98]. This method is very fast as the problem is reduced to the solution of a linear system. Similar to this method, we can also cite the deterministic variational 4D-Var approach widely used for weather forecasting [109, 142, 83] which leans on a functional minimization. However, this method may lack of robustness as only an error on the initial state can be considered. Kalman filters may also lack of robustness due to the simplification made in form of the uncertainties and the model. That is why it was coupled to the modified constitutive relation error method in some recent works [99, 1]. To avoid the lack of robustness, particle filters/sequential Monte-Carlo methods are widely used to deal with sequential inference problems [132, 5]. The purpose of those methods is to represent the posterior density with samples distributed according to the posterior density. At each assimilation step, the samples are re-weighted (sequential importance sampling) and can be coupled with MCMC algorithms [69]. However, as presented in Section 1.1.2.4, the cost associated to the generation of the samples and the slow convergence can lead to high computational costs.

1.1.3 Limitations of current approaches for real-time simulation

In the DDDAS framework, the use of a Bayesian method was already applied in [118, 119]. This work addressed sequential data assimilation aspects alone (no feedback control loop) and focused on uncertainty quantification and the dynamical selection of models. The authors used a full stochastic Bayesian framework for inverse analysis and dynamical model

updating in order to take experimental uncertainties and modeling errors [111] into account in a straightforward manner. Nevertheless, it appears that this probabilistic framework (involving the costly computation of a likelihood function as well as multiple samplings of the forward model using MCMC techniques) remains much too heavy for considering real-time applications on large scale structures. Concurrent methods for the stochastic calibration of models are provided by means of nonlinear extensions of Kalman filtering and for example, investigated in [34, 87, 102, 101, 151]. Even though Kalman filtering has been used for a long time for structural identification [63], it is not really suited when considering complex structures with a large number of degrees of freedom and identification of distributed model parameters. Indeed, in addition to expensive algorithms due to the large number of sought parameters, modeling errors are generally *a priori* poorly known in this case. Consequently, the identification problem becomes an ill-conditioned problem and may lead to divergence or unstable solutions. The statement is similar when performing model updating from deterministic least-square methodologies and Tikhonov regularization, due to non-convex cost functions and the lack of continuity producing instabilities. In the application where full-field measurements are available, alternative variational data assimilation methods (Equilibrium Gap, Modified Constitutive Relation Error) are more robust. However, these approaches are not considered for real-time applications due to their computational costs.

Consequently, the robustness and real-time constraint that appear in data assimilation motivate the joint use of Bayesian inference and reduced order models that permit a low cost during data assimilation processes.

1.2 Model reduction methods serving inverse problems

In the majority of inverse methods, the prediction of the forward model is computed in order to compare model outputs to observations. This prediction is obtained by interrogating the model for a given value of the parameter set. In order to find the values of parameters that give the minimum of discrepancy between the model prediction and the observations, iterative methods are used. Consequently, the output ot the forward model needs to be computed for a large amount of parameters values. This is emphasized in the context of Bayesian inference where Monte-Carlo methods are used, which requires a large amount of samples.

In the targeted engineering applications, numerical models are based on the solution of complex Partial Differential Equations (PDE). In this context, parameterized PDEs have to be solved in a multi-query and computationally intensive manner along the inversion process. Hence, inverse problems have become more and more coupled with model order reduction techniques. These techniques do not simplify models but rather decrease their computational complexity by using specific mathematical techniques that generate an adequate numerical approximate solution from a low-dimensional basis (manifold). Consequently, model order reduction techniques have the potential to dramatically reduce CPU costs and memory resources without sacrificing too much of the solution accuracy.

The purpose of model reduction methods is to approximate a field $\mathbf{u}$ depending on d variables $(p_1, ..., p_d)$ by a low rank approximation. Using the canonical decomposition, this

approximation reads:

$$\mathbf{u}(p_1, ..., p_d) \approx \mathbf{u}_m(p_1, ..., p_d) = \sum_{k=1}^{m} \prod_{i=1}^{d} \alpha_{ik}(p_i) \tag{1.29}$$

Thus, a relevant reduced order model is a model $\mathbf{u}_m$ that gives a good approximation of u with a few number of modes m. Model reduction methods are very useful to speed-up the computation of the solution of inverse problems as the parameters can be chosen as spatial coordinates, time and parameters to be identified. In this context, the solution $\mathbf{u}_m$ can be computed in the offline phase, before performing data assimilation and evaluated during the inverse problem solution when data are assimilated.

1.2.1 The Proper Orthogonal Decomposition

Maybe the most common model order reduction method is the Proper Orthogonal Decomposition [91] which shares similar ideas with the Karhunen-Loeve decomposition [88] and the Principal Component Analysis (PCA) [116]. The common idea is to use an approximation of a matrix $\mathbf{U}$ by its truncated Singular Value Decomposition (SVD). Applied to time-dependent problems, the matrix $\mathbf{U}$ is built with snapshots of the model solution:

$$\mathbf{U} = [\mathbf{u}(\mathbf{x}, t_1), ..., \mathbf{u}(\mathbf{x}, t_{N_t})] \tag{1.30}$$

where $\mathbf{u}(\mathbf{x}, t_i)$, $i \in \{1, ..., N_t\}$ are vectors of size the number of discretized spatial coordinates N_s. In practice, when the considered problem is governed by a time-dependent PDE, the vector $\mathbf{u}(\mathbf{x}, t_i)$ represents the numerical solution of the PDE at time step t_i. This leads to a matrix $\mathbf{U}$ of size $N_s \times N_t$ and its SVD reads:

$$\mathbf{U} = \mathbf{Z}\mathbf{S}\mathbf{V}^T \tag{1.31}$$

The matrix $\mathbf{S}$ is a diagonal matrix composed by the singular values arranged in descending order. The orthogonal matrices $\mathbf{Z} \in \mathbb{R}^{N_s \times N_s}$ and $\mathbf{V} \in \mathbb{R}^{N_t \times N_t}$ are composed with spatial and time modes respectively. Thus, the reduced order model reads:

$$\mathbf{U} \approx \sum_{k=1}^{m} \sigma_i \mathbf{z}_i \otimes \mathbf{v}_i \tag{1.32}$$

where $\mathbf{z}_i$ and $\mathbf{v}_i$ are the basis column vectors of $\mathbf{Z}$ and $\mathbf{V}$, respectively. Here, the truncation order $m \leq \min(N_s, N_t)$ can be chosen by studying how the singular values decrease and quantifying the truncation error. Consequently, the discretized field $\mathbf{U}$ can be represented only with m vectors of size N_s and m vector of size N_t. That way, the solution of a time dependent PDE can be computed in an offline phase and the POD basis of the spatial modes is kept to evaluate the solution in the online phase.

This method can be applied for any matrix associated to the snapshots of any parameters. However, in inverse problems the models of interest have more than two parameters. Thus classical POD can be limited to solve inverse problems. To take into account a parametric dependency of the model of interest to d parameters, the reduced model (1.29) can be computed with snapshots and enrichment methods such as the High-Order Singular

Value Decomposition (HOSVD) [11]. This approach can be associated to the decomposition of high-order tensors like the Canonical Polyadic (CP) decomposition [62] and its variants [33].

In the context of inverse problems, POD was used in [15] for the identification of material properties, in the context of sequential data assimilation with time-dependent problem [37, 46], or in the context of Bayesian inference in [86].

Nevertheless, all these methods based on snapshots are very costly as they require the evaluation of the model for a large number of parameter sets. Even if in the context of inverse problems this cost is offline, such computations can be difficult to perform.

1.2.2 The Reduced Basis method

An improvement of the generation of reduced order models with snapshots is given by the Reduced Basis (RB) method [126, 95]. The principle of this method is a greedy algorithm which selects the best snapshots to enrich the reduced order model. That way, only the most relevant computations are performed. To build the reduced order model, a first parameter set $\mathbf{p}^1 = (p_1^1, ..., p_2^1) \in S_{\mathbf{p}}$ is chosen with a prior knowledge or in a random way. At this stage, the reduced basis is $\{\mathbf{u}(\mathbf{p}^1)\}$. Then, each new best snapshot $\mathbf{p}^k$, $k \in \{2, .., m\}$, is defined as the one for which the associated error is the largest:

$$\mathbf{p}^k = \underset{\mathbf{p} \in S_{\mathbf{p}}}{\operatorname{argmax}} \, ||\mathbf{u}(\mathbf{p}) - \Pi_{k-1}\mathbf{u}(\mathbf{p})||_2 = \Delta_{k-1}(\mathbf{p}) \tag{1.33}$$

where Π_{k-1} is the orthogonal projection on the space $\mathcal{U}_{k-1} = \text{span}\left\{\mathbf{u}(\mathbf{p}^1), ..., \mathbf{u}(\mathbf{p}^{k-1})\right\}$ is stopped when $\Delta_k(\mathbf{p})$ achieves a suitably small tolerance. Once the reduced basis is computed, the evaluation of the solution for an arbitrary value of the parameters $\mathbf{p}$ requires a Galerkin projection.

The RB method was used in many applications to solve inverse problems. In [93], it was coupled with the Genearalized Empirical Interpolation Method, and in [82] it was applied in a biomedical context. The RB method was also coupled with Bayesian inference in [36] where the snapshots required to compute the basis were optimized with respect to the posterior density. In [97], Bayesian inference was coupled with reduced basis and model error estimation.

1.2.3 Polynomial chaos expansion

Coupled with Bayesian inference, the polynomial chaos expansion is also widely used [104, 105]. The purpose is to replace the forward model in the expression of the likelihood function by a truncated polynomial chaos approximation. In this case, the polynomial basis is chosen according to the prior density $\pi(\mathbf{p})$. First, a polynomial basis $\Phi_{\boldsymbol{\alpha}}(\mathbf{p})$, $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_d)$ $(\alpha_i \in \mathbb{N})$ is chosen as the product of univariate polynomials:

$$\Phi_{\boldsymbol{\alpha}}(\mathbf{p}) = \prod_{i=1}^{d} H_{\alpha_i}(p_i) \tag{1.34}$$

where $\{H_{\alpha_i}, i \in \{1, ..., d\}\}$ are a polynomial orthogonal basis (Hermite, Legendre, ...). chosen according to the prior density $\pi(\mathbf{p}$. Then, the mth-order polynomial expansion of the

model $\mathcal{M}$ is defined by:

$$\mathcal{M}_m(\mathbf{p}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}^m} a_{\boldsymbol{\alpha}}\Phi_{\boldsymbol{\alpha}}(\mathbf{p}) \tag{1.35}$$

The coefficients $a_{\boldsymbol{\alpha}}$ are found solving:

$$a_{\boldsymbol{\alpha}} = \mathbb{E}[\mathcal{M}(\mathbf{p})\Phi_{\boldsymbol{\alpha}}(\mathbf{p})] = \int \mathcal{M}(\mathbf{p})\Phi_{\boldsymbol{\alpha}}(\mathbf{p})\pi(\mathbf{p})\mathrm{d}\mathbf{p} \tag{1.36}$$

In the case where the model $\mathcal{M}$ found by solving a partial differential equation, stochastic finite element method can also be used to found the coefficients $a_{\boldsymbol{\alpha}}$.

This method has two main limitations [90]. First, the polynomial chaos expansion is based on the prior density of the Bayesian formulation. Then, when the posterior is strongly shifted from the prior (case when the data are highly informative), the approximation can become inaccurate. Second, when dealing with highly non-linear models, the number m of modes needed to accurately represent the solution is very large.

1.2.4 The Proper Generalized Decomposition method

The Proper Generalized Decomposition is a model order reduction technique used in a wide range of applications [28]. It constructs an *a priori* modal approximate solution of a PDE using the separation of variables on a low number of modes. The classical approach called progressive Galerkin PGD [110] is based on the Galerkin orthogonality built from the global weak form of the PDE. This PDE is here assumed to be defined on a spatial domain Ω and time domain $I = [0, T_f]$. Considering for instance a Laplacian operator and transient regime, we introduce the Sobolev space $\mathcal{S} = H_0^1(\Omega)$ (space of H^1 functions on Ω verifying homogeneous Dirichlet boundary conditions), as well as the Lebesgue space $\mathcal{T} = L^2(I)$ and the Bochner space $\mathcal{V} = L^2(I;\mathcal{S}) \simeq \mathcal{S}\otimes\mathcal{T}$. We thus consider the space-time weak formulation of the PDE defined as follows:

$$\text{Find } \mathbf{u}\in\mathcal{V}, \text{ such that } a(\mathbf{u},\mathbf{v}) = l(\mathbf{v}),\ \forall\mathbf{v}\in\mathcal{V} \tag{1.37}$$

In the context of parameter inference, the PDE is assumed to be dependent on model parameters $\mathbf{p}\in P = P_1\times ...\times P_d$. The idea of the PGD method is to find the solution of the problem globally in terms of space, time, and parameters under the canonical tensor format:

$$\mathbf{u}_m(\mathbf{x},t,\mathbf{p}) = \sum_{k=1}^{m}\Lambda_k(\mathbf{x})\lambda_k(t)\prod_{i=1}^{d}\alpha_{ik}(p_i) \tag{1.38}$$

After introducing functional spaces $\mathcal{P}_i = L^2(P_i)$, $i\in\{1,...,d\}$, the global bilinear form $A(.,.)$ and linear form $L(.)$ are defined as:

$$A(\mathbf{u},\mathbf{v}) = \int_P a(\mathbf{u},\mathbf{v})\mathrm{d}\mathbf{p},\ L(\mathbf{v}) = \int_P l(\mathbf{v})\mathrm{d}\mathbf{p} \tag{1.39}$$

In the progressive approach, modes are computed sequentially; the decomposition at order $m-1$ being known, the mode m (i.e. functions $\Lambda_m, \lambda_m, \alpha_{im}$, $i\in\{1,...,d\}$) is searched such that:

$$\mathbf{u}_m(\mathbf{x},t,\mathbf{p}) = u_{m-1}(\mathbf{x},t,\mathbf{p}) + \Lambda_m(\mathbf{x})\lambda_m(t)\prod_{i=1}^{d}\alpha_{im}(p_i) \tag{1.40}$$

Using the Galerkin orthogonality on the tangent space, the problem reads:

Find $\Lambda_m \in \mathcal{S}$, $\lambda_m \in \mathcal{T}$, $\alpha_{im} \in \mathcal{P}_i$, $i \in \{1, ..., d\}$ such that:

$$A\left(\mathbf{u}_{m-1} + \Lambda_m \lambda_m \prod_{i=1}^{d} \alpha_{im}, \mathbf{v}^*\right) = L(\mathbf{v}^*) \tag{1.41}$$

$$\text{with } \mathbf{v}^* = \Lambda^* \lambda \prod_{i=1}^{d} \alpha_{im} + \Lambda \lambda^* \prod_{i=1}^{d} \alpha_{im} + \Lambda \lambda \sum_{j=1}^{d} \alpha_{jm}^* \prod_{\substack{i=1 \\ i \neq j}}^{d} \alpha_{im}$$

$$\forall \Lambda_m^* \in \mathcal{S}, \lambda_m^* \in \mathcal{T}, \alpha_{jm}^* \in \mathcal{P}_j,\ j \in \{1, ..., d\}$$

This leads to the solution of $d + 2$ coupled equations:

$$A\left(\mathbf{u}_{m-1} + \Lambda_m \lambda_m \prod_{i=1}^{d} \alpha_{im}, \Lambda^* \lambda \prod_{i=1}^{d} \alpha_{im}\right) = L\left(\Lambda^* \lambda \prod_{i=1}^{d} \alpha_{im}\right), \forall \Lambda_m^* \in \mathcal{S} \tag{1.42a}$$

$$A\left(\mathbf{u}_{m-1} + \Lambda_m \lambda_m \prod_{i=1}^{d} \alpha_{im}, \Lambda \lambda^* \prod_{i=1}^{d} \alpha_{im}\right) = L\left(\Lambda \lambda^* \prod_{i=1}^{d} \alpha_{im}\right), \forall \lambda_m^* \in \mathcal{T} \tag{1.42b}$$

$$A\left(\mathbf{u}_{m-1} + \Lambda_m \lambda_m \prod_{i=1}^{d} \alpha_{im}, \Lambda \lambda \alpha_{jm}^* \prod_{\substack{i=1 \\ i \neq j}}^{d} \alpha_{im}\right) = L\left(\Lambda \lambda \alpha_{jm}^* \prod_{\substack{i=1 \\ i \neq j}}^{d} \alpha_{im}\right), \forall \alpha_{im}^* \in \mathcal{P}_i,\ i \in \{1, ..., d\} \tag{1.42c}$$

Once spaces $\mathcal{P}_i$, $i \in \{1, ..., d\}$, $\mathcal{T}$ and $\mathcal{S}$ are discretized, sub-problems (1.42a), (1.42c) are linear problems solved using the finite element method and sub-problem (1.42b) is an ordinary differential equation solved using a time scheme. As the equations of the global system (1.42) are coupled, they are in practice solved by means of a fixed-point algorithm (alternated directions strategy). At the difference of aforementioned model reduction methods, the PGD model is computed while solving the PDE. Consequently, the model output can directly be evaluated for any parameter value by evaluating finite element functions.

The PGD technique was widely applied to the solution of inverse problems as in [135] in a geophysical context, or in the DDDAS framework [54, 50]. It was coupled to Kalman filtering in [53] and in [99] where the modified constitutive relation error was also used. In [17], PGD was coupled to modified constitutive relation in the context of real-time control of a machining process. In [10], a first use of the PGD with Bayesian inference and MCMC sampling was proposed to identify thermal conductivity in a time-dependent problem.

1.3 Conclusions

In the literature, many developments were dedicated to the solution of inverse problems. While the deterministic methods consist in minimizing a functional, stochastic approaches based on the Bayesian inference aim to find the posterior densities on the parameters of interest. In this context, all the uncertainties are considered as random variables and can be propagated to estimate the parameters of interest. The Bayesian inference method has proven to be robust and can give better results than the classical least-squares formulation

[51]. In the context of health monitoring of civil engineering structures, regularized least-squares, modified constitutive error and Bayesian approaches were compared [148]. In this application these methods seem to provide equivalent accurate estimations of parameters. However, the computation of those approaches may not be compatible with real-time data assimilation. It is especially the case in the Bayesian framework where the posterior density needs to be characterized with many Monte-Carlo samples.

To circumvent this issue, reduced order models are commonly used. Among the *a posteriori* techniques, the Proper Orthogonal Decomposition and Reduced Basis are the most used techniques. In those approaches, particular solutions of the problem (snapshots) are used to build the reduced order model. Another approach often described as *a priori* is the Proper Generalized Decomposition. In this method, the multi-parametric solution of the studied problem is obtained directly by solving the governing PDE and assuming the separated structure of the solution. This approach is well suited for real-time applications as it gives a quasi-analytical formulation of the model solution depending on all the variables (space, time, parameters).

As a result, the approach to solve inverse considered here is placed in the Bayesian inference framework. This choice is motivated by the robustness of the method and the stochastic framework that allows to propagate all uncertainties in the estimation of the parameters. In order to reduce the computation time required to compute Bayesian solutions, the starting idea will be to use a PGD model computed in an offline phase (time before the physical system is observed) and evaluated with a low cost in the online phase (during the assimilation of data). This is the topic of Chapter 2.

Chapter 2

Fast Bayesian inference with PGD model order reduction

In this chapter, the purpose is to couple a Bayesian formulation of inverse problems to the PGD model reduction method. Different types of uncertainties will be considered as measurement error, model error and stochastic parameters. This chapter is adapted from the published paper [129].

2.1 Bayesian updating

2.1.1 Building the likelihood function

In this chapter, 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 $\pi(\mathbf{d}^{\text{obs}}|\mathbf{p})$. This function represents the probability to have the model output equal to the measurements knowing parameters $\mathbf{p}$. Measurements being known, the likelihood function is seen as a function of $\mathbf{p}$. This probability is directly linked with with intrinsic uncertainties (stochastic parameters $\mathbf{i}$) considered in the model environment (see Figure 4), model and measurements errors. Here, few parameters are considered so that the likelihood function will be built by sampling directly the discretized parametric space. Other sampling alternatives will be discussed in Section 2.4.

2.1.2 Management of 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:

$$\mathbf{d}^{\text{obs}} = \mathbf{d} + \mathbf{e} \text{ , with } \mathbf{d} = \mathcal{M}(\mathbf{p}, \mathbf{i}) \tag{2.1}$$

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

$$\pi(\mathbf{d}^{\text{obs}}|\mathbf{p}) = \int \pi(\mathbf{d}|\mathbf{p}).\pi_{\text{err}}(\mathbf{d}^{\text{obs}} - \mathbf{d})\text{d}\mathbf{d} \tag{2.2}$$

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

uncertainty in the model (no stochastic parameter $\mathbf{i}$ and no model error) the probability density function reads:

$$\pi(\mathbf{d}|\mathbf{p}) = \delta(\mathbf{d} - \mathcal{M}(\mathbf{p})) \tag{2.3}$$

with δ the Dirac function. Then the computation of the integral (2.2) is explicit and the likelihood function is given by:

$$\pi(\mathbf{d}^{\text{obs}}|\mathbf{p}) = \pi_{\text{err}}(\mathbf{d}^{\text{obs}} - \mathcal{M}(\mathbf{p})) \tag{2.4}$$

2.1.3 Management of uncertain parameters/modeling 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 $\mathbf{d}_{\text{mc}}$ are evaluated in the model with values of the uncertainties $\mathbf{i}_{\text{mc}}$ drawn according to the probability density function assumed in the modeling:

$$\mathbf{d}_{\text{mc}} = \mathcal{M}(\mathbf{p}, \mathbf{i}_{\text{mc}}) \tag{2.5}$$

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

$$\pi(\mathbf{d}|\mathbf{p}) \approx \frac{1}{N_{\text{mc}}h} \sum_{\text{mc}=1}^{N_{\text{mc}}} K\left(\frac{\mathbf{d} - \mathbf{d}_{\text{mc}}}{h}\right) \tag{2.6}$$

K is a kernel function and h is a smoothing parameter called the bandwidth. For example, K can be chosen 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 $\pi(\mathbf{d}|\mathbf{p}) = \pi_{\text{mod}}(\mathbf{d})$, with $\pi_{\text{mod}}(\mathbf{d})$ 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.

2.1.4 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 substantial information on the parameter identified as a probability density function. The post-processing of this density can be performed by means of 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: $\sigma = \sqrt{\int p^2.\pi(p|\mathbf{d}^{obs})\mathrm{d}p - p_M^2}$

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

$$\pi_k(p_k|\mathbf{d}^{\text{obs}}) = \int_{\substack{\otimes_{l=1}^{m} I_l \\ l\neq k}} \pi_{\text{joint}}\left((p_l)_{\substack{l\in\{1,..,m\} \\ l\neq k}}|\mathbf{d}^{\text{obs}}\right) \mathrm{d}\left((p_l)_{\substack{l\in\{1,..,m\} \\ l\neq k}}\right) \tag{2.7}$$

When a model updating process is considered from successive measurements (e.g. at some time steps), the Bayesian inference naturally allows one to take successive pieces of information into account and keep an history of the previous identification steps by taking the prior probability density function at the current step as the posterior probability density function of the previous step. Finally, the global algorithm of the general Bayesian framework for a given model $\mathcal{M}$ is summed up in the Algorithm 2.

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 Algorithm 2 the model $\mathcal{M}$ needs to be evaluated $N \times \dim(I^h) \times N_{\text{mc}}$ times. Furthermore, once the posterior probability density function is built, a post-processing needs to be done and this adds 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.

2.2 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 [74] as "radial approximation" to solve nonlinear problems in structural mechanics. Since [29], this method was used in many fields: model verification and validation [22], virtual charts for the engineering [146] [35] etc. PGD was also used for identification problems in a deterministic framework [99] [17] [89] and the great possible number of parameters types that can be considered [28] seems to be well suited for Bayesian inference. A first PGD-Bayesian inference approach is given in [10] where a PGD model is used in a Monte Carlo Markov Chain (MCMC) framework. In this chapter, we consider stochastic parameters which lead to additional sources of uncertainties. Those uncertainties are considered with the PGD model.

2.2.1 Progressive Galerkin PGD

We consider the general case 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:

$$\text{Find } u \in \mathcal{U} / \forall v \in \mathcal{V} : A(u, v) = L(v) \tag{2.8}$$

Algorithm 2 Bayesian identification - Monte-Carlo

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

for j=1 to N **do**
  for $\mathbf{p} \in I^h$ **do** ▷ Discretized parameter space
    Monte-Carlo sampling :
    for mc = 1 to N_{mc} **do**
      $\mathbf{i}_{\text{mc}} \sim \pi(\mathbf{i})$
      $\mathbf{d}_{\text{mc}} = \mathcal{M}(\mathbf{i}_{\text{mc}}, \mathbf{p})$
    end for
    end

    Uncertainty propagation with kernel density estimation :

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

    Computation of the current likelihood function point $\mathbf{p}_{\text{fixed}}$:

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

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

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

end for
end

where A is a bilinear form, L a linear form, u is the unknown field, and v the test field. l variables $p_1, ..., p_l$ are considered. Those variables can be space variables, time, loads parameters, material properties, etc. The following separated solution of the problem (2.8) is searched (tensorized representation with canonical format):

$$u(p_1, ...p_l) \approx u_{PGD}(p_1, ...p_l) = \sum_{n=1}^{m} \prod_{k=1}^{l} \lambda_{kn}(p_k) \tag{2.9}$$

m is the number of modes and $u \in \mathcal{U} = \otimes_1^l U_k$. The computation of modes is performed incrementally, so that the order m PGD decomposition reads:

$$u_{PGD}(p_1, ...p_l) = \sum_{n=1}^{m-1} \prod_{k=1}^{l} \lambda_{kn}(x_k) + \prod_{k=1}^{l} \lambda_{km}(p_k) \tag{2.10}$$

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

$$v = \sum_{k=1}^{l} \lambda_{km}^* \prod_{\substack{j=1 \\ j \neq k}}^{l} \lambda_{jm} \ , \ \lambda_{km}^* \in U_k \ , \ k \in \{1, ..., l\} \tag{2.11}$$

Verifying (2.8) thus leads to n coupled equations:

$$\lambda_{km} = \mathcal{S}_k((\lambda_{jm})_{\substack{j \in \{1,...,l\} \\ j \neq k}}) \ , \ k \in \{1, ..., l\} \tag{2.12}$$

This system of equations is solved by a fixed-point algorithm. This way, the 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, $k \in \{1, ..., l\}$.

2.2.2 PGD model reduction for Monte-Carlo sampling

The PGD model order reduction method allows one to have the response of a system modelled with partial differential equations for all values of parameters. The goal is to build such a PGD model to speed up the Bayesian process described in Algorithm 2. According to the model environment set in previous sections (Figure 4), the following PGD representation is searched:

$$\mathcal{M}(\mathbf{p}, \mathbf{x}, \mathbf{i}) = \mathbf{d} \approx \sum_{n=1}^{m} \Lambda_n(\mathbf{x}) \alpha_{1n}(\mathbf{p}) \alpha_{2n}(\mathbf{i}) \tag{2.13}$$

The model is built by separation of each parameter/variable in the loop of Algorithm 2. 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 types 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 due to the evaluation of the PGD model (which is explicit) for each Monte-Carlo draw, to each iteration on the parameter to identify, and to each integral computation (for model/measurement errors, or post-processing).

2.2.3 PGD for analytical uncertainty quantification

The PGD model reduction is also a very interesting approach as it provides an analytical form of the solution for all 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.

In this case, the uncertainty propagation in the model can be found by a change of variables. 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 zero derivative, then the probability density function $\pi_Y(y)$ of the variable change is given by:

$$\pi_Y(y) = \left| \frac{1}{f'(f^{-1}(y))} \right| .\pi_X(f^{-1}(y)) \tag{2.14}$$

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

$$\pi_Y(y) = \sum_{k=1}^{N_y} \left| \frac{1}{f'(f_k^{-1}(y))} \right| .\pi_X(f_k^{-1}(y)) \tag{2.15}$$

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 be zero-measured.

This theorem can be applied with the PGD function:

$$f : i \to \sum_{n=1}^{m} \Lambda_n(\mathbf{x})\alpha_{1n}(\mathbf{p})\alpha_{2n}(i) \tag{2.16}$$

The probability density function estimated by Monte-Carlo sampling and kernel density estimation is thus replaced with the change of variables:

$$\pi(\mathbf{d}|\mathbf{p}) = \pi_{f(\mathbf{p},\mathbf{x},i)}(\mathbf{d}) \tag{2.17}$$

However, this change of variables 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 3.

Avoiding the Monte-Carlo sampling seems to be very promising as 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.

2.3 Numerical examples

2.3.1 Welding control quality example

In the industrial context, the welding process control is a practical issue and a process controlled by a numerical model through DDDAS framework can be an interesting approach.

Algorithm 3 Bayesian identification - Analytical PGD

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

1: **for** j=1 to N **do**

2: **for** $\mathbf{p}_{\text{fixed}} \in I^h$ **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 computation of corresponding function inverses f_k^{-1}.

5: Uncertainty propagation by analytical change of variables:

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

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

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

7: **end for**

8: **end**

9: $\pi(\mathbf{p}|\mathbf{d}^{\text{obs}}) = \pi(\mathbf{d}_j^{\text{obs}}|\mathbf{p}).\pi_j^{\text{prior}}(\mathbf{p})$

10: $\pi_{\text{j+1}}^{\text{prior}}(\mathbf{p}) = \pi(\mathbf{p}|\mathbf{d}^{\text{obs}})$

11: **end for**

12: **end**

13: Post-processing for the multi-parametric case:

14: **for** k=1 to m **do**

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

15: **end for**

16: **end**

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 initially proposed in [57]. Even if the industrial problem has been simplified, it is still sufficiently complex to evaluate the performance of the presented methodology.

2.3.1.1 Definition of the problem

The welding control quality problem is described in Figure 2.1. Two metal plates are welded by a heat source whose 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.

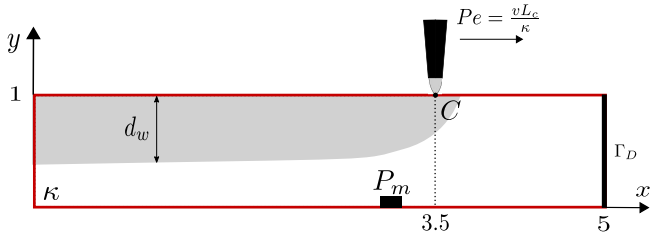


Figure 2.1: Joint-section view of the welding process

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 numerically and dynamically modified to improve the quality of the process.

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

- A 2D unsteady convection-diffusion problem is considered with geometry described in Figure 2.1. A homogeneous isotropic material is assumed.

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

  $$T(x,y;t) \equiv \frac{\overline{T}(x,y;t) - \overline{T}_\infty}{\overline{T}_f - \overline{T}_\infty} \tag{2.18}$$

  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:

  $$\frac{\partial T}{\partial t} + \underline{v}.\underline{\text{grad}}T - \kappa\Delta T = s \tag{2.19}$$

with s the volume heat input, $\underline{v} = [Pe; 0]$, and $Pe = \frac{v.L_c}{\kappa}$ 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 point $C = (x_c; y_c) = (3.5; 1)$:

$$s(x, y; \sigma) = \frac{u}{2\pi\sigma^2} \exp\left(-\frac{(x - x_c)^2 + (y - y_c)^2}{2\sigma^2}\right) \tag{2.20}$$

  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 2.1). 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 room temperature and the remaining boundaries are assumed insulated. Eventually, the initial condition is $T(x, y; t = 0) = 0$.

2.3.1.2 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 (2.19). The kinematically admissible temperature field space is defined as follows:

$$\mathcal{T} = \{T \in H^1(\Omega =]0; 5[\times]0; 1[), T = 0 \text{ on } \Gamma_D\} \tag{2.21}$$

The equation (2.19) is equivalent to finding $T \in \mathcal{T}$ such that $\forall T^* \in \mathcal{T}$:

$$a(T, T^*) = l(T^*) \tag{2.22}$$

with:

$$a(T, T^*) = \int_\Omega \left(\frac{\partial T}{\partial t} + \underline{v}.\underline{\text{grad}}T\right).T^* + \kappa.\underline{\text{grad}}T.\underline{\text{grad}}T^* d\Omega \tag{2.23}$$

$$l(T^*) = \int_\Omega s.T^* d\Omega \tag{2.24}$$

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

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

$$[M]\{\dot{T}\} + [C_H]\{T\} = \{S\} \tag{2.25}$$

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 element matrices read:

$$[M] = \int_{\Omega_h} [N]^T [N] d\Omega \tag{2.26}$$

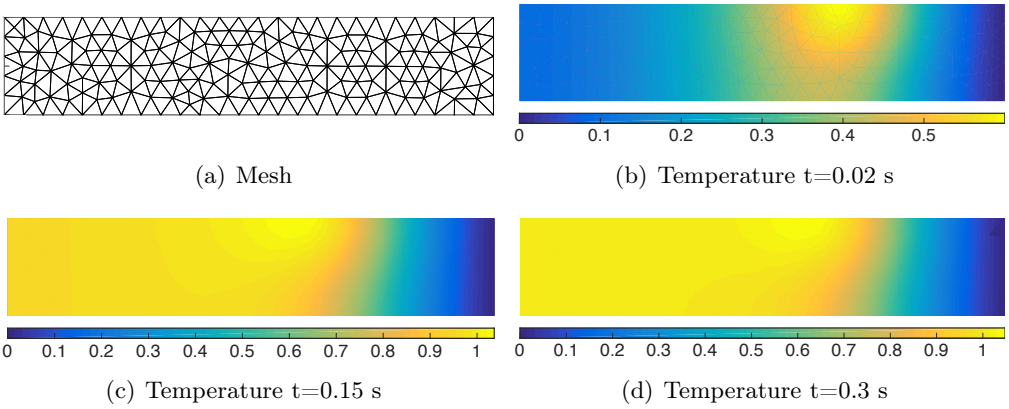
$$[H] = \int_{\Omega_h} \{V\}^T [dN]^T [N] d\Omega \tag{2.27}$$

$$[C] = \kappa. \int_{\Omega_h} [N]^T [N] d\Omega \tag{2.28}$$

$$[C_H] = [C] + [H] \tag{2.29}$$

The equation (2.25) is solved owing to a time discretization coupled with an implicit first-order time scheme.

The previous solution scheme is implemented in a MATLAB code. The temperature field obtained at different time steps is shown in Figure 2.2.



(a) Mesh (b) Temperature t=0.02 s

(c) Temperature t=0.15 s (d) Temperature t=0.3 s

Figure 2.2: Finite element solution

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

2.3.1.3 PGD solution

In order to apply the Bayesian identification described in the previous sections, this section 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:

$$T(x, y, t; \sigma) \approx T_m(x, y, t; \sigma) = \sum_{n=1}^{m} \Lambda_n(x, y) \lambda_n(t) \alpha_n(\sigma) \tag{2.30}$$

In the Bayesian-DDDAS framework, the model needs to be evaluated at each time step, and for all values of the parameter σ as explained in Algorithm 2. 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 2.2.1 is applied to the global weak form of the problem, obtained from the integration of (2.22) over the parametric domain. The PGD

modes are given by the fixed-point Algorithm 4. In Appendix A, the details of the PGD solution are explained.

Algorithm 4 Fixed point algorithm for PGD solution

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

for $p = 1$ to m **do** ▷ Number of modes

  $\lambda = \frac{1}{\sqrt{T_f}}.\{1\}_I$

  $\alpha = \frac{1}{\sqrt{\sigma_{\max} - \sigma_{\min}}}.\{1\}_\Sigma$

  for $k = 1$ to $k_{\max}$ **do**

    $\Lambda = S_m(\lambda, \alpha)$

    $\lambda = T_m(\Lambda, \alpha)$

    $\alpha = P_m(\Lambda, \lambda)$

  end for

  end

  $\Lambda_p = \Lambda$

  $\alpha_p = \frac{\alpha}{\sqrt{\int_\Sigma \alpha^2 d\sigma}}$ ▷ Normalization of parametric modes

  $\lambda_p = \frac{\lambda}{\sqrt{\int_I \lambda^2 d\lambda}}$ ▷ Normalization of time modes

end for

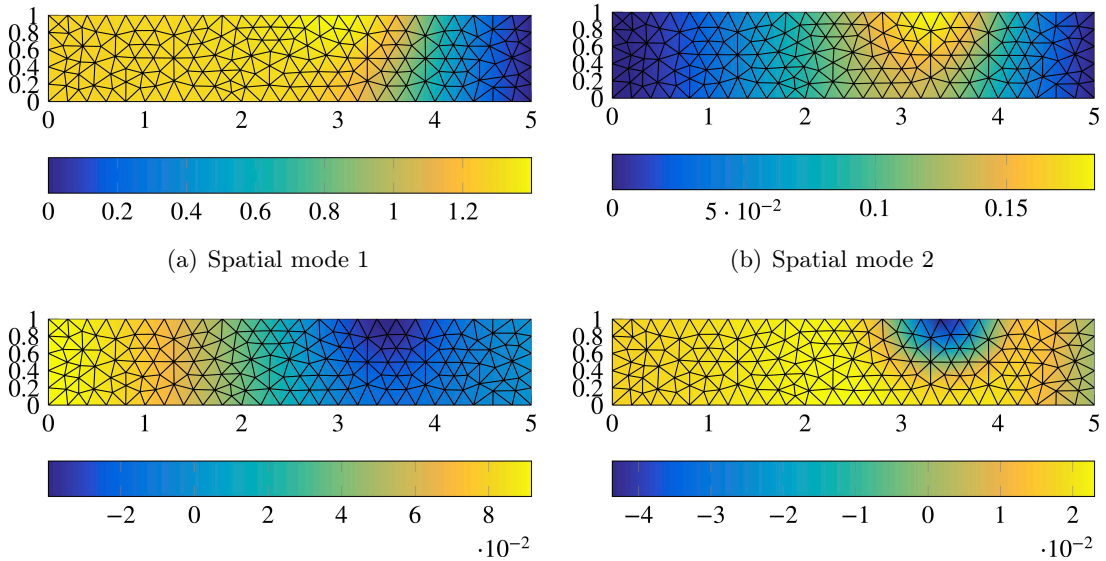
end

In Algorithm 4, 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.

The first four spatial modes are represented in Figure 2.3. In comparison with the finite element simulation in Figure 2.2, 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 near the geometry edges and around the heat input. As the time modes and parametric modes are normalized, the energy norm of spatial modes gives the contribution of each mode to the global solution. This quantity is represented in Figure 2.4. We observe that the six first modes are contributing for the most of the solution.

The normalized time modes are plotted in Figure 2.5. We observe that the modes have a strong gradient at the beginning which requires a fine time discretization.

Eventually, in Figure 2.6 normalized parametric modes are represented. This figure shows that the 4th mode has a higher dependency on σ than the others. This result seems conform to the corresponding spatial mode 4 (Figure 3.4(d)) which exhibits a strong gradient near the torch center.



(a) Spatial mode 1

(b) Spatial mode 2

(c) Spatial mode 3

(d) Spatial mode 4

Figure 2.3: The four first spatial modes

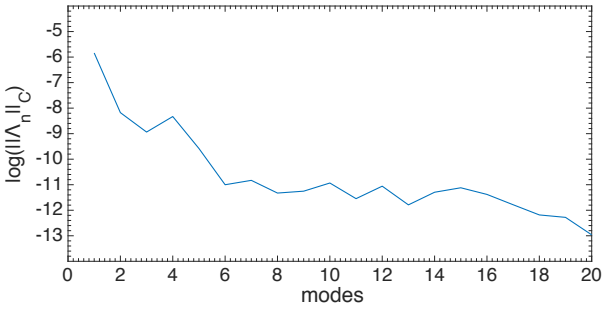


Figure 2.4: Energy norm of the spatial modes

2.3.1.4 Error quantification

The quality of the PGD model is studied and error quantification is performed in this section in order to use it in the Bayesian identification. 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 $T_{\text{FE}}(x, y, t, \sigma)$. The T_1 functions are the temperatures at the measurement point. Here, the discretization error made by the finite element method itself is not considered.

The following errors are defined :

- Local error integrated in space:

$$\epsilon(t, \sigma) = \frac{\|T_{\text{PGD}}(x, y, t, \sigma) - T_{\text{FE}}(x, y, t, \sigma)\|_{L^2(\Omega)}}{\|T_{\text{FE}}(x, y, t, \sigma)\|_{L^2(\Omega)}} \tag{2.31}$$

- Local error on the output T_1:

$$\epsilon_1(t, \sigma) = \frac{|T_{1\text{PGD}}(t, \sigma) - T_{1\text{FE}}(t, \sigma)|}{|T_{1\text{FE}}(t, \sigma)|} \tag{2.32}$$

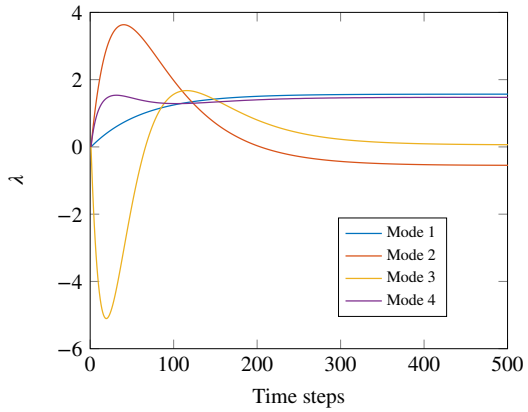


Figure 2.5: Time modes $\lambda_i(t)$, $i \in \{1, ..., 4\}$

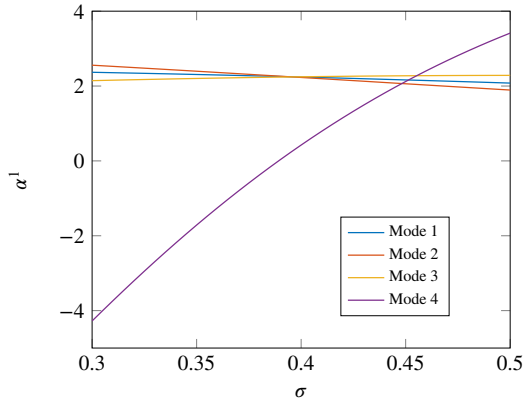


Figure 2.6: Parametric modes $\alpha_i(\sigma)$, $i \in \{1, ..., 4\}$

- Global error on the output T_1:

$$\bar{\epsilon} = \frac{\|T_{1\text{PGD}}(t,\sigma) - T_{1\text{FE}}(t,\sigma)\|_{L^2(\Sigma,I)}}{\|T_{1\text{FE}}(t,\sigma)\|_{L^2(\Sigma,I)}} \tag{2.33}$$

Figure 2.7 shows the error $\bar{\epsilon}$ depending on the number of modes in the PGD solution. 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 2.8 shows the local errors $\epsilon(t,\sigma)$ and $\epsilon_1(t,\sigma)$ on a discretized map of parametric space and time. We observe that the error is high for the first time steps. Then, it strongly decreases during the transient phase up to a very low value during the steady-state phase.

Those error quantifications can be partially justified with [110, 2, 28] in which it is explained that the progressive Galerkin PGD method can encounter 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 also be encountered for some values of the Peclet number. The studied case is far from those instabilities but some

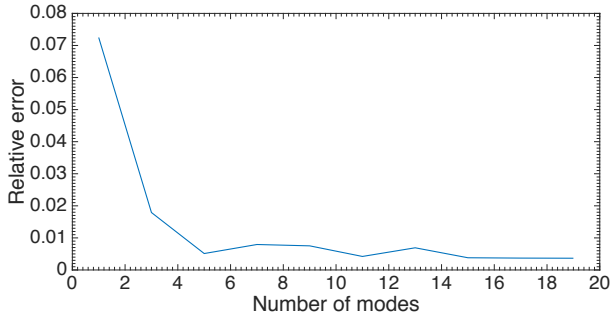
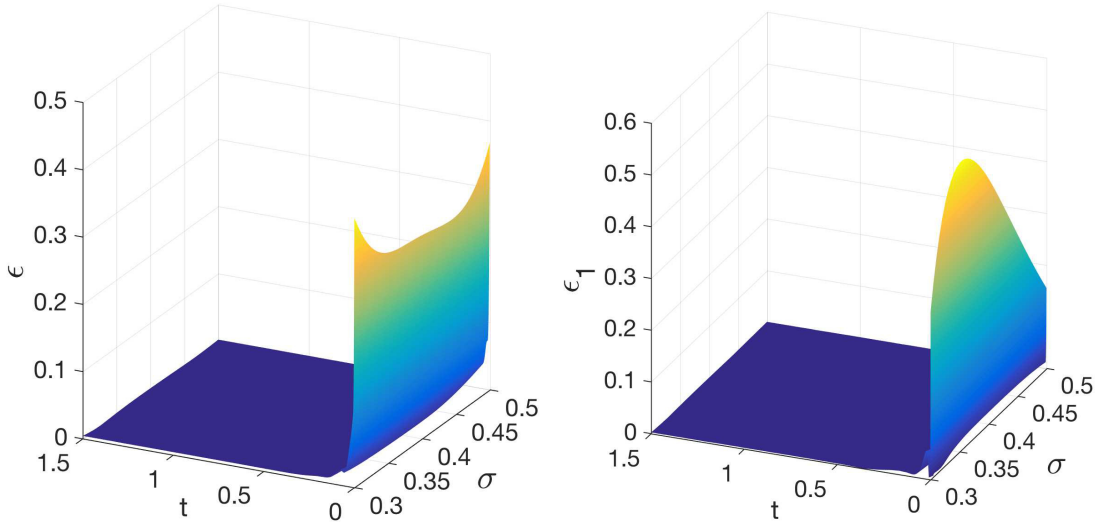


Figure 2.7: Error $\bar{\epsilon}$ depending on the number of modes



(a) Relative error integrated on the geometry

(b) Relative error on the output T_1

Figure 2.8: Relative local errors ϵ and ϵ_1 on $\Sigma \times I$

other formulations (SUPGD [28, 92]) can be used to improve the stability of the solution on the opposite case.

2.3.1.5 Bayesian inference with measurement error

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

The probability density function of the measurement noise $\pi_{\text{err}}(e)$ is 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 2.1, for a given value of the parameter σ the corresponding likelihood function value is given by (2.34):

$$\pi(T_1^{\text{obs}}|\sigma) = \int \pi(T_1|\sigma).\pi_{\text{err}}(T_1^{\text{obs}} - T_1)\mathrm{d}T_1 \tag{2.34}$$

As the model is deterministic, we get:

$$\pi(T_1|\sigma) = \delta(T_1 - \mathcal{M}(x_1, y_1, t; \sigma)) \tag{2.35}$$

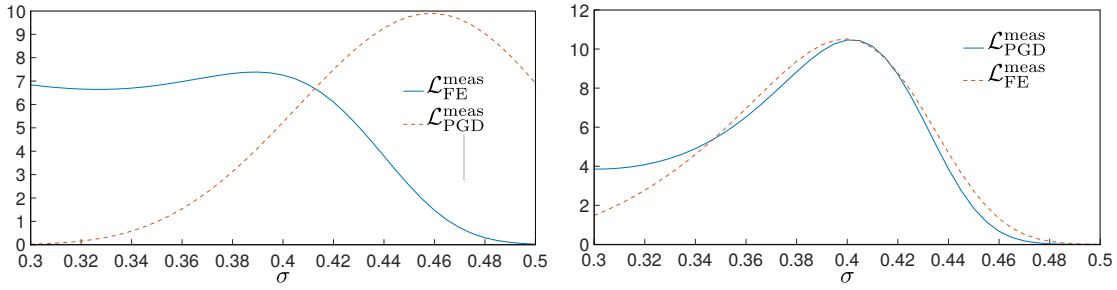
with δ the Dirac delta function and $\mathcal{M}$ the considered model (Finite Element or PGD).

In this case, the computation of the integral (2.34) reads:

$$\mathcal{L}^{\text{meas}}(\sigma) \equiv \int \pi(T_1|\sigma).\pi_{\text{meas}}(T_1^{\text{obs}} - T_1)\mathrm{d}T_1 \tag{2.36}$$

$$= \pi_{\text{err}}(T_1^{\text{obs}} - \mathcal{M}(x_1, y_1, t; \sigma)) \tag{2.37}$$

Figure 2.9 shows the likelihood functions obtained with the finite element model ($\mathcal{L}^{\text{meas}}_{FE}$) and the PGD model ($\mathcal{L}^{\text{meas}}_{PGD}$). The PGD model used is built with 20 modes. The computation is made at two different time steps for comparison.



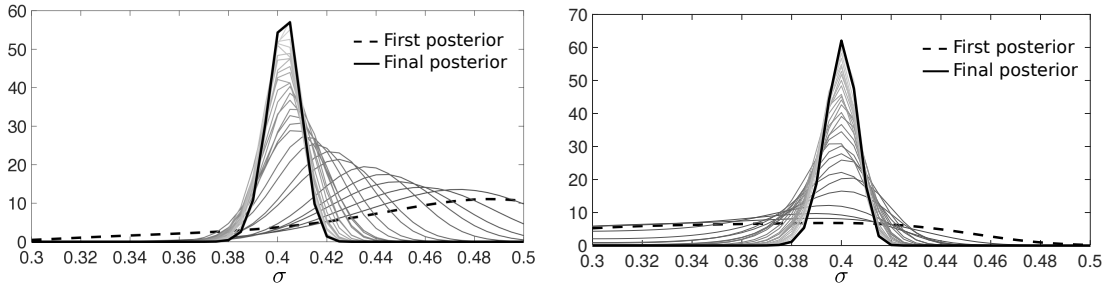
(a) Measurement at $t = 0.032$ s

(b) Measurement at $t = 0.142$ s

Figure 2.9: Likelihood functions $\mathcal{L}^{\text{meas}}$ for a single measurement at two different time steps. Functions built with PGD and Finite Element models are compared

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 2.1, the posterior probability function at a given time step is improved considering the previous posterior as the current prior. In Figure 2.10, a 25 measurements set process is considered. At each iteration, the likelihood function $\mathcal{L}^{\text{meas}}$ 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. We observe that 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.



(a) Inference made with PGD model

(b) Inference made with finite element model

Figure 2.10: 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.

2.3.1.6 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, the use of a pre-computed finite element meta-model (as the one used to compare with the PGD results) can still be used. Now, a source of uncertainty is considered in order to apply the methods of uncertainty propagation (Algorithms 2 and 3). 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 (2.20)) is considered as an uncertain parameter (parameter **i** in Figure 4). 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 2.2.3. As the parameter u has a proportional influence on the output, the change of variable function f is linear:

$$f(i) = i.T_{1u} \tag{2.38}$$

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:

$$T_m(x, y, t, \sigma, u) = u * \sum_{n=1}^{m} \Lambda_n(x, y)\lambda_n(t)\alpha_n(\sigma) \tag{2.39}$$

As the change of variable is strictly monotonic, the change of variable theorem (2.14) is applied.

Figure 2.11 shows the probability density $\pi(T_1|\sigma = 0.345)$ built with the Monte-Carlo sampling (40,000 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.

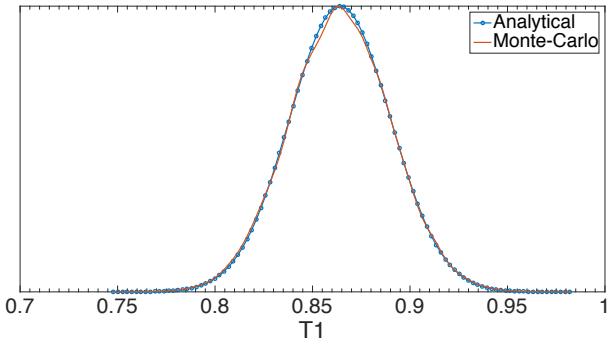


Figure 2.11: Probability density function $\pi(T_1|\sigma = 0.345)$

The probability density functions are known analytically so that the integration (2.34) to compute one point of the likelihood function with measurement error is still performed numerically.

In a second time, we consider the Peclet number (Pe) as an additional uncertain parameter. The influence of the Peclet number on the output being no longer proportional, a new PGD model (2.40) is computed:

$$T_m(\mathbf{x}, t, \sigma, Pe) = \sum_{n=1}^{m} \Lambda_n(\mathbf{x})\lambda_n(t)\alpha_{1n}(\sigma)\alpha_{2n}(Pe) \tag{2.40}$$

The approach to build this new PGD model is the same as in Algorithm 4. 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 2.12, the modes associated with the parameter Pe are plotted.

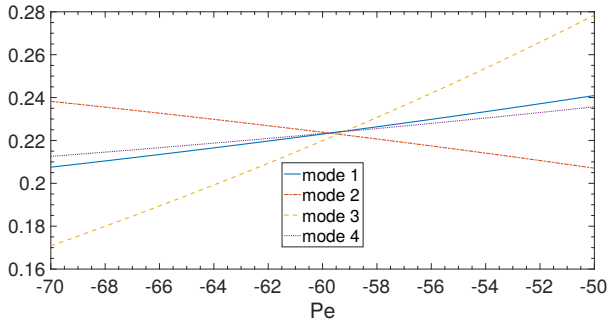


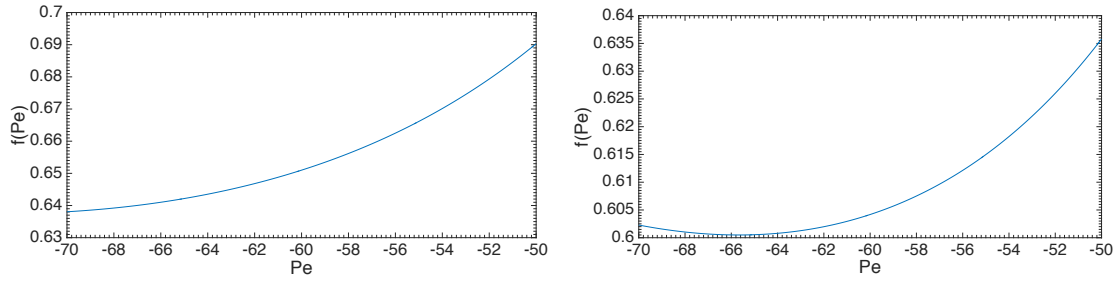
Figure 2.12: Parametric modes related to the Peclet number

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:

$$f : i \rightarrow \sum_{n=1}^{m} \Lambda_n(x_1, y_1)\lambda_n(t)\alpha_{1n}(\sigma)\alpha_{2n}(i) \tag{2.41}$$

However, this function is not necessarily monotonic. In Figure 2.13 the f functions are plotted at the same time step $t = 0.034$, with the parameter values $\sigma = 0.3$ (Figure 2.13(a)) and $\sigma = 0.35$ (Figure 2.13(b)).



(a) Monotonic change of variable f for $\sigma = 0.3$ (b) Nonmonotonic change of variable f for $\sigma = 0.35$

Figure 2.13: Change of variable function f given by two values of parameter σ at $t = 0.034$ s

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 $\frac{df}{di}$ 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 (2.15) is used to compute the probability density function given by the propagation of the uncertainty i through the model.

Figure 2.14 shows the probability density function $\pi(T_1|\sigma)$ computed with the same set of parameters as in Figure 2.13(b). As seen before, the densities given by both analytical and Monte-Carlo uncertainty propagation are very similar leading to the same likelihood functions.

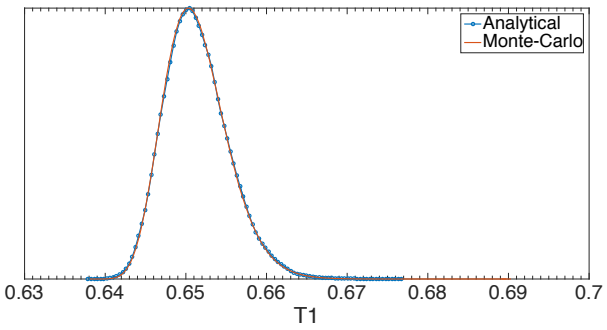


Figure 2.14: Probability density function $\pi(T_1|\sigma = 0.35)$ at $t = 0.034$ s

Eventually, in Figure 2.15 the posterior density functions obtained with both methods are plotted for a given time step $t = 0.034$ s. The prior is still chosen uniform and both methods give similar results.

With the change of variable method, studying the function f is costly especially if the function is not monotonic. Table 2.1 shows the results of the computation time required to compute one point of the likelihood function with Finite Element model/Monte-Carlo method (40,000 samples), PGD model/Monte-Carlo method (40,000 samples) and PGD model/analytical method.

For the sampling of one point of the likelihood function, the analytical approach leads to a factor 10 speed-up. Furthermore, this gain is multiplied by the number of likelihood

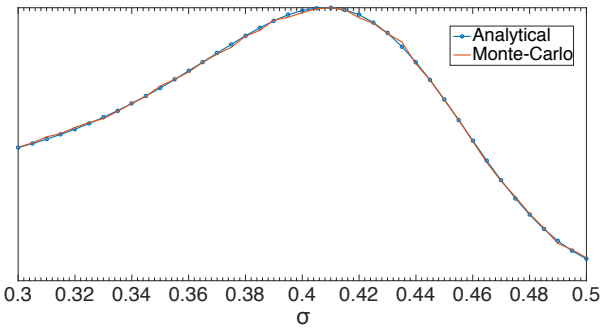


Figure 2.15: Final posterior built with analytical and Monte-Carlo methods with a single measurement at $t = 0.034$ s

Table 2.1: Comparison of the computational cost to build one point of the likelihood function

Average computation time		
Analytical PGD	Monte-Carlo PGD	Monte-Carlo FE
0.2678 s	2.5579 s	$\sim$ 4 h

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 which is larger 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.

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

2.3.2.1 Definition of the problem

The considered problem is a glued assembly adapted from [89]. 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 2.16).

The purpose of this study is to identify the variability of the assembly process leading to the structure presented in Figure 2.16. 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 known (here $E_0 = 1$ GPa). For each structure made by the assembly process, a simple test is performed: a load p is applied at the top of the structure and the vertical displacement of the point A is measured. Based on tests with 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

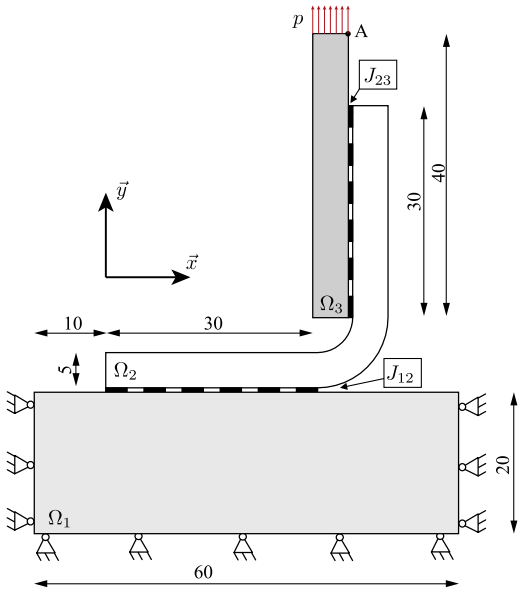


Figure 2.16: The geometry of the glued assembly

distribution with mean μ_e and standard deviation σ_e as unknown parameters:

$$\pi(e) = \frac{1}{\sigma_e\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{e-\mu_e}{\sigma_e}\right)^2\right) \tag{2.42}$$

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

2.3.2.2 PGD solution

First, the PGD model of the problem is built in order to perform the Bayesian inference detailed in Algorithm 2. 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

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

$$U_m(x_A, y_A, e) = \sum_{n=1}^{m} \Lambda_n(x_A, y_A).\alpha_n(e) \tag{2.43}$$

where (x_A, y_A) are coordinates of Point A. The reduced model is found thanks to the fixed-point Algorithm 4. In Figure 2.17, the first four parametric modes are plotted.

The PGD model is compared to a direct finite element solution for a given value of the Young modulus amplitude $e = 0.8$ (Figure 2.18). With only one mode, the relative error is $10^{-3.75}$ and the minimum error is $10^{-5.5}$. Consequently, the PGD model for this static problem is very efficient.

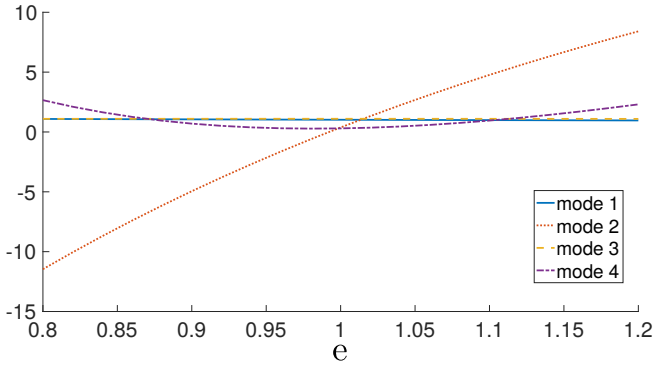


Figure 2.17: Parametric modes

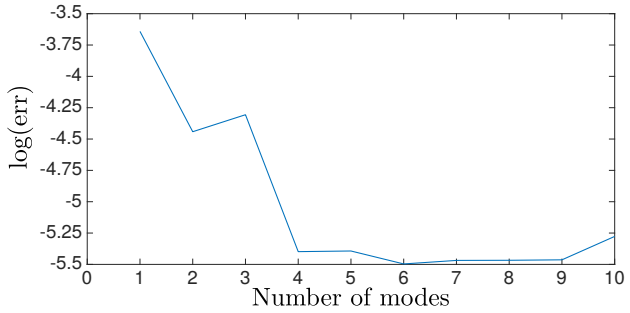


Figure 2.18: Relative error with respect to the number of modes

2.3.2.3 Bayesian inference

The main difference between this example and the previous one is that the parameter e to identify directly leads to uncertainties in the model. In order to estimate the probability density function of the random variable e, the Bayesian inference method described in Algorithm 4 is applied to the deterministic parameters to identify μ_e and σ_e. The Bayes formulation reads:

$$\pi(\mu_e, \sigma_e | u^{\text{obs}}) = \frac{1}{C}.\pi(u^{\text{obs}} | \mu_e, \sigma_e).\pi(\mu_e, \sigma_e) \tag{2.44}$$

First, to compute the likelihood function, and 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 density function with mean μ_e and standard deviation σ_e. Here also, both analytical and Monte-Carlo uncertainty propagation methods give the same density $\pi(u|\mu_e, \sigma_e)$. 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 samples according to the probability density function with mean $\mu_e = 1$ and standard deviation $\sigma_e = 0.05$. An example of a 2D likelihood function given for a single measurement is plotted in Figure 2.19.

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

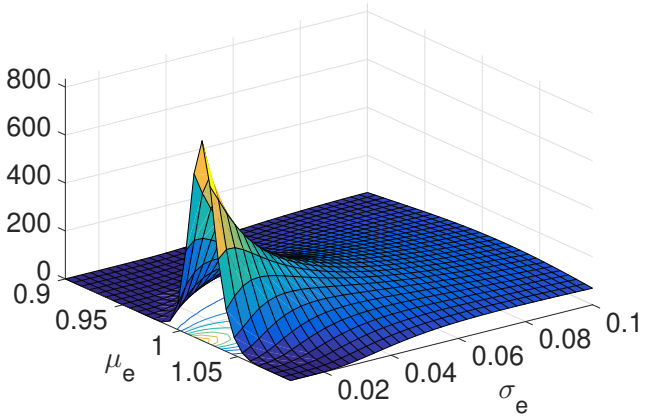
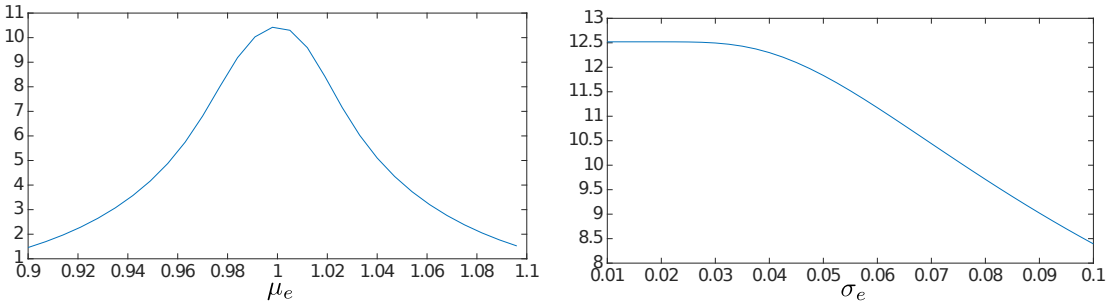


Figure 2.19: Likelihood function $\pi(u^{\text{obs}}|\mu_e, \sigma_e)$ for a single measurement

computed:

$$\pi(\mu_e|u^{\text{obs}}) = \int_{0.01}^{0.1} \pi(\mu_e, \sigma_e|u^{\text{obs}}) d\sigma_e \tag{2.45}$$

$$\pi(\sigma_e|u^{\text{obs}}) = \int_{0.9}^{1.1} \pi(\mu_e, \sigma_e|u^{\text{obs}}) d\mu_e \tag{2.46}$$



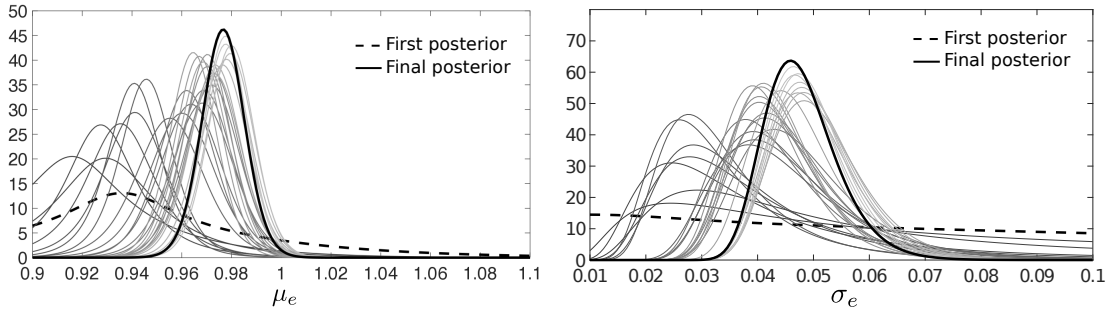
(a) Marginal density of the mean parameter.

(b) Marginal density of the standard deviation parameter

Figure 2.20: Marginal densities given by a single measurement

The marginal densities given by the Bayesian inference from one measurement are shown in Figure 2.20. We observe that the marginal density for the mean parameter is maximum for $\mu_e = 1$ which is the value used to simulate the measurements. In addition, 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 2.21, the resulting posterior densities with consecutive simulated measurements are plotted. The first marginal density 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 2.21: 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 2.22 shows the variability of the identification regarding the measurements (with 20 successive measurements). Figures 2.22(a) and 2.22(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 2.22(c) and 2.22(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.

$$\mu_e^{\text{MAP}} = \max_{\mu_e} \pi(\mu_e | u^{\text{obs}}) \tag{2.47}$$

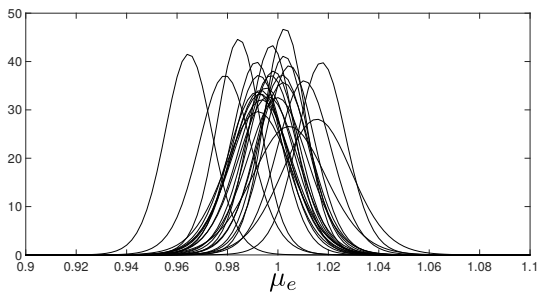
$$\sigma_e^{\text{MAP}} = \max_{\sigma_e} \pi(\sigma_e | u^{\text{obs}}) \tag{2.48}$$

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

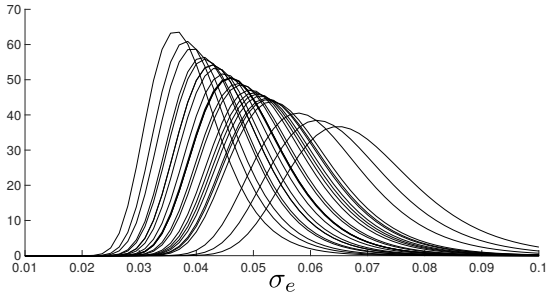
$$c_V = \frac{\text{stdd}(\pi(x))}{\text{mean}(\pi(x))} \tag{2.49}$$

where stdd represents the standard deviation of the density.

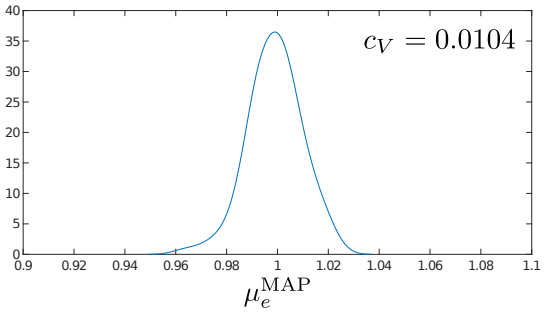
The most likely estimation values with MAP estimator are $\mu_e^{\text{MAP}} = 1$ with $c_V = 1.0 \times 10^{-2}$, and $\sigma_e^{\text{MAP}} = 0.046$ with $c_V = 1.7 \times 10^{-1}$. The estimations of the parameter with



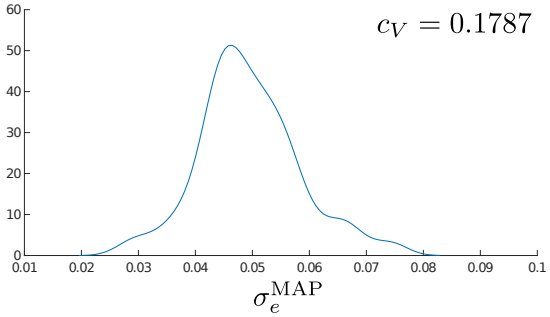
(a) 25 posterior marginal densities of the mean parameter



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



(c) Density of the MAP estimator on 200 posterior marginal densities of the mean parameter



(d) Density of the MAP estimator on 200 posterior marginal densities of the standard deviation parameter

Figure 2.22: Results of the identification process with 20 successive measurements

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.

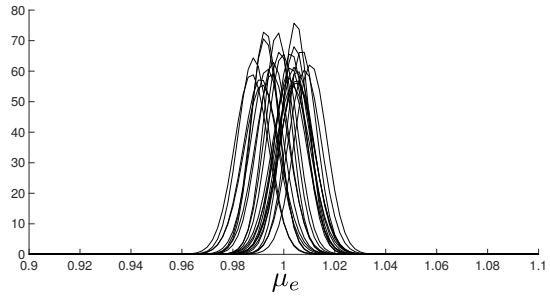
In order to show the influence of the size of the measurements sets, the same densities are plotted in Figure 2.23 with sets of 60 successive measurements.

The results are significantly improved with coefficients 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.

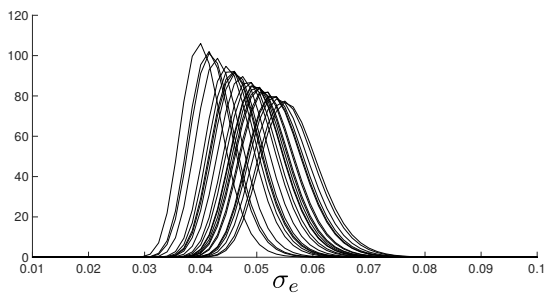
Consequently, the estimation of hyper-parameters from a stochastic parameter is very difficult and the result highly depends on the quantity of information (measurements) available. However, this is related to the a huge advantage of Bayesian identification, with a low amount of information (60 measurements), 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.

Similarly, to the welding example, the model reduction error is studied. In Figure 2.24, the relative error between the PGD models defined (2.43) with 10 modes and 1 mode is plotted:

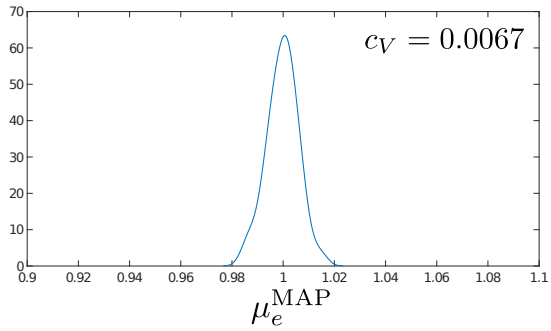
$$\text{err} = \frac{|U_{10} - U_1|}{|U_1|} \tag{2.50}$$



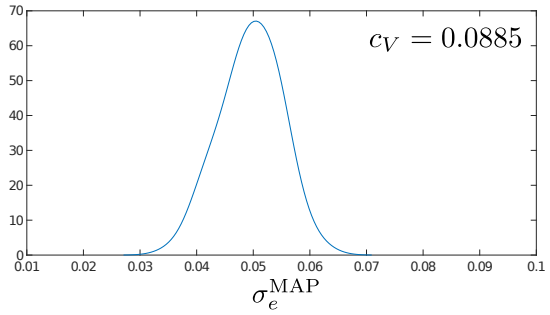
(a) 25 posterior marginal densities of the mean parameter



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



(c) Density of the MAP estimator on 200 posterior marginal densities of the mean parameter



(d) Density of the MAP estimator on 200 posterior marginal densities of the standard deviation parameter

Figure 2.23: 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 2.25 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 $\sigma_e = 0.05$. Here, a Gaussian centered noise is added on the output with a $\sigma_{\text{err}} = 0.003$ 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:

$$\pi(u^{\text{obs}}|\mu_e, \sigma_e) = \int \pi(u|\mu_e, \sigma_e).\pi_{\text{err}}(u^{\text{obs}} - u)\text{d}u \tag{2.51}$$

with $\pi_{\text{err}}(e)$ the Gaussian probability density function with standard deviation of σ_{err} and zero mean. In Figure 2.26, three likelihood functions are plotted. First, in Figure 2.26(a) the likelihood function is built from a noisy measurement without considering the measurement error. In Figure 2.26(b) the likelihood function built from the simulated measurement without the noise is represented. Finally, in Figure 2.26(c) the likelihood function is built from the same noisy measurement but taking into account the measurement error.

Figure 2.27 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

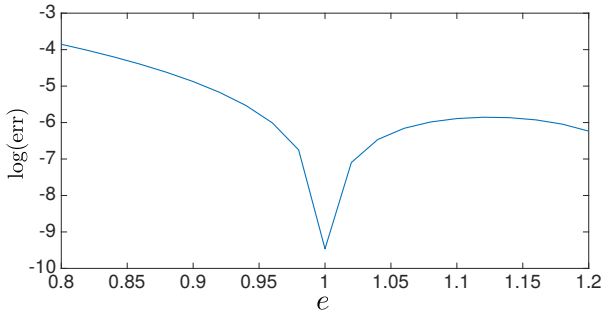
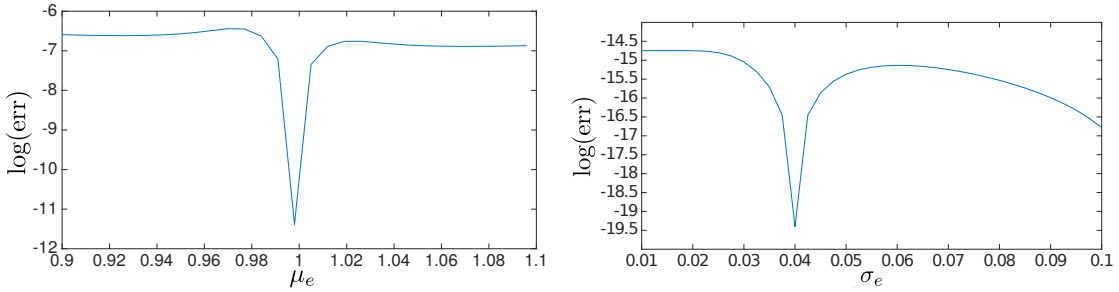


Figure 2.24: Relative error between PGD model with 1 and 10 modes



(a) Relative error for mean marginal density

(b) Relative error for standard deviation marginal density

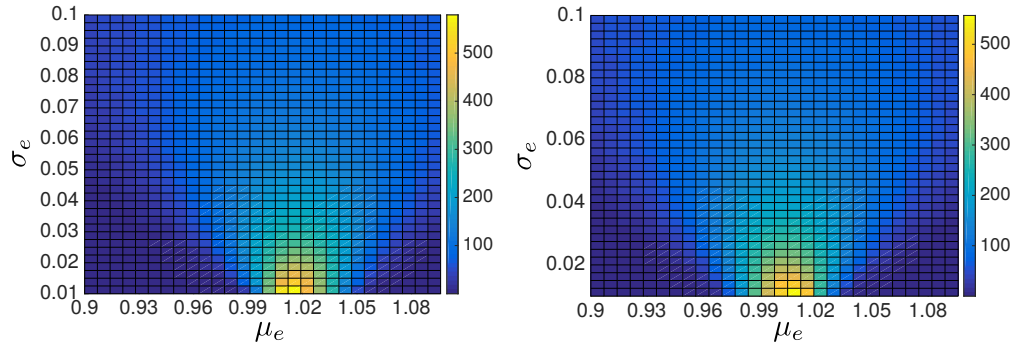
Figure 2.25: Relative errors for posterior marginal densities

represented in Figure 2.26(b) (respectively the "meas" graph in Figure 2.27). The comparison between this likelihood and the likelihood function in Figure 2.26(a) (respectively the "meas+noise" graph in Figure 2.27) 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 2.26(c) (respectively the "meas+noise+meas err" graph in Figure 2.27) represents the likelihood function built from the noisy measurement but considering the measurement error. We observe that to consider measurement error allows to have a much wider density. Thus, less trust is given to the likelihood maximum. In addition, we can observe that this maximum is closer to the likelihood maximum obtained with the true measurement.

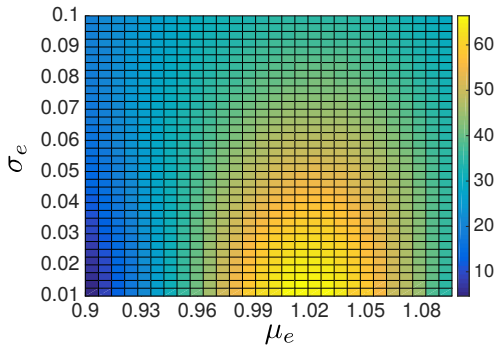
2.4 Conclusions

Through the examples studied, this chapter showed 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 result in 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 the 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



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

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

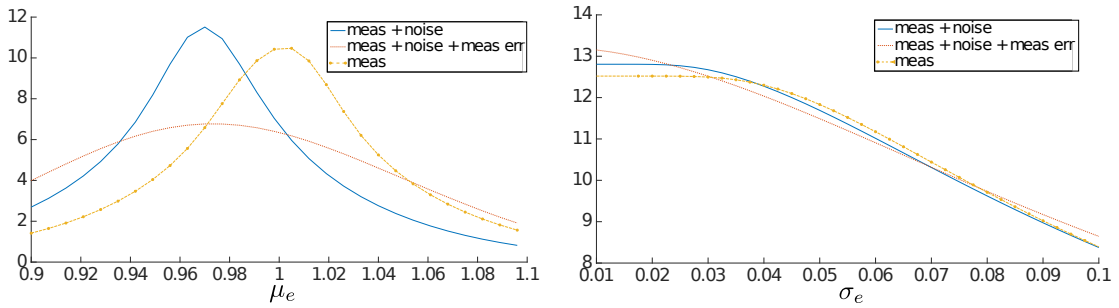


(c) Likelihood function built from measurement with noise and considering measurement error

Figure 2.26: Likelihood functions regarding measurement error

all discretized values of parameters. This way, the likelihood function is built "point by point". In order to reduce the uncertainty propagation cost, a PGD reduced model was 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 showed great benefits for Bayesian updating leading to analytical computation of integrals, or uncertainty propagation at a very low cost (summation of modes).

Nevertheless, 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 [67, 36, 56]. 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 50,000 which is a lower bound on the values found in the literature [10]. The computation times to build posterior marginal densities are compared between the "grid" and the MCMC methods



(a) Marginal densities for mean parameter

(b) Marginal densities for standard deviation parameter

Figure 2.27: Likelihood marginal densities regarding measurement error. "meas" represents the marginal densities corresponding to the likelihood function in Figure 2.26(b), "meas + noise" to Figure 2.26(a) and "meas+noise+meas err" to Figure 2.26(c).

in Table 2.2. For each method the PGD model is used with the "PGD-analytic" method presented in Section 2.2 to propagate uncertainties. For the grid method, the number of elements in the grid is specified. 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 50,000 (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 2.2: 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	$\sim 12,000$ s	6.63 s	~ 1300 s

However, the "grid sampling" is limited when the number of parameters is larger than one due to the exponential growth of the computational cost of integrals required to derive useful characterization of the posterior densities. Furthermore, the support of the integrals can be difficult to know *a priori* which makes sequential inference difficult. To palliate these limitations, the next chapter deals with the characterization of the posterior densities by means of sampling method adapted to the sequential inference of larger number of parameters.

Chapter 3

Transport Map sampling with PGD model reduction for fast dynamical Bayesian data assimilation

For full Bayesian inference (sequential or not), and as an alternative to the MCMC framework, a promising sampling technique is the so-called Transport Map method [44, 103, 138]. This recent approach aims at defining a deterministic map between a reference probability measure and the posterior measure resulting from the Bayesian inference. That way, all integrals with respect to the posterior measure are "transported" to integrals with respect to the reference measure. The map can itself characterize the posterior density, or samples can be drawn according to the posterior density by transporting samples drawn from a reference density. In another framework, full Bayesian inference problems was solved by characterizing the posterior density by means of polynomial chaos expansion and conditional expectation [113, 124, 107]. Another alternative is also presented in [43] where a hierarchical tensor representation is used to solve the forward problem. This chapter aims to couple the Transport Map method with PGD model reduction to solve sequential Bayesian inference problems. Its content is adapted from the published paper [131].

3.1 Concept of Transport Map sampling

The main idea of the Transport Map strategy is to build a deterministic coupling (using a map M) between a reference probability measure ν_ρ and a target probability measure ν_π. The purpose is to implement the change of variables such that:

$$\int g \mathrm{d}\nu_\pi = \int g \circ M \mathrm{d}\nu_\rho$$

In this framework, it is possible to transport samples drawn according to a reference density for them to become samples drawn according to the target density (Figure 3.1). This idea can be compared to the isoparametric finite element formulation, where all integral computations are performed with respect to a reference element by means of the Jacobian transformation.

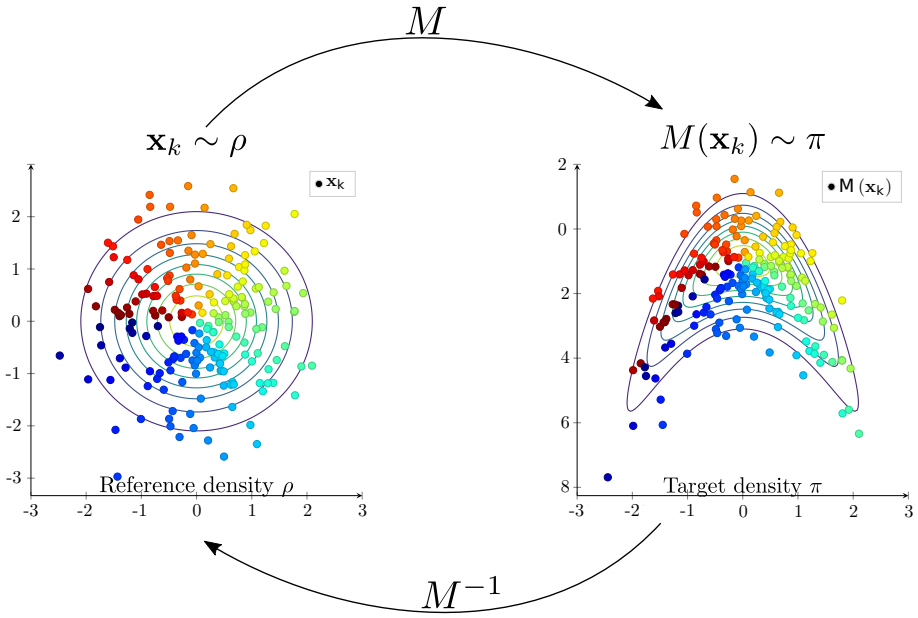


Figure 3.1: Transport Map principle

A preliminary work on optimal transports was developed in [145]. Recently, this work was adapted to Bayesian inference [138] with effective computation tools (see http://transportmaps.mit.edu). In this framework, the target density is the posterior density and the reference density is taken as a classical density which is easy to sample (e.g. standard normal Gaussian).

3.1.1 Laplace approximation

A first approach to build a map is to use the Laplace approximation described below. The analytical expression of the non-normalized density $\tilde{\pi}(\mathbf{p})$ of $\pi(\mathbf{p})$ with $\mathbf{p} \in \mathbb{R}^d$ is assumed to be known:

$$\pi(\mathbf{p}) = \frac{1}{C}.\tilde{\pi}(\mathbf{p}) \tag{3.1}$$

The stationary point $\mathbf{p}_0$ is defined as the solution of the minimization problem:

$$\mathbf{p}_0 = \min_{\mathbf{p}}(-\log(\tilde{\pi}(\mathbf{p}))) \tag{3.2}$$

Then, a second-order Taylor expansion of the function $\log(\tilde{\pi}(\mathbf{p}))$ at point $\mathbf{p}_0$ gives:

$$\log\left(\tilde{\pi}(\mathbf{p})\right) \approx \log\left(\tilde{\pi}(\mathbf{p_0})\right) + \frac{1}{2}(\mathbf{p}-\mathbf{p_0})^T\mathbb{H}(\mathbf{p_0})(\mathbf{p}-\mathbf{p_0})$$

where the matrix $\mathbb{H}(\mathbf{p_0})$ represents the Hessian matrix of $\log(\tilde{\pi}(\mathbf{p}))$ at point $\mathbf{p}_0$. An approximation of $\tilde{\pi}(\mathbf{p})$ is thus given by:

$$\tilde{\pi}(\mathbf{p}) \approx \tilde{\pi}(\mathbf{p_0})\exp\left\{\frac{1}{2}(\mathbf{p}-\mathbf{p_0})^T\mathbb{H}(\mathbf{p_0})(\mathbf{p}-\mathbf{p_0})\right\}$$

As a result, the normalization constant C of (3.1) is given by the integration of the Gauss function:

$$\begin{aligned} C &= \int_P \tilde{\pi}(\mathbf{p})\mathrm{d}\mathbf{p} \\ &\approx \int_P \tilde{\pi}(\mathbf{p_0})\exp\left\{\frac{1}{2}(\mathbf{p}-\mathbf{p_0})^T\mathbb{H}(\mathbf{p_0})(\mathbf{p}-\mathbf{p_0})\right\}\mathrm{d}\mathbf{p} \\ &= \tilde{\pi}(\mathbf{p_0})\frac{(2\pi)^{\frac{d}{2}}}{|\mathbb{H}(\mathbf{p}_0)|^{\frac{1}{2}}} \end{aligned}$$

Finally, an approximation of the density $\pi(\mathbf{p})$ is:

$$\begin{aligned} \pi(\mathbf{p}) &\approx \frac{(2\pi)^{\frac{d}{2}}}{|\mathbb{H}|^{\frac{1}{2}}}\exp\left\{\frac{1}{2}(\mathbf{p}-\mathbf{p_0})^T\mathbb{H}(\mathbf{p_0})(\mathbf{p}-\mathbf{p_0})\right\} \\ &= \mathcal{N}\left(\mathbf{p_0}, -\mathbb{H}(\mathbf{p}_0)^{-1}\right) \end{aligned}$$

The Laplace approximation method thus finds the best Gaussian approximation of $\pi(\mathbf{p})$ with mean corresponding to the density mode $\mathbf{p}_0$. It is defined by the linear map $L(\mathbf{p}) = \mathbf{p}_0 - \mathbb{H}^{-\frac{1}{2}}\mathbf{p}$. To build $\mathbb{H}^{\frac{1}{2}}$, eigenvectors $\mathbf{P}$ and diagonal matrix $\mathbf{D}$ are computed from $\mathbb{H}$ such that $\mathbb{H} = \mathbf{PDP}^{-1}$. Then, $\mathbb{H}^{\frac{1}{2}} = \mathbf{PD}^{\frac{1}{2}}\mathbf{P}^{-1}$ with $\mathbf{D}^{\frac{1}{2}}$ the diagonal matrix built from the square roots of eigenvalues.

Once the map L is computed, if samples $\mathbf{p_i}$ are drawn from the standard normal distribution $\mathcal{N}(0, \mathbb{I}_d)$, then the samples $L(\mathbf{p_i})$ are drawn according to $\mathcal{N}(\mathbf{p_0}, -\mathbb{H}^{-1})$ which is an approximation of π. This method is a first approach to transform samples drawn from a reference density (here standard normal Gaussian) to samples which are approximately distributed according to the target density. The method is fully deterministic and only requires the solution of the minimization problem (3.2) and the computation of the second-order derivatives of density $\tilde{\pi}(\mathbf{p})$ to compute the Hessian matrix $\mathbb{H}$.

3.1.2 Computation of transport maps

In a general framework, the posterior densities of interest are not necessarily Gaussian. A way to build more accurate transport maps from a reference distribution ν_ρ to the target distribution ν_π is to use the methodology presented in [138]. In this section, we highlight the principles of this methodology. From the standard normal reference density ρ, the purpose (see Figure 3.1) is to build the invertible map $M : \mathbb{R}^d \to \mathbb{R}^d$ such that:

$$\nu_\pi \approx M_\sharp \nu_\rho = \rho \circ M^{-1}|\det \nabla M^{-1}| \tag{3.3}$$

To quantify the difference between the two distributions ν_π and $M_\sharp\nu_\rho$, the Kullback-Leibler divergence $\mathcal{D}_{KL}$ is used:

$$\mathcal{D}_{KL}(M_\sharp\nu_\rho||\nu_\pi) = \mathcal{D}_{KL}(\nu_\rho||M_\sharp^{-1}\nu_\pi) \tag{3.4}$$

$$= \mathbb{E}_\rho\left[\log\frac{\nu_\rho}{M_\sharp^{-1}\nu_\pi}\right] \tag{3.5}$$

$$= \int_P \left[\log(\rho(\mathbf{p})) - \log([\pi\circ M](\mathbf{p})) - \log(|\det\nabla M(\mathbf{p})|)\right]\rho(\mathbf{p})\mathrm{d}\mathbf{p} \tag{3.6}$$

Maps M are sought among Knothe-Rosenblatt rearrangements (i.e lower triangular and monotonic maps). This particular choice of structure is motivated by the properties of unique minimizer of (3.6), optimality regarding the weighted quadratic cost, and computational feasibility. Detailed motivations are further described in [103, 138] and references within. Then, maps M are parameterized as:

$$M(\mathbf{p}) = \begin{bmatrix} M^1(\mathbf{a}_c^1, \mathbf{a}_e^1, p_1) \\ M^2(\mathbf{a}_c^2, \mathbf{a}_e^2, p_1, p_2) \\ \vdots \\ M^d(\mathbf{a}_c^d, \mathbf{a}_e^d, p_1, p_2, ..., p_d) \end{bmatrix} \tag{3.7}$$

with:

$$M^k(\mathbf{a}_c^k, \mathbf{a}_e^k, \mathbf{p}) = \Phi_c(\mathbf{p})\mathbf{a}_c^k + \int_0^{p_k} (\Phi_e(p_1, ..., p_{k-1}, \theta)\mathbf{a}_e^k)^2 \mathrm{d}\theta \tag{3.8}$$

Functions Φ_c and Φ_e are the Hermite polynomials with coefficients $\mathbf{a_c}$ and $\mathbf{a_e}$. The order of the map is defined by the maximum order of the Hermite polynomials.

Finally, with the parametrization described in (3.7) and (3.8), the map M is constructed from the minimization of the functional (3.8) with respect to the coefficients $\mathbf{a}_e^k$ and $\mathbf{a}_c^k$. The integral in (3.6) can be approximated by a Gaussian quadrature rule (with sets $(\omega_i, \mathbf{p}_i)_{i=1}^N$) such that the minimization problem to solve reads:

$$\min_{\mathbf{a}_c^{1,...,d}, \mathbf{a}_e^{1,...,d}} \sum_{i=1}^{N} \omega_i \left[-\log(\tilde{\pi} \circ M(\mathbf{a}_c^{1,...,d}, \mathbf{a}_e^{1,...,d}, \mathbf{p}_i) - \log(|\det \nabla M(\mathbf{a}_c^{1,...,d}, \mathbf{a}_e^{1,...,d}, \mathbf{p}_i))|) \right] \tag{3.9}$$

In (3.9), the target density π can be replaced by its non-normalized version as the problem to solve consists in a minimization. This method allows to find a deterministic coupling between a reference density ρ and a target density π. Once the map M is found, it can be used for sampling purposes by transporting samples drawn from ρ to samples drawn from π. Similarly, Gaussian quadrature $(\omega_i, \mathbf{p}_i)_{i=1}^N$ for ρ can be transported to quadrature $(\omega_i, M(\mathbf{p}_i))_{i=1}^N$ for π.

3.1.3 Sequential Bayesian inference using transport maps

Another interesting advantage of the Transport Map sampling is that it can exploit the Markov structure of the target density by building the transport map by composition of low-order maps [138] (see Figure 3.2). This is particularly adapted to the sequential updating of stationary parameters. Indeed, at each assimilation of a new measurement $\mathbf{d}_i^{\text{obs}}$, the previous posterior is multiplied by the likelihood function π_{t_i} involving the new measurement:

$$\pi(\mathbf{p}|\mathbf{d}_1^{\text{obs}}, ..., \mathbf{d}_i^{\text{obs}}) \propto \prod_{j=1}^{i} \pi_{t_j}(\mathbf{d}_j^{\text{obs}}|\mathbf{p}).\pi(\mathbf{p}) \tag{3.10}$$

Instead of computing at each assimilation step a full map from reference ρ to target $\pi(\mathbf{p}|\mathbf{d}_1^{\text{obs}}, ..., \mathbf{d}_i^{\text{obs}})$, the idea is to compute the coupling between $\pi(\mathbf{p}|\mathbf{d}_1^{\text{obs}}, ..., \mathbf{d}_{i-1}^{\text{obs}})$ and $\pi(\mathbf{p}|\mathbf{d}_1^{\text{obs}}, ..., \mathbf{d}_i^{\text{obs}})$. Then, the coupling between the reference ρ and the posterior density at assimilation step t_i will read:

$$(M_1 \circ .... \circ M_i)_\sharp \, \rho(\mathbf{p}) = (\mathbb{M}_i)_\sharp \, \rho(\mathbf{p}) \approx \pi(\mathbf{p}|\mathbf{d}_1^{\text{obs}}, ..., \mathbf{d}_i^{\text{obs}})$$

The map M_1 is the map coupling the reference density $\rho(\mathbf{p})$ to the first posterior $\pi(\mathbf{p}|\mathbf{d}_1^{\text{obs}}) \propto \pi_{t_1}(\mathbf{d}_1^{\text{obs}}|\mathbf{p}).\pi(\mathbf{p})$. Each map M_i, $i \in \{2, ..., N_t\}$, is found by computing the transport map between the reference ρ and the target π_i^* defined as:

$$\pi_i^*(\mathbf{p}) = \pi_{t_i}(\mathbf{d}_i^{\text{obs}}|\mathbb{M}_{i-1}(\mathbf{p})).\rho(\mathbf{p}) \tag{3.11}$$

The advantage of this formulation is that the theoretical computation complexity of each assimilation step is constant.

3.1.4 Convergence criteria

Once a map M is computed, the Transport Map method allows one to assess the quality of the approximation by defining a clear convergence criterion. In [44, 138], the variance diagnostic ϵ_σ is defined as:

$$\epsilon_\sigma = \frac{1}{2}\mathbb{V}\text{ar}_\rho\left[\log\frac{\nu_\rho}{M_\sharp^{-1}\nu_\pi}\right] \tag{3.12}$$

The integration involved in the variance diagnostic is performed with respect to the density ρ such that the same quadrature rule as in the computation of the Kullback-Liebler divergence is used. In that sense the computation cost of the variance diagnostic is very low. Thanks to the convergence criterion based on variance diagnostic, an automatic algorithm can be proposed to deal with sequential parameter inference problems (Algorithm 5). The schematic principle of the sequential maps computation is presented in Figure 3.2. For the assimilation of the first measurement $\mathbf{d}_1^{\text{obs}}$, a linear map L is computed thanks to the Laplace approximation described in Section 3.1.1. Then, this map is applied to build an intermediate density which is closer to the reference density with approximately a zero mean and a unit covariance matrix. This step acts as a normalization over the parametric space. Then, the map M_1 is computed between the reference and this intermediate density. The linear transformation step helps the convergence as the target density becomes closer to the standard normal density. For the sequential assimilation of the other measurements, the maps are computed between the reference density (standard normal) and the target densities π_i^* (see (3.11)) which are the posterior densities affected by the inverse transformation of maps already computed for all previous assimilation steps.

The computations for each assimilation step are described in Algorithm 5. The adaptivity of the maps is performed thanks to the variance diagnostic. While the tolerance on the diagnostic is not fulfilled, a higher-order map is computed and composed with the previous computed map. The purpose here is to initialize the computation of the new increased order map by the previous lower-order map computed. As the target density is pulled-back by an approximated map, the new map should be less complex to compute.

In Algorithm 5, the number of minimization steps (and therefore the computational cost) increases with the iterations on the maps order. In the next sections, the algorithms used to solve the minimization problem will be discussed.

3.2 Transport Map sampling with PGD models

Here, as in Chapter 2, a PGD formulation of the forward model is used. The model output is assumed to be a field $\mathbf{u}$ (temperature, displacement,...) related to the solution of a

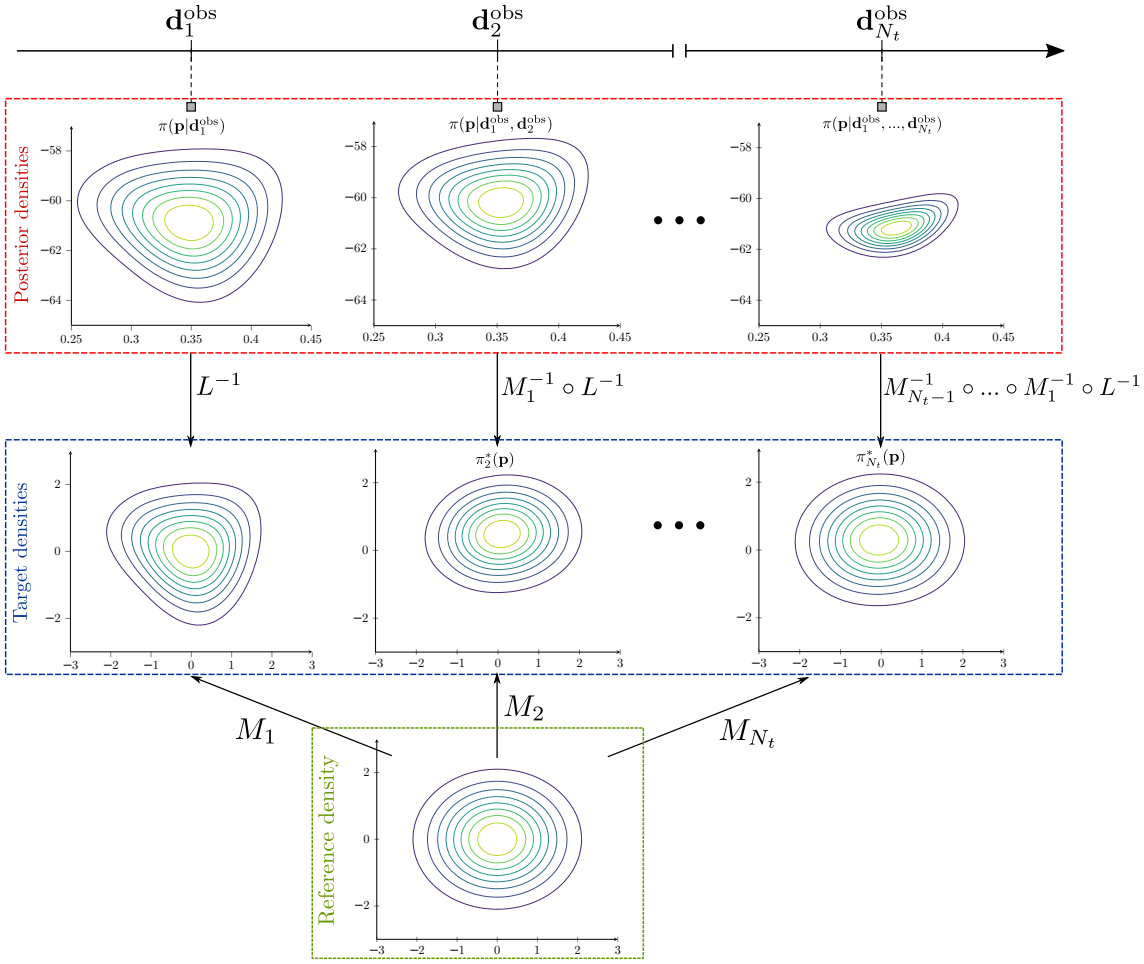


Figure 3.2: Sequential inference with transport maps

PDE. This model is supposed to be time-dependent in addition to a dependency on some parameters $\mathbf{p} \in \mathbb{R}^d$. Then, the PGD approximation $\mathbf{u}_m$ of the field $\mathbf{u}$ reads:

$$\mathbf{u}_m(\mathbf{x}, t, \mathbf{p}) = \sum_{k=1}^{m} \Lambda_k(\mathbf{x}) \lambda_k(t) \prod_{i=1}^{d} \alpha_{ik}(p_i) \tag{3.13}$$

This decomposition can be computed with the progressive-Galerking PGD method presented in Chapter 1.

3.2.1 Use of PGD models to compute transport maps

Once the PGD approximation $\mathbf{u}_m(\mathbf{x}, t, \mathbf{p})$ is built, an explicit formulation of the inference non-normalized posterior density can be derived. By the observation operator $\mathcal{O}$, the output $\mathbf{d}(\mathbf{p}, t) = \mathcal{O}(\mathbf{u}_m(\mathbf{x}, t, \mathbf{p}))$ is extracted from the field $\mathbf{u}_m(\mathbf{x}, t, \mathbf{p})$. If the observed data are N local sensors of the field u on the space coordinates $\mathbf{x}_k$, $k \in \{1, ..., N_o\}$, the output model reads:

$$\mathbf{d}_m(\mathbf{p}, t) = (u_m(\mathbf{x}_1, \mathbf{p}, t), ..., u_m(\mathbf{x}_{N_o}, \mathbf{p}, t)) \tag{3.14}$$

with explicit dependency on $\mathbf{p}$. In the end, the non-normalized posterior density $\bar{\pi}$ reads:

$$\bar{\pi}\left(\mathbf{p}|\mathbf{d}_1^{\text{obs}},...,\mathbf{d}_i^{\text{obs}}\right) = \prod_{j=1}^{i} \pi_{\text{meas}}\left(\mathbf{d}_j^{\text{obs}} - \mathbf{d}\left(\mathbf{p},t_j\right)\right).\pi(\mathbf{p}) \tag{3.15}$$

This leads to cost effective evaluations of the non-normalized posterior as highlighted in Chapter 2. Moreover, PGD is highly beneficial in the sampling procedure using transport maps. Indeed, as presented in Section 3.1, transport maps are deterministic couplings between probability densities computed by solving the minimization problem (3.9). This problem can be addressed using function information alone (Simplex method [117], or Simulated Annealing [72]). However, with PGD models, first- and second-order derivatives can also be exploited as the partial derivatives of the model with respect to parameters can be directly computed as:

$$\frac{\partial^n \mathbf{u}_m}{\partial p_j^n}(\mathbf{x},t,\mathbf{p}) = \sum_{k=1}^{m} \Lambda_k(\mathbf{x})\lambda_k(t)\frac{\partial^n \alpha_{jk}}{\partial p_j^n}(p_j)\prod_{\substack{i=1\\ i\neq j}}^{d}\alpha_{ik}(p_i) \tag{3.16}$$

The parametric modes being finite element functions, the derivations are easily performed on one-dimensional shape functions:

$$\frac{\partial^n \alpha_{jk}}{\partial p_j^n}(p_j) = \frac{\partial^n}{\partial p_j^n}\left[N(p_j)\right]\left\{\alpha_{jk}\right\} \tag{3.17}$$

with $[N(p_j)]$ the shape functions matrix for the parametric space $\mathcal{P}_j$ and $\{\alpha_{jk}\}$ the nodal values vector for mode k. To compute gradient and Hessian quantities, the first- and second-order derivatives of the parametric modes are computed and stored in the offline phase. Thanks to the separated representation of the PGD, cross-derivatives can also be computed by combinations of univariate modes derivatives.

As a result, the problem (3.9) can be effectively solved by means of minimization algorithms using gradient information (such as BFGS [21]) or Hessian information (such as trust-region algorithms [73]), which speeds up the transport maps and Laplace approximation computations.

3.2.2 Uncertainty quantification with PGD models

A quantity of interest (QoI) q is introduced by applying the operator $\mathcal{Q}$ on the PGD model:

$$q(\mathbf{p},t) = \mathcal{Q}\left(\mathbf{u}_m(\mathbf{x},t,\mathbf{p})\right) \tag{3.18}$$

This QoI is defined over the whole parametric and time-space domains. Once the measurements are assimilated with Bayesian inference, the Transport Map method returns samples $\mathbf{p}_k$ ($k \in \{1,...,N_{MC}\}$) drawn from the posterior density. Since the QoI is derived from the PGD model, samples q_k ($k \in \{1,...,N_{MC}\}$) of the probability density of the QoI $\pi(q)$ can be easily computed as:

$$q_k = q(\mathbf{p}_k,t),\ k \in \{1,...,N_{MC}\} \tag{3.19}$$

Then, the pdf of q can be approximated with the kernel density estimation (KDE). This pdf can be evaluated for all time steps of the PGD model. In real-time application computations, it would be computed for time steps posterior to the assimilation time step so

that a prediction of the QoI with uncertainties is given for control purposes (see Section 3.3.2.3).

3.3 Numerical examples

In this section, the goal is to characterize posterior densities in practical examples with the general method presented in previous sections. The implementation and cost of such approaches will be discussed. All transport maps are computed thanks to the Python package available at: http://transportmaps.mit.edu.

3.3.1 3-bar truss example

3.3.1.1 Formulation of the inference problem

This example is a 3-bar truss presented in [89] with the geometry described in Figure 3.3. This is an illustrative example where the analytical formulation of the model is known. Bars (1) and (3) are assumed to have the same Young modulus e_1, the bar (2) has a Young modulus denoted e_2. The truss is loaded with force $\mathbf{F} = F_x\mathbf{e_x} + F_y\mathbf{e_y}$. From the measurement of the vertical displacement u_y^{obs} of point D, the objective is to infer the values of Young moduli e_1 and e_2.

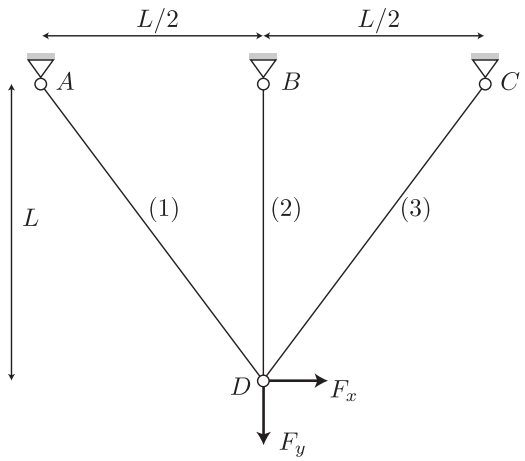


Figure 3.3: 3-bar truss geometry

For this example, the Bayes formulation reads:

$$\pi(e_1, e_2|u_y^{\text{obs}}) \propto \pi_{\text{meas}}(u_y^{\text{obs}} - u_y(e_1, e_2)).\pi(e_1, e_2) \tag{3.20}$$

The function $u_y(e_1, e_2)$ represents the vertical displacement of the truss at point D under the load $\mathbf{F}$, depending on the Young moduli e_1 and e_2. The analytical static solution reads:

$$u_y(e_1, e_2) = \frac{20F_yLe_1(16\sqrt{5}e_1^2 + 25e_1e_2)}{S(5\sqrt{5}e_1e_2 + 32e_1^2 + 50e_1e_2)^2} \tag{3.21}$$

The measurement error and the prior density are chosen to be Gaussian:

$$\pi_{\text{err}}(u_y^{\text{obs}} - u_y(e_1, e_2)) = \frac{1}{\sigma_{\text{meas}}\sqrt{2\pi}} \exp\left(-\frac{(u_y(e_1, e_2) - u_y^{\text{obs}})^2}{2\sigma_{\text{meas}}^2}\right) \tag{3.22}$$

Variables σ_{meas}, σ_1, and σ_2 represent standard deviations of the measurement error, prior density on parameter e_1, and prior density on parameter e_2, respectively. Variables μ_1 and μ_2 represent the mean of prior densities on parameter e_1 and e_2, respectively.

$$\pi(e_1, e_2|u_y^{\text{obs}}) = \frac{1}{\sigma_1\sqrt{2\pi}} \exp\left(-\frac{(e_1-\mu_1)^2}{2\sigma_1^2}\right) . \frac{1}{\sigma_2\sqrt{2\pi}} \exp\left(-\frac{(e_2-\mu_2)^2}{2\sigma_2^2}\right) \tag{3.23}$$

This way, we can define the log-gradient and the log-Hessian of the posterior density:

$$\nabla \log \pi(e_1, e_2|u_y^{\text{obs}}) = -\frac{1}{\sigma_{\text{meas}}^2} \begin{pmatrix} (u_y(e_1,e_2) - u_y^{\text{dobs}}) \dfrac{\partial u_y}{\partial e_1}(e_1,e_2) \\ (u_y(e_1,e_2) - u_y^{\text{dobs}}) \dfrac{\partial u_y}{\partial e_2}(e_1,e_2) \end{pmatrix} - \begin{pmatrix} \dfrac{1}{\sigma_1^2}(e_1-\mu_1) \\ \dfrac{1}{\sigma_2^2}(e_2-\mu_2) \end{pmatrix} \tag{3.24}$$

$$\nabla^2 \log \pi(e_1, e_2|u_y^{\text{obs}}) = -\frac{1}{\sigma_{\text{meas}}^2} \begin{pmatrix} \left(\dfrac{\partial u_y}{\partial e_1}\right)^2 + (u_y - u_y^{\text{dobs}}) \dfrac{\partial^2 u_y}{\partial e_1^2} & \dfrac{\partial u_y}{\partial e_1}\dfrac{\partial u_y}{\partial e_2} + (u_y - u_y^{\text{dobs}}) \dfrac{\partial^2 u_y}{\partial e_1 \partial e_2} \\ \text{sym} & \left(\dfrac{\partial u_y}{\partial e_2}\right)^2 + (u_y - u_y^{\text{dobs}}) \dfrac{\partial^2 u_y}{\partial e_2^2} \end{pmatrix} - \begin{pmatrix} \dfrac{1}{\sigma_1^2} & 0 \\ 0 & \dfrac{1}{\sigma_2^2} \end{pmatrix} \tag{3.25}$$

In Table 3.1 are given all the parameter values used for this example.

L	F_y	S	μ_1	σ_1	μ_2	σ_2	σ_{meas}	u_y^{obs}
500	-500	100	11	9	11	4	10	-105

Table 3.1: Values used in the 3-bars truss example

3.3.1.2 Influence of derivative information on transport map computations

A transport map is computed in order to sample the posterior density (3.20). In Table 3.2, we detail the computation times depending on the information provided to the Transport Map solver. With 0-order information, only the log-pdf function is used to solve the minimization problem. With first-order information, the minimization algorithm exploits the log-pdf function (3.23) and its gradient denoted log-gradient (3.24). Finally, with second-order information, it exploits the function, its gradient and its Hessian denoted log-Hessian (3.25). With zero- and first-order information, a BFGS algorithm [21] is used. With zero order information, the functional gradient is computed numerically while with first order information, it is explicitly given. With second-order information, a conjugate-gradient algorithm [73] is used. All calculations are performed with a third-order map. The Gauss-Hermite quadrature with a grid of 10 points on each dimension (giving 100 points overall) is chosen to solve (3.9). The convergence criterion of the minimization algorithm (which is the norm of the gradient) is fixed at 10^{-3}.

The number of function evaluations is much higher with zero-order information than with first-order information as the function is also used to numerically evaluate the gradient. Overall, we observe that adding derivatives information speeds-up the computation

Order information:	0	1	2
Iterations	33	33	11
Function evaluations	743	41	12
Gradient evaluations	-	39	22
Hessian evaluations	-	-	11
Computation time	0.90 s	0.128 s	0.111 s

Table 3.2: Computation times of the transport map using an analytical model

of the map. Adding the gradient provides for a speed-up of 7 compared with the use of the pdf alone, while the use of Hessian gives a speed-up of 1.15 compared with the use of the gradient. The poor speed-up obtained from the use of the Hessian can be explained by the low dimensionality of the problem (2 parameters) and the low evaluation cost of the analytical model.

However, in complex engineering problems, the relation between the physical output of interest and the parameters to infer is not explicit. Here, we propose to build a PGD model of the output u_y depending on the parameters e_1 and e_2:

$$u_y^{\text{PGD}}(e_1, e_2) = \sum_{n=1}^{m} \Lambda_n \alpha_{1n}(e_1)\alpha_{2n}(e_2) \tag{3.26}$$

The computation of functions Λ_n, α_{1n} and α_{2n} is detailed in [89]. Hence, the derivations needed to compute the gradient and the Hessian can be performed by deriving the functions α_{1n} and α_{2n}. As those functions are computed by the Finite Element method, the derivations are performed on the Finite Element shape functions.

In Table 3.3 are given the computation times to build the transport maps with the PGD model. The gradient information gives a 4.9 speed-up and the Hessian a 1.25 additional speed-up compared with the use of the function information alone. The numbers of necessary iterations are very close between Tables 3.2 and 3.3 as PGD and analytical models are quite similar. However, using the PGD model is more expensive in terms of computation time; indeed, the number of operations is much higher in the latter case due to the summations and products of modes. The computation cost is also increased due to the finite element structure of the parametric modes which requires interpolations. Computation times given in Tables 3.2 and 3.3 are taking into account the computation of the variance diagnostic presented in Section 3.1.4.

Order information:	0	1	2
Iterations	33	33	10
Function evaluations	703	37	11
Gradient evaluations	-	37	20
Hessian evaluations	-	-	10
Computation time	4.18 s	0.85 s	0.68 s

Table 3.3: Computation times of the transport map using PGD model

3.3.1.3 Influence of maps order

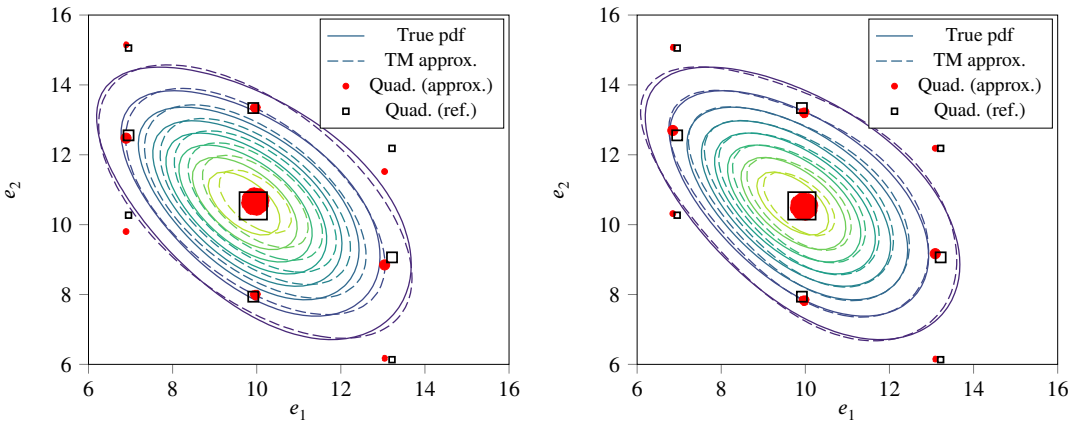
The impact of the order and of the value of the variance diagnostic on the Transport Map approximation of the posterior density is shown in Figure 3.4. Contours with continuous lines represent the non-normalized posterior density with the explicit formulation given in (3.20). Contours with dashed lines represent the posterior density given by Transport Map approximation. The comparison of dashed and continuous lines gives an idea of the approximation as the normalization constant is not involved in the contour representation. As mentioned in Section 3.1.4, transport maps allow the transport of quadrature rules. In Figure 3.4, we can also compare the quadrature Transport Map approximation. The level 3 reference Gauss-Hermite quadrature points are computed with an order 9 map and a variance diagnostic of 10^{-5}; they are represented with square marks. Approximate quadrature points are represented with circle marks. The size of the marks is proportional to the weight of the quadrature points. Subfigures 3.4(a), 3.4(b), 3.4(c) and 3.4(d) represent approximations made with maps of order 1, 2, 3 and 4, respectively. The variance diagnostic varies from 1.45×10^{-2} to 5.62×10^{-4}. The first-order approximation with a variance diagnostic of 1.45×10^{-2} gives the general shape of the density but the approximated quadrature points (circle marks), especially the smallest ones, are quite far from their reference positions (square marks). With a variance diagnostic smaller than 10^{-3} (order 3 and 4 maps), the approximation is very close to the reference in terms of contours and quadrature points.

Figure 3.5 represents the evolution of the variance diagnostic with respect to the order of the map. The computations involving the PGD model are represented with dashed lines and continuous lines represent computations performed with the analytical model. In Figure 3.5(b), the approximation of (3.9) is performed by the level 10 Gauss-Hermite quadrature (overall 100 points). Increasing the order of the transport maps permits to reduce the variance diagnostic to a certain limit imposed by the number of quadrature points chosen to compute the functional to minimize. Indeed, the higher is the order of the map, the more quadrature points are needed to compute the functional in (3.9) accurately. Figure 3.5(a) shows that with a level 5 quadrature the variance diagnostic can be decreased from 1.45×10^{-2} to 1.32×10^{-4} by increasing the map order from 1 to 5. After the 5th-order, the variance diagnostic increases due to the lack of accuracy of the quadrature. With a level 10 quadrature, Figure 3.5(b) shows that the variance diagnostic can be decreased to 10^{-5}. We can observe in both figures that the variance diagnostic is smaller by using a conjugate-gradient algorithm with Hessian information than using a BFGS algorithm with gradient information.

3.3.1.4 Computation times

For fast computation purposes, it seems preferable to use the smallest possible order as the computation cost of the map increases with the order. Furthermore, the use of high-order maps leads to costly transports of samples. In Figure 3.6, we represent the computation cost of the map added to the computation of the transport of 20,000 samples drawn according to the reference distribution (normal standard). With both PGD and analytical models, and with both gradient and Hessian information, the computation cost grows exponentially.

Another aspect of the efficiency of Transport Map sampling compared with MCMC



(a) Map order: 1, $\epsilon_\sigma = 1.45 \times 10^{-2}$

(b) Map order: 2, $\epsilon_\sigma = 3.90 \times 10^{-3}$

(c) Map order: 3, $\epsilon_\sigma = 6.61 \times 10^{-4}$

(d) Map order: 4, $\epsilon_\sigma = 5.62 \times 10^{-4}$

Figure 3.4: Comparison between contours of the true pdf (dashed lines) and Transport Map approximation (solid lines), and comparison between estimated quadrature points (circles) and reference quadrature points (squares). Subfigures 3.4(a), 3.4(b), 3.4(c) and 3.4(d) represent the approximations with order 1, 2, 3 and 4, respectively

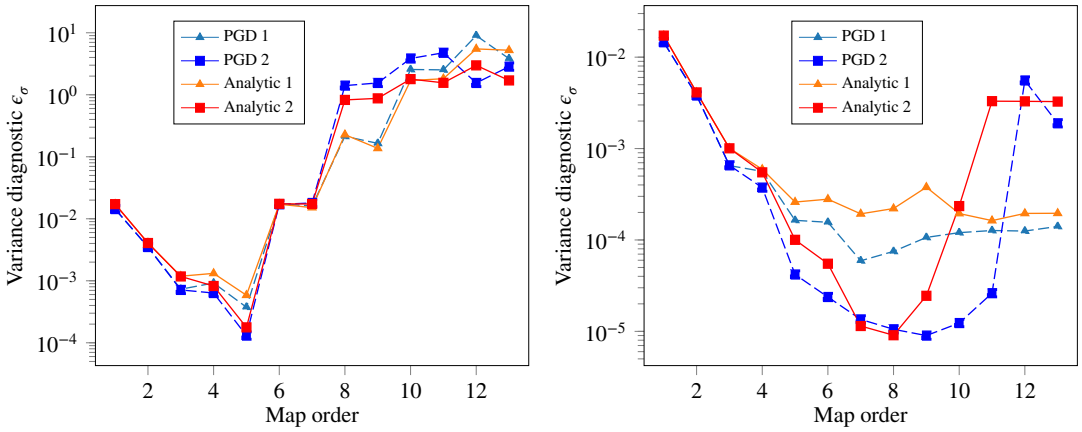
sampling is the independence of generated samples presented in [44]. In Figure 3.7, the auto-correlation functions for samples generated by MCMC (dashed line) and transport maps (continuous line) are plotted. MCMC samples are generated by the Adaptative Metropolis method described in Algorithm 1. Transport Map samples are computed by the transport of samples generated from the standard normal distribution by means of a 3rd-order map. The autocorrelation function $\rho(s)$ at lag s for samples $\{X_n\}_{n=0}^{N_s}$ is defined as:

$$\rho(s) = \frac{\text{cov}(X_n, X_{n+s})}{\text{cov}(X_0, X_s)} \tag{3.27}$$

The quantity $\text{cov}(X_n, X_{n+s})$ represents the covariance between the values of two random variables X_n and X_{n+s} in the chain separated by lag s.

In addition to that, the Integrated AutoCorrelation Time (IACT) is defined as:

$$\tau = 1 + 2\sum_{s=1}^{\infty} \rho(s) \tag{3.28}$$



(a) Computations with a level 5 quadrature (b) Computations with a level 10 quadrature

Figure 3.5: Evolution of the variance diagnostic of the maps computed with PGD and analytical model with respect to the approximation order. Continuous lines correspond to the use of gradient information and dashed lines to the use of both gradient and Hessian information. Square marks correspond to computations with an analytical model and triangle marks to computations with a PGD model. In Figure 3.5(a), computations are performed with a level 10 quadrature and in Figure 3.5(b) with a level 5 quadrature

The IACT can be interpreted as the number of necessary updates of the chain to give one effective independent sample. Here the sum involved in (3.27) is truncated at $s = 200$ to compute the IACT. The MCMC sampling gives an IACT of 8.49 and the Transport Map sampling an IACT of 1.12. On the one hand, and as expected, Transport Map sampling gives independent samples (with a theoretical IACT of 1) as it transports independent samples drawn with a random number generator. On the other hand, despite the good mixing provided by the Adaptative Metropolis algorithm, the chain produces approximately 8.49 times less independent samples. In order to have an idea of the cost efficiency of the Transport Map sampling, we compare the computation cost for generating 20,000 independent samples drawn according to the posterior density (3.20) with both Transport Map and MCMC samplings. For the MCMC sampling, we consider the time to generate $169,800$ samples which is the number of independent samples wanted $(20,000)$ times the IACT of 8.49. The transport map sampling is implemented with a t3rd-order map computed with Hessian information. The value given in Table 3.4 takes into account the computation of the map and the transport of $20,000$ samples. The computation times for both MCMC and Transport Map sampling with analytical and PGD models are also detailed.

	Transport Maps	MCMC
Analytic model	0.21 s	41.65 s
PGD model	0.83 s	42.75 s

Table 3.4: Comparison of computation times between MCMC and Transport Maps to generate $20,000$ independent posterior samples

The speed-up obtained between MCMC and Transport samplings is about 195 using the posterior density with analytical model. With the PGD model, the speed-up is about

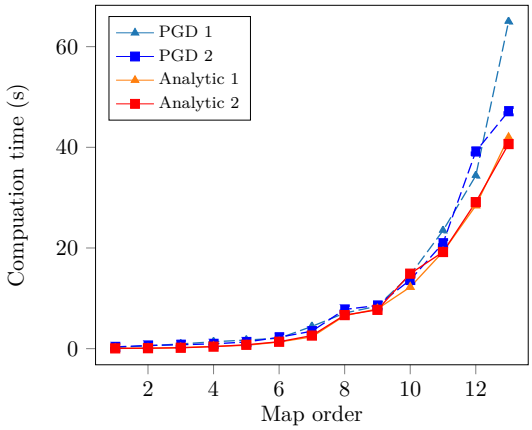


Figure 3.6: Computation time to build the transport map and to use it to transport 20,000 samples depending on the map order. Continuous lines correspond to the use of gradient information and dashed lines to the use of both gradient and Hessian information. Square marks correspond to computations with an analytical model and triangle marks to computations with a PGD model

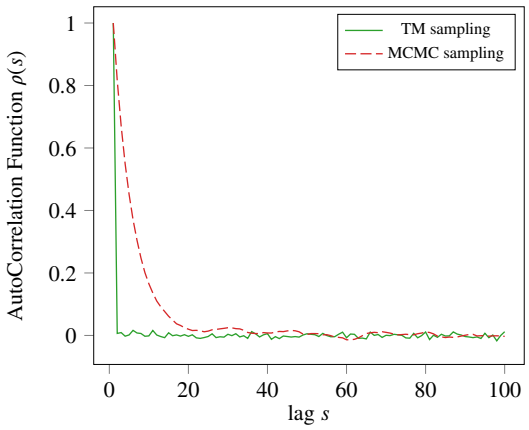
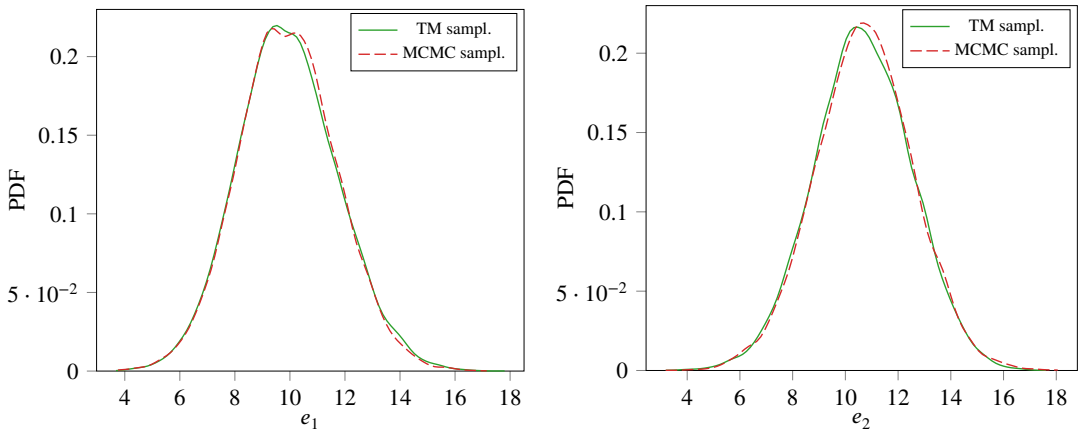


Figure 3.7: Auto-correlation functions

51. Due to the fact that the PGD model is defined with finite element functions, the computation cost is concentrated on finding and evaluating the shape functions corresponding to the points where the model output needs to be evaluated. Using transport maps, this cost appears in the evaluation of the function, its gradient and its Hessian whereas it appears only in the function evaluation using MCMC. Despite this, the computation of the map still involves less model evaluations than in MCMC sampling and the transport of the $20,000$ samples drawn according to the standard normal distribution by a third-order polynomial is still cost effective. It is worth noting that the computed transport map is numerically approximated. Consequently, transported samples are not exactly distributed according to the target density. However, by reaching a small variance diagnostic, this bias is assumed to be negligible.

Finally, we observe that marginals computed with MCMC (dashed lines) and Transport Map sampling (continuous lines) are very similar in this example as shown in Figure 3.8.



(a) Marginals on e_1 (b) Marginals on e_2

Figure 3.8: Marginals on each parameter with 20, 000 samples from MCMC and Transport Map samplings

3.3.2 Welding control example

3.3.2.1 Formulation of the inference problem

In this example, the purpose is to illustrate the use of the PGD-Transport Map algorithm and to compare it with a PGD-MCMC approach. The considered example is a welding control example already treated in Chapter 2. This application was also addressed in [127, 130]. The geometry of the problem is described in Figure 3.9.

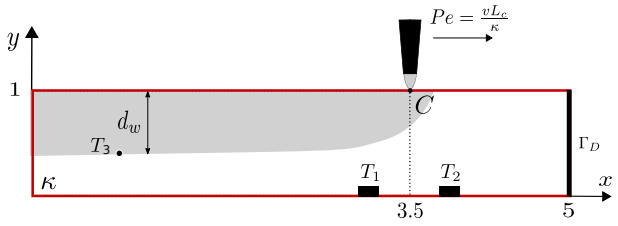


Figure 3.9: Welding control problem

The parametric PGD solution of the conduction-convection problem detailed in Chapter 2 and Appendix A reads:

$$T_k(x_k, y_k, t, \sigma, Pe) = \sum_{n=1}^{m} \Lambda_n(x_k, y_k)\lambda_n(t)\alpha_n^1(\sigma)\alpha_n^2(Pe) \tag{3.29}$$

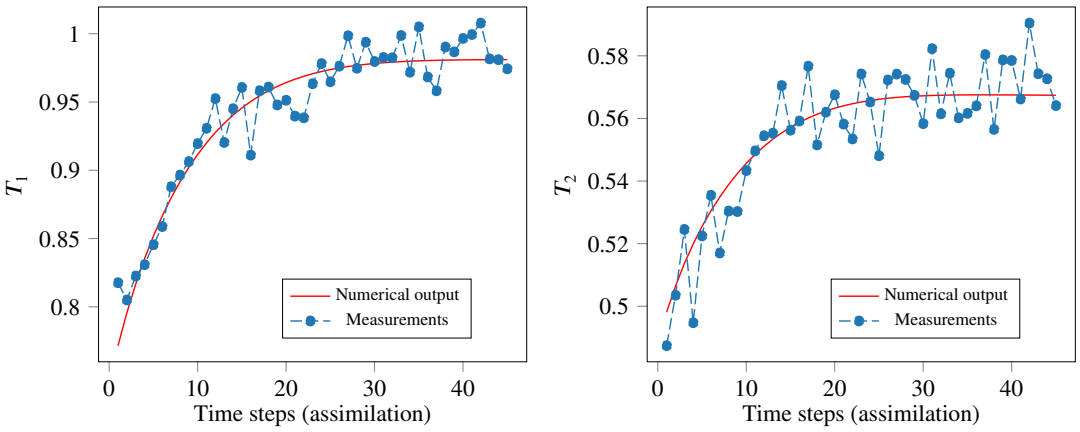
In the considered inference context, the output of interest is the temperature T_3 (see Figure 3.9) which is inaccessible by direct measurement. The knowledge of this temperature gives information about the welding depth and consequently on the welding quality in the Region of Interest (RoI). We consider the Peclet number Pe and the standard deviation of the Gaussian heat source input σ as unknown parameters of the model. From successive measurements of temperatures T_1 and T_2, we want to predict the temperature T_3 in order to know if the welding quality will be sufficient. The parameters σ and Pe are assumed to be constant over the time domain. At each time step, the temperatures T_1 and T_2 are

assimilated to refine the knowledge on those parameters. We denote τ_i, $i \in \{1, ..., N_\tau\}$, the time steps related to the discretization of the physical time used for the numerical solution. The assimilation time steps are denoted t_i, $i \in \{1, ..., N_t\}$, which correspond to the times when a measurement is acquired. In this example, physical time steps coincide with assimilation steps. The prior knowledge is modeled by independent Gaussian random variables with means $(\mu_\sigma = 0.4, \mu_{Pe} = -60)$ and variances $(\sigma_\sigma^2 = 0.003, \sigma_{Pe}^2 = 7)$.

At the considered assimilation time step t_i, $i \in \{1, ..., N_t\}$, the posterior pdf of having the parameters knowing the measurements reads:

$$\pi(\sigma, Pe|T_1^{\text{obs},1:i}, T_2^{\text{obs},1:i}) = \prod_{j=1}^{i} \pi_{t_j}(T_1^{\text{obs},j}, T_2^{\text{obs},j}|\sigma, Pe).\pi(\sigma, Pe) \tag{3.30}$$

The measurements are simulated using the PGD model with reference values $(\sigma = 0.4, Pe = -60)$. Then, independent random normal noises are added with zero means and standard deviations $\sigma_1^{\text{meas}} = 0.01925$ and $\sigma_2^{\text{meas}} = 0.01245$. Figure 3.10 shows the model output for each time step and the perturbed output which gives the measurements used for the considered example.



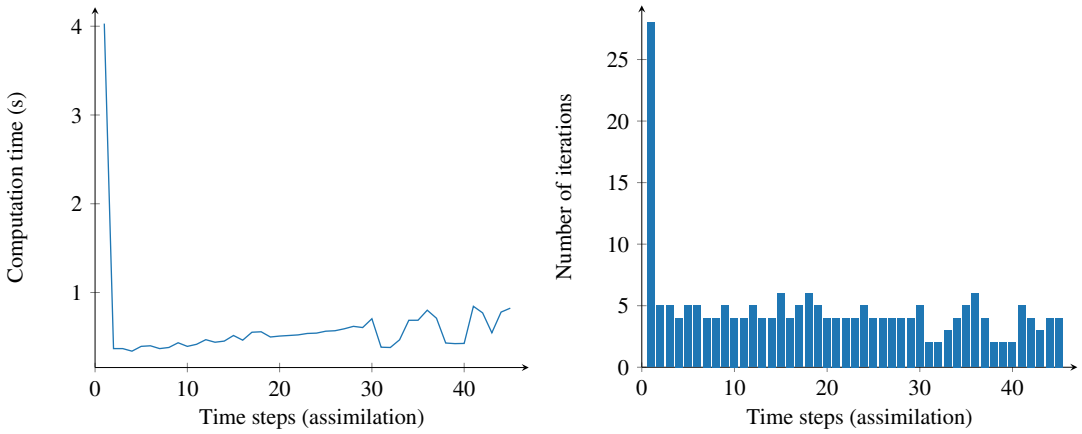
(a) Output T_1 (b) Output T_2

Figure 3.10: Measurements simulated with the numerical model

3.3.2.2 Solution of the sequential inverse problem

Algorithm 5 is used to find the successive maps $M_1, ..., M_{N_t}$. In Figure 3.11, two computation costs over the time steps are shown. Figure 3.11(a) shows the computation time to build each map M_i, $i \in \{1, ..., N_t\}$. Figure 3.11(b) shows the cost in terms of model evaluations to compute each map. Computations of the transport maps are performed using order 2 information (gradient and Hessian given explicitly to the minimization algorithm) with a level 10 Gauss-Hermite quadrature. The first step is the more costly because the order required to meet the variance diagnostic criteria is 4. After the first, step a 1st-order map is sufficient to satisfy the stopping criterion. Indeed, the first map transports the reference measure (standard normal) to the first posterior which requires the most complex map as the reference and target densities are the most unlikely. The other maps are built between intermediate posteriors which slightly differ at each step. From the second step

to the final step, we observe that the computation time slowly increases (Figure 3.11(a)) while the evaluation cost slowly decreases (Figure 3.11(b)). This is due to the fact that the evaluation of the composition of maps grows with the number of steps. One way to circumvent this issue would be to perform regression on the map composition if the number of time steps tends to be large [138]. Another issue that the method could encounter is due to the fact that the computed maps are not exact. In the sequential inference using Algorithm 5, the target density for the current time step t_i is given by (3.11) as the map $\mathbb{M}_{i-1}$ is exact. This leads to effective computations as all likelihood functions of previous times steps do not need to be computed. The counterpart is that the errors could accumulate with the number of assimilation time steps growing. One way to fix this issue would be to add a map computed between the reference density and the true posterior density given by (1.24) pulled-back by the current approximated map $\mathbb{M}_{i-1}$. This operation can be performed from time to time in order to recover the error accumulation. In the current example where the maps are low-order and computed with a small variance diagnostic, the accumulated error is neglected.



(a) Computation time for each time step

(b) Number of iterations of the minimization algorithm for each time step

Figure 3.11: Costs of the transport maps computations performed with Hessian information for each assimilation time step

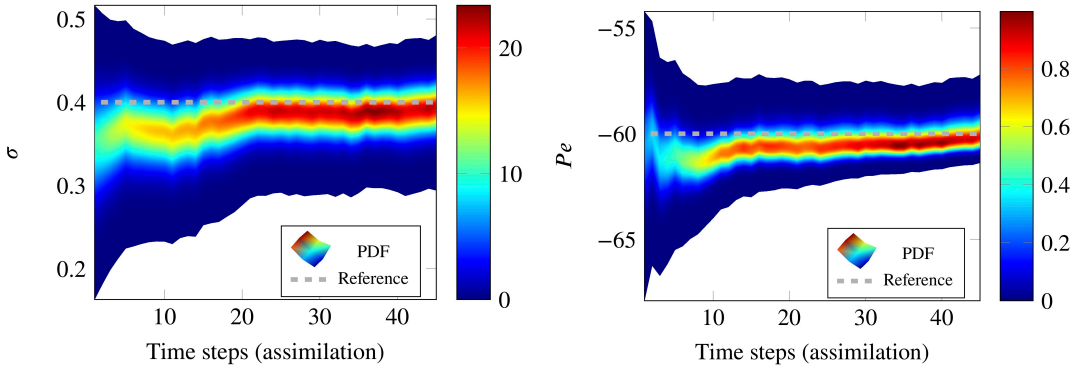
In Table 3.5, we illustrate the influence of the order information given to the minimization algorithm on computation costs of the transport maps. The computation costs of the first time step is detailed as it requires an order 4 map. For the other time steps, mean computation costs are indicated. The speed-up for the first iteration is about 5.5 between 0th-order information and 1st-order information. Between the 1st-order information and the 2nd-order information, the speed-up is about 1.34. As for the other time steps, the computed map is very simple, the speed-up is very small.

In the computation costs presented in Figure 3.11 and Table 3.5, successive transports of samples for Monte-Carlo integration are included. First, 20,000 samples are drawn according to the 2D standard normal distribution (reference measure), then successive transports of those samples are computed through the successive maps. A KDE on each coordinate separately gives an estimation of the posterior marginals. Figure 3.12 represents all the marginals for each time step. The X-axis represents the time steps and Y-axis the

Order information:	0	1	2
Number of iterations for step 1	107	33	10
Computation time for step 1	33.85 s	6.18 s	4.60 s
Average number of iterations for steps $\{2, .., 45\}$	4.2	4.16	4.13
Average computation time for steps $\{2, .., 45\}$	1.24 s	0.92 s	0.90 s

Table 3.5: Computation costs of the transport maps depending on the order information given in the minimization algorithm

parameter values. The color map gives the information on the probability density function values.



(a) Marginals on σ

(b) Marginals on Pe

Figure 3.12: Marginals computed with 20,000 samples and KDE for each assimilation time step

During the iterations over the time steps, marginals become thinner with higher pdf values giving more confidence on the parameter estimation. We can also observe that the parameter σ (Figure 3.12(a)) is less sensitive than the parameter Pe regarding the inference process.

After the 45 assimilation time steps, the algorithm gives a maximum estimator $[0.394, -60.193]$ and a mean estimator $[0.392, -59.949]$. These values are very close to the reference values $[0.40, -60]$ used to simulate the measurements.

3.3.2.3 Post-processing: uncertainty quantification

In addition to the mean and maximum *a posteriori* estimates, another post-processing application may be the temperature prediction in the RoI. Once the parameters are inferred in a probabilistic way, a post-process can be the propagation of uncertainties *a posteriori* in order to know the impact of the uncertainties on a Quantity of Interest (QoI) (here the temperature T_3 in Figure 3.9). The idea is to know the uncertainties on the value of the QoI during the process in order to assess the welding depth and consequently the welding quality. For that purpose, samples of the last posterior density taken from the Transport Map sampling are used (after the assimilation of the N_t measurements). As the PGD model gives the temperature field globally as a function of time and parameters over the whole space domain, we compute the output T_3 for all values of the parameters samples at each

physical time step τ_i, $i \in \{1, .., N_\tau\}$. For a given physical time step τ_i, the set of outputs T_3 is used to build the KDE of the pdf $\pi(T_3|\sigma, Pe, \tau_i)$ of having the temperature T_3 knowing the uncertainties on the parameters σ and Pe. Figure 3.13 represents the uncertainty map on the output T_3 depending on the physical time. The X-axis represents time so that each vertical slice represents the pdf $\pi(T_3|\sigma, Pe, \tau_i)$. The Y-axis represents temperature values and the color map gives pdf values. The discontinuous line represents the evolution of the temperature T_3 with no uncertainty on the parameters (i.e. output of the PGD with the reference values ($\sigma = 0.4, Pe = -60$), which is unavailable in real application without simulated measurements). This computation can be used to determine at which time the plates are correctly welded (i.e: $T_3 > 1$) and with which confidence. With this knowledge, a stochastic computation of the structural stiffness can be obtained in real-time.

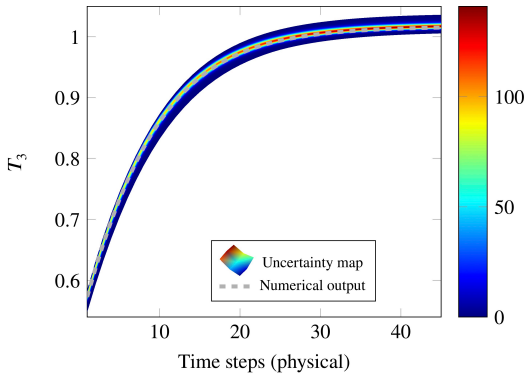


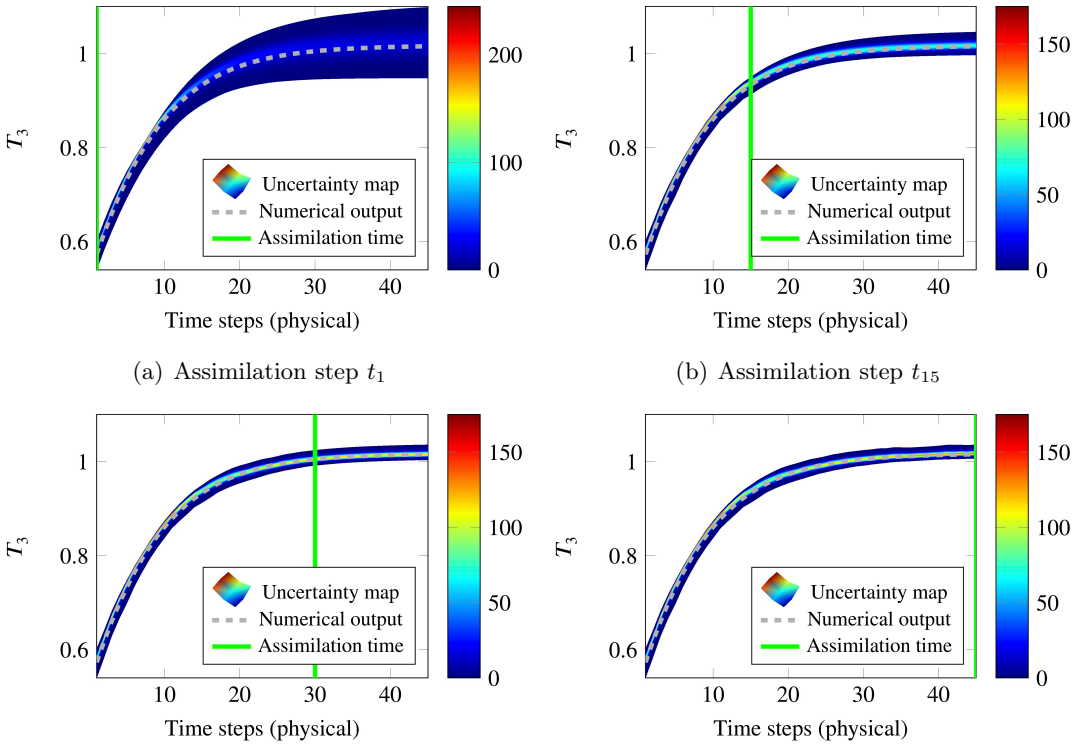
Figure 3.13: Inference of temperature T_3 computed *a posteriori* after the last assimilation step

Uncertainty propagation can also be performed in real-time in a purpose of temperature prediction in the RoI. Knowing the uncertainties on the parameters, the purpose is to predict at each assimilation time step the evolution of the temperature T_3 for the next physical time steps. Similarly to the previous example, this can be done thanks to the PGD model as the temperature field is known globally in time. Thus, at each time step, samples of the posterior density are available from the Transport Map sampling. Evaluating the output of the PGD on these parameter samples at the spatial coordinates of the temperature T_3 and for a given time step provides for output samples. A KDE on these samples provides for the pdf of having the output T_3 knowing the uncertainties on the inferred parameters. This computation is performed after each assimilation time step for all the physical time steps after the considered assimilation time step.

Figure 3.14(a) shows the result of prediction with uncertainty propagation after the first assimilation step t_1 for all the physical steps τ_i, $i > 1$. To that end, samples are drawn according to the first posterior: $\pi(\sigma, Pe|T_1^{\text{obs},1}, T_2^{\text{obs},1}) = \pi_{t_1}(T_1^{\text{obs},1}, T_2^{\text{obs},1}|\sigma, Pe).\pi(\sigma, Pe)$. The slice $[\tau_0, \tau_1]$ represents the guess on the temperature T_3 knowing the uncertainties on the parameters (σ, Pe) after the first assimilation step t_1. For $\tau_i > \tau_1$ the graph represents the prediction of the output T_3 considering the current knowledge on the parameters uncertainty (i.e. with the assimilation of the first set of measurements $T_1^{\text{obs},1}$ and $T_2^{\text{obs},1}$ only). The discontinuous line represents the evolution of the temperature T_3 with the true value of parameters ($\sigma = 0.4, Pe = -60$).

Other graphs (3.14(b), 3.14(c), 3.14(d)) show the refinement of the prediction with the refinement of the parameters uncertainty knowledge. The current measurement assimilation step is indicated by the vertical cursor. On the right of the cursor $\tau = \tau_i$, the graphs represent the prediction of the temperature T_3 after the assimilation of the measurements $T_1^{\text{obs},1:i}$ and $T_2^{\text{obs},1:i}$.

If the temperature prediction on the RoI is not satisfying in terms of uncertainties, a control procedure on input parameters (for example the heat source intensity) can be made with on-the-fly modification of control variables. On the left of the cursor graphs, each slice $[t_{j-1}, t_j]$ $(j < i)$ represents the prediction made at the assimilation time t_j (the predictions of the temperature T_3 for physical time steps anterior to the assimilation time step t_i are not updated). This procedure performed online can be used in the context of welding control.



(a) Assimilation step t_1

(b) Assimilation step t_{15}

(c) Assimilation step t_{30}

(d) Assimilation step t_{45}

Figure 3.14: Prediction of the output T_3 for all time steps after the considered assimilation step

Figure 3.15 shows the convergence of the prediction of temperature T_3 in the steady state regime ($\tau = 45$) with respect to the assimilation steps. We observe that, as foreseen, more confidence is given to the output of interest (evaluated at final time) along the real-time data assimilation process.

3.4 Conclusions

In this work, we presented a data-assimilation procedure in the full Bayesian inference context. In order to perform fast computations, the Bayesian approach was coupled with

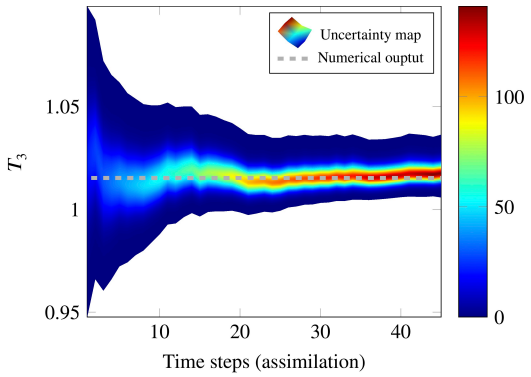


Figure 3.15: Prediction of temperature T_3 at physical time step $\tau = 45$ after each assimilation time step $t_i, i \in \{1, ..., 45\}$

PGD model reduction and Transport Map sampling. The computation of transport maps permits a fast sampling from the posterior. This latter technique is particularly suited for sequential inference as it exploits the Markov structure of the posterior to build low-order maps. Owing to PGD models, large-scale multi-parameter engineering problems can be effectively addressed and information on derivatives can be straightforwardly added to speed-up the computation of the transport maps. Eventually, the global time-space definition of the PGD models allows one to post-process the posterior density and predict quantities of interest with uncertainty propagation included.

The proposed approach seems to be suited to real-time applications and is an appealing candidate for data assimilation in the general stochastic Bayesian context as the sampling method leans on deterministic computations alone (which do not vary with respect to the random number generator seed), with a clear convergence criterion. Moreover, a trade-off between speed-up and quality can be obtained using order adaptivity of the maps knowing the error by the variance diagnostic criterion.

This chapter paves the way for future works dealing with improvements of the data assimilation algorithm. In particular, the growing computation cost of the maps composition for sequential updating could be improved as mentioned in [138]. Consideration of the modeling error in the Bayesian formulation could also be investigated following the works on PGD error estimation [75, 23]. Correcting the model bias will be the topic of Chapter 5

Algorithm 5 Procedure for sequential inference with transport maps

$\pi_1(\mathbf{p}|\mathbf{d}_1^{\text{obs}}) = \pi_{t_1}(\mathbf{d}_1^{\text{obs}}|\mathbf{p}).\pi(\mathbf{p})$: Assimilation of first measurement
$L = \textbf{LaplaceApproximation}(\pi_1)$ (see 3.1.1)
$\hat{\mathbb{M}}_1 = \mathbb{I}_d$
order $= 0$
while $\epsilon_\sigma >$ tolerance **do**
    order $=$ order $+ 1$
    M=**IntegratedSquaredMap**(order): Parametrization defined in (3.8)
    $M_1 = \underset{M}{\textbf{minDKL}}(\rho, (L \circ \hat{\mathbb{M}}_1 \circ M)_\sharp^{-1}\pi_1)$: Minimization with gradient and Hessian information
    $\hat{\mathbb{M}}_1 = \hat{\mathbb{M}}_1 \circ M_1$
    $\epsilon_\sigma = \frac{1}{2}\mathbb{V}\text{ar}_\rho\left[\frac{\rho}{(L \circ \hat{\mathbb{M}}_1)_\sharp^{-1}\pi_1}\right]$: computation of variance diagnostic
end while
$\mathbb{M}_1 = L \circ \hat{\mathbb{M}}_1$
$\mathbf{X} = \textbf{randn}(N_{mc})$: Samples from ρ
$\mathbf{Y}_1 = \mathbb{M}_1(\mathbf{X})$: Samples from π_1
for $k = 2...N_t$ **do**
    $\pi_k^*(\mathbf{p}) = \pi(\mathbf{d}_k^{\text{obs}}|\mathbb{M}_{k-1}(\mathbf{p})).\rho(\mathbf{p})$: Assimilation of k-th measurement
    $\hat{\mathbb{M}}_k = \mathbb{I}_d$
    order $= 0$
    while $\epsilon_\sigma >$ tolerance **do**
        order $=$ order $+ 1$
        M=**IntegratedSquaredMap**(order): Parametrization defined in (3.8)
        $M_k = \underset{M}{\textbf{minDKL}}(\rho, (\hat{\mathbb{M}}_k \circ M)_\sharp^{-1}\pi_k^*)$
        $\hat{\mathbb{M}}_k = \hat{\mathbb{M}}_k \circ M_k$
        $\epsilon_\sigma = \frac{1}{2}\mathbb{V}\text{ar}_\rho\left[\frac{\rho}{(\hat{\mathbb{M}}_k)_\sharp^{-1}\pi_k^*}\right]$: computation of variance diagnostic
    end while
    $\mathbb{M}_k = \mathbb{M}_{k-1} \circ \hat{\mathbb{M}}_k$
    $\mathbf{Y}_k = \mathbb{M}_k(\mathbf{X})$: Samples from π_k
end for

Chapter 4

Realistic application to the experimental characterization of damage in a concrete beam

In this chapter, the objective is to illustrate the methodology presented in the previous chapter on a real test-case. The considered experimental test is a three-points bending test on a notched concrete beam. The experimental campaign is a joint work with Dr. François Soleilhet, post-doctoral student at LMT. The overall objective of the campaign was to characterize the influence of the drying on the concrete properties through different experimental tests. Here, the focus is only made on the numerical study of the bending tests where parameters of the damage law are inferred from measurements of the displacement field in order to predict the crack propagation into the beams, and possibly the collapse of the structure.

4.1 Experimental campaign

4.1.1 Manufacturing of the specimens

Specimens used in this campaign are prismatic 840 mm × 100 mm × 100 mm beams notched at mid-span on 20% of their height (Figure 4.1). The beam are supported by two supports on the bottom side and the displacement u_{imp} is imposed on the top side. This geometry

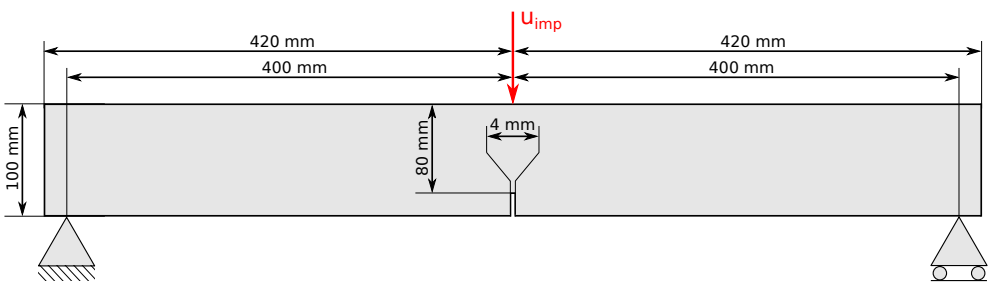
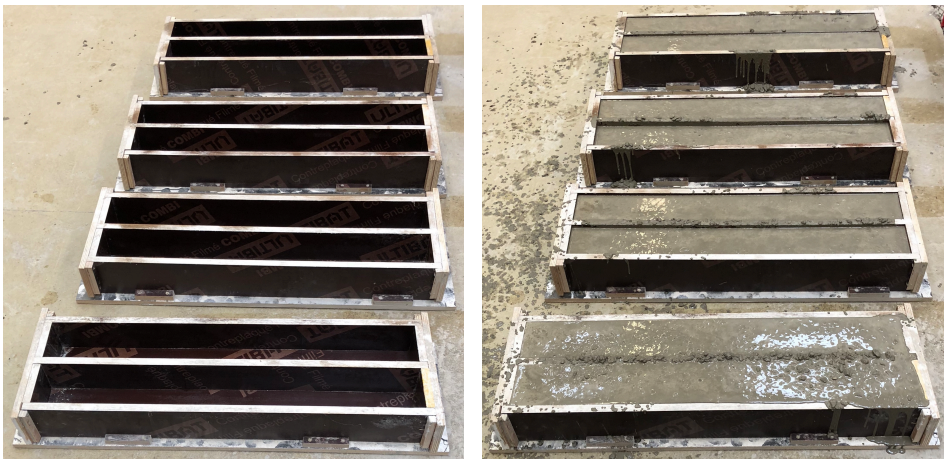


Figure 4.1: Geometry and boundary conditions of the reference test



(a) Manufacturing of the formworks

(b) Casting of the beams

(c) Uncasting of the beams

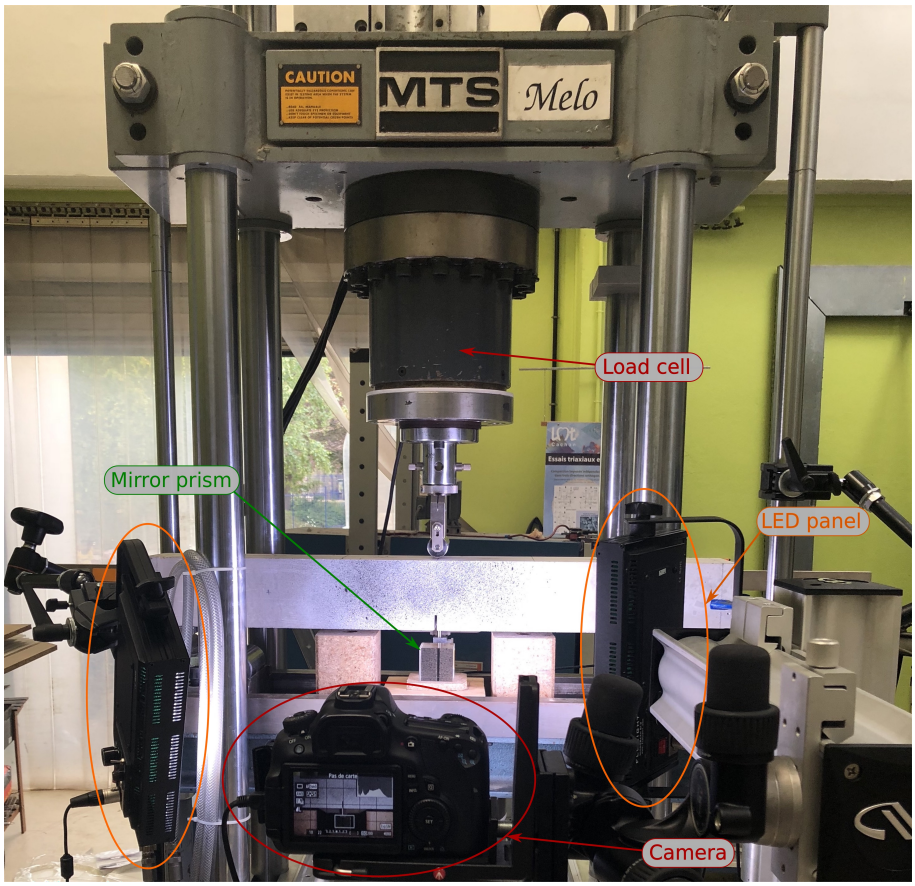
(d) Storage of the beams in controlled environment

Figure 4.2: Preparation stages of the beams

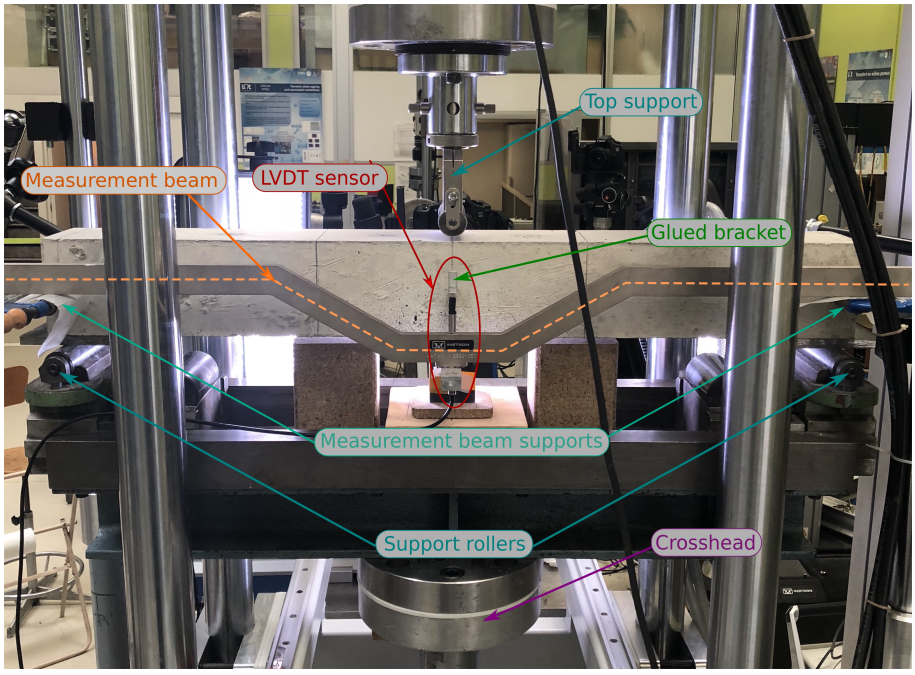
and boundary conditions follow the recommendation of [61] for the characterization of fracture energy. First, eight wood formworks are built with the shape of a 840 mm × 100 mm × 100 mm prism (Figure 4.2(a)). Second, the concrete is cast on these formworks (Figure 4.2(b)). After one week, the beams are stripped of their formworks (Figure 4.2(c)). Then, specimens are separated in two equal groups. The first group of beams is stored under water while the other group is stored in a controlled climate environment (Figure 4.2(d)) with temperature set to 25° C and a relative humidity of 30%.

4.1.2 Designing the experimental protocol

After a drying time of 70 days, beams are notched with a circular saw according to the geometry described in Figure 4.1. The experimental settings are shown in Figure 4.3. A tensile test machine with a capacity of 100 kN is used to apply the imposed boundary conditions described in Figure 4.1. At the bottom of the structure, the supports are made with steel rollers positioned according to the geometry described in Figure 4.1. Steel plates



(a) Front view of the experimental settings



(b) Back view of the experimental settings

Figure 4.3: Experimental settings

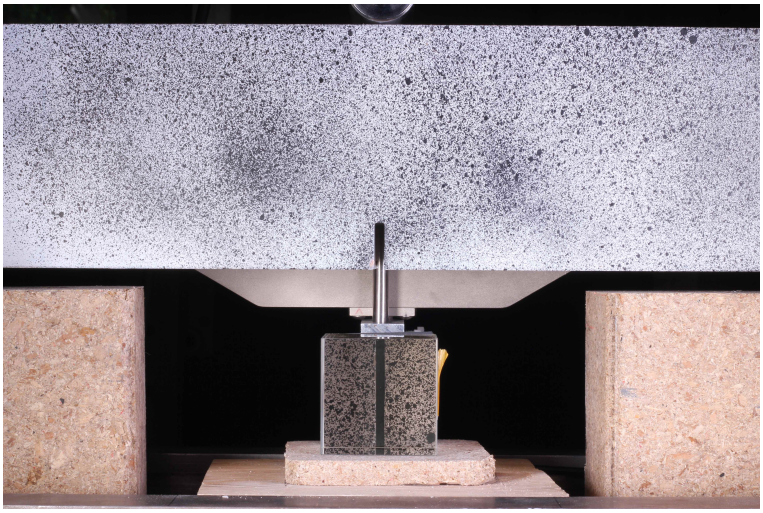
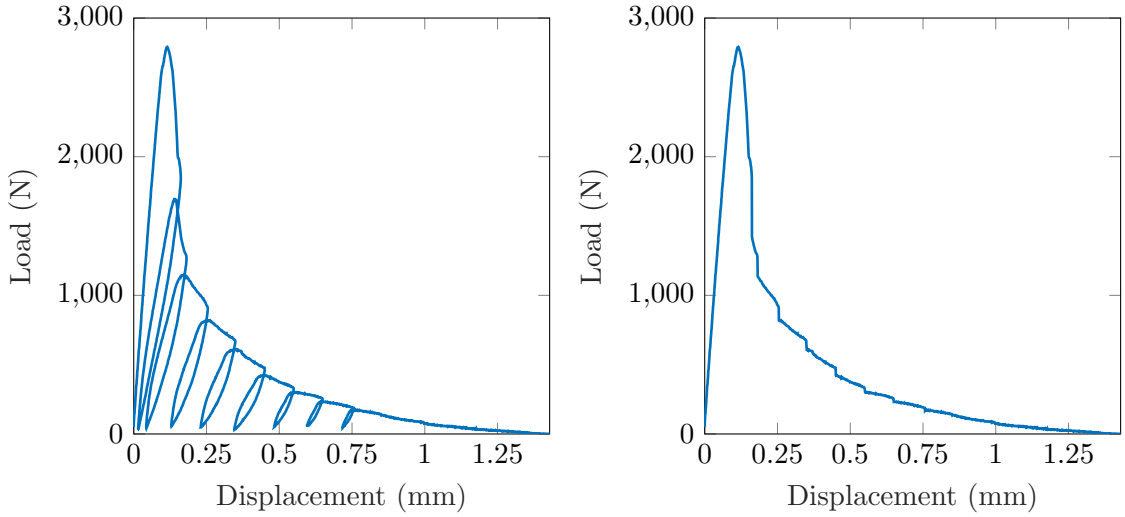


Figure 4.4: Example of an image post-processed with DIC

are inserted between the cylinders and the beam to prevent the indentation of the concrete due the concentrated stresses. The displacement is imposed with an hydraulic cylinder at the bottom side of the beam. The upper support is fixed and the load cell measures the reaction force in this support. The test is controlled by the displacement prescribed to the hydraulic cylinder. Adjustment cycles are carried out in order to fill all gaps related to positioning. After three cycles, a monotonic increasing displacement is imposed at the rate of 3×10^{-3} mm/s until the complete fracture of the specimen is observed. During the test, unloading/loading cycles are performed in order to observe the evolution of the fracture dissipated energy. This study is beyond the framework of this thesis so that only the monotonic part will be considered (no further damage appears in the unloading phase).

In addition to the load measurement from the testing machine, the test is instrumented with a displacement sensor (Linear Variable Differential Transformer (LVDT)) to precisely measure the bending of the beam. In order to do so, a measurement beam is installed on two supports at the mid-span height of the specimen corresponding to its neutral axis. Thus, one side of the LVDT sensor is fixed on the measurement beam, the other side is in contact with a bracket glued at the center of the specimen. This measurement is performed on one side of the specimen (see Figure 4.3(b)). On the other side (Figure 4.3(a)), Digital Image Correlation is performed on the central part of the specimen. For this purpose, a painted speckle pattern is sprayed on the considered surface of the concrete beam. LED panels are added in order to adjust the exposition and contrast of the picture. Pictures for DIC are taken with a camera linked to the acquisition computer. Then, during the test, the camera is triggered to take one picture every 5 s. An example of a picture taken with the right settings is given in Figure 4.4.

A prism including a 45 degree-oriented mirror is placed under the notch in order to be able to use DIC on the bottom side with the same camera. That way, the reflection of the bottom of the beam is in the camera axis and the crack opening is visible.



(a) Results with loading cycles

(b) Results with monotonic part of the loading alone

Figure 4.5: Experimental results

4.1.3 Experimental results

Tests are performed on the eight specimens named B_i, $i \in \{1, ..., 8\}$ with the settings presented in the previous section. Figure 4.5 shows the result of the test for the specimen B_5 in terms of the load/displacement curve. The load is measured by the load cell of the test machine, and the displacement is the one measured by the LVDT sensor. Data are post-processed in order to set all initial zeros, especially all the setting cycles are removed. The curve in Figure 4.5(a) is plotted with the raw data, while the curve in Figure 4.5(b) is plotted by only keeping the monotonic part of the test.

Figure 4.6 shows the monotonic curves for all the specimens except specimen B_8. For unexplained reasons the curve for this specimen is not consistent with the other curves. The specimens B_i, $i \leq 4$, are specimens stored in wet conditions (under water). The other specimen B_i, $i > 4$, are stored under drying conditions during 70 days. We can observe that curves show two different behaviors; wet specimens have less resistance and have a softer post-peak behavior than dry beams. Inside a group of beams with the same drying condition, the post-peak behavior seems to be quite similar.

For the DIC measurements, we use the *Corelli* software [84] developed at the LMT lab by the Eikology research team. This software permits to select a reference image and build a finite element mesh (here with linear triangular elements) on the zone of interest. The code then computes the displacement of each node of the mesh between each picture taken during the test and the reference image. Specific details of the DIC method will be discussed in the next sections/chapters. Figure 4.7 shows an example of the DIC displacement field along the horizontal coordinate computed for the test of specimen B_5 in an advanced damaging time. The discontinuity of the displacement field induces the presence of a crack in the specimen. However, the zone of interest of the DIC does not include the translation of the bottom supports. Yet, the displacement is imposed by the test machine by means of the bottom supports. As a result, the vertical displacement field obtained by DIC has to be shifted by the displacement measured by the LVDT sensor in order to have the vertical

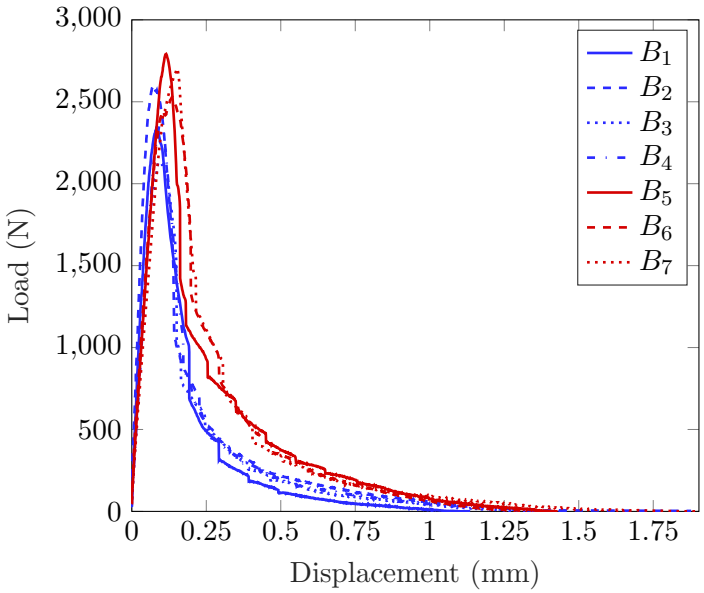


Figure 4.6: Monotonic part of experimental results for all specimens

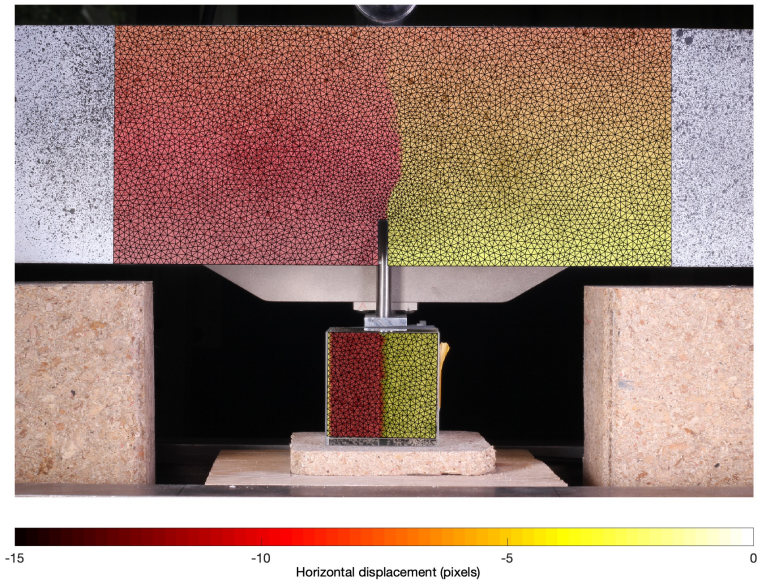


Figure 4.7: Example of a DIC result

displacement of the beam according to its global initial configuration.

4.2 Mathematical and numerical models

The model used to predict the behavior of the structure is adapted from [120] in [147]. We highlight here the main equations of the behavior and the formulation of the corresponding PGD solution.

4.2.1 Damage law

The behavior of the crack closing is not considered here. Thus, the material behavior is supposed to follow the constitutive equation:

$$\sigma = (1 - d)\mathbf{C} : \epsilon \tag{4.1}$$

where σ and ϵ are stress and strain respectively and $\mathbf{C}$ is the hook tensor. The scalar d is the isotropic damage variable that follows the evolution law:

$$d(Y, A_d, Y_0) = 1 - \frac{1}{1 + A_d(Y - Y_0)} \tag{4.2}$$

where Y is the part of energy rate released due to damage defined as:

$$Y = \frac{1}{2} < \epsilon >_+ : \mathbf{C} :< \epsilon >_+ \tag{4.3}$$

$< \epsilon >_+$ denoting the positive part of ϵ. Thus, the damage law depends on two parameters Y_0 and A_d: Y_0 represents the initial threshold, while A_d is a parameter driving the brittleness of the post-peak. In order to see their influence, Figure 4.8 shows the model behavior $\sigma_t = f(\epsilon_t)$ for different values of (Y_0, A_d). The variables σ_t and ϵ_t represent the scalar stress and strain values of one element submitted to traction.

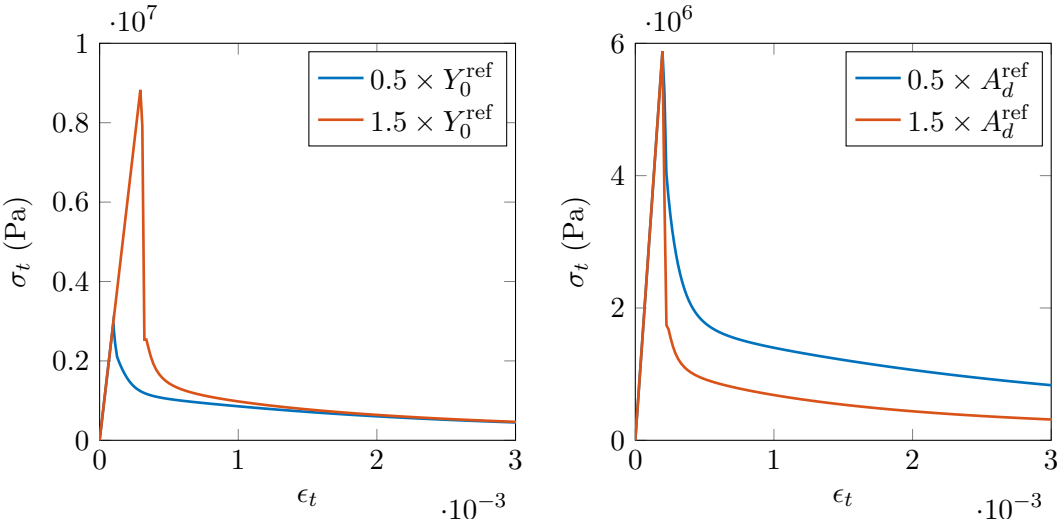
In Figure 4.8(a), the solutions of the model are computed for two values of Y_0 around a reference value Y_0^{ref} and for a reference value A_d^{ref}. Conversely, Figure 4.8(b) shows the model solutions for two values of A_d with a value of Y_0 fixed at Y_0^{ref}.

4.2.2 PGD formulation

In our application, the parameters of interest are the damage parameters Y_0 and A_d. In order to perform dynamical Bayesian inference of those parameters in real-time during the bending test, the model behavior of the structure has to be known with respect to space coordinates $\mathbf{x}$, time t, and parameters (Y_0, A_d). Following the works [147, 146] on the formulation of multi-parametric nonlinear PGD models, the solution in terms of displacement u^{PGD} of the structure can be found by summation of univariate functions:

$$u^{\text{PGD}}(\mathbf{x}, t, Y_0, A_d) = \sum_{n=1}^{m} \Lambda_n(\mathbf{x})\lambda_n(t)\alpha_n^1(Y_0)\alpha_n^2(A_d) \tag{4.4}$$

This solution is computed thanks to the LMT in-house code ROMlab in a black-box way. The 3D mesh geometry represented in Figure 4.9 is given to the input file as well as the boundary conditions. The geometry considered is a half beam as the problem is



(a) Influence of the parameter Y_0 (b) Influence of the parameter A_d

Figure 4.8: Influence of the damage law parameters on the behavior of an element under tension

symmetrical with respect to the $(z = 0)$ plane. Thus, displacement in the z direction is fixed to zero for all nodes in the plane $(z = 0)$. The displacements in both x and y directions are imposed to zero for all the nodes with coordinates $(x = 0.02, y = 0, z)$ corresponding to the left support. For the right support, the nodes with coordinates $(x = 0.82, y = 0, z)$ have their displacements imposed to zero only in the y direction. Finally, the prescribed displacement in the y direction is imposed to the nodes at the top center of the beam. The geometry is discretized with linear cubic elements (Q8).

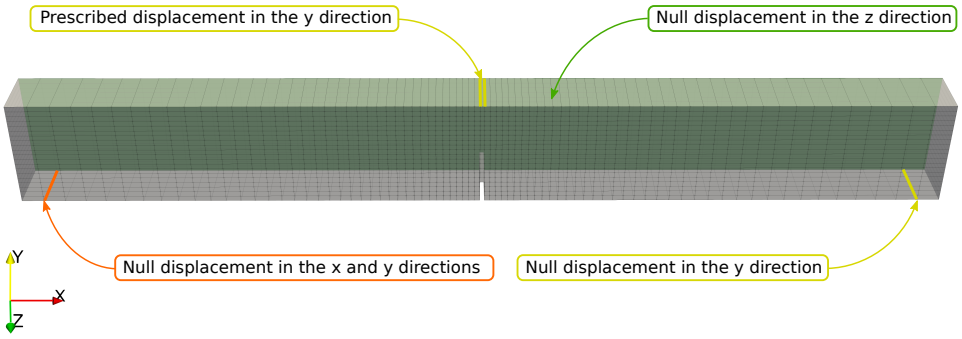


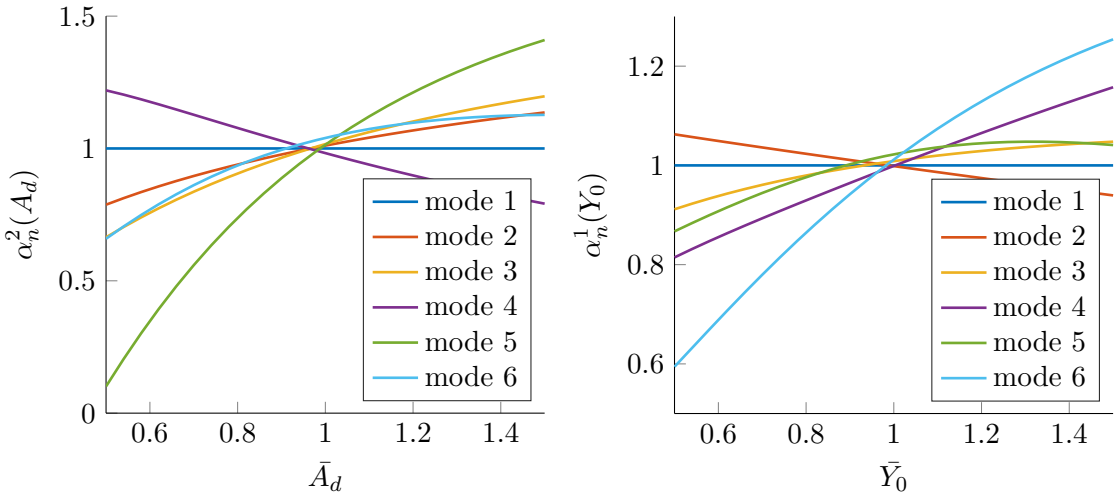
Figure 4.9: Mesh and boundary conditions used for the computation of the PGD model

The reference material properties for this simulation are summarized in Table 4.1. All values come from educated guess thanks to previous experimental campaigns on similar materials in [137]. The imposed displacement u_{imp} increases linearly from 0 to 4×10^{-4}

Poisson ratio	Young modulus (GPa)	Y_0^{ref} (J.m^{-3})	A_d^{ref} (J^{-1}.m^3)
0.23	30	216	2.25×10^{-3}

Table 4.1: Material coefficients used to build the PGD model

m. The loading is discretized in 100 steps. The computation of the PGD solution is based on a LATIN-PGD algorithm [74]. The PGD solution is also computed with a variability of the damage parameters. The variation is chosen to be centered at the reference values $(Y_0^{\text{ref}}, A_d^{\text{ref}})$ with an amplitude of variation of 50%. We define the variables normalized quantities $\bar{Y}_0 = Y_0/Y_0^{\text{ref}}$ and $\bar{A}_d = A_d/A_d^{\text{ref}}$ for comparison purposes. Both intervals of variation of the parameters are discretized with 26 values. The number of iterations of the LATIN-PGD algorithm is fixed to 11. An update stage is used to improve the convergence of the algorithm. After 11 hours of computation on the Fusion cluster with $35 \times$ Intel Xeon CPUs at 2.4 GHz and 450 Go of RAM, the ROMlab code produces 6 PGD modes. Figure 4.10 shows the parametric modes, Figure 4.11 the time modes, and 4.12 the first four spatial modes. All modes except time modes are normalized. That is why the amplitude of time modes rapidly decreases with m as the contribution of each becomes smaller. We can notice that the first mode is a full elastic mode as the corresponding



(a) Modes for parameter Y_0

(b) Modes for parameter A_d

Figure 4.10: PGD modes associated to the damage law parameters

parameter modes are null. Further discussions about the shape of the modes will be given in Chapter 6.

4.3 Dynamical inference of the damage law parameters

4.3.1 Formulation of the inference problem

Here, the parameters to be updated are the initial damage threshold Y_0 and the brittleness coefficient A_d which drives the post-peak behavior. Although the experimental tests are all performed at the same time without real-time post-processing due to practical constraints, the purpose here is to sequentially update the knowledge on the parameters Y_0 and A_d from *in-situ* DIC measurements within the Bayesian framework. One could think of also using the available load measurement. However, only the displacement field is formulated in the PGD format. Building a meta-model of the reaction forces depending of the damage law parameters can be conceivable but we choose here to restrict the identification problem

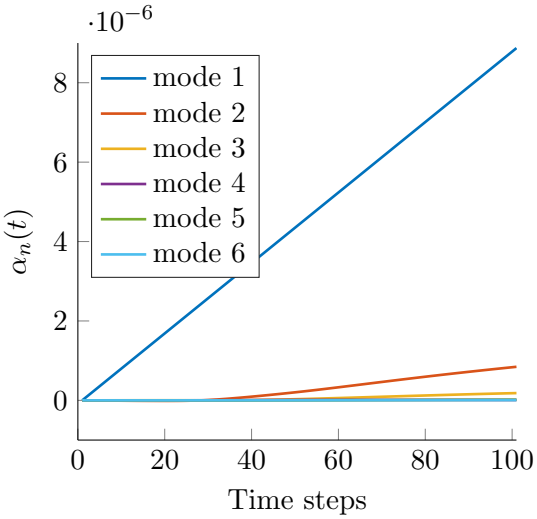


Figure 4.11: PGD modes for time dependency

to displacement measurement, which seems to be the most likely available information in the monitoring of real structures. This numerical example was also treated in [128].

Then, at each time step t_i, we consider the knowledge of the measurements $\mathbf{u}^{\text{obs},i}$ which represent the displacements of the degrees of freedom of the DIC mesh. In this context, the posterior density function at time t_i reads:

$$\pi(Y_0, A_d|\mathbf{u}^{\text{obs},1:i}) \propto \prod_{j=1}^{i} \pi(\mathbf{u}^{\text{obs},j}|Y_0, A_d).\pi(Y_0, A_d) \tag{4.5}$$

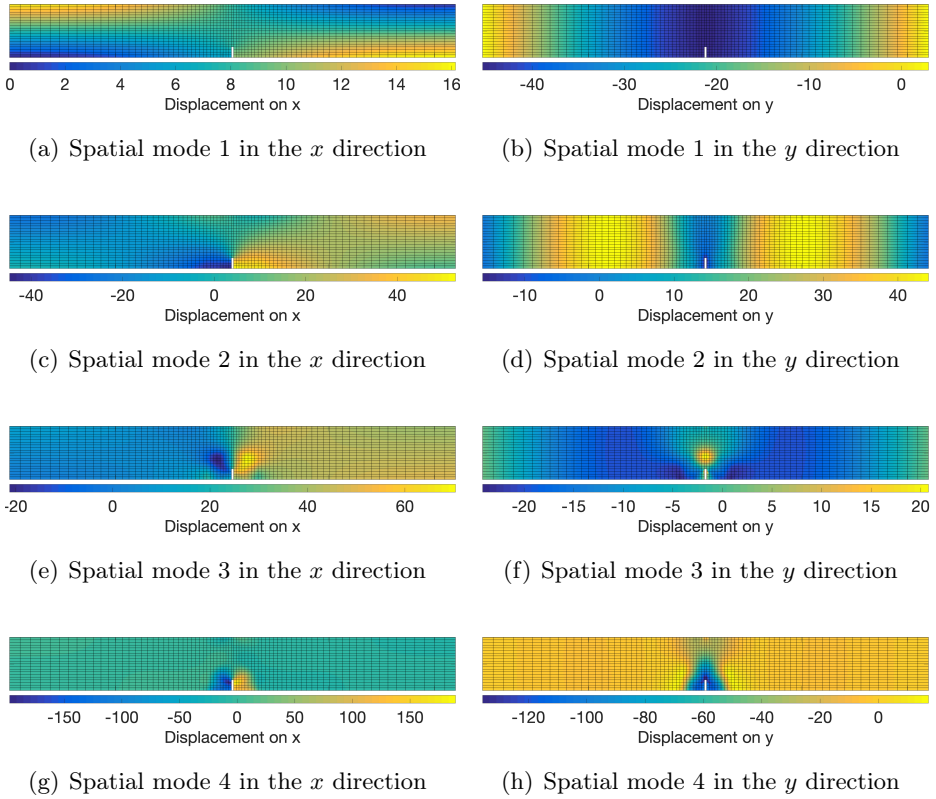
4.3.2 Application with simulated measurements

In this section, the problem is solved with simulated measurements from the PGD model. We choose 30 random nodes on the central part of the beam as measurements points (see Figure 4.13). Then the measurements used are the displacements of those nodes given by the PGD model with parameters set to $Y_0 = 1 \times Y_0^{\text{ref}}$ and $A_d = 1 \times A_d^{\text{ref}}$. A Gaussian white noise with a standard deviation of 10^{-6} m is added to those displacements in order to simulate the DIC noise. Further discussions about actual DIC measurement noise will be addressed in Chapter 6. The dynamical Bayesian inference with transport maps is performed with Algorithm 5 presented in Chapter 3. Then, the likelihood function $\pi(\mathbf{u}^{\text{obs},j}|Y_0, A_d)$ in the formulation of the posterior density in Equation (4.5) reads:

$$\pi(\mathbf{u}^{\text{obs},j}|Y_0, A_d) = \prod_{k=1}^{N_{\text{meas}}} \pi_{\text{err}}(u_k^{\text{obs},j} - u^{\text{PGD}}(\mathbf{x}_k, t_j, Y_0, A_d)) \tag{4.6}$$

The pdf π_{err} is Gaussian with zero mean and standard deviation of 10^{-6} , and the $\mathbf{x}_k$, $k \in \{1, ..., N_{\text{meas}}\}$, are the coordinates of the measurements nodes (here $N_{\text{meas}} = 30$). The prior density $\pi(\bar{Y_0}, \bar{A_d})$ is chosen as a Gaussian distribution with mean $(0.9, 1.1)$ and covariance matrix:

$$\Sigma_{\text{prior}} = \begin{pmatrix} 0.2 & 0 \\ 0 & 0.2 \end{pmatrix} \tag{4.7}$$



(a) Spatial mode 1 in the x direction

(b) Spatial mode 1 in the y direction

(c) Spatial mode 2 in the x direction

(d) Spatial mode 2 in the y direction

(e) Spatial mode 3 in the x direction

(f) Spatial mode 3 in the y direction

(g) Spatial mode 4 in the x direction

(h) Spatial mode 4 in the y direction

Figure 4.12: First four PGD spatial modes

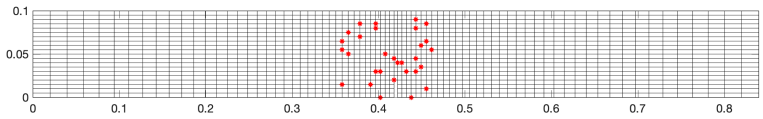
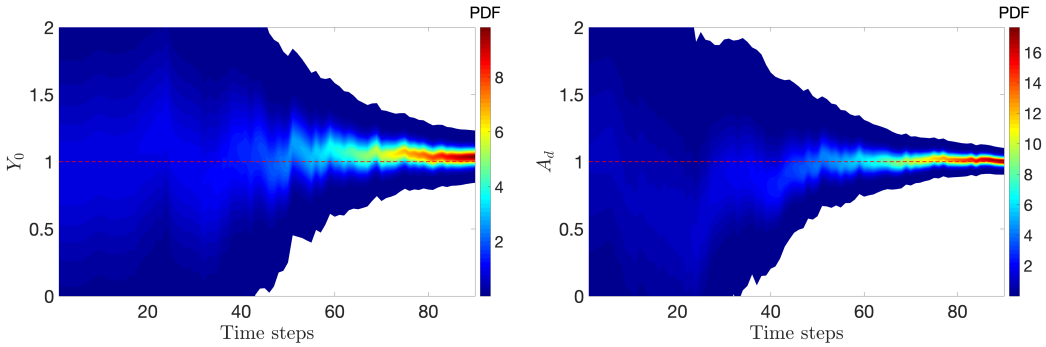


Figure 4.13: Position of the nodes considered as measurement points

The results in terms of marginals are shown in Figure 4.14. During the first time steps, the posterior densities are very wide due to the fact that the structure is in the elastic domain so that the displacement gives poor information on the damage parameters. However, at the final time step, maxima of the marginals give an estimation of the parameters of $(\bar{Y}_0, \bar{A}_d) = (1.04, 1.00)$ which is very close to the reference value $(1, 1)$ despite the relative wrong first estimation $(0.9, 1.1)$ given by the prior. We can also see that the estimation of the parameter $\bar{Y}_0$ is slower than the estimation of the parameter $\bar{A}_d$. At the final time step, the standard deviation of the marginal on the parameter $\bar{Y}_0$ is 0.041 while the standard deviation of the marginal on the parameter $\bar{A}_d$ is equal to 0.022. The computation costs of the inference algorithm are shown in Figure 4.15. The dashed line shows the time required to compute each map using only the explicit formulation of the functional in the solution of the minimization problem (order derivative equal to 0). The solid line represents the time required to compute the maps by using an explicit formulation of both functional and its gradient (order derivative equal to 1) computed thanks to the derivatives of the PGD modes. As shown in the examples of Chapter 3, the use of the functional gradient strongly speed up the computation of the transport maps especially when the maps order



(a) Marginals on $\bar{Y_0}$ (b) Marginals on $\bar{A_d}$

Figure 4.14: Successive posterior marginals obtained after each time step of measurement assimilation

becomes large. Indeed, the bar chart represents the final order of the maps computed in order to achieve the given tolerance of the variance diagnostic (10^{-3}). We can see that the highest computation costs correspond to the highest order of map and the highest speed-up obtained by using the transport map algorithm with 1st-order derivatives.

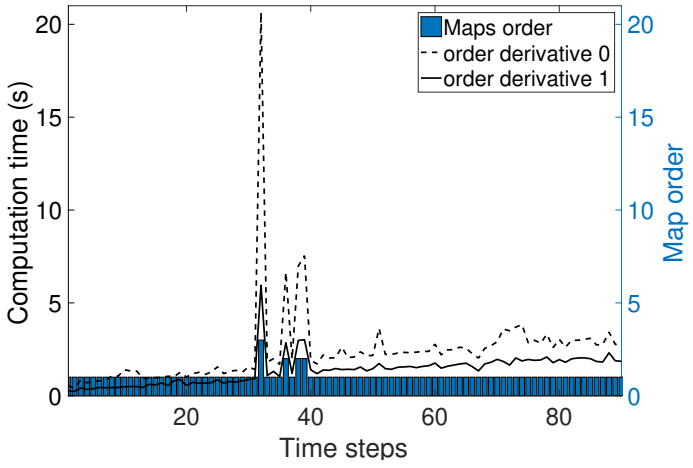


Figure 4.15: Computation cost of each map for each time step. The dashed line represents the computation time using order 0 derivatives and the continuous line represents the computation time using order 1 derivatives. The bar chart represents the order of each computed transport map

We can also explain the change of maps order for time steps 32, 36, 38 and 39. For the first time step, the structure is in the elastic domains so that the measured displacements do not provide any knowledge on the parameters. Hence, the posterior densities for the first time steps are very close to the prior density which is Gaussian. Then, a first order map is sufficient to represent the map between the reference standard normal density and the approximately Gaussian posterior density. Between time steps 30 and 40, the change of the posterior density is the most brutal as the structure follows a nonlinear damage behavior. Then, higher order maps are necessary to represent the coupling between two consecutive posterior densities. After this critical domain, as the time steps are quite small,

the evolution of the posterior densities is smooth and consequently easily representable by 1st-order maps.

4.4 On-the-fly estimation of the crack length

As seen in Chapter 3, once the posterior density on parameters is known at a given time step, uncertainties can be propagated trough the PGD model to predict the behavior of the system for the other time steps. Here, the PGD model gives the displacement field of the structure. The idea of this part is to post-process the displacement field obtained after uncertainty propagation in order to derive an estimation of the cracking state of the beam and predict the potential collapse of the specimen.

4.4.1 Simple meta-model of crack opening

In order to do the link between the PGD damage model formulated in terms of displacements, and the collapse of the structure, we propose here to build a kinematic bridge between the PGD model and the elastic solution of the beam under the same boundary conditions and including a crack. In the following, we assume that the crack is initiated in the right corner of the notch and that the crack propagates straightly in the vertical direction. Then, different meshes are created corresponding to different discretized values of crack lengths. The elastic solutions on those meshes are computed with the same boundary conditions as the reference problem and a unit imposed displacement. The interval of crack lengths varies from 0 to 80 mm. This space is discretized in 80 values so that 80 elastic solutions are computed. Figure 4.16 shows the elastic solutions for four particular crack lengths.

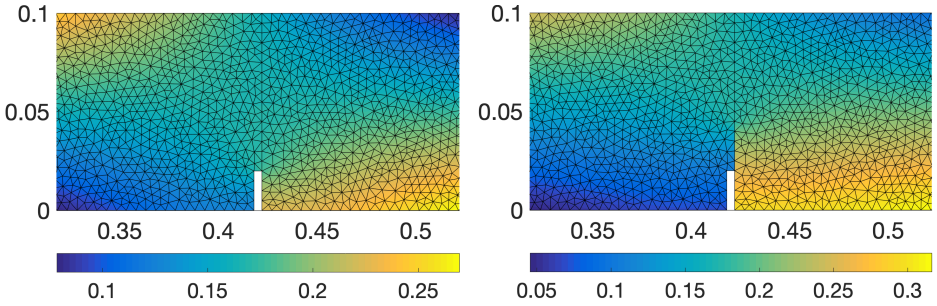
To efficiently estimate the crack length in the beam in real-time after the inference of the parameters from successive *in-situ* measurements, a meta-model is computed from those 80 snapshots computed in the *offline* phase. Therefore, all the 80 elastic solutions are stored in a matrix $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_{80}\}$ with $\mathbf{y}_j \in \mathbb{R}^{n_{\text{dof}}}$ the displacement vectors computed on the finite element mesh composed of n_{dof} degrees of freedom. Then, the Singular Value Decomposition (SVD) of the matrix $\mathbf{Y}$ is computed such that:

$$\mathbf{Y} = \mathbf{U}\Sigma\mathbf{V}^T \tag{4.8}$$

where $\mathbf{U} = \{U_1, ..., U_{n_{\text{dof}}}\}$ is a $n_{\text{dof}} \times n_{\text{dof}}$ matrix, $\mathbf{V} = \{V_1, ..., V_{80}\}$ is a 80×80 matrix, and Σ the $n_{\text{dof}} \times 80$ diagonal matrix containing the singular values σ_j, $j \in \{1, ..., 80\}$. By truncating the SVD decomposition to N_{SVD} modes, the displacement $u_{\text{SVD}}(\mathbf{x}, l)$ of the structure submitted to a unitary displacement can be approximated as a function of the crack length l as

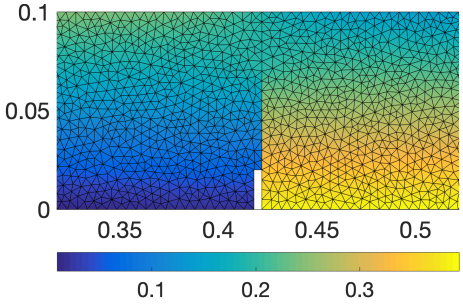
$$u_{\text{SVD}}(\mathbf{x}, l) = \sum_{k=1}^{N_{\text{SVD}}} \sigma_k U_k(\mathbf{x}) V_k(l) \tag{4.9}$$

The functions $U_k(\mathbf{x})$ and $V_k(l)$ are built from the linear interpolation of the corresponding vectors U_k and V_k verifying $U_k(x_i) = U_{ki}$ and $V_k(l_j) = V_{kj}$. Figure 4.17 shows the first six modes $V_k(l)$ and Figure 4.18 the corresponding horizontal displacement modes $U_k(\mathbf{x})$.
The first two spatial modes represent the displacement with the full opening of the crack and the other modes can be seen as localized "stitching" patches along the crack.

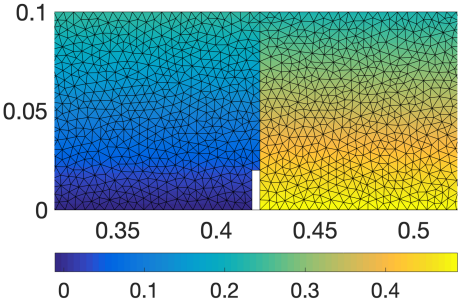


(a) Elastic solution for $l = 0$

(b) Elastic solution for $l = 0.25$



(c) Elastic solution for $l = 0.50$



(d) Elastic solution for $l = 0.75$

Figure 4.16: Elastic solution in terms of displacement in the x direction for different crack openings

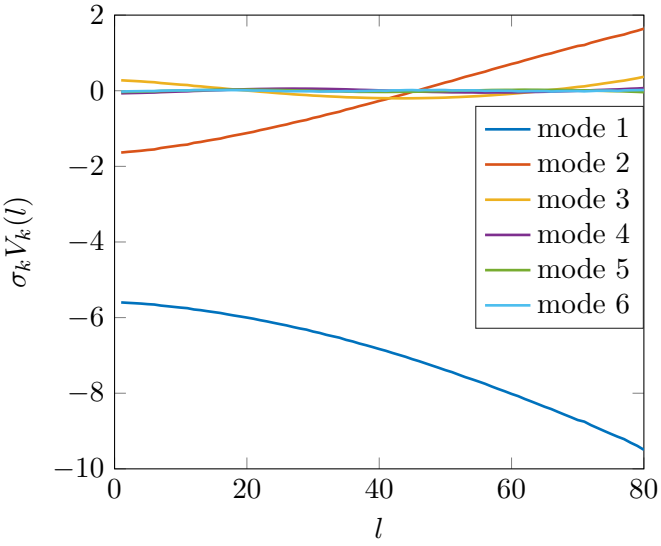
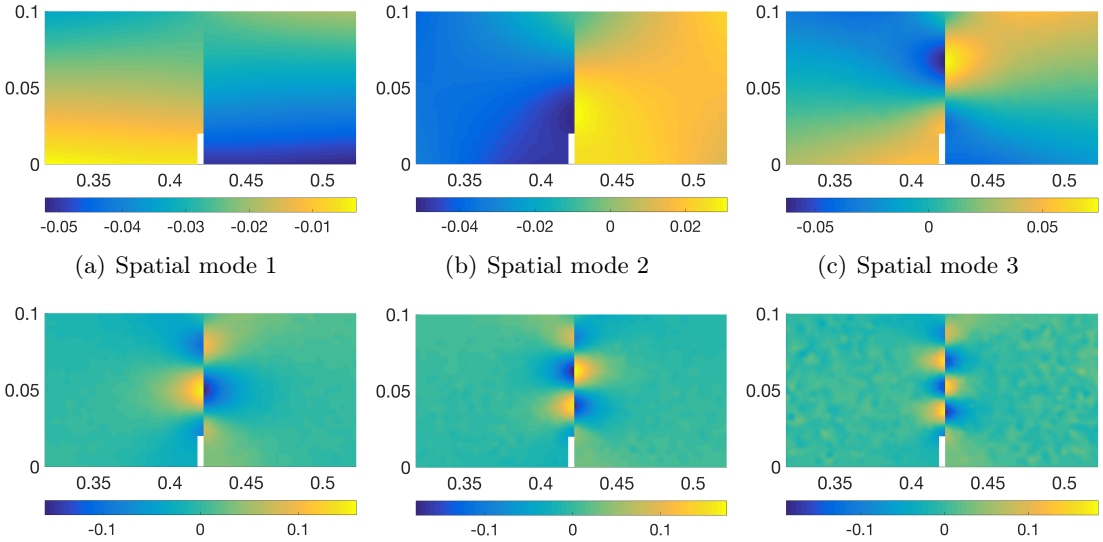


Figure 4.17: Spatial SVD modes associated to the parameter l



(a) Spatial mode 1 (b) Spatial mode 2 (c) Spatial mode 3

(d) Spatial mode 4 (e) Spatial mode 5 (f) Spatial mode 6

Figure 4.18: Spatial SVD modes in terms of displacements in the x direction

4.4.2 Estimation of the crack length with uncertainty propagation

Once the damage model is updated, several quantities of interest can be defined thanks to the global formulation of the PGD model. Here the quantities of interest will be the crack length at the final time step t_f of the simulation. An associated application can be the monitoring of the structure at this time step in order to assess whether the structure will collapse or not. Of course, this study could be done for another specific time step or for all time steps to study the behavior of the crack propagation along time.

From the knowledge of the displacement field $u^{\text{PGD}}(\mathbf{x}, t_f, Y_0, A_d)$, the crack length parameter l can be identified by comparison with the displacement field coming from the SVD model $u^{\text{SVD}}(l)$. As the displacement of the damage PGD model depends on the uncertainties on the parameters Y_0 and A_d, the Bayesian inference is well suited to perform the stochastic identification of the parameter l. Thus, the random displacement field u^{PGD} needs to be compared with the SVD model u^{SVD}. To be consistent with the Bayesian formulation of Chapter 1 with additive noise measurement, we can write the equality:

$$\mathbf{0} = -u^{\text{SVD}}(l) + u^{\text{PGD}} \tag{4.10}$$

with u^{PGD} seen as the random variable of "noise" with density $\pi_{\text{u}}(u^{\text{PGD}})$ and $\mathbf{0}$ the realization of the "measurement" random variable. The likelihood function then reads:

$$\pi(l) = \pi_{\text{u}}(u^{\text{SVD}}(l)) \tag{4.11}$$

The pdf $\pi_{\text{u}}(u^{\text{PGD}})$ is obtained by the propagation of the uncertainties on the parameters Y_0 and A_d through the PGD model evaluated at time t_f and at the considered space coordinates. Here the purpose is to perform this second inference in real-time so the uncertainty propagation has to be as fast as possible.

If we consider one measurement point, for example the point with space coordinates $\mathbf{x}_1 = (x_1, y_1) = (0.4, 0)$, the propagation of the uncertainties through the PGD model gives

a random variable u_1. Samples u_1^k of this random variable can be computed as:

$$u_1^k = u^{\text{PGD}}(\mathbf{x}_1, t_f, Y_0^k, A_d^k) \tag{4.12}$$

with (Y_0^k, A_d^k) samples drawn according to the posterior density $\pi(Y_0, A_d|\mathbf{u}^{\text{obs},1:j})$ resulting from the Bayesian inference on damage parameters at time step t_j.

However, in order to know explicitly the likelihood function (4.11), the pdf π_{u_1} should also be known explicitly. To compute the pdf π_{u_1}, the fastest way seems to approximate it with the two first moments and the maximum entropy principle (i.e. approximation by a Gaussian). Indeed, after one step t_j of data assimilation, the solution of the first inference problem gives a transport map M_j between the standard normal distribution ν_ρ and the posterior distribution with density $\pi(Y_0, A_d|\mathbf{u}^{\text{obs},1:j})$. Then, from the Gaussian-Hermite quadrature $(w_k, \mathbf{p}_k)$, $k \in \{1, ..., N_q\}$, and for the density ρ, the n first moments of the random variable u_1 can be computed as:

$$m_{u_1}^n = \mathbb{E}_{\pi_{u_1}}\left[u_1^n\right] = \sum_{k=1}^{N_q} w_k \left(u^{\text{PGD}}(\mathbf{x}_1, t_f, M_t(\mathbf{p}_k))\right)^n \tag{4.13}$$

Then we can deduce that u_1 is approximately distributed according to the normal distribution $\mathcal{N}(\mu_{u_1}, \sigma_{u_1})$ with $\mu_{u_1} = m_{u_1}^1$ and $\sigma_{u_1}^2 = m_{u_1}^2 - \left(m_{u_1}^1\right)^2$. This method is effective as it only requires the evaluation of the PGD model N_q times (typically 100 times for a level 10 Gauss quadrature). However, this method can be a coarse approximation of the pdf and we want to compare it with a more accurate method based on finding a transport map S that pushes forward the reference distribution ν_ρ to the unknown distribution $\nu_{\pi_{u_1}}$ of the random variable u_1. This topic called inverse transport is presented in [115] with the application to build non-Gaussian proposal distributions for MCMC algorithms. The main idea is the same as the one presented in Chapter 3, and the map S is found by solving the problem:

$$S = \underset{M}{\text{argmin}}\ \mathcal{D}_{KL}\left(\nu_{\pi_{u_1}} || M_\sharp \nu_\rho\right) \tag{4.14}$$

$$= \underset{M}{\text{argmin}}\ \mathbb{E}_{\pi_{u_1}}\left[\log\left(\frac{\pi_{u_1}}{M_\sharp \rho}\right)\right] \tag{4.15}$$

$$= \underset{M}{\text{argmin}}\ \mathbb{E}_{\pi_{u_1}}\left[-\log\left(M_\sharp \rho\right)\right] \tag{4.16}$$

Thus, the function to minimize does not involve the pdf of π_{u_1} and the expectation with respect π_{u_1} can be computed by Monte-Carlo with the available samples (Equation (4.12)). Furthermore, this expectation can also be computed using a Gauss quadrature. Knowing the Gauss quadrature $(w_k, M_j(\mathbf{p}_k))$, $k \in \{1, ..., N_q\}$ for $\pi(Y_0, A_d|\mathbf{u}^{\text{obs},1:j})$ the quadrature points for π_{u_1} are $(w_k, u^{\text{PGD}}(\mathbf{x}_1, t_f, M_t(\mathbf{p}_k)))$. Consequently, the computation of the transport map S can be done in a cost effective way. Once the map S is computed, the pdf of the random variable u_1 reads by definition:

$$\pi_{u_1}(u_1) = \left[\rho \circ S^{-1}\right](u_1) |\det \nabla S^{-1}(u_1)| \tag{4.17}$$

In Table 4.2, the computation times for each uncertainty propagation method is presented: first the Gaussian method is used; second the transport map method for density

estimation (with a 3d-order map); and third, as a reference method, a kernel density estimation computed with a Gaussian kernel and 20,000 Monte-Carlo samples u_1^k. All computations times also take into account the evaluation of 100 points by the estimated pdf. Figure 4.19 shows the corresponding pdfs. The Gaussian estimation is in this case quite far from the reference pdf computed with the kernel density estimation. However the approximation by Transport Map sampling is quite good, leading to a speed-up of 25.7 by comparison with the kernel density estimation and trading a speed-up of 3.8 for accuracy by comparison with the Gaussian approximation.

Gaussian estimation	Kernel density estimation	Transport map
0.014 s	1.39 s	0.054 s

Table 4.2: Computation times for different density estimation techniques

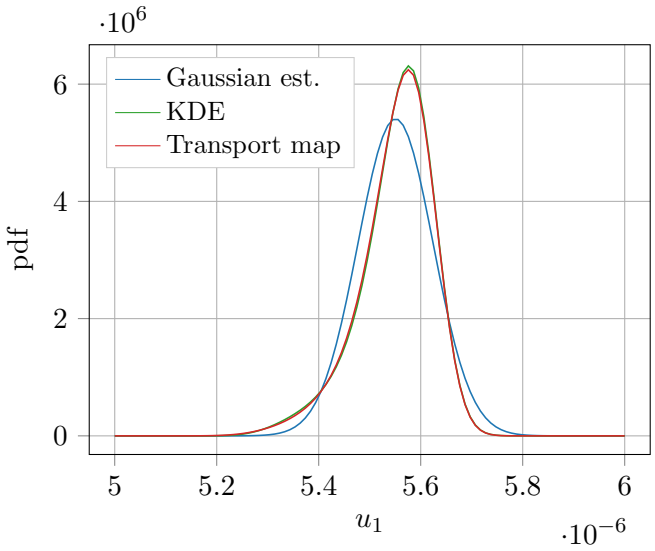


Figure 4.19: Comparison between the different density estimation techniques

Finally, the Bayesian inference problem for the identification of the crack length reads:

$$\pi(l) = \pi_{u_1}(u^{\mathrm{SVD}}(l)).\pi_{\mathrm{pr}}(l) \tag{4.18}$$

The prior density π_{pr} is chosen to be Gaussian with a mean of 0.04 and standard deviation of 0.013. The inference of the crack length l is done at each time step t_j of the simulation, after each determination of the posterior density $\pi(Y_0, A_d|\mathbf{u}^{\mathrm{obs},1:j})$. The results of all successive posterior densities $\pi(l)$ are represented in Figure 4.20. Thanks to the formulation of the second Bayesian inference problem, the uncertainties on the parameters A_d and Y_0 are propagated in the identification of the length of an equivalent straight crack. During the first time steps, the estimation of the crack length is very coarse with a large variance then it becomes better and converges at the final time step to the mean estimation of 0.0372 and a maximum a posteriori estimation of 0.0376.

Finally, Figure 4.21 shows the cumulative computation time required to perform the successive inferences for each time step. We can notice that all iterations can be done in less than 5 s which is the duration between two successive picture acquisitions. Therefore,

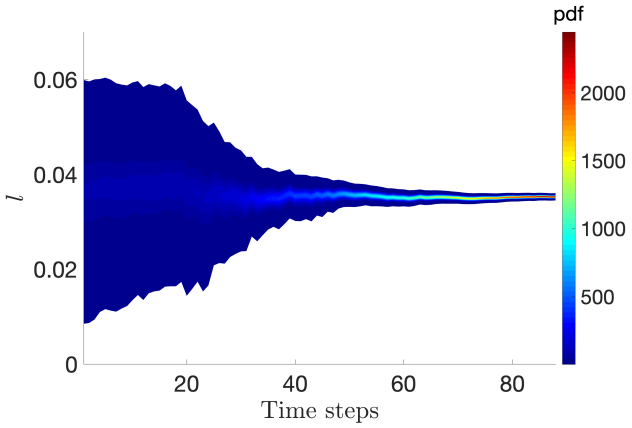


Figure 4.20: Successive posterior densities on parameter l

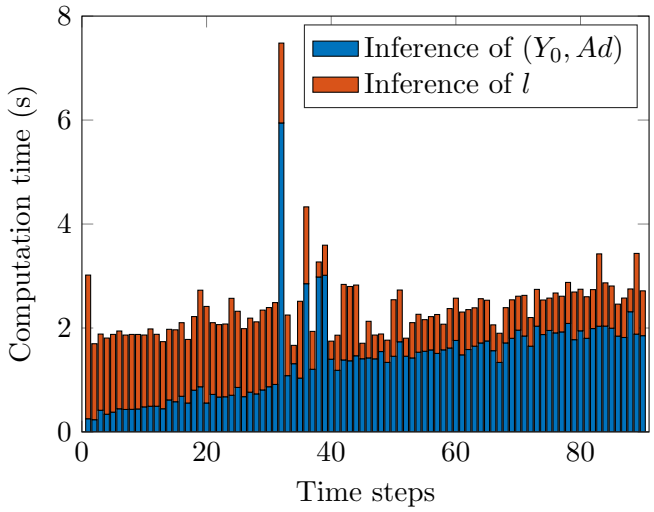


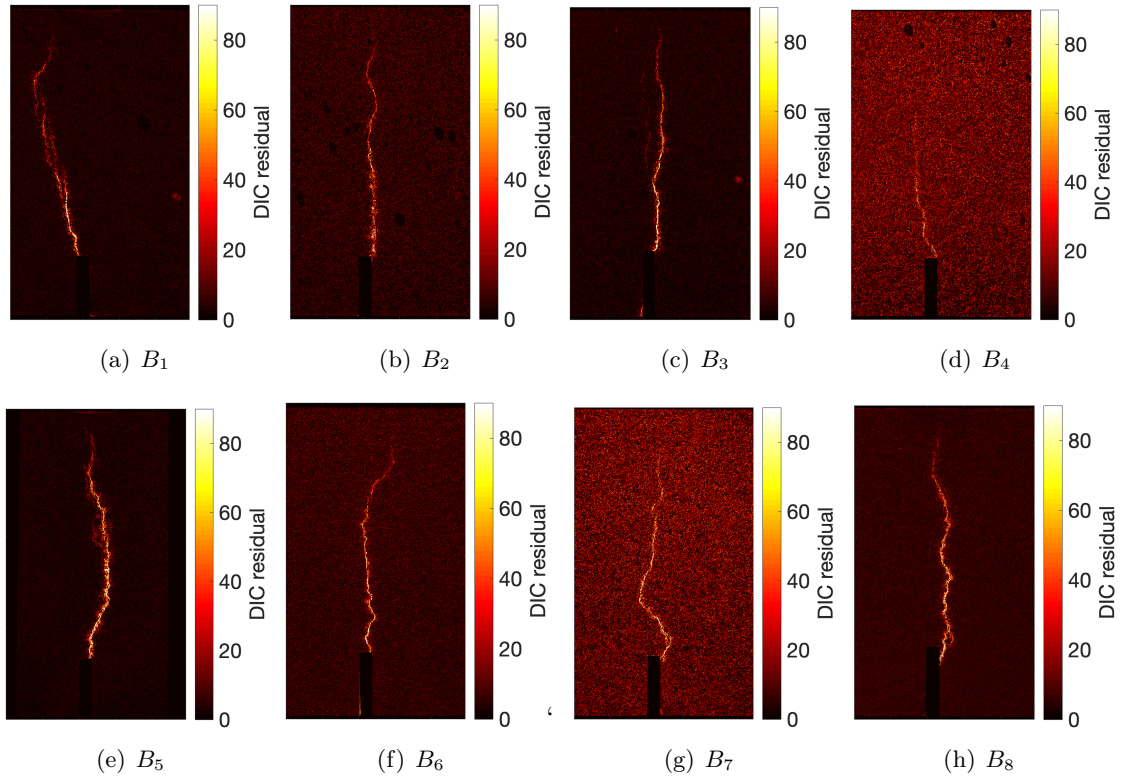
Figure 4.21: Total computation cost for each time step. The red part represents the computation time relative to the estimation of the parameter l, and the blue part to the estimation of the damage law parameters (Y_0, A_d)

in this context this approach could be considered in real-time. However, in this example the computation time required to perform live-DIC is not discussed. This time has to be added to the total computation time. Although we have no reason to believe that this computation time will not be particularly high (see [85] for an online use of DIC), in the case where the computation time for an assimilation time step is higher than the acquisition time, measurements can be stored in order to catch up the delay by simultaneously assimilating two or more sets of measurements.

4.4.3 Other model of crack propagation

The model of a straight planar crack propagation initiated from a specific region is quite restrictive. Indeed, Figure 4.22 shows the results of the computation of the DIC residuals for all the beams of this experimental campaign at the final time step. As the DIC meshes are continuous, they do not take into account the crack opening kinematics. Then, around

the crack the DIC algorithm cannot converge properly and the residual is quite high. Therefore, the crack path in the plane of the DIC can be estimated by looking for the highest values of the residual.



(a) B_1 (b) B_2 (c) B_3 (d) B_4

(e) B_5 (f) B_6 (g) B_7 (h) B_8

Figure 4.22: DIC residual at the final step of the experiment of all specimens of the campaign

We can see that the straight hypothesis of the crack is quite valid for specimens B_2 and B_3 but questionable for the other specimens. A first improvement can be made by enriching the characterization of the crack by adding more kinematic parameters. For example, in addition to the crack length, an angle of propagation could be considered. Then, the meta-model could be computed by HO-SVD with all the parameters and spatial coordinates. Another perspective would be to consider a database of DIC residual results and do manifold learning on this database in order to extract the relevant kinematic parameters of any crack geometry. A preliminary work has been done by using the same methodology as in [52] with disappointing results probably due to the small size of the data base (only 8 specimens).

4.4.4 Application with real measurements

So far, the inference problems are considered with measurements directly simulated from the PGD model. The PGD model is computed with nominal damage parameter values with the same boundary conditions as in the real test-case. However, the PGD model presented in this chapter is not able to represent the real displacements observed in the tests. Indeed, we can see that in the application of the crack length estimation, at the final time step, the crack is barely at mid-span of the beam height while in the real test-case the

beam is collapsed. This discrepancy can be explained by the fact that the nominal values of the damage parameters give less damage than in the reality. However, when modifying the damage law to have more damage in the beam, the PGD code seems to no longer converge. This can be due to the strong localization of the damage in this example. To circumvent this problem, both spatial and time discretizations can be refined. However, in this application, it leads to intractable computations due to high RAM capacity which is required.

4.5 Conclusions

In the first part of this chapter, the experimental campaigns carried out in this thesis were presented. The experimental tests aim to study the cracking behavior of a concrete beam. To that end, interesting experimental sets have been used with *in-situ* measurements performed with DIC. In the second part of this chapter, the Bayesian inference with PGD model reduction and Transport Map sampling was applied. The purpose is to update the knowledge on the damage law parameters in order to predict the crack propagation in the concrete beams. In order to do so, the PGD model output was compared with an elastic meta-model with a crack length dependency. As the model output is considered as a random field due to the uncertainties on the damage law parameters, the link with the meta-model was performed with the formulation of a second inference problem. This second inference problem was built with a non-Gaussian likelihood function obtained by fast uncertainty propagation with transport maps. Hence, the approximated crack length can be estimated in real-time at different time steps thanks to the assimilation of the displacement fields.

However, in this work the assimilation of measurements coming from the experimental test has not been possible due to a convergence issue of the PGD code. An immediate prospect will be to solve this issue in order to fully use experimental tests. A short term prospect will be to improve the convergence of the code or the storage of the operators required to compute the PGD solution. Another prospect will be to enrich the model by a model correction in order to compensate all the experimental uncertainties that can be encountered. This latter prospect is addressed in the next chapter.

Chapter 5

Data assimilation with on-the-fly correction of model bias

In this chapter, the purpose is to study the influence a model bias on the solution of a sequential Bayesian inference problem. Moreover, a method to dynamically identify and correct this model bias is proposed in order to improve the solution of the Bayesian inference problem. The proposed approach is applied to two numerical examples.

5.1 Computation of the model bias

5.1.1 Context and definition of the model bias

In the classical Bayesian inversion formulation, the forward model that is evaluated in the likelihood function is supposed to be exact. This model is supposed to be able to generate the observations without bias. However, the inaccuracy of the model can have strong effects on the accuracy of the posterior density [71]. Then, introducing a modeling error term in the Bayesian formulation of inverse problem can improve its solution [134]. For example, in the recent work [20], the model error between low and high fidelity models known *a priori* is introduced to improve the Bayesian formulation computed with importance sampling.

However, this model error can be difficult to know *a priori* (when a high-fidelity model is unknown) and an appealing idea is to learn the model correction term directly from the data. This "hybrid" approach consists of performing data assimilation in order to estimate the best values of the model parameters that fit the physics, then compute the data-based model correction as the residual discrepancy between the model and the observed physics .

In the litterature, integration of a model correction can be found in the developments of the PBDW method where a reduced order model is enriched with orthogonal modes which represent the discrepancy between the model prediction and the data [94]. In another recent work [27], a parametric model $\mathcal{M}(\mathbf{p})$ is updated in a deterministic way thanks to the data $\mathbf{d}^{\text{obs}}$ to find the best parameters fit $\mathbf{p}^{\text{est}}$, then the model correction B is computed as the estimation residual:

$$B(\mathbf{x}^{\text{obs}}, t_i) = \mathbf{d}^{\text{obs}} - \mathcal{M}(\mathbf{x}^{\text{obs}}, t_i, \mathbf{p}^{\text{est}}) \tag{5.1}$$

where $\mathbf{x}^{\text{obs}}$ represents the spatial coordinates of the measurements points and t_i the assimilation time steps. In Bayesian inference, the result of the parameter estimation at time t_i is the posterior pdf $\pi(\mathbf{p}|\mathbf{d}_1^{\text{obs}}, ..., \mathbf{d}_i^{\text{obs}})$. Thus, the model correction can be computed as presented in (5.1) by computing the estimation $\mathbf{p}^{\text{est}}$ with post-processing of the posterior pdf (i.e. mean or maximum *a posteriori*).

However, we consider here the estimation residual as a random variable due to the uncertainties on the parameters $\mathbf{p}$ and the measurement noise $\mathbf{e}$:

$$B(\mathbf{x}^{\text{obs}}, t_i) = \mathbf{d}_i^{\text{obs}} - \mathbf{e} - \mathcal{M}(\mathbf{x}^{\text{obs}}, t_i, \mathbf{p}) \tag{5.2}$$

Then, the pdf of the random variable B is computed by uncertainty propagation and sum of independent random variables. This estimation residual is computed after data assimilation at time t_i; it represents the discrepancy between the data and the model. In the case of sequential inference, this discrepancy can be taken into account in the data assimilation at time t_{i+1} by writing:

$$\mathbf{d}_{i+1}^{\text{obs}} = \mathcal{M}(\mathbf{x}^{\text{obs}}, \mathbf{p}, t_{i+1}) + B(\mathbf{x}^{\text{obs}}, t_{i+1}) \tag{5.3}$$

The model bias $B(\mathbf{x}^{\text{obs}}, t_{i+1})$ is taken as the extrapolation of the estimation residual computed at time t_i. Thus, the likelihood function at time t_{i+1} reads:

$$\pi(\mathbf{d}_{i+1}^{\text{obs}}|\mathbf{p}) = \pi_B(\mathbf{d}_{i+1}^{\text{obs}} - \mathcal{M}(\mathbf{x}^{\text{obs}}, \mathbf{p}, t_{i+1})) \tag{5.4}$$

where π_B is the pdf of the random variable of the model bias B.

5.1.2 Extrapolation

There are several methods to extrapolate the model bias. The first idea is to perform a constant extrapolation and keep the pdf π_B equal to the pdf of the random variable of the residual computed at time t_{i+1}. Another idea is to linearly extrapolate mean and standard deviation of the random variable B and approximate the pdf π_B by a Gaussian. In any case, this extrapolation can be extremely wrong as it relies only on the data without considering physics. However, in the targeted application the model is supposed to represent the studied physics quite accurately so the bias remains small or at least evolves slowly. The more the model is accurate to represent the physics, the less the contribution of the data-based bias and the error committed in the extrapolation will be important. In the case where lots of measurements are assimilated at once for a given time step (for example in DIC context), the model bias is a random vector with size the number of the discretized displacement field that is measured. Then, at each assimilation time step, we can compute the mean vector and the standard deviation vector of the model bias. The extrapolation can be computed on those deterministic quantities, coordinate by coordinate but the chosen approach here is to perform extrapolation on the SVD modes of the mean and standard deviation. The purpose here is to extrapolate the displacement in a global manner in order to bring more physics and to filter the noise. To compute these extrapolations, at time step t_i, we build the matrices of mean and standard deviation:

$$B_{\text{mean}} = \left[\text{mean}\left(B(\mathbf{x}^{\text{obs}}, t_1)\right), ..., \text{mean}\left(B(\mathbf{x}^{\text{obs}}, t_i)\right)\right] \tag{5.5}$$

$$B_{\text{std}} = \left[\text{std}\left(B(\mathbf{x}^{\text{obs}}, t_1)\right), ..., \text{std}\left(B(\mathbf{x}^{\text{obs}}, t_i)\right)\right] \tag{5.6}$$

with mean(.) and std(.) the functions that compute the mean and the standard deviation of a random variable with a quadrature rule. The SVD of those matrices are computed such that:

$$B_{\text{mean}} = U_{\text{mean}} S_{\text{mean}} V_{\text{mean}}^T \tag{5.7}$$

$$B_{\text{std}} = U_{\text{std}} S_{\text{std}} V_{\text{std}}^T \tag{5.8}$$

The matrices V_{mean} and V_{std} represent the time dependency of the model bias. Thus, the extrapolated mean and standard deviation vectors for time t_{i+1} are computed with the linear extrapolation $\hat{V}_{\text{mean}}$ and $\hat{V}_{\text{std}}$ of the matrices V_{mean} and V_{std}. The linear extrapolation of the matrices is performed column by column such that an additional line is added to the matrices V_{mean} and V_{std}.

Finally, the pdf of the bias at time t_{i+1} is the multidimensional Gaussian distribution with mean $\hat{B}_{\text{mean}}^{i+1}$ computed with the extrapolated matrix $\hat{V}_{\text{mean}}$ and a diagonal covariance matrix filled with the values of the standard deviation vector $\hat{B}_{\text{std}}^{i+1}$ computed with the extrapolated matrix $\hat{V}_{\text{std}}$. Consequently, truncation can be applied and only the first SVD modes may be used to build the mean and standard deviation vectors of the model bias.

In this approach, the computation and storage costs of the model bias increase during the assimilation with the number of assimilated time steps. To circumvent this problem, the idea is to use the so-called Sequential Karhunen-Loeve (SKL) algorithm [125]. This algorithm allows to compute a SVD of a matrix piece-wisely. This is particularly useful when dealing with a large set of data or, in our case, when the data are acquired sequentially. The purpose of the algorithm is to compute the SVD of $[C\ D]$, which is the horizontal concatenation of the matrices C and D, knowing the SVD of C. The main idea of the algorithm is to do a QR decomposition in order to retrieve the span of the SVD modes already computed. Then the SVD is computed on a matrix of size $N_{\text{trunc}} + m$ where N_{trunc} is the number of modes kept in the SVD of C and m the size of D. Here, the matrix C is either the mean or the standard deviation of the bias at time t_i and D is either the mean or the standard deviation computed at time t_{i+1}. That way, here m is equal to 1 and N_{trunc} is in practice very small if the model is quite close to the data. The sequential computation of the bias moments with the SKL method is presented in Algorithm 6.

The advantage of doing the extrapolation with respect to the SVD modes is that the truncation allows to filter the SVD noise modes (with small singular values). Moreover, thanks to the SKL method, this operation is compatible with the constraint of real-time simulation.

5.1.3 Detection of the model bias with model evidence

In Bayesian inference, the model evidence can be seen as an indicator of the quality of the model; this is another advantage of this stochastic framework. Therefore, this quantity can be used for model selection [31] in order to determine which model among a class of models is the most likely to have generated the data.

Algorithm 6 Computation of the model bias moments with the SKL method

Require: Moment B_1 after the first assimilation time step. B_1 represents either the mean or standard deviation of the model bias.

$C = B_1$
$[U, S, V] = \mathbf{svd}(C)$ ▷ SVD computation
for $t = 2$ to N_t **do** ▷ Assimilation time steps
  $D = B_t$ ▷ Computation of the model bias moment for time t
  $[U\ \tilde{D}]R = \mathbf{qr}([US\ D])$ ▷ QR decomposition
  $[\tilde{U}, \tilde{S}, \tilde{V}] = \mathbf{svd}(R)$ ▷ SVD computation
  $U' = [U\ \tilde{D}]\tilde{U}$
  $S' = \tilde{S}$
  $V' = [U\ \tilde{D}]\tilde{U}$
  $U = U'(:, 1 : N_{\text{trunc}})$ ▷ Truncation of the modes
  $S = S'(1 : N_{\text{trunc}}, 1 : N_{\text{trunc}})$
  $V = V'(:, 1 : N_{\text{trunc}})$
  $\hat{B} = US\hat{V}^T$ ▷ Extrapolated moment
end for

In the case of Bayesian updating of a model $\mathcal{M}$ depending on parameters $\mathbf{p}$, the posterior pdf reads again:

$$\pi(\mathbf{p}|\mathbf{d}^{\text{obs}}) = \frac{1}{c}\pi(\mathbf{d}^{\text{obs}}|\mathbf{p}).\pi(\mathbf{p}) \tag{5.9}$$

In this formulation where a single model is considered, the model dependency of the posterior pdf is hidden in the formulation of likelihood function. The model evidence c is then computed as:

$$c = \pi(\mathbf{d}^{\text{obs}}) = \int \pi(\mathbf{d}^{\text{obs}}|\mathbf{p}).\pi(\mathbf{p})\mathrm{d}\mathbf{p} \tag{5.10}$$

If now, N models $\{\mathcal{M}_1, ..., \mathcal{M}_N\}$ depending on the parameters $\mathbf{p}_k$ are considered, the posterior probability that $\mathcal{M}_k$ generated the data $\mathbf{d}^{\text{obs}}$ is:

$$\pi(\mathcal{M}_k|\mathbf{d}^{\text{obs}}) = \frac{\pi(\mathbf{d}^{\text{obs}}|\mathcal{M}_k).\pi(\mathcal{M}_k)}{\sum_{k=1}^{N}\pi(\mathbf{d}^{\text{obs}}|\mathcal{M}_k).\pi(\mathcal{M}_k)} \tag{5.11}$$

where $\pi(\mathcal{M}_k)$ is the prior probability on the model and $\pi(\mathbf{d}^{\text{obs}}|\mathcal{M}_k)$ is the integrated likelihood of $\mathcal{M}_k$. This pdf is computed by marginalization:

$$\pi(\mathbf{d}^{\text{obs}}|\mathcal{M}_k) = \int \pi(\mathbf{d}^{\text{obs}}, \mathbf{p}_k|\mathcal{M}_k)d\mathbf{p}_k \tag{5.12}$$

$$= \int \pi(\mathbf{d}^{\text{obs}}|\mathbf{p}_k, \mathcal{M}_k).\pi(\mathbf{p}_k|\mathcal{M}_k)d\mathbf{p}_k \tag{5.13}$$

$$= c_k \tag{5.14}$$

where c_k is the model evidence associated to the Bayesian inference performed with the model $\mathcal{M}_k$.

That way, in the case where the prior pdfs $\pi(\mathcal{M}_k)$, $k \in \{1, ..., N\}$ are equal, the Bayes factor b_{ij} between two models $\mathcal{M}_i$ and $\mathcal{M}_j$ reads:

$$b_{ij} = \frac{\pi(\mathcal{M}_i|\mathbf{d}^{\text{obs}})}{\pi(\mathcal{M}_j|\mathbf{d}^{\text{obs}})} = \frac{c_i}{c_j} \tag{5.15}$$

The interpretation of this factor is that when $b_{ij} > 1$ the model $\mathcal{M}_i$ is more likely than the model $\mathcal{M}_j$ to produce the considered data $\mathbf{d}^{\text{obs}}$. The Jeffreys scale of evidence [65] gives an empirical interpretation of the Bayes factor (see Table 5.1.3).

b_{ij}	Strength of evidence
$< 10^0$	negative (supports $\mathcal{M}_j$)
10^0 to $10^{1/2}$	barely worth mentioning
$10^{1/2}$ to 10^1	substantial
10^1 to $10^{3/2}$	strong
$10^{3/2}$ to 10^2	very strong
$> 10^2$	decisive

Table 5.1: Jeffreys' scale of evidence for Bayes factor

The purpose of this study being to identify when a model becomes less reliable, the idea here is to monitor the evolution of the model evidence during the sequential inference problem. In the case where the model evidence drops drastically within a certain range, the addition of a model correction can be considered. However, comparing the model evidence with two successive measurements can be difficult because the observations are not affected with the same noise. Consequently, the model evidence monitoring is a rough idea to assess the quality of the model as the Bayes factor will be polluted by measurement noise.

The computation of the model evidence is not straightforward. In classical sampling methods that allow the characterization of the posterior density, this quantity is not available. However, by using the Transport Map sampling method, the model evidence can be interpreted as the distance between the reference density ρ and the posterior density pulled-back by the transport map M [44]:

$$c = \exp\left(\mathbb{E}_\rho\left[\log\left(M_\sharp^{-1}\pi\right) - \log(\rho)\right]\right) \tag{5.16}$$

This quantity is easy to compute as a quadrature rule with respect to ρ is supposed to be known. Another advantage of the Transport Map method as presented in Chapter 3 is that if the model is found to be inaccurate in a given time range the corresponding maps can be removed in order to retrieve the error committed in the estimation of the parameters. Then, missed measurements can be re-assimilated when the model bias is corrected.

5.2 Applications

5.2.1 Error in the description of the material behavior

The first example is an academic problem of structural mechanics. The studied structure is a rectangular beam with size 0.1 m $\times$ 1 m submitted to bending. The geometry and boundary conditions are described in Figure 5.1. The structure is submitted to an increasing load at point A and and clamped in its left side. Two behavior models are considered for the structure. The first one is a homogeneous isotropic elastic model and the second one is an elasto-plastic model with linear isotropic hardening.

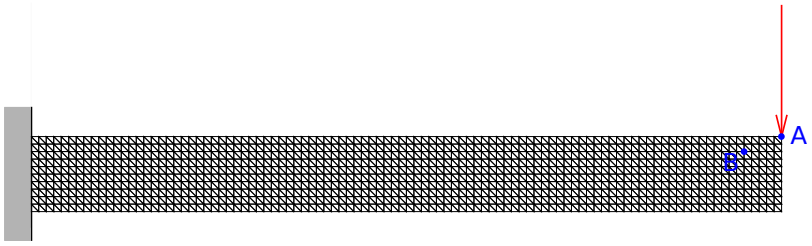
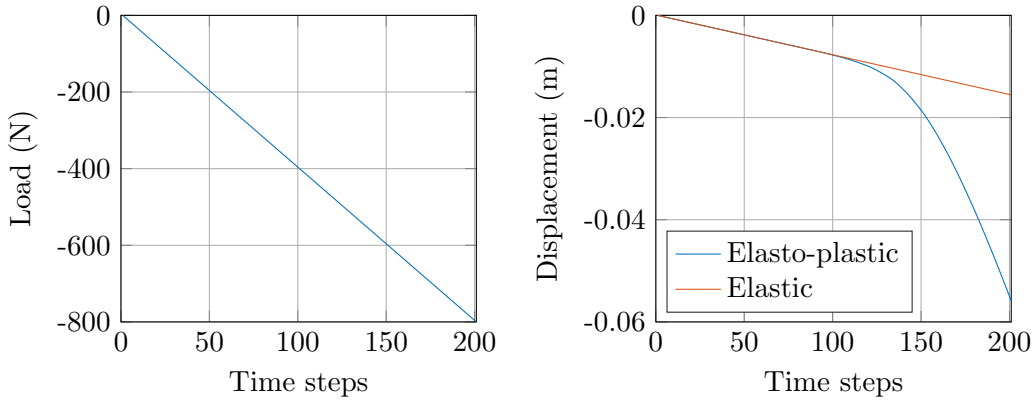


Figure 5.1: Mesh and boundary conditions

Figure 5.2(a) shows the load imposed at point A with respect to the time steps. Figure 5.2(b) shows the result of the numerical solution in terms of vertical displacement of the point A with the two considered models.



(a) Load intput

(b) Vertical displacement of point A

Figure 5.2: Time dependence of the imposed load and models outputs

Both models have the same elastic behavior with a Young modulus set to 200 GPa and a Poisson ratio to 0.3. That way, the displacement of the structure with both models coincide at the beginning (until the time step 100 approximately) then the elasto-plastic model gives larger displacements than the elastic one.

5.2.1.1 Influence of model bias

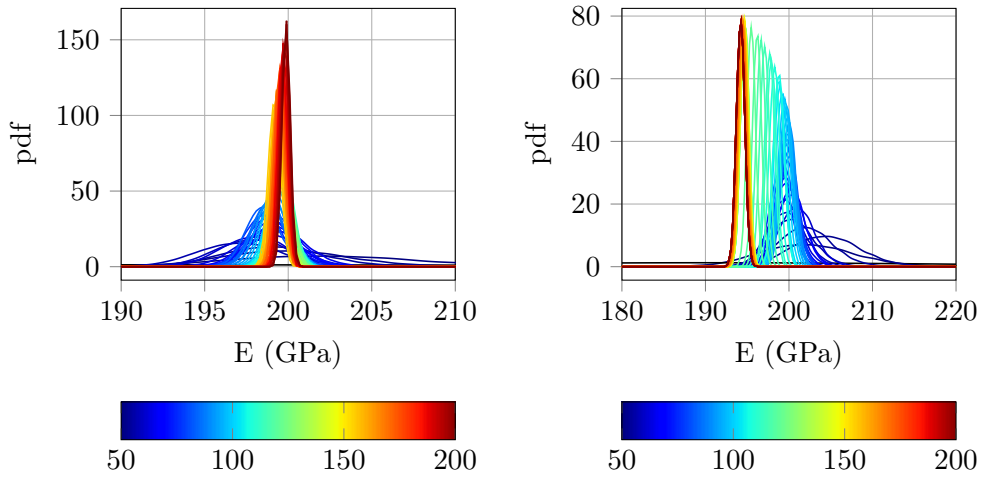
The first purpose here of is to see the influence of using a wrong model and a correction to identify a parameter in the context of Bayesian inference. The parameter of interest is the Young modulus E and we compare the result of the Bayesian inference by using a parameterized elastic model with simulated measurements from elastic and elasto-plastic models.

First, Bayesian inference is performed with the elastic model and simulated measurements from the elastic model evaluated with a Young modulus of 200 GPa polluted with a Gaussian noise with a standard deviation of 10^{-4} m. Only one measurement point is considered (point B in Figure 5.1) which gives two measurements in both x and y directions. Time steps from 50 to 200 are considered and the acquisition of the measurement is performed every two time steps so that the amount of assimilation time steps is 75. The

prior density is Gaussian with a mean of 190 GPa and a standard deviation of 10 GPa.

Figure 5.3(a) shows the result in terms of marginals of the Bayesian inference with elastic simulated measurements. The marginals are computed with transport of samples by computing transport maps with the same methodology as in Chapters 3 and 4. The color map indicates the correspondence with the physical time steps (same reference as in Figure 5.2). The result is classical: marginals become thinner with the assimilation of the measurements and the maximum/mean of the density become closer to the reference value of 200 GPa.

On the other hand, Figure 5.3(b) shows the result of the Bayesian inference with elasto-plastic simulated measurements. In this case, the marginals for the first time steps follow the correct measurement as the beam still has an elastic behavior. However, from time step 100, marginals are shifted from the reference value and the estimation diverges. As the structure becomes more compliant due to local plasticity, the identified Young modulus value becomes smaller than the reference. Consequently, the estimation is diverging as the mean/maximum of each marginals are continuously during the iterations.



(a) Posterior marginals computed with elastic simulated measurements

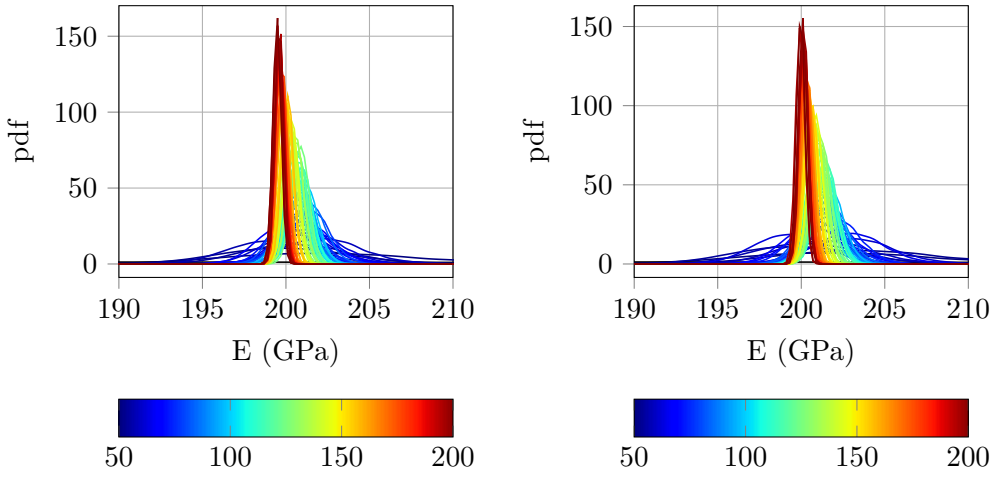
(b) Posterior marginals computed with elasto-plastic simulated measurements

Figure 5.3: Reference solutions without correction (color map indicates the correspondence with time steps)

5.2.1.2 Dynamical correction of model bias

In order to improve the convergence of the inference process, a correction of the model bias is performed. Figure 5.4 shows the result of the inference of the Young modulus from elasto-plastic measurements with the addition of a model bias. Figure 5.4(a) shows the result with a deterministic correction and Figure 5.4(b) the result with a stochastic correction. The deterministic bias is computed following (5.1) using the mean estimator. The stochastic bias is computed following (5.2). Thus, both corrections are computed by linear extrapolation directly for the deterministic correction and by means of mean and standard deviation for the stochastic one.

The inference process is highly improved by the addition of the correction of model

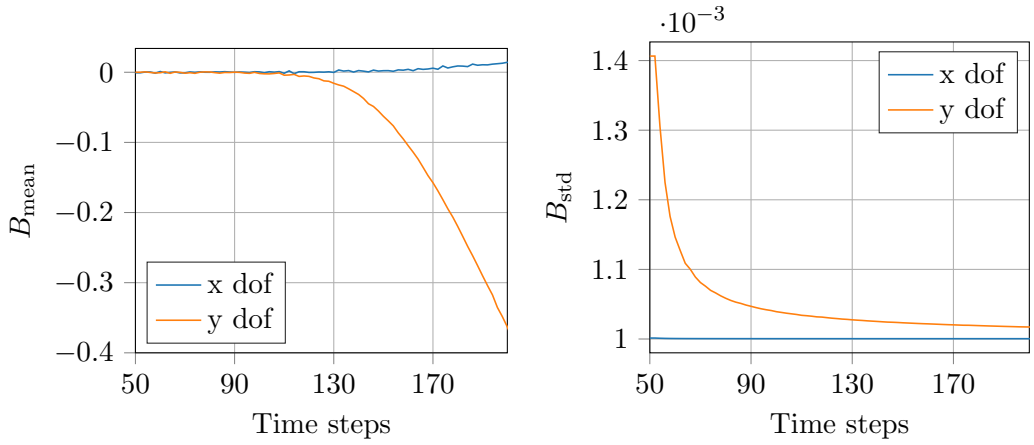


(a) Posterior marginals computed with deterministic correction

(b) Posterior marginals computed with stochastic correction

Figure 5.4: Solutions computed with elasto-plastic measurements and model correction (color map indicates the correspondence with time steps)

bias with maximum of the marginals converging to reference value of 200 GPa. In this example both corrections lead to similar results due to the low level of noise. In Figure 5.5, the means and standard deviations of the model correction are shown. The mean of the model correction is null for the first time steps then increases to compensate the extra-deformation of the structure due to plasticity. The standard deviation of the model bias decreases with the assimilation due to the fact that the uncertainty on the parameter decreases at the same time. Then, the standard deviation converges toward the standard deviation of the measurement noise. However, this variation is very small due to the fact that the sensitivity of the model with respect to the parameter is negligible when compared to the level of the measurement noise.

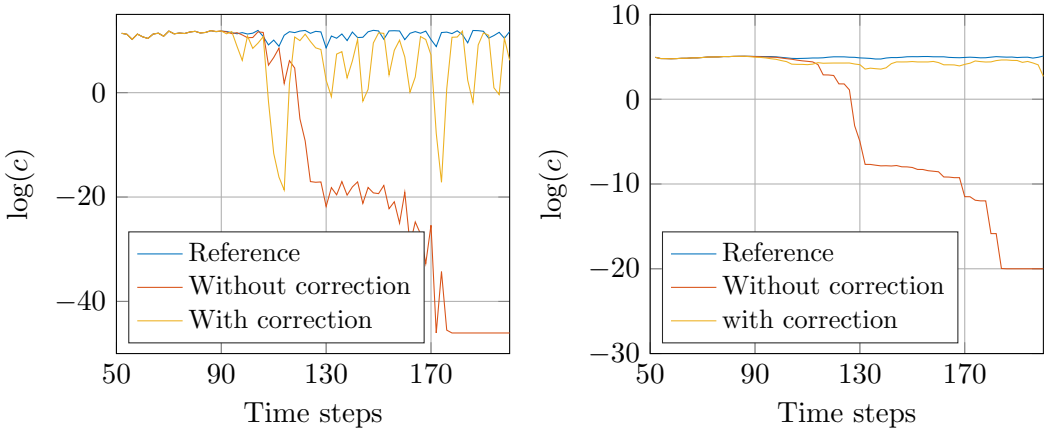


(a) Evolution of the mean bias for both measured degrees of freedom

(b) Evolution of the standard deviation of bias for both measured degrees of freedom

Figure 5.5: Evolution of mean and standard deviation of model bias

Figure 5.6(a) shows the evolution of the log-model evidence corresponding to the different types of inference: inference with measurements simulated with the elastic model (corresponding to Figure 5.3(a)), inference with measurements simulated with the elasto-plastic model (corresponding to Figure 5.3(b)), and inference with measurements simulated with the elasto-plastic model and stochastic correction (corresponding to Figure 5.4(b)). The evolution of the model evidence is noisy due to the fact that the measurements are not



(a) Evolution of raw model evidences

(b) Evolution of smoothed model evidences

Figure 5.6: Evolution of model evidences

polluted with the same noise at each time step. Each model evidence is computed for one realization of the measurement random variable. The model evidence also depends on the accuracy of the computed transport map even if the same tolerance in terms of variance diagnostic is imposed for each time step. In Figure 5.6(b), a 10-points moving average smoothing is applied in order to filter the noise. For the reference inference computation with elastic measurements, the model evidence is roughly constant. In the case of elasto-plastic measurements, we can observe that without correction the model evidence drops off drastically. However, the addition of the model correction seems to have a good impact on the model evidence even if it is still lower than the reference.

The monitoring of the model evidence quantity can be used to compute the model correction only when needed. Figure 5.7 shows the result of the inference process performed with the elastic model and the correction of the model bias while measurements are also simulated with the elastic model. In this case, the model bias is null and the addition of the correction induces an artificial bias due to the measurement noise. This leads to a slight inaccuracy of the computation of the marginals in comparison with the reference without correction presented in Figure 5.3(a).

5.2.1.3 Extrapolation on quantities of interest with SVD

In this section, the quantity of interest is the global kinematics of the beam. Thus, the model correction is still computed at the measurement point to improve the convergence of the inference solution but also on the entire beam boundary in order to be able to describe the global motion of the beam during time. In this example, one measurement point is used to perform inference on the Young modulus parameter, then the estimation residual

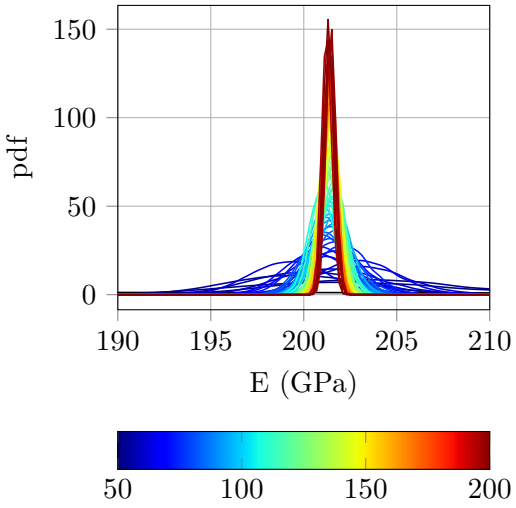


Figure 5.7: Posterior marginals computed with simulated elastic measurements and model correction (color map indicates the correspondence with time steps)

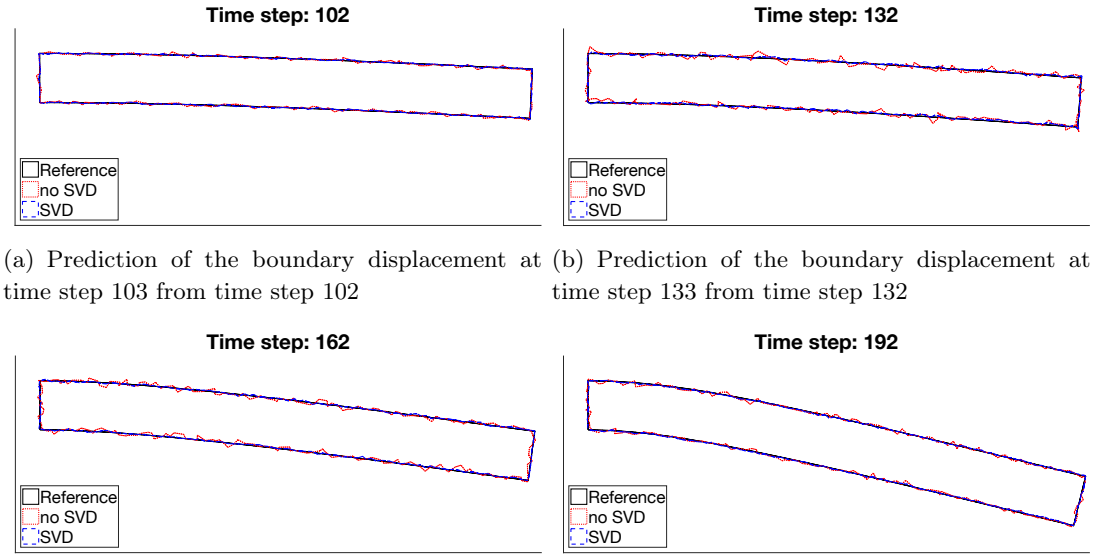
is computed at both boundaries and measurement points then extrapolated to compute the model correction for the next time step. The number of boundary nodes is 220 and the extrapolation is performed by extrapolating time SVD modes.

At each time step the elastic posterior pdf on the Young modulus is updated and the residual estimation is computed at each boundary nodes. Then, this residual is extrapolated to give the model correction for the next time step. Figure 5.8 shows the estimation of the displacement of the beam for different time steps with the estimation of the model correction performed at the previous time step. The predicted displacement at each node is a random vector, for readability reasons only the mean is represented. The black solid line represents the reference given by the elasto-plastic simulation with a Young modulus of 200 GPa. The blue dashed line represents the mean of the estimated displacement with the extrapolation of the residual computed on SVD time modes (here only one mode is kept). As a comparison, the red dotted line represents the mean of the estimated displacement with point-wise extrapolation.

Overall, the estimations are quite good and the estimated displacement with the updated elastic model coupled with the correction follows the true elasto-plastic displacement of the beam. However the point-wise correction is much more disturbed. In this application, no mechanical regularization or smoothing is applied in the predicted displacement which could greatly improve the results. However the purpose here is to quantify the quality of the raw results.

In Figure 5.9, the quantity of interest is the global displacement of the beam at time step 200. In this case the displacement is computed with the displacement of the updated elastic model evaluated at time step 200 and the linear extrapolation of the model bias at this time step.

As in the previous figure, the black solid line represents the reference, the blue dashed line the prediction with SVD extrapolation and the red dotted line with the point-wise extrapolation. In this example, the extrapolation of the noise by doing point-wise extrapolation is magnified. For all time steps, the SVD extrapolation gives much better results



(a) Prediction of the boundary displacement at time step 103 from time step 102

(b) Prediction of the boundary displacement at time step 133 from time step 132

(c) Prediction of the boundary displacement at time step 163 from time step 162

(d) Prediction of the boundary displacement at time step 193 from time step 192

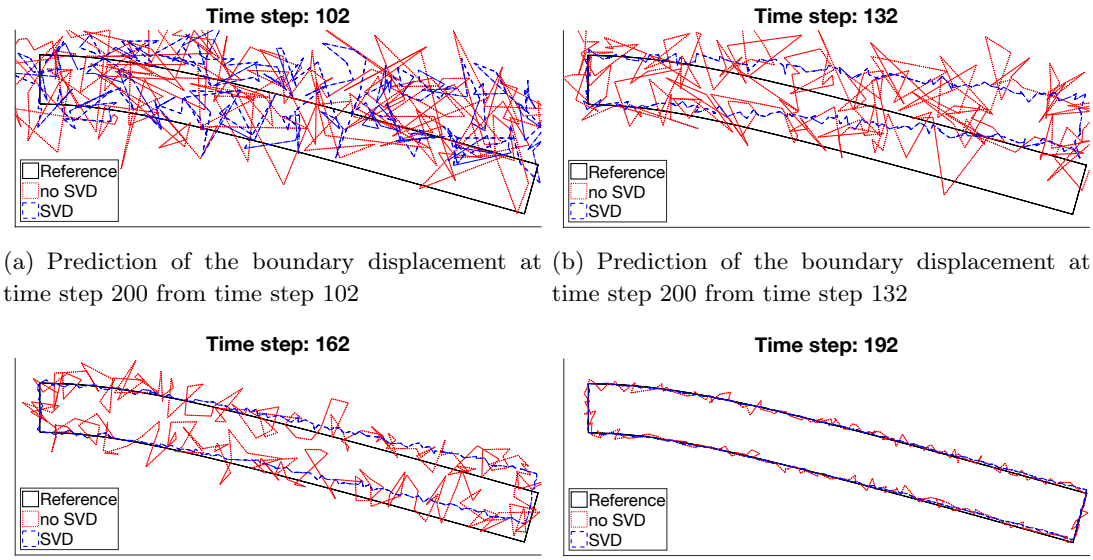
Figure 5.8: Prediction of the global displacement of the beam at different time steps

in terms of consistency. However, for time step 102 (Figure 5.9(a)) both results are quite wrong. The reasons are first that at this time step the extrapolation to be done is the largest. Second, at this time step the model bias is quite low because the plasticity is just beginning. Thus, only noise is extrapolated and motivates the use of a monitoring the model evidence in order to prevent this situation. For the other time steps, the error between the prediction and the reference is due to the linear approximation of the model bias between the considered time step and time step 200.

Here, the extrapolation of the model bias is computed on all boundary nodes (220 nodes, 440 degrees of freedom). In Figure 5.10, times to sequentially compute the SVD of the model bias with the SKL algorithm 6 and the classical SVD are compared. Both computations are performed with a MATLAB code with the built-in functions of SVD and QR decomposition. For the SKL algorithm, only one mode is kept at each time step. In Figure 5.10(a), the SVD of the mean model bias is computed for the 440 degrees of freedom of the beam boundary; we observe that the classical SVD seems to outperform the SKL algorithm. However, in the case where the sequential SVD of a larger matrix needs to be computed, the SKL algorithm seems to be cost efficient (Figure 5.10(b)). Here, the SVD is computed on all 2222 degrees of freedom of the beam mesh. Overall, the computation time of the SVD is quite low making the approach suitable for real-time applications.

5.2.2 PGD truncation error

The objective of this second example is to study the influence of the truncation error in the PGD decomposition and its correction. The test on a concrete beam with damage presented in Chapter 4 is again considered. The system is studied between time step 40 (where the damage of the structure begins) and the final time step. The reference Bayesian inference solution is computed with simulated measurements from the PGD model with 6



(a) Prediction of the boundary displacement at time step 200 from time step 102

(b) Prediction of the boundary displacement at time step 200 from time step 132

(c) Prediction of the boundary displacement at time step 200 from time step 162

(d) Prediction of the boundary displacement at time step 200 from time step 192

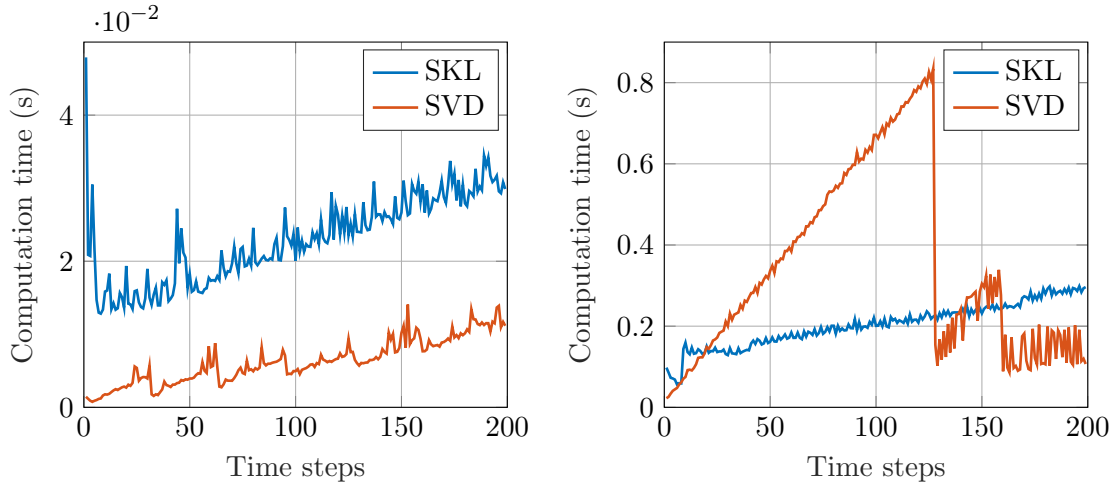
Figure 5.9: Prediction of the global displacement of the beam at time step 200 from different time steps

modes and the reference values $(\bar{Y}_0^{\text{ref}}, \bar{A}_d^{\text{ref}}) = (1, 1)$. The prior density is a Gaussian with mean $(\bar{Y}_0, \bar{A}_d) = (0.9, 1.1)$ and diagonal covariance matrix with values $(0.2, 0.2)$. The same 30 measurements points as the ones presented in Figure 4.13 are used. The same standard deviation of the Gaussian noise equal to 10^{-6} is also used. The result of this inference problem in terms of marginals computed with Transport Map sampling is shown in Figure 5.11.

As a comparison, Figure 5.12 shows the result of the Bayesian inference by using the same measurements but using the PGD model truncated to 3 modes for the formulation of the likelihood functions. The influence of the truncation error in the computation of the marginals is low for the first time steps. Indeed, as shown in Figure 4.11, only the first mode contributes to the PGD solution during the elastic part (corresponding to the first time steps). Then, when damage begins, the error committed by truncation becomes larger and the marginals are shifted giving a wrong estimation of the parameters.

In order to overcome this issue, the stochastic correction is added to the formulation of the Bayesian inference. The model bias is estimated at the 30 measurements points (i.e. 60 degrees of freedom) then the stochastic correction is computed by the extrapolation of the SVD modes of the mean and variance. In this application, only two modes are kept to represent the model correction. The result of the Bayesian inference with correction of the model bias is shown in Figure 5.13. Here also, model correction improves the result of the inference by withdrawing the error bias in the posterior densities.

In this example, the error committed is known and formulated in terms of PGD modes. Figure 5.14(a) compares the first SVD time mode of the mean correction with the 4th PGD time mode which is the first mode of the true truncation error. Here, the Sequential Karhunen-Loeve algorithm is not used in order to see the influence of high-order modes. It appears that the mode of the mean correction is a noisy version of the PGD mode. The



(a) SVD computed for a matrix of size 440 × number of time steps

(b) SVD computed for a matrix of size 2222 × number of time steps

Figure 5.10: Comparison of the computation times between SKL and MATLAB SVD algorithms for different sizes of matrices

effect of the noise can also be observed in the graph of the singular values of the mean correction in Figure 5.14(b). Although the theoretical number of non-zero values is 3 as the number of PGD modes withdrawn from the model is 3, singular values slowly decrease from the second to the sixtieth. Thus, it is necessary to truncate the SVD in order to limit the extrapolation of noise when computing the correction from the estimation residual.

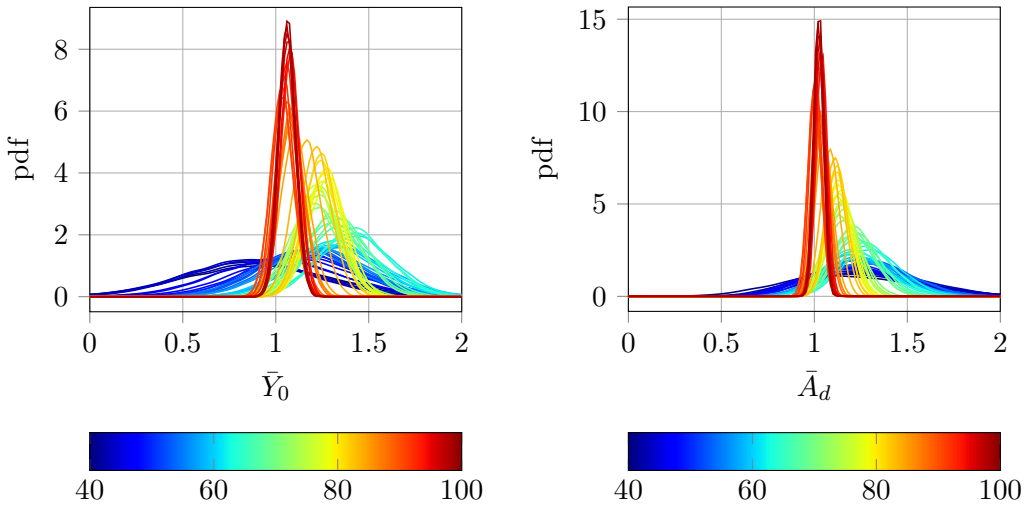
However, the computation of the estimation residual depends on the realization of the random variable of measurement noise as no filtering is performed at this stage. Figure 5.15 shows the same result than Figure 5.13 but using a different random seed, which gives another set of simulated measurements. This solution is quite different from the first one and the last marginals maxima are farther to the reference values.

In Figure 5.16, the first SVD time mode of the mean correction is plotted with the SVD mode of the previous solution and the reference which is the 4th PGD mode. Thus, this new computation gives another realization of the noisy PGD mode reference. Consequently, this explain the differences between Figures 5.13 and 5.15.

Figure 5.17 shows the evolution of model evidences for the reference inference with the 6 mode-PGD model, the inference with the 3-mode PGD model without correction, and the inference with the 3 mode-PGD model and the correction. All evolutions are smoothed with a moving average filter of span 10. We can see that the model evidence slowly drops off from the reference when using the 3 mode-PGD model without correction. Adding the correction, improves the model evidence except in the first time steps when the model bias is the lowest. This is due to the fact that the correction is introduced from the beginning of the Bayesian inference procedure.

5.3 Conclusions

In this chapter, the influence of the model bias and its correction on the Bayesian inference solution have been highlighted. The choice made here was to build a model correction based



(a) Posterior marginals on parameter Y_0

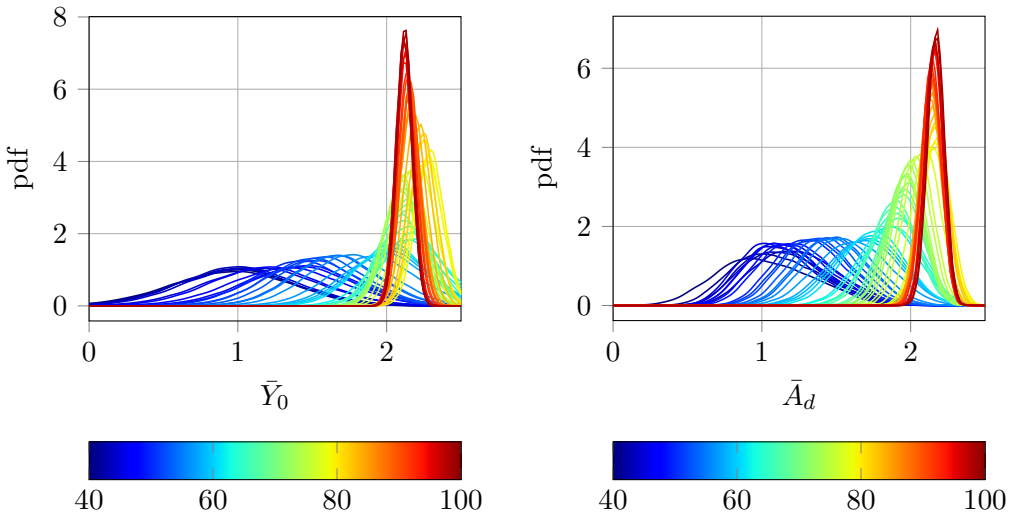
(b) Posterior marginals on parameter A_d

Figure 5.11: Reference solution computed with the full 6 mode-PGD model (color map indicates the correspondence with time steps)

on the extrapolation of the stochastic estimation residual. This extrapolation gives an independent random variable of model bias used to improve the Bayesian inference process. This data-based approach is performed without making any assumptions on the type of model bias so that all kinds of uncertainty sources can be considered (time-dependent boundary conditions, geometry defects, etc.).

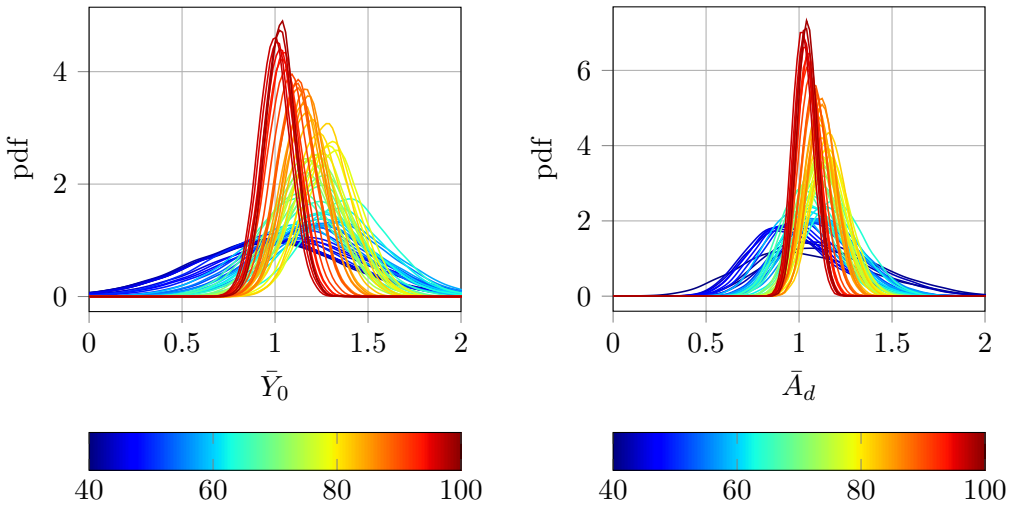
However, in both treated examples the model bias followed the same pattern, namely slowly increasing from a null value. In the opposite case, the knowledge of the model bias from error estimation procedures [3, 75] should be used as prior for the model bias random variable as done in [97]. Monitoring the model evidence seems also to be an interesting idea to identify when a model becomes less accurate and a model correction is necessary. This monitoring has not been included in the presented example and future works will be to set an automatic procedure to control the model bias. This idea can easily be coupled with the Transport Map methodology where the inference result is represented by a composition of maps. In the case where a model bias is detected with the monitoring of the model evidence, the maps corresponding to the assimilation of data with an inaccurate model can be removed.

In the presented approach, the model correction is descriptive as it represents the gap between the model prediction sensitive to noise. Another more robust approach with respect to the noise would consist in parametrizing the model correction and update those hyper-parameters along the parameters of interest of the model. This approach however leads to additional computation cost and complexity as the dimension of the posterior will be larger.



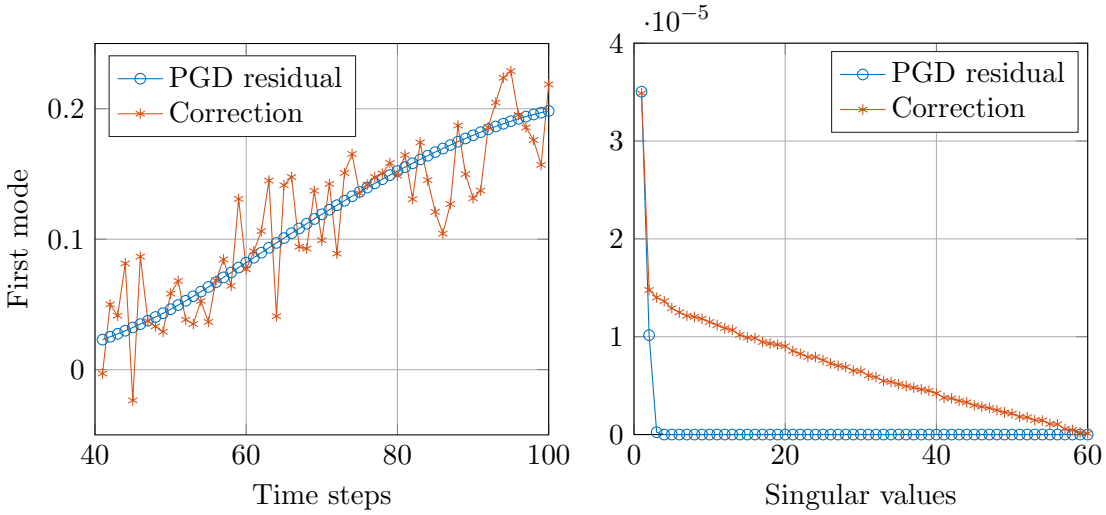
(a) Posterior marginals on parameter Y_0 (b) Posterior marginals on parameter A_d

Figure 5.12: Solution computed with the 3 mode-PGD model for simulated measurements from the 6 mode-PGD model (color map indicates the correspondence with time steps)



(a) Posterior marginals on parameter Y_0 (b) Posterior marginals on parameter A_d

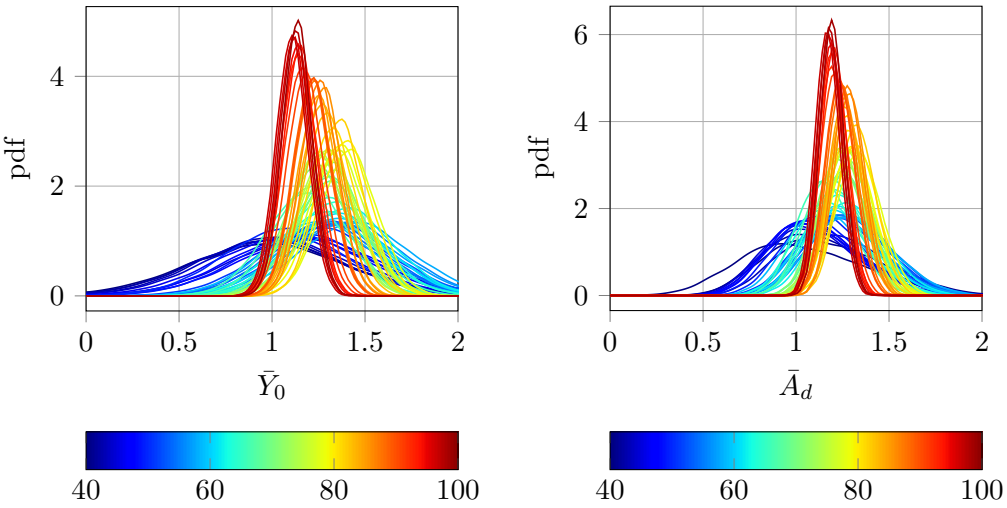
Figure 5.13: Solution computed with the 3 mode-PGD model and model correction for simulated measurements from the 6 mode-PGD model (color map indicates the correspondence with time steps)



(a) First model correction time modes

(b) Singular values

Figure 5.14: SVD of the mean correction



(a) Posterior marginals on parameter Y_0

(b) Posterior marginals on parameter A_d

Figure 5.15: Solution with model correction with a different seed of the random number generator

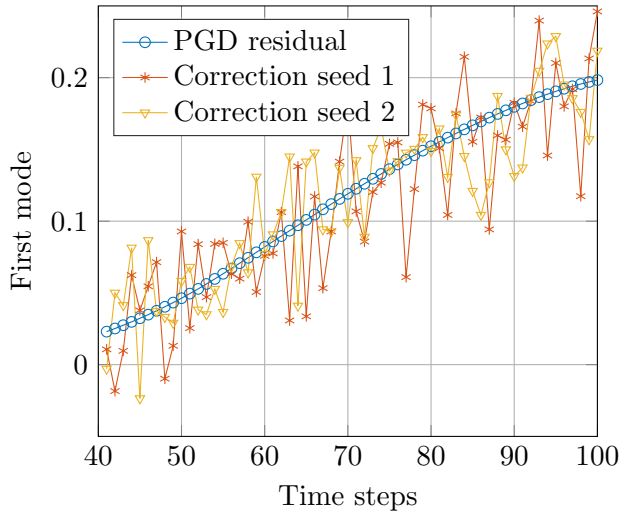


Figure 5.16: Comparison between the first time modes of the model correction

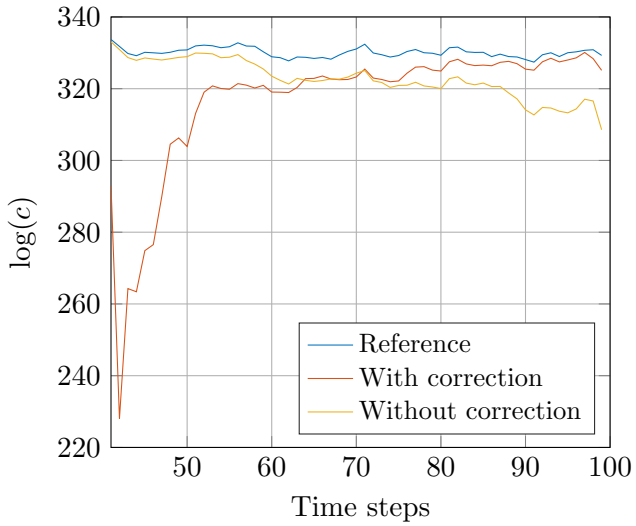


Figure 5.17: Evolution of the smoothed model evidences

Chapter 6

Selection of Digital Image Correlation data for the Bayesian updating of PGD models

Digital Image Correlation is a measurement method which is widely used in solid mechanics to identify full-field displacements during mechanical tests on materials and structures [60, 141]. This method was performed during the experimental campaign presented in Chapter 4. This measurement technique provides a rich information but can be difficult to handle in the context of real-time Bayesian data assimilation. Indeed, as measurements are compared with model outputs, the more measurements are considered, the more model outputs need to be evaluated leading to additional computational cost and questioning the real-time constraint. Then, this computation cost can be reduced by only considering the most relevant information from the full-field measurements. In this chapter, the purpose is to optimize the use of DIC measurements in the context of the Bayesian updating of PGD models. Two ways are explored: filtering a large amount of available data, or performing experimental design to optimize the quality of obtained data.

6.1 The global Digital Image Correlation method

6.1.1 Correlation principles

The purpose of Digital Image Correlation is to identify a displacement field $\mathbf{u}(\mathbf{x})$ that links two pictures $f(\mathbf{x})$ and $g(\mathbf{x})$. The pictures $f(\mathbf{x})$ and $g(\mathbf{x})$ are matrices with the number of pixels ($\mathbf{x}$ represents the pixels coordinates) with gray level values. f is the reference picture and g is the picture deformed by the displacement field $\mathbf{u}(\mathbf{x})$ such that the local gray level conservation reads:

$$f(\mathbf{x}) = g(\mathbf{x} + \mathbf{u}(\mathbf{x})) \tag{6.1}$$

In practical cases, this local conservation is not strictly satisfied due to noise. The displacement $\mathbf{u}(\mathbf{x})$ is then determined by solving a more global problem that consists of minimizing the following scalar quantity:

$$||f(\mathbf{x}) - g(\mathbf{x} + \mathbf{u}(\mathbf{x}))|| \tag{6.2}$$

Usually the L^2-norm is chosen for $||.||$. The domain where the correlation residual (6.2) is defined leads to two categories of DIC approaches.

First, the local DIC approach aims at minimizing the correlation residual on sub-images or Zones Of Interest (ZOI) treated independently. The local residual to minimize then reads:

$$\sum_{\mathbf{x}\in ZOI} \left(f(\mathbf{x}) - g(\mathbf{x}+\mathbf{u}(\mathbf{x}))\right)^2 \tag{6.3}$$

In this approach, the output of the analysis is the mean displacement at the middle of the ZOI. This averaging naturally introduces a noise filtering. Large ZOI will lead to small uncertainties but will be restricted to the identification of smooth displacement fields.

Second, in the global DIC approach used here, the correlation residual is defined in the whole measurement zone (Region Of Interest or ROI). This ROI contains a large number N_p of pixels, so that a global nonlinear correlation residual is minimized:

$$\frac{1}{N_p}\sum_{\mathbf{x}\in ROI} \left(f(\mathbf{x}) - g(\mathbf{x}+\mathbf{u}(\mathbf{x}))\right)^2 \tag{6.4}$$

In this variational formulation the displacement field is expressed in a chosen basis:

$$\mathbf{u}(\mathbf{x}) = \sum_n u_n \psi_n(\mathbf{x}) \tag{6.5}$$

where ψ_n are chosen vector functions and u_n the corresponding degrees of freedom. This approach helps to regularize the problem by considering physics knowledge (as the displacement continuity). Usually, the displacement field is described with Finite Element (FE) functions which is convenient when the measurements are compared with FE models (e.g with FEMU methods [70, 85, 106]). In this case, u_n are the nodal values of the displacement field $\mathbf{u}(\mathbf{x})$.

To solve (6.4) a first Taylor expansion is used to linearize $g(\mathbf{x}+\mathbf{u}(\mathbf{x}))$:

$$g(\mathbf{x}+\mathbf{u}(\mathbf{x})) \approx g(\mathbf{x}) + \mathbf{u}(\mathbf{x}).\nabla g(\mathbf{x}) \tag{6.6}$$

$$\approx g(\mathbf{x}) + \mathbf{u}(\mathbf{x}).\nabla f(\mathbf{x}) \tag{6.7}$$

Then, an iterative sequential construction of the $\mathbf{u}(\mathbf{x})$ is performed by minimizing:

$$\sum_{\mathbf{x}\in ROI} \left(f(\mathbf{x}) - \tilde{g}(\mathbf{x}) - \nabla f.\mathrm{d}\mathbf{u}^{(k)}(\mathbf{x})\right)^2 \tag{6.8}$$

where $\tilde{g}(\mathbf{x}) = g(\mathbf{x}+\mathbf{u}^{(k)}(\mathbf{x}))$ is the updated deformed image, and $\mathrm{d}\mathbf{u}^{(k)} = \mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}$ is the correction at iteration k. In order to have sub-pixel resolution a (spline) gray level interpolation is performed in the construction of $\tilde{g}$.

Finally, by writing the stationary of (6.8), the minimization is equivalent to solve the linear system:

$$\mathbb{M}\mathrm{d}u^{(k)} = \mathbf{b}^{(k)} \tag{6.9}$$

with

$$\begin{aligned} \mathbb{M}_{mn} &= \sum_{\mathbf{x}\in ROI} \nabla f(\mathbf{x}).\psi_m(\mathbf{x}) \nabla f(\mathbf{x}).\psi_n(\mathbf{x}) \\ \mathbf{b}_m^{(k)} &= \sum_{\mathbf{x}\in ROI} (f(\mathbf{x}) - \tilde{g}(\mathbf{x})) \nabla f(\mathbf{x}).\psi_m(\mathbf{x}) \end{aligned} \tag{6.10}$$

The vector $\mathbf{b}^{(k)}$ is the residual vector updated at iteration k and $\mathbb{M}$ is the symmetric positive DIC matrix computed once for all as it only depends on the reference image f and the chosen basis ψ.

6.1.2 Measurement error in DIC

When dealing with measurements, a crucial point is the associated uncertainties. This is particularly the case when using Bayesian inference where the pdf of the measurement is supposed to be known. In this context, the global DIC method seems to be well suited as it provides a clear estimation of the measurement noise.

First, a white noise with zero mean variance of $2\gamma_f^2$ is supposed to affect the deformed image g while the reference image f is considered as noiseless (equivalent to assume that both image are affected by the same noise). This white noise can be identified by computing the variance of $f - g$ with two different reference pictures when no displacement is prescribed. Then the covariance matrix $\mathbb{C}_{\mathbf{u}}$ of the measured degrees of freedom can be computed from (6.9) and leads to:

$$\mathbb{C}_{\mathbf{u}} = 2\gamma_f^2 \mathbb{M}^{-1} \tag{6.11}$$

Due to the fact that $\mathbb{M}^{-1}$ is not diagonal, the displacement noise is spatially correlated. This correlation depends on the chosen basis ψ_n. For example, using FE functions with small elements leads to larger measurements uncertainties than when using larger elements. However, using large elements will compromise the accuracy of displacements with strong gradients.

Finally, global DIC provides the displacement field over the basis ψ as well as the residual map (see Figure 4.22 for example) and noise covariance/sensitivity with the matrix $\mathbb{M}$.

6.2 Selection of DIC data for Bayesian updating of PGD models

6.2.1 Computation of sensitivity fields

The DIC provides the full-field measurement of the displacement $\mathbf{u}$. However, this large amount of data can be difficult to handle and post-process especially in the context of real-time model updating due to the large number of model queries that the method requires. Thus, the purpose is to use the most relevant data in order to update the model of interest. In model updating applications, the relevance of the data is the sensitivity of the data with respect to the variation of the parameters of interest. For example, in the application presented in Chapter 4, the model depends on parameters $(p_1, p_2) = (Y_0, A_d)$. The sensitivity of the model output $\mathbf{u}$ (which is the displacement field in the structure) with respect to the parameter p_i reads:

$$\mathbf{S}_i(\mathbf{x}, p_1, p_2) = \frac{\partial \mathbf{u}(\mathbf{x}, p_1, p_2)}{\partial p_i} \quad i \in \{1, 2\} \tag{6.12}$$

These sensitivity fields can be easily computed as the displacement field of the structure is known in the PGD form:

$$\mathbf{u}^{\mathrm{PGD}}(\mathbf{x}, t, Y_0, A_d) = \sum_{n=1}^{m} \Lambda_n(\mathbf{x}) \lambda_n(t) \alpha_n^1(Y_0) \alpha_n^2(A_d) \tag{6.13}$$

Thus, the sensitivity fields $\mathbf{S}_{p_1}$ and $\mathbf{S}_{p_2}$ can be computed with the derivation of parameter modes as:

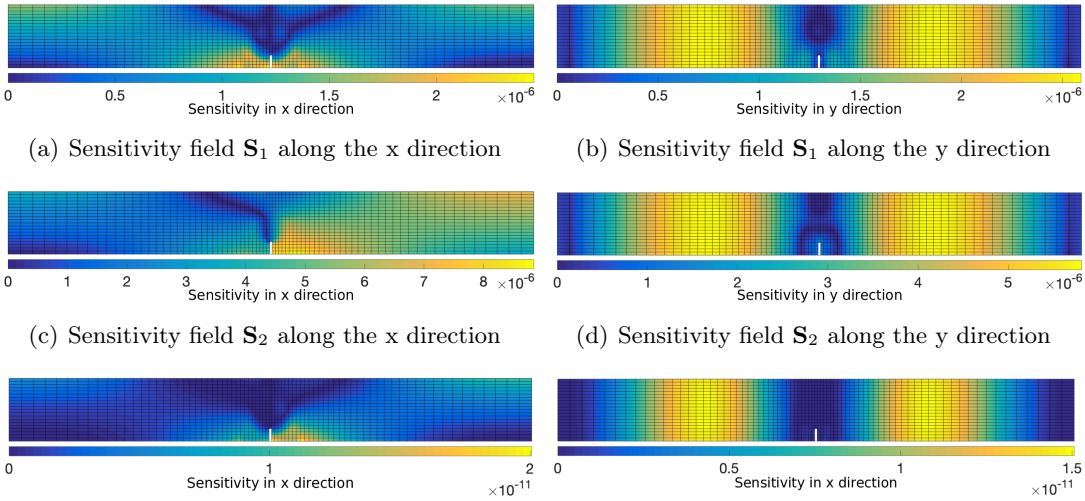
$$\mathbf{S}_1(\mathbf{x}) = \left| \sum_{n=1}^{m} \Lambda_n(\mathbf{x}) \lambda_n(\bar{t}) \frac{\partial \alpha_n^1}{\partial Y_0}(\bar{Y}_0) \alpha_n^2(\bar{A}_d) \right| \tag{6.14}$$

$$\mathbf{S}_2(\mathbf{x}) = \left| \sum_{n=1}^{m} \Lambda_n(\mathbf{x}) \lambda_n(\bar{t}) \alpha_n^1(\bar{Y}_0) \frac{\partial \alpha_n^2}{\partial A_d}(\bar{A}_d) \right| \tag{6.15}$$

Here, the mean sensitivity fields are computed with the mean values of time $\bar{t}$ and parameters $\bar{Y}_0$ and $\bar{A}_d$.

In addition to the sensitivities $\mathbf{S}_1$ and $\mathbf{S}_2$, the cross sensitivity $\mathbf{S}_{12} = \mathbf{S}_1 \times \mathbf{S}_2$ can be defined in order to have an average sensitivity of the model with respect to both parameters. Those sensitivities can be computed in the offline phase, once for all, and used to select the location of the most sensitive measurements for parameters identification. That is why defining $\mathbf{S}_{12}$ can be useful when dealing with the identification of strictly independent parameters; in this case, the data selected with respect to the sensitivity of one parameter may indeed be insensitive to the other parameters.

Considering the practical application addressed in Chapter 4, Figure 6.1 shows the sensitivity fields in both displacement directions x (horizontal) and y (vertical). Overall,



(a) Sensitivity field $\mathbf{S}_1$ along the x direction

(b) Sensitivity field $\mathbf{S}_1$ along the y direction

(c) Sensitivity field $\mathbf{S}_2$ along the x direction

(d) Sensitivity field $\mathbf{S}_2$ along the y direction

(e) Sensitivity field $\mathbf{S}_{12}$ along the x direction

(f) Sensitivity field $\mathbf{S}_{12}$ along the y direction

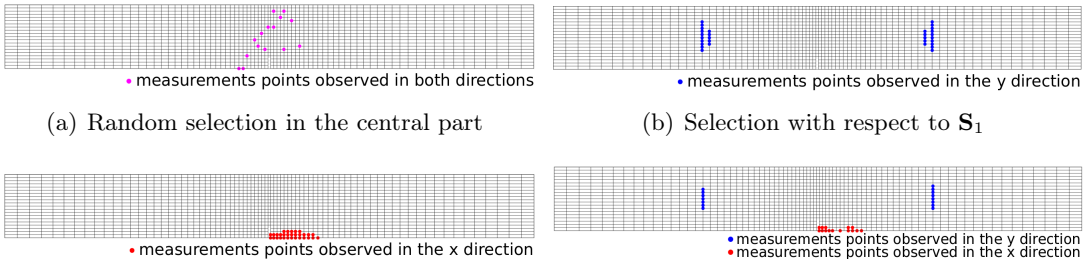
Figure 6.1: Sensitivity fields

the most sensitive points are the points with the largest displacement (for example around the notch for the sensitivity in the x direction). However, as in the considered problem the displacement of the structure is prescribed in the y direction at the mid span and at both bottom ends of the beam, those regions are quite insensitive to the parameter variation.

6.2.2 Influence of the measured degrees of freedom on the Bayesian solution

From the computation of sensitivities, the most sensitive degrees of freedom can be selected to compute the Bayesian solution. Figure 6.2 shows the selection of the best 30 degrees of

freedom with respect to the chosen sensitivity field. As a comparison, Figure 6.2(a) shows a random selection of degree of freedom constrained in the central part of the beam.



(a) Random selection in the central part

(b) Selection with respect to $\mathbf{S}_1$

(c) Selection with respect to $\mathbf{S}_2$

(d) Selection with respect to $\mathbf{S}_{12}$

Figure 6.2: Selection of the 30 most sensitive degrees of freedom

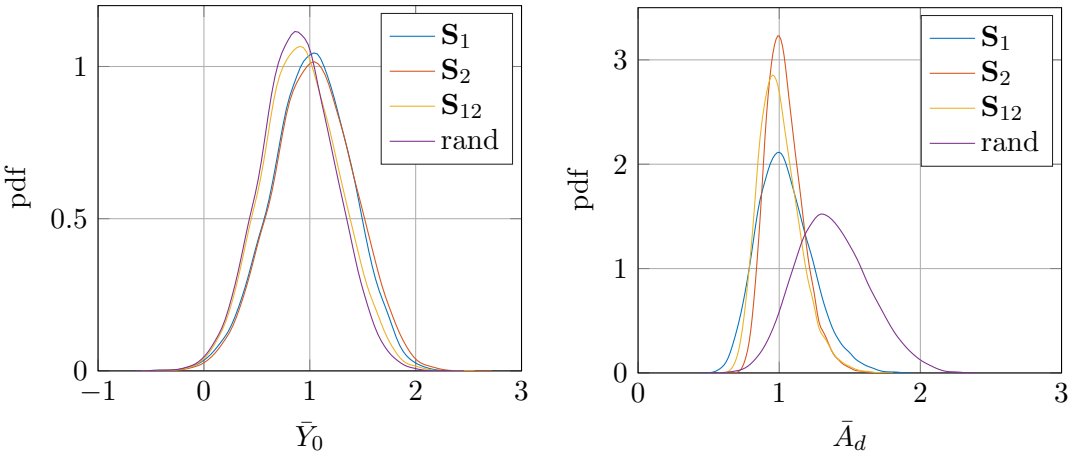
The Bayesian inference problem of interest is here the assimilation of the measured degrees of freedom for only one time step. Thus, the posterior density of interest reads:

$$\pi(Y_0, A_d) \propto \pi(\mathbf{u}^{\text{obs},45}|Y_0, A_d).\pi(Y_0, A_d) \tag{6.16}$$

where $\mathbf{u}^{\text{obs},45}$ is the set of measured degrees of freedom at time step 45 which is a time step where the damage of the beam is beginning. Those measurements are simulated from the numerical PGD model with reference $(\bar{A}_d, \bar{Y}_0) = (1.0, 1.0)$ and the addition of a Gaussian white noise with zero mean and standard deviation of 10^{-6} m. The prior density $\pi(Y_0, A_d)$ is chosen to be Gaussian with mean $(0.9, 1.1)$ and a diagonal covariance matrix with variances $(0.2, 0.2)$.

The results in terms of marginals for time step 45 with respect to the measured degrees of freedom are shown in Figure 6.3. For the marginals on parameter Y_0, we observe that the influence is quite low for this simulation as the densities are very similar. However, we can notice that the maximum of marginals computed with the data selection with respect to $\mathbf{S}_1$ and $\mathbf{S}_2$ is closer to the reference $\bar{Y}_0 = 1$ than the one computed with a random data selection. For the marginals on parameter A_d, the influence of the data selection is more visible. The estimation given by the marginal with the random data selection is quite far with a maximum *a posteriori* estimation of $\bar{A}_d = 1.3$ and large variance. This estimation is improved by selecting data with the sensitivities $\mathbf{S}_1$, $\mathbf{S}_2$ and $\mathbf{S}_{12}$. In terms of variance, the data selection with $\mathbf{S}_2$ gives the smaller variance leading to a better noise filtering. Data selection with $\mathbf{S}_1$ gives a larger variance but still better than the one obtained with the random selection. Data selection with $\mathbf{S}_{12}$ gives an intermediate result between the data selection with $\mathbf{S}_1$ and $\mathbf{S}_2$. Overall, data selection with respect to sensitivity fields greatly improves the Bayesian inference solution compared with a random selection. Cross-sensitivity can be a good intermediate to select data for the identification of several parameters. In the considered example, the data selection with sensitivity $\mathbf{S}_{12}$ is equivalent to the selection of half the data with $\mathbf{S}_1$ and half the data with $\mathbf{S}_2$.

Figure 6.4 shows the comparison of the marginals obtained with different data selections in the case of sequential inference. Marginals in Figures 6.4(a) and 6.4(b) are computed with the random data selection. Marginals in Figures 6.4(c) and 6.4(d) are computed with data selection with respect to sensitivity $\mathbf{S}_{12}$. With sequential inference, the data selection



(a) Posterior marginals on parameter Y_0

(b) Posterior marginals on parameter A_d

Figure 6.3: Marginals at time step 45 for the different choices of measured degrees of freedom

gives a smoother evolution of the marginals and smaller variances especially in the first time steps where the signal-to-noise ratio is the lowest.

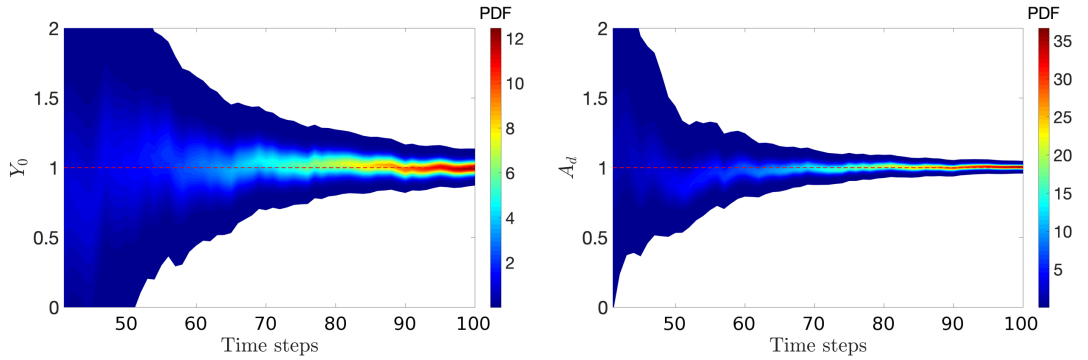
6.2.3 Application to DIC data selection

The selection of data with respect to model sensitivity is useful when the measurements are subjected to the same spatial distribution of noise. Here, simulations are performed with a white Gaussian noise with zero mean and standard deviation of 10^{-6}. However, matrix $\mathbb{M}$ in DIC highlights spatial correlation due to pictures texture and finite element basis. Then, the sensitivity fields for parameter identification should be weighted by the measurement sensitivity which represents the signal-to-noise ratio. In global DIC, this information is available a priori and given by the correlation matrix $\mathbb{M}$ of (6.9). Figure 6.5 represents the diagonal of matrix $\mathbb{M}$ which is the sensitivity in x and y of all nodes of the DIC mesh. The picture used to compute this matrix is one of the experimental campaigns presented in Chapter 4. The FE mesh used to represent the measured displacement is quite fine and subjected to high measurement uncertainties.

In order to define sensitivity with respect to model identification and measurement uncertainties, the matrix $\mathbb{M}$ is multiplied by the sensitivity $\mathbf{S}_1$, $\mathbf{S}_2$ and $\mathbf{S}_{12}$:

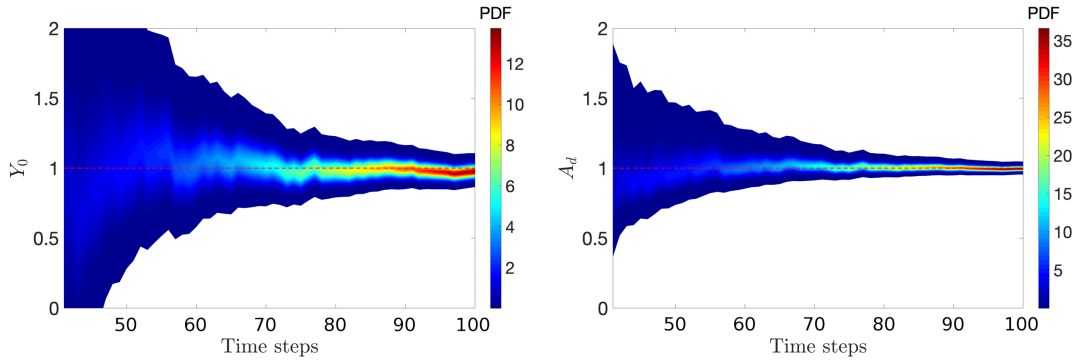
$$\mathbf{S}_i^{\mathrm{DIC}} = \mathbb{M}\mathbf{S}_i \quad i \in \{1, 2, 12\} \tag{6.17}$$

That way the measurement uncertainties are propagated to the sensitivity fields. In order to compute sensitivity fields $\mathbf{S}_i^{\mathrm{DIC}}$, the original sensitivity fields $\mathbf{S}_i$ are projected in the DIC mesh then the product of $\mathbf{S}_i$ by the correlation matrix $\mathbb{M}$ is performed. Figure 6.5 shows the sensitivity field $\mathbf{S}_{12}^{\mathrm{DIC}}$. The most "highlighted" nodes will be the measurements points with the highest sensitivity with respect to the identification of parameters under measurement uncertainties. Finally, the most sensitive degrees of freedom can be selected in order to do the identification procedure. Figure 6.7 shows the 30 most sensitive degrees of freedom. These degrees of freedom are located on the right side of the notch which is the area with the larger horizontal displacement. The displacement is not symmetric due



(a) Posterior marginals for Y_0 with random data selection

(b) Posterior marginals for A_d with random data selection



(c) Posterior marginals for Y_0 with data selection w.r.t $\mathbf{S}_{12}$

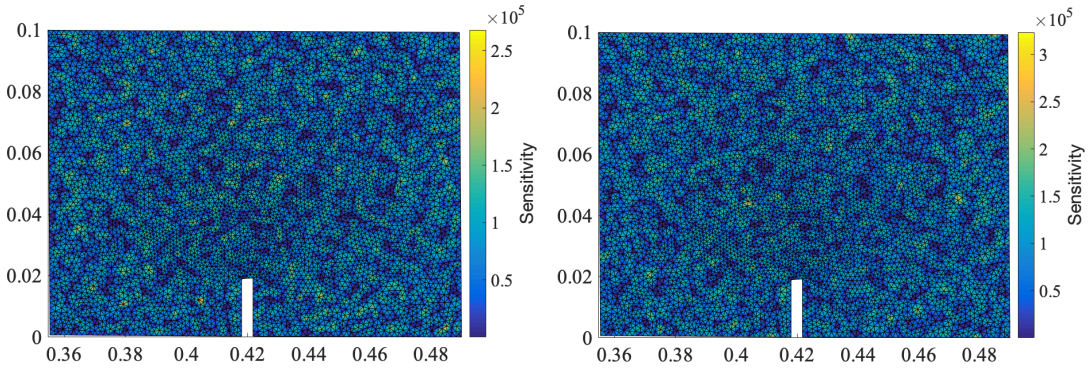
(d) Posterior marginals for A_d with data selection w.r.t $\mathbf{S}_{12}$

Figure 6.4: Marginals in the case of sequential inference for different choices of measured degrees of freedom

to the fact that the horizontal displacement on the left side of the beam is prescribed to zero (see Figure 4.9).

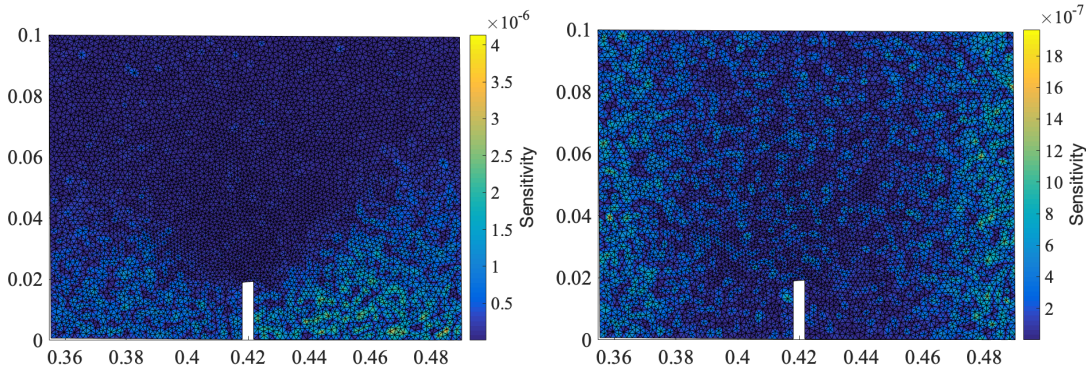
Once the best data are selected, the covariance matrix of measurement error can be computed thanks to (6.11). First, the standard deviation γ_f is computed by comparison of two reference pictures. The difference of gray levels is computed pixel by pixel, then the variance of this quantity is computed and gives $2\gamma_f^2$. The factor 2 is due to the fact that both reference pictures are assumed to be affected by the same noise. Figure 6.8 shows the histogram of the gray levels difference between two reference pictures of the considered experimental test. We can see that the noise is approximately Gaussian with zero mean. Here the standard deviation is about 2.3/255 (gray level). Then the covariance measurement error matrix restricted to the 30 chosen degrees of freedom is represented in Figure 6.9.

We observe that the covariance matrix is approximately diagonal due to the small size of the finite element mesh and the linear shape functions used. Indeed, in this case only nodes sharing the same patch are correlated. For the selected points, the mean standard deviation of noise is about 5×10^{-7} m. This precision is quite high and justifies the level of noise used to simulate the measurement with the numerical model and a standard deviation of 10^{-6} m.



(a) Sensitivity along the x direction (b) Sensitivity along the y direction

Figure 6.5: Sensitivity of the displacements at each degree of freedom independently of the displacement of the other degrees of freedom (diagonal of the sensitivity matrix $\mathbb{M}$)



(a) Sensitivity along the x direction (b) Sensitivity along the y direction

Figure 6.6: Sensitivity to measurement and model identification S_{12}^{DIC}

6.3 Speckle optimization for DIC with PGD models

This section presents an on-going joint work with Dr. Clément Jailin, post-doctoral student at LMT, about speckle optimization for parameter identification. This is an additional way to reduce the amount of information necessary to effectively update model parameters in the Bayesian inference context.

6.3.1 Definition of the problem

The prior knowledge of the parametric PGD model allows to identify the most relevant DIC data for the identification of parameters. This combination of techniques is even more interesting as DIC offers the possibility to compute the sensitivity matrix *a priori* knowing the reference image (the speckle pattern) and the basis over the displacement field. However, additional preliminary work can be done in order to optimize the speckle pattern of the reference picture with respect to the identification of parameters.

The idea is to maximize the correlation matrix $\mathbb{M}$ and therefore the scalar product $\nabla f(\mathbf{x}).\psi(\mathbf{x})$ with respect to the image f. The basis vector $\psi(\mathbf{x})$ is here chosen as the

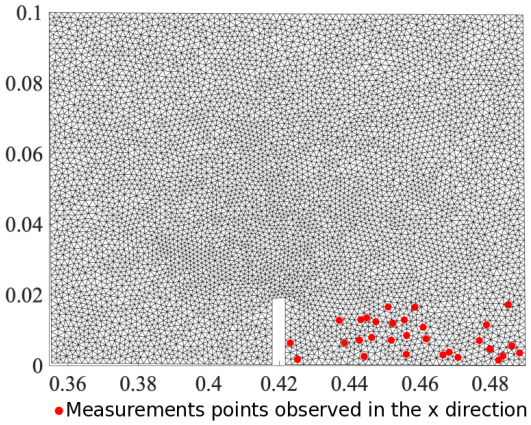


Figure 6.7: Selection of the 30 most sensitive degrees of freedom w.r.t to measurement and model identification

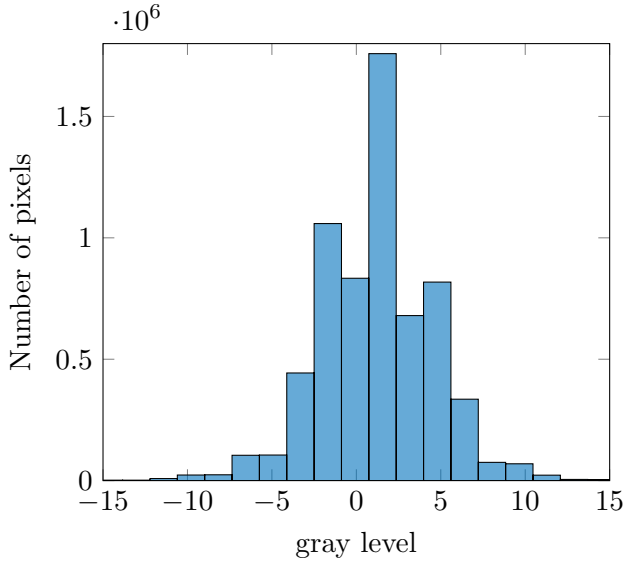


Figure 6.8: Histogram of the gray level difference between two reference pictures

sensitivity field $\mathbf{S}(\mathbf{x}) = \dfrac{\partial \mathbf{u}}{\partial p}(\mathbf{x}, \bar{p})$ in order to maximize the sensitivity of the measure with respect to the variation of the parameter p. Then, the functional J_{sens} is defined as follows:

$$J_{\text{sens}}(f) = ||\nabla f(\mathbf{x}).\mathbf{S}(\mathbf{x})||_{\text{ROI}} \tag{6.18}$$

The functional J_{sens} needs to be maximized and this maximum is given when the norm of the picture gradient is large and collinear to the sensitivity field ψ.

Another constraint of the DIC, especially in the real-time simulation context, is the computation time required to solve the minimization problem (6.8). This problem is solved thanks to a Newton-Raphson based on the linear approximation (6.6) $g(\mathbf{x} + \mathrm{d}\mathbf{u}(\mathbf{x})) \approx g(\mathbf{x}) + \mathrm{d}\mathbf{u}(\mathbf{x}).\nabla g(\mathbf{x})$. Thus, the more accurate this approximation is, the fewer iterations will be needed to solve the problem. In this case, the functional J_{lin} to minimize reads:

$$J_{\text{lin}}(f) = ||f(\mathbf{x} + \mathbf{S}(\mathbf{x})) - f(\mathbf{x}) - \mathbf{S}(\mathbf{x}).\nabla f(\mathbf{x})||_{\text{ROI}} \tag{6.19}$$

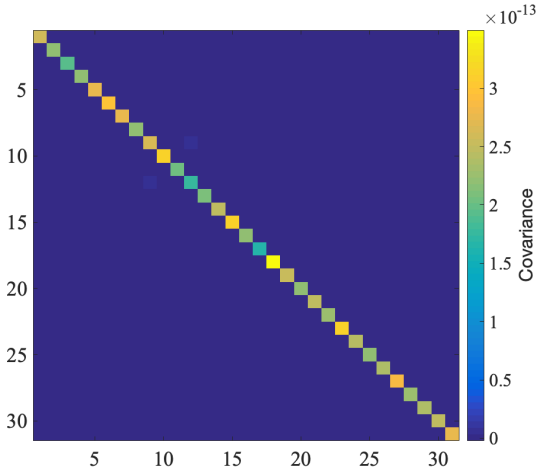


Figure 6.9: Covariance matrix of measurement noise for the 30 selected degrees of freedom

Simultaneous maximization of J_{sens} and minimization of J_{lin} seems to have opposite effects. A legitimate choice of functional is to consider a weighted functional J_α:

$$J_\alpha = (1-\alpha)J_{\text{lin}} - \alpha J_{\text{sens}} \tag{6.20}$$

That way, $\alpha = 0$ gives a functional to minimize in order to have fast DIC computations, and $\alpha = 1$ a functional to minimize in order to have high sensitivity of the measure of the data with respect to the variation of the model parameter.

6.3.2 Application to a 1D picture

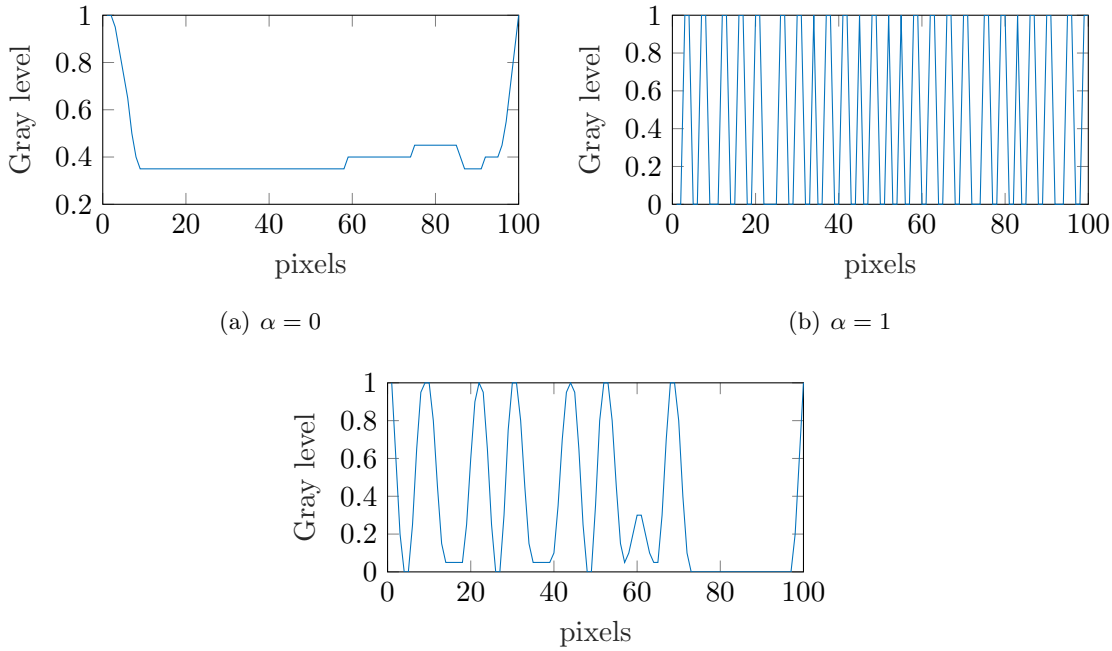
The first application is the optimization of a 1D picture with respect to a uniform sensitivity field ($\mathbf{S}(\mathbf{x}) = \mathbf{1}$, $\forall \mathbf{x} \in$ ROI). This can occur for example when the parameter of interest is the Young modulus of a steel plate under tension. In order to minimize J_α, an iterative algorithm is used. The idea is that at each iteration of the algorithm, the best location to add an unitary increment of gray level is found. In order to do so, N_r pixel candidates $\{\mathbf{x}_i\}_{i\in\{1,..,N_r\}}$ are drawn randomly in the picture. Then N_r picture candidates f_i are built by incrementing the gray level of the pixel $\mathbf{x}_i$ by an unitary gray level:

$$f_i(\mathbf{x}) = f(\mathbf{x}) \quad \forall \mathbf{x}\{\mathbf{x}_1, ..., \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, ..., \mathbf{x}_{N_p}\} \tag{6.21}$$

$$f_i(\mathbf{x}_i) = f(\mathbf{x}_i) + \delta w \tag{6.22}$$

where δw is the unitary increment of gray level. Then the functional J_α is computed for all picture candidates f_i and the picture that gives the lowest value of the functional is kept. In practice, the initial picture is uniformly black and the algorithm finds the best location of the drop of white paint (quantity δw) that minimizes J_α. The algorithm stops when a given quantity of paint is used.

In Figure 6.10, the result of the gray level of the final picture after 1000 iterations is given with different values of α. The number of tested candidates at each iteration is 30 and the unitary gray level increment δw is equal to 0.05. Gray level values vary from 0 (black) to 1 (white). When the gray level of the best candidate is already equal to 1, the gray level stays at 1 (the algorithm stagnates). Figure 6.10(a) shows the result of the 1D



(a) $\alpha = 0$

(b) $\alpha = 1$

(c) $\alpha = 0.5$

Figure 6.10: Optimized pictures for different values of α

picture with $\alpha = 0$. In this case the gray level is quite uniform, which correctly satisfies the linearity condition as $f(\mathbf{x} + \mathbf{S}(\mathbf{x})) - f(\mathbf{x}) - \mathbf{S}(\mathbf{x}).\nabla f(\mathbf{x})$ if f is uniform. White pixels at the end of the picture are due to the numerical computation of the image gradient ∇f. The opposite case with $\alpha = 1$ is shown in Figure 6.10(c). In this case, the picture has a strong gradient with a periodic pattern with the alternation of clear white and black zones (chessboard pattern). Figure 6.10(b) shows the picture result computed with $\alpha = 0.5$. In this case, the patterns are coarser with a smooth variation between white and black zones. This kind of multiscale pattern seems to be the most efficient as the fine details help the sensitivity of the measure and coarser patterns help the convergence of the DIC.

6.3.3 Application to a 2D picture

This methodology is now applied to the identification of the damage model parameters. The sensitivity field used is the sensitivity field $\mathbf{S}_1$ corresponding to parameter Y_0. The size of the optimized pictures is 150×150 pixels that represents a square of 0.1×0.1 m at the central part of the beam. In the optimization algorithm, the number of candidates drawn at each iteration is equal to 150 and the number of total iterations is equal to $69,000$ with an increment of gray level δw of 0.1. This choice of total number of iterations is made by comparison with the speckle used in the experimental settings. With $69,000$ iterations, and with an increment of 0.1, the computed pictures will have the same mean gray level than real pictures taken during the experimental campaign.

Figure 6.11 shows the result of the pictures computed with different values of α. For small values of α, the picture highlights a white pattern where the sensitivity is null (see Figure 6.1(a) and Figure 6.1(b)). For $\alpha = 1$, the picture is covered with a chessboard

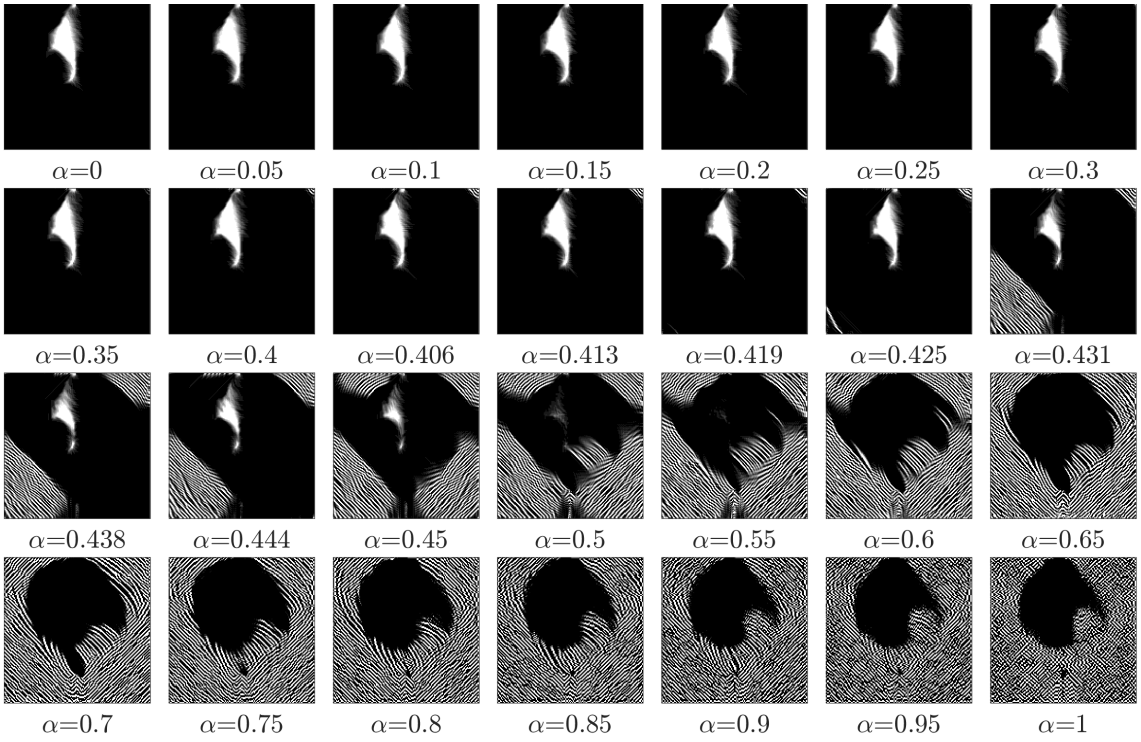


Figure 6.11: Optimized pictures w.r.t $\mathbf{S}_1$ for different values of α

pattern except in the top-central zone where the sensitivity is null. For the intermediate values of α, patterns highlight straight lines orthogonal to the displacement field.

In order to define the optimal value of the parameter α, all pictures are represented in the diagram $(J_{\text{sens}}, J_{\text{lin}})$. The optimal value of α is the one that gives a minimum value of J_{lin} and a maximum of J_{sens} (bottom right corner). In this representation, it seems that the optimal value of α is 0.5. As a comparison, the points that represent random pictures and an example of picture used for the experimental campaign are plotted. Random pictures are computed with gray levels drawn according to the random uniform distribution. Then, the point that represents the random speckle in the diagram is the mean point for 10000 random generated pictures. An example of random picture is shown in Figure 6.13(c). In order to compute the real-test point, the picture of a real-test speckle in a square of 0.1×0.1 m is taken (Figure 6.13(a)) then the resolution is lowered to 150×150 pixels (Figure 6.13(b)) in order to compute the value of $(J_{\text{sens}}, J_{\text{lin}})$. Both random and real-test speckles seem to be much less optimal than the computed picture for $\alpha = 0.5$. Here the method is illustrated by considering the sensitivity $\mathbf{S}_1$. In order to have optimality with respect to both parameters, the sensitivity field $\mathbf{S}_{12}$ can be used instead.

6.4 Conclusions

In this chapter, the focus was made on the prior work that could be done to optimize data assimilation when a large amount of data is available (such as when using DIC or DVC). This optimization is performed thanks to multi-parametric PGD models that are computed prior to the Bayesian inference procedure. Thus, sensitivity fields can be easily computed

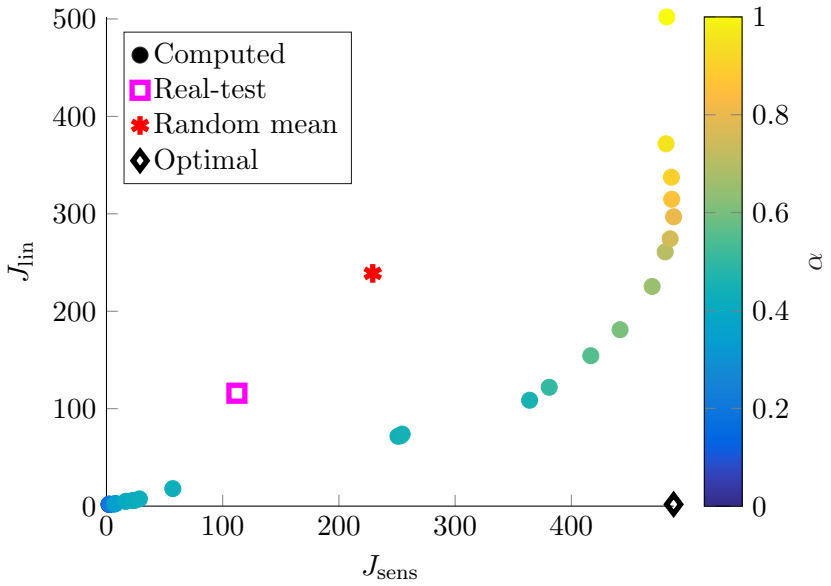
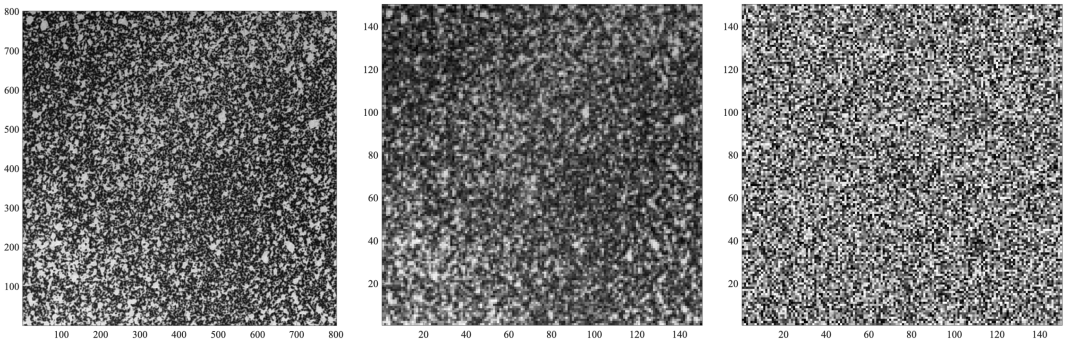


Figure 6.12: Diagram $(J_{\text{sens}}, J_{\text{lin}})$ for the speckle optimization



(a) Real-test speckle

(b) Real-test speckle with resolution 150×150

(c) Random speckle

Figure 6.13: Comparison of speckles

and the most sensitive spatial coordinates can be selected. Reducing the number of degrees of freedom is particularly useful in identification procedure such as Bayesian inference in order to save computation cost. It also improves the robustness of the identification as measurements with a low signal-to-noise ratio will be discarded. However, the measurement uncertainties may be not uniformly distributed and the sensitivity field can be perturbed by the spatial distribution of noise. Thanks to the global DIC method, this spatial distribution of noise can be determined with two reference pictures and the data selection can be done with respect to parameter variation and measurement noise. Pictures themselves can be optimized by building a speckle pattern sensitive with respect to the parameters variation thanks to sensitivity fields computed with PGD modes. In practical terms, this speckle can be printed into the studied specimen using a stencil cut according to the computed speckle and paint spraying or by following the technique described in [26].

General conclusions and future works

Introducing a dialog between numerical models and *in-situ* measurement seems to be a path of the future to precisely predict behaviors of connected physical systems. It is especially the case in the DDDAS framework where the numerical model is used to control the physical system in real-time. However, coupling data and models is a challenging task as the updating of models with indirect corrupted data requires a special computation care. Indeed, finding useful characterization of model parameters from noisy measurements leads to inverse problems that often require regularization procedures and costly numerical techniques. As the result, this thesis proposed innovative computational tools in order to address real-time data assimilation problems.

Among all the inversion methods, the choice in this thesis was to used Bayesian inference. This framework has two main advantages. First, the Bayesian formulation of inverse problem is quite convenient as it leads to an automatic regularized problem. Second, it is the ideal framework to take into account all the uncertainties surrounding the studied physical system. Thus, uncertainties can be propagated to the estimation of the model parameters without assumptions. With this method, the prediction of quantities of interest is performed in an exact manner as random variables. However, the continuous dialog between model and *in-situ* measurements require that the Bayesian updating can be done according to the characteristic time of the physical system. In order to so, the PGD technique was used to produce a multi-parametric model that can be evaluated in the online phase with quasi analytical computations. This ROM technique was chosen because it can be applied to all kinds of parameters, and the online computations are quite inexpensive. However, at this stage, the Bayesian-PGD can not be envisioned to deal with real-time sequential data assimilation. Indeed, the exploration of the posterior density which is the result of Bayesian inference is still very costly using the classical MCMC-based algorithms. Thus, to address this issue, the Bayesian-PGD framework was enriched with the Transport Map method. This method allows to build a deterministic coupling (map) between a reference density and the target posterior density. The method is particularly adapted to sequential inference as the maps are computed by iterative compositions. Hence, this thesis work highlighted the performance of such an approach to deal with sequential updating of numerical models and the associated computations difficulties.

Figure 6.14 shows a schematic representation of the computations tasks associated to the sequential model updating of a numerical model to predict and/or control a physical system with the approach aforementioned. Those tasks are divided into two categories:

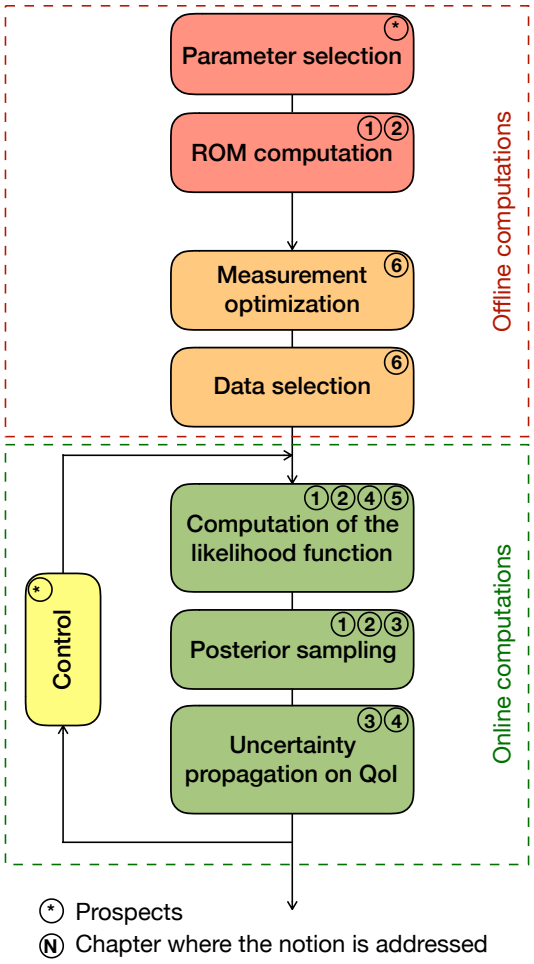


Figure 6.14: Summary of the computation tasks associated to the DDDAS framework

the offline and online computations. The offline computations are all pre-computations that can be done without real-time constraints. Conversely, the online computations deal with the computations that are done in real-time when data are assimilated while the physical system is evolving. In the offline computations, the first step is to define the model and the most relevant parameters that will be identified with the data assimilation. In this work, the choice of relevant parameters was arbitrarily made. In practice, this choice may be guided by expert knowledge or by a proper sensitivity analysis. Once the parameters of interest are identified, the PGD model was built with a time, space, and parameter dependency. The building of PGD reduced order models was addressed in Chapter 1 where the classical methodology was presented. This methodology was applied in Chapter 2 in order to model a welding process. It is worth noting that these the tasks of parameter selection and ROM computation can be coupled with the approach of manifold learning presented in [52]. In this latter paper, most relevant geometrical parameters of human livers are computed thanks to a manifold learning technique (the kPCA). Then, the multi-parametric mechanical response of the livers depending on the identified geometrical parameters is computed with PGD.

Another task that can be addressed in the offline phase is the optimization of the

measurements with respect to the Bayesian identification. This may be very useful when a large amount of data is available. It is the case when using the Digital Image Correlation. This method allows to have a large amount of data that can be costly to assimilate. In Chapter 6, this problematic was addressed by using the sensitivity fields that can be computed *a priori* with DIC and PGD models. The measurement errors being known at the beginning of the DIC computation and constant over the time, this allows to select the best sensitive degrees of freedom to use in the identification process. In this chapter, it was shown that using optimally selected measurements improves the accuracy of the Bayesian inference. Furthermore, it was also shown that the sensitivity of the measures with respect to the identification of the parameters can be directly improved by the optimization of the speckle pattern required for DIC measurements. To that end, an optimized functional was proposed in order to perform the most sensitive and fastest measurements as possible.

Concerning the online phase, the core of the Bayesian inference is to build the likelihood function. While in Chapter 1 the classical expression of the likelihood function was developed, Chapter 2 showed that the use of the PGD allows to have in this case an explicit formulation of the likelihood function. In this chapter, a formulation of the likelihood function was also developed when stochastic parameters are considered. In this case, the PGD model is computed with respect of those parameters. Hence, the computation was performed with uncertainty propagation thanks to Monte-Carlo simulations or in the one dimensional case analytically. This permitted fast evaluations of the posterior density in the presence of different sources of uncertainty. In Chapter 4, a particular case of Bayesian inference was defined in order to couple two different models: the damage PGD model and a SVD meta-model of crack opening. In this context the likelihood function is non-Gaussian and computed with uncertainty propagation. Finally, Chapter 5 showed Bayesian computations with the enrichment by means of a data-based model correction. In this context, the computation of the likelihood function was performed by considering a biased Gaussian error that improves the model updating process. Once the likelihood function is computed, the next step is to sample the posterior density in order to derive useful characterizations. When the problem of interest involves only one parameter with no sequential update, the posterior density can be explored on a regular grid (as in Chapter 2). However, when dealing with the identification of many parameters, the integration with respect to the posterior density become intractable with classical quadrature. That is why in Chapter 1 the MCMC method was presented as the reference method to sample the posterior density. Although this method allows to perform Monte-Carlo integration, it suffers from many drawbacks. Indeed, many parameters have to set *a priori* and the convergence can be difficult to estimate. This leads to high computation costs. In order to address those issues, the promising Transport Map method was introduced in Chapter 3. This method proved to be very efficient and cheap especially owing to its coupling with the PGD which enables to speed-up the computation of the different optimizations required to compute the transport maps. This allows to condense the Bayesian inference solution into a deterministic application (map) that pushes the reference density to the posterior density. As a result, with this method, uncertainty propagation on quantities of interest can be easily computed. This was shown in Chapter 3 where the uncertainties on the model parameters were propagated spatially to a temperature point which is not reachable by direct measurement and temporally by predicting this temperature for all the

time steps. In Chapter 4, the uncertainty propagation was performed on the displacement field in order to compute and predict the crack length in the studied concrete beams.

Finally, the last computation that can be done in the online phase is the control of the system according to prediction of the quantities of interest aforementioned. This step requires a specific and has not been addressed in this work.

Overall, using PGD model reduction and Transport Map sampling significantly reduced the computation cost of the dynamical Bayesian model updating under uncertainties. In addition to all the advantages of using a PGD model already mentioned, one advantage of the PGD formulation is that automatic Bayesian procedure can be envisioned. Indeed, thanks to the modal representation, all the information concerning the model and the physics is condensed inside modes. Therefore, the Bayesian algorithm only requires entries on modes, their types (space, time or parameters), the observations and the spatial coordinates concerned by the observation (equivalent to defining an observation operator). Thus, once the general algorithm is computed, it is very easy to adapt it to each applications. This was the case in this thesis as the main algorithm was used for each numerical example, from a linear thermal model to a non-linear damage model. Although the online use of the PGD model is very convenient, the computation of the PGD model itself can be difficult due to fact that it is directly computed from a PDE formulation. Thus, commercial codes are hardly usable to build PGD models in the classical progressive Galerkin approach. However, some promising developments [52, 152, 144] could be used to build non-intrusive PGD algorithms and alleviate this difficulty. Another limitation can also be the number of parameters that can be taken into account as extra coordinates. Here also recent works [79, 112] could be interesting ideas to use PGD models to identify material property fields.

The presented work has also many other prospects that can be split according to their estimated complexity:

- The short term prospects deal with some further developments in the studied examples. In Chapter 4, the prediction of crack length in the concrete beam was identified thanks to an elastic model of the beam with the inclusion of a linear crack. However, this is a strong approximation and more complex geometries of the cracks should be used. An appealing idea would be to use manifold learning on picture cracks and derive the inherent parameters of the trajectory of the beam in concrete media. The principle is to follow the method described in [52]. From cracks pictures, the purpose is to discretize cracks and find the transformation between a reference trajectory and the other trajectories of the data set. In order to build a large data set, a large number of experiments has to be available which is possible as this kind of test is quite common. Then, an automatic procedure has to be elaborated in order to identify the crack trajectories and the kinematic transformation. This can for example be done by post-processing the DIC residuals of the experimental tests. Another development that could be envisioned in the same context is the speckle optimization mentioned in Chapter 6. Improvements can be done in the formulation of the objective function and its minimization. Furthermore, an experimental campaign has to be set to validate the approach.

- The middle term prospects are more focused on the improvement of the sequential

computation of the transport maps. Indeed, in the presented work transport maps are computed by iterating on the order until the variance diagnostic is satisfied. However, this approach can lead to the computation of full high-order maps which can be very costly. Associated to the previous one, another issue is the computation time scalability of the presented procedure with respect to the number of parameters. Indeed, the computation of full order maps is intractable when the parametric dimension increases. A way to circumvent this issue would be to use a composition of low-order maps in order to recover the possible sparse structure of target densities. One way to do that is to follow the ideas presented in [44]. In this paper, the composition of maps is performed by finding a sequence of maps between the reference density and target densities that smoothly moves from the reference to the target density. The purpose is that the maps between two close densities can be represented with low-order polynomials. This property can also be observed in the numerical examples treated in this thesis. In order to build this sequence of intermediate target densities, data can be assimilated sequentially even when they are available simultaneously. In this case an approach such as the one presented in Chapter 6, (where the influence of the measurements on the Bayesian solution is controlled with selection of the most relevant data) could be followed. This can be done when many measurements are obtained simultaneously. Another way to construct the sequence of intermediate densities is to build the posterior densities with increasing accuracy of the forward model. This can be well suited with the use of PGD models as the accuracy of the forward model can simply be controlled with the number of modes used. Otherwise, an interesting recent method is [14] where the transport maps are computed with the sequential computation of "lazy maps". Another way to build low-order maps consists of studying the link between the sparse structure of the forward PGD model and the sparse structure of the corresponding posterior densities.

- In the longer term, the approach developed in this thesis could be adapted and improved to deal with real-life engineering applications in the context of DDDAS. Indeed, the approach presented here is restricted to the prediction of quantities of interest. Thus, a future work will be to introduce a control loop between the quantities of interest and some entry parameters of the studied framework. Here again, the PGD model reduction seems to be well suited due to the diversity of parameter types that can be integrated in the PGD formulation [30, 50]. Furthermore, as in the framework presented in this thesis the quantities of interest are predicted as random variables, it will be interesting to use stochastic control [13] coupled with transport maps and PGD models.

Appendix A

Computation of the PGD model for the welding example

In this appendix, details on the PGD solution are given and illustrated on the welding example of Chapter 2.

A.1 Problem

The same problem as in Section 2.3 is considered with the convection-diffusion equation:

$$\frac{\partial T}{\partial t} + \underline{v}.\underline{\text{grad}}T - \Delta T = s \tag{A.1}$$

with:

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

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

A.2 Progressive Galerkin PGD

As presented in Section 2.2, the PGD modes are built recursively thanks to the Galerkin orthogonality. The spaces of variation of each parameter are defined as follows: $I = [0, T_f]$ is the time interval and $\Sigma = [\sigma_{\min}, \sigma_{\max}]$ is the space of variation of σ. The admissible field spaces are defined:

$$\mathcal{T} = \{T \in H^1(\Omega =]0; 5[\times]0; 1[), T = 0 \text{ on } \Gamma_D\} \tag{A.3}$$

$$\mathcal{I} = \{T, \int_I \|T(x, y, .; \sigma)\|^2_{H^1(\Omega)} < \infty, \forall (x, y, \sigma) \in \Omega \times \Sigma\} \tag{A.4}$$

$$\mathcal{E} = \{T, \int_\Sigma \|T(x, y, t; .)\|^2_{H^1(\Omega)} < \infty, \forall (x, y, t) \in \Omega \times I\} \tag{A.5}$$

The weak formulation of (A.1) on each space reads:

Find $T \in \mathcal{T} \otimes \mathcal{I} \otimes \mathcal{E}$, such that $\forall T^* \in \mathcal{T} \otimes \mathcal{I} \otimes \mathcal{E}$:

$$a(T, T^*) = l(T^*) \tag{A.6}$$

with:

$$a(T, T^*) = \int_{I \times \Sigma \times \Omega} \frac{\partial T}{\partial t}.T^* + \underline{v}.\underline{\text{grad}}T.T^* + \underline{\text{grad}}T.\underline{\text{grad}}T^* dt d\sigma d\Omega \tag{A.7}$$

$$l(T^*) = \int_{I \times \Sigma \times \Omega} s.T^* dt d\sigma d\Omega \tag{A.8}$$

The solution is searched in the separated form:

$$T_m(x, y, t; \sigma) = \sum_{n=1}^{m} \Lambda_n(x, y)\lambda_n(t)\alpha_n(\sigma) \tag{A.9}$$

The $m-1$ first modes are supposed being known, the mode m is searched. Then the solution reads:

$$\{T_m(x, y, t; \sigma)\} = \sum_{n=1}^{m-1} \lambda_n(t)\alpha_n(\sigma)\Lambda_n(x, y) + \lambda(t)\alpha(\sigma)\Lambda(x, y) \tag{A.10}$$

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:

$$T^* = \lambda^*\alpha\Lambda + \lambda\alpha^*\Lambda + \lambda\alpha\Lambda^* \tag{A.11}$$

Using this form, the variational formulation (A.6) leads to coupled problems with the applications S_m, T_m, P_m such that:

$$\Lambda = S_m(\lambda, \alpha) \tag{A.12}$$
$$\lambda = T_m(\alpha, \Lambda) \tag{A.13}$$
$$\alpha = P_m(\lambda, \Lambda) \tag{A.14}$$

A.2.1 Spatial application S_m

The weak formulation for the space problem reads:

$$a\left(T_{m-1} + \lambda\alpha\Lambda, \lambda\alpha\Lambda^*\right) = l(\lambda\alpha\Lambda^*) \tag{A.15}$$

with:

$$a\left(\lambda\alpha\Lambda, \lambda\alpha\Lambda^*\right) = \int_{I \times \Sigma \times \Omega} \lambda\dot{\lambda}\alpha^2\Lambda\Lambda^* + \alpha^2\lambda^2\underline{v}.\underline{\text{grad}}\Lambda\Lambda^* + \alpha^2\lambda^2\underline{\text{grad}}\Lambda\underline{\text{grad}}\Lambda^* d\Omega d\sigma dt \tag{A.16}$$

The use of a P1 discretization on all fields reads:

$$\Lambda = [N_x]\{\Lambda\} \tag{A.17}$$
$$\lambda = [N_t]\{\lambda\} \tag{A.18}$$
$$\alpha = [N_\sigma]\{\sigma\} \tag{A.19}$$

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

$$a\left(\lambda\alpha\Lambda, \lambda\alpha\Lambda^*\right) = \{\Lambda^*\}^T[A_\Lambda]\{\Lambda\} \tag{A.20}$$

with:

$$[A_\Lambda] = \left(\{\alpha\}^T[M_\sigma]\{\alpha\}\right).\left[\left(\{\lambda\}^T[H_t]\{\lambda\}\right)[M] + \left(\{\lambda\}^T[M_t]\{\lambda\}\right)[C_H]\right] \tag{A.21}$$

where the following matrices are defined:

$$[M_\bullet] = \int_\bullet [N_\bullet]^T[N_\bullet]d\bullet \tag{A.22}$$

$$[H_\bullet] = \int_\bullet [N_\bullet]^T[N_\bullet]d\bullet \tag{A.23}$$

$$[C_H] = \kappa.[M_x] + [H_x] \tag{A.24}$$

Likewise:

$$a\left(T_{m-1}, \lambda\alpha\Lambda^*\right) = \{\Lambda^*\}^T \sum_{n=1}^{m-1} [A_{\Lambda_n}]\{\Lambda_n\} \tag{A.25}$$

with:

$$[A_{\Lambda_n}] = \left(\{\alpha\}^T[M_\sigma]\{\alpha_n\}\right).\left[\left(\{\lambda\}^T[H_t]\{\lambda_n\}\right)[M] + \left(\{\lambda\}^T[M_t]\{\lambda_n\}\right)[C_H]\right] \tag{A.26}$$

$$l(\{\lambda\alpha\Lambda^*\}) = \int_{I\times\Sigma\times\Omega} \lambda(t)\alpha(\sigma)s(x,y;\sigma).\Lambda^*(x,y)dtd\sigma d\Omega \tag{A.27}$$

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

$$l(\lambda\alpha\Lambda^*) = \{\Lambda^*\}^T\{Q_\Lambda\} \tag{A.28}$$

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 order r is used.

The volumic load s is approximated as:

$$s(x,y;\sigma_0+\delta\sigma) \approx \sum_{i=0}^{r} \frac{\partial^i s}{\partial\sigma^i}(x,y;\sigma_0)\frac{\delta\sigma^i}{i!} \tag{A.29}$$

which leads to:

$$\int_{\sigma_{\min}}^{\sigma_{\max}} s(x,y;\sigma)\alpha(\sigma)d\sigma = \sum_{k=1}^{N}\int_{\sigma_k}^{\sigma_{k+1}} s(x,y;\sigma)\alpha(\sigma)d\sigma \tag{A.30}$$

$$= \sum_{k=1}^{N}\int_{\sigma_k-\sigma_{0k}}^{\sigma_{k+1}-\sigma_{0k}} s(x,y;\sigma_{0k}+\tilde{\sigma})\alpha(\sigma_{0k}+\tilde{\sigma})d\tilde{\sigma} \tag{A.31}$$

$$= \sum_{k=1}^{N}\sum_{i=0}^{r}\int_{\sigma_k-\sigma_{0k}}^{\sigma_{k+1}-\sigma_{0k}} \frac{\tilde{\sigma}^i}{i!}\alpha(\sigma_{0k}+\tilde{\sigma})d\tilde{\sigma}\frac{\partial^i s}{\partial\sigma^i}(x,y;\sigma_{0k}) \tag{A.32}$$

$$= \sum_{k=1}^{N}\sum_{i=0}^{r}\int_{\sigma_k}^{\sigma_{k+1}} \frac{(\sigma-\sigma_{0k})^i}{i!}\alpha(\sigma)d\sigma\frac{\partial^i s}{\partial\sigma^i}(x,y;\sigma_{0k}) \tag{A.33}$$

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

The right-hand side reads now:

$$l(\lambda\alpha\Lambda^*) = \{1_t\}^T[M_t]\{\lambda\}.\sum_{k=1}^{N}\sum_{i=0}^{r}\int_{\sigma_k}^{\sigma_{k+1}}\frac{(\sigma-\sigma_{0k})^i}{i!}\alpha(\sigma)d\sigma\{\Lambda^*\}^T\{S_{ik}\} \tag{A.34}$$

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

A.2.2 Time application 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 on the unknown λ:

$$a.\dot{\lambda}(t) + b.\lambda(t) = c - \sum_{n}^{m-1} a_n.\dot{\lambda_n}(t) + b_n.\lambda_n(t) \tag{A.35}$$

with:

$$a = \int_{\Sigma\times\Omega}\alpha^2\Lambda^2 d\sigma d\Omega = \left(\{\alpha\}^T[M_\sigma]\{\alpha\}\right)\left(\{\Lambda\}^T[M]\{\Lambda\}\right) \tag{A.36}$$

$$b = \int_{\Sigma\times\Omega}\alpha^2\left(\underline{v}.\underline{\text{grad}}\Lambda\Lambda + \underline{\text{grad}}\Lambda\underline{\text{grad}}\Lambda\right)d\sigma d\Omega = \left(\{\alpha\}^T[M_\sigma]\{\alpha\}\right)\left(\{\Lambda\}^T[C_H]\{\Lambda\}\right) \tag{A.37}$$

$$c = \int_{\Sigma\times\Omega}\alpha(\sigma)s(x,y;\sigma).\Lambda(x,y)dtd\sigma d\Omega = \sum_{k=1}^{N}\sum_{i=0}^{r}\int_{\sigma_k}^{\sigma_{k+1}}\frac{(\sigma-\sigma_{0k})^i}{i!}\alpha(\sigma)d\sigma\{\Lambda\}^T\{S_{ik}\} \tag{A.38}$$

$$a_n = \int_{\Sigma\times\Omega}\alpha\alpha_n\Lambda\Lambda_n d\sigma d\Omega = \left(\{\alpha\}^T[M_\sigma]\{\alpha_n\}\right)\left(\{\Lambda\}^T[M]\{\Lambda_n\}\right) \tag{A.39}$$

$$b_n = \int_{\Sigma\times\Omega}\alpha\alpha_n\left(\underline{v}.\underline{\text{grad}}\Lambda\Lambda_n + \underline{\text{grad}}\Lambda\underline{\text{grad}}\Lambda_n\right)d\sigma d\Omega = \left(\{\alpha\}^T[M_\sigma]\{\alpha_n\}\right)\left(\{\Lambda\}^T[C_H]\{\Lambda_n\}\right) \tag{A.40}$$

A.2.3 Parametric application P_m

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

$$a\left(T_{m-1} + \lambda\alpha\Lambda, \lambda\alpha^*\Lambda\right) = l(\lambda\alpha^*\Lambda) \tag{A.41}$$

$$a\left(\lambda\alpha\Lambda, \lambda\alpha^*\Lambda\right) = \int_{I\times\Sigma\times\Omega}\lambda\dot{\lambda}\alpha\alpha^*\Lambda^2 + \alpha\alpha^*\lambda^2\underline{v}.\underline{\text{grad}}\Lambda\Lambda + \alpha\alpha^*\lambda^2\underline{\text{grad}}\Lambda\underline{\text{grad}}\Lambda dtd\sigma d\Omega \tag{A.42}$$

A P1 discretization leads to:

$$a\left(\lambda\alpha\Lambda, \lambda\alpha^*\Lambda\right) = \{\alpha^*\}^T[A_\alpha]\{\alpha\} \tag{A.43}$$

with:

$$[A_\alpha] = [\left(\{\Lambda\}^T[M]\{\Lambda\}\right)\left(\{\lambda\}^T[H_t]\{\lambda\}\right) + \left(\{\Lambda\}^T[C_H]\{\Lambda\}\right)\left(\{\lambda\}^T[M_t]\{\lambda\}\right)][M_\sigma] \quad \text{(A.44)}$$

The contribution of the previous modes reads:

$$a\left(T_{m-1}, \lambda\alpha^*\Lambda\right) = \{\alpha^*\}^T \sum_{n=1}^{m-1} [A_{\alpha_n}]\{\alpha_n\} \quad \text{(A.45)}$$

with:

$$[A_{\alpha_n}] = [\left(\{\Lambda\}^T[M]\{\Lambda_n\}\right)\left(\{\lambda\}^T[H_t]\{\lambda_n\}\right) + \left(\{\Lambda\}^T[C_H]\{\Lambda_n\}\right)\left(\{\lambda\}^T[M_t]\{\lambda_n\}\right)][M_\sigma] \quad \text{(A.46)}$$

Finally the right-hand side reads:

$$l(\lambda\alpha^*\Lambda) = \{1_t\}^T[M_t]\{\lambda\}.\sum_{k=1}^{N}\sum_{i=0}^{r}\int_{\sigma_k}^{\sigma_{k+1}} \frac{(\sigma-\sigma_{0k})^i}{i!}\alpha^*(\sigma)d\sigma\{\Lambda\}^T\{S_{ik}\} \quad \text{(A.47)}$$

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

Bibliography

[1] A. Alarcon. *A combined Kalman Filter and Error in Constitutive Relation approach for system identification in structural dynamics.* PhD thesis, École Polytechnique, 2012.

[2] P. E. Allier, L. Chamoin, and P. Ladevèze. Proper Generalized Decomposition computational methods on a benchmark problem: introducing a new strategy based on Constitutive Relation Error minimization. *Advanced Modeling and Simulation in Engineering Sciences*, 2(1), 2015.

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

[4] S. Andrieux, A. B. Abda, and H. D. Bui. Reciprocity principle and crack identification. *Inverse Problems*, 15(1):59–65, 1999.

[5] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/nongaussian bayesian tracking. In *Bayesian Bounds for Parameter Estimation and Nonlinear Filtering/Tracking*, pages 723–737. IEEE Transactions on Signal Processing, 2007.

[6] S. Avril, M. Bonnet, A. S. Bretelle, M. Grédiac, F. Hild, P. Ienny, F. Latourte, D. Lemosse, S. Pagano, E. Pagnacco, and F. Pierron. Overview of identification methods of mechanical parameters based on full-field measurements, 2008.

[7] T. Bayes and R. Price. An essay towards solving a problem in the doctrine of chances. *Resonance*, 8(4):80–88, 2003.

[8] J. L. Beck. Bayesian system identification based on probability logic. *Structural Control and Health Monitoring*, 17(7):825–847, 2010.

[9] M. Ben Azzouna, P. Feissel, and P. Villon. Robust identification of elastic properties using the Modified Constitutive Relation Error. *Computer Methods in Applied Mechanics and Engineering*, 295:196–218, oct 2015.

[10] J. Berger, H. R. Orlande, and N. Mendes. Proper Generalized Decomposition model reduction in the Bayesian framework for solving inverse heat transfer problems. *Inverse Problems in Science and Engineering*, 25(2):260–278, 2017.

[11] G. Bergqvist and E. Larsson. The higher-order singular value decomposition: Theory and an application. In *IEEE Signal Processing Magazine*, volume 27, pages 151–154. Institute of Electrical and Electronics Engineers Inc., 2010.

[12] M. Bertero, C. De Mol, and G. A. Viano. The Stability of Inverse Problems. In *Doklady Akademii Nauk Sssr*, pages 161–214. 1980.

[13] G. Bierman. Stochastic models, estimation, and control. *IEEE Transactions on Automatic Control*, 28(8):868–869, 1983.

[14] D. Bigoni, O. Zahm, A. Spantini, and Y. Marzouk. Greedy inference with layers of lazy maps. *arXiv preprint*, arXiv:1906:1–15, may 2019.

[15] M. Bocciarelli, V. Buljak, C. K. Moy, S. P. Ringer, and G. Ranzi. An inverse analysis approach based on a POD direct model for the mechanical characterization of metallic materials. *Computational Materials Science*, 95:302–308, 2014.

[16] M. Bonnet and W. Aquino. Three-dimensional transient elastodynamic inversion using the modified error in constitutive relation. In *Journal of Physics: Conference Series*, volume 542, page 012003, oct 2014.

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

[18] J. M. Brownjohn. Structural health monitoring of civil infrastructure. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2007.

[19] A. Buljac, C. Jailin, A. Mendoza, J. Neggers, T. Taillandier-Thomas, A. Bouterf, B. Smaniotto, F. Hild, and S. Roux. Digital Volume Correlation: Review of Progress and Challenges. *Experimental Mechanics*, 58(5):661–708, 2018.

[20] D. Calvetti, M. Dunlop, E. Somersalo, and A. Stuart. Iterative updating of model error for Bayesian inversion. *Inverse Problems*, 34(2), 2018.

[21] L. G. Chambers and R. Fletcher. Practical Methods of Optimization. *The Mathematical Gazette*, 85(504):562, may 2001.

[22] L. Chamoin, P. E. Allier, and B. Marchand. Synergies between the constitutive relation error concept and PGD model reduction for simplified V&V procedures. *Advanced Modeling and Simulation in Engineering Sciences*, 3(1), 2016.

[23] L. Chamoin, F. Pled, P. E. Allier, and P. Ladevèze. A posteriori error estimation and adaptive strategy for PGD model reduction applied to parametrized linear parabolic problems. *Computer Methods in Applied Mechanics and Engineering*, 327:118–146, dec 2017.

[24] P. E. Charbonnel, P. Ladevèze, F. Louf, and C. Le Noac'h. A robust CRE-based approach for model updating using in situ measurements. *Computers and Structures*, 129:63–73, dec 2013.

[25] H.-P. Chen and Y.-Q. Ni. *Structural Health Monitoring of Large Civil Engineering Structures*. Wiley-Blackwell, 2018.

[26] Z. Chen, C. Quan, F. Zhu, and X. He. A method to transfer speckle patterns for digital image correlation. *Measurement Science and Technology*, 26(9), 2015.

[27] F. Chinesta, E. Cueto, E. Abisset-Chavanne, J. L. Duval, and F. E. Khaldi. Virtual, Digital and Hybrid Twins: A New Paradigm in Data-Based Engineering and Engineered Data. *Archives of Computational Methods in Engineering*, 2018.

[28] F. Chinesta, R. Keunings, and A. Leygue. The proper generalized decomposition for advanced numerical simulations: A primer. In *SpringerBriefs in Applied Sciences and Technology*, number 9783319028644, pages iv–v. Springer, 2014.

[29] F. Chinesta, P. Ladevèze, and E. Cueto. A Short Review on Model Order Reduction Based on Proper Generalized Decomposition. *Archives of Computational Methods in Engineering*, 18(4):395–404, 2011.

[30] F. Chinesta, A. Leygue, F. Bordeu, J. V. Aguado, E. Cueto, D. Gonzalez, I. Alfaro, A. Ammar, and A. Huerta. PGD-Based Computational Vademecum for Efficient Design, Optimization and Control. *Archives of Computational Methods in Engineering*, 20(1):31–59, 2013.

[31] H. Chipman, E. I. George, and R. E. McCulloch. The Practical Implementation of Bayesian Model Selection. In *Model selection*, pages 65–116. Institute of Mathematical Statistics, 2001.

[32] D. Claire, F. Hild, and S. Roux. A finite element formulation to identify damage fields: The equilibrium gap method. *International Journal for Numerical Methods in Engineering*, 61(2):189–208, 2004.

[33] P. Comon, X. Luciani, and A. L. de Almeida. Tensor decompositions, alternating least squares and other tales. *Journal of Chemometrics*, 23(7-8):393–405, 2009.

[34] A. Corigliano and S. Mariani. Parameter identification of a time-dependent elastic-damage interface model for the simulation of debonding in composites. *Composites Science and Technology*, 61(2):191–203, 2001.

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

[36] T. Cui, Y. M. Marzouk, and K. E. Willcox. Data-driven model reduction for the Bayesian solution of inverse problems. *International Journal for Numerical Methods in Engineering*, 102(5):966–990, 2015.

[37] D. N. Daescu and I. M. Navon. Efficiency of a POD-based reduced second-order adjoint model in 4D-Var data assimilation. *International Journal for Numerical Methods in Fluids*, 53(6):985–1004, 2007.

[38] F. Daghia, S. de Miranda, F. Ubertini, and E. Viola. Estimation of elastic constants of thick laminated plates within a Bayesian framework. *Composite Structures*, 80(3):461–473, oct 2007.

[39] F. Darema. Dynamic Data Driven Applications Systems: A New Paradigm for Application Simulations and Measurements. In M. Bubak, G. D. van Albada, P. M. A. Sloot, and J. Dongarra, editors, *Computational Science - ICCS 2004*, pages 662–669, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[40] F. Darema. DDDAS, A Key Driver for Large-Scale-Big-Data and Large-Scale-Big-Computing. *Procedia Computer Science*, 51:2463, 2015.

[41] M. I. Diaz, W. Aquino, and M. Bonnet. A modified error in constitutive equation approach for frequency-domain viscoelasticity imaging using interior data. *Computer Methods in Applied Mechanics and Engineering*, 296:129–149, nov 2015.

[42] M. Ehrhart and W. Lienhart. Monitoring of Civil Engineering Structures using a State-of-the-art Image Assisted Total Station. *Journal of Applied Geodesy*, 9(3):174–182, 2015.

[43] M. Eigel, M. Marschall, and R. Schneider. Sampling-free Bayesian inversion with adaptive hierarchical tensor representations. *Inverse Problems*, 34(3):035010, mar 2018.

[44] T. A. El Moselhy and Y. M. Marzouk. Bayesian inference with optimal maps. *Journal of Computational Physics*, 231(23):7815–7850, oct 2012.

[45] G. Evensen. The Ensemble Kalman Filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, 2003.

[46] F. Fang, C. C. Pain, I. M. Navon, M. D. Piggott, G. J. Gorman, P. E. Farrell, P. A. Allison, and A. J. Goddard. A POD reduced-order 4D-Var adaptive mesh ocean modelling approach. *International Journal for Numerical Methods in Fluids*, 60(7):709–732, 2009.

[47] P. Feissel and O. Allix. Modified constitutive relation error identification strategy for transient dynamics with corrupted data: The elastic case. *Computer Methods in Applied Mechanics and Engineering*, 196(13-16):1968–1983, mar 2007.

[48] R. Ferrier, M. L. Kadri, and P. Gosselet. Planar crack identification in 3D linear elasticity by the reciprocity gap method. *Computer Methods in Applied Mechanics and Engineering*, 355:193–215, 2019.

[49] A. Gelman and D. B. Rubin. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 7(4):457–472, nov 1992.

[50] C. Ghnatios, F. Masson, A. Huerta, A. Leygue, E. Cueto, and F. Chinesta. Proper Generalized Decomposition based dynamic data-driven control of thermal processes. *Computer Methods in Applied Mechanics and Engineering*, 213-216:29–41, 2012.

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

[52] D. González, J. V. Aguado, E. Cueto, E. Abisset-Chavanne, and F. Chinesta. kPCA-Based Parametric Solutions Within the PGD Framework. *Archives of Computational Methods in Engineering*, 2018.

[53] D. González, A. Badías, I. Alfaro, F. Chinesta, and E. Cueto. Model order reduction for real-time data assimilation through Extended Kalman Filters. *Computer Methods in Applied Mechanics and Engineering*, 326:679–693, 2017.

[54] D. González, F. Masson, F. Poulhaon, A. Leygue, E. Cueto, and F. Chinesta. Proper Generalized Decomposition based dynamic data driven inverse identification. *Mathematics and Computers in Simulation*, 82(9):1677–1695, 2012.

[55] M. Grédiac, F. Pierron, S. Avril, and E. Toussaint. The virtual fields method for extracting constitutive parameters from full-field measurements: A review. *Strain*, 42(4):233–253, 2006.

[56] P. L. Green and K. Worden. Bayesian and Markov chain Monte Carlo methods for identifying nonlinear systems in the presence of uncertainty. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 373(2051):20140405–, 2015.

[57] M. A. Grepl. *Reduced-Basis Approximation and A Posteriori Error Estimation for Parabolic Partial Differential Equations*. PhD thesis, Massachusetts Institute of Technology, 2005.

[58] H. Haario, E. Saksman, and J. Tamminen. adaptive Metropolis algorithm. *Bernouilli*, 7(2):223–242, 2014.

[59] P. C. Hansen. The L-Curve and its Use in the Numerical Treatment of Inverse Problems. *in Computational Inverse Problems in Electrocardiology, ed. P. Johnston, Advances in Computational Bioengineering*, 4:119–142, 2000.

[60] F. Hild and S. Roux. Digital image correlation: From displacement measurement to identification of elastic properties - A review, 2006.

[61] A. Hillerborg. The theoretical basis of a method to determine the fracture energy G F of concrete. *Materials and Structures*, 18(4):291–296, 1985.

[62] F. L. Hitchcock. The Expression of a Tensor or a Polyadic as a Sum of Products. *Journal of Mathematics and Physics*, 6(1-4):164–189, 1927.

[63] M. Hoshiya and E. Saito. Structural identification by extended Kalman filter. *Journal of Engineering Mechanics*, 110(12):1757–1770, 1984.

[64] C. Jailin, A. Carpiuc, K. Kazymyrenko, M. Poncelet, H. Leclerc, F. Hild, and S. Roux. Virtual hybrid test control of sinuous crack. *Journal of the Mechanics and Physics of Solids*, 102:239–256, 2017.

[65] H. Jeffreys. *Theory of probability*. Oxford: Clarendon Press, 3rd ed edition, 1961.

[66] H. A. Jensen, C. Vergara, C. Papadimitriou, and E. Millas. The use of updated robust reliability measures in stochastic dynamical systems. *Computer Methods in Applied Mechanics and Engineering*, 267:293–317, 2013.

[67] J. P. Kaipio, E. Somersalo, E. S. S. Antman, J. E. M. L. Sirovich, A. J. K. Hale, P. H. J. Keener, J. K. B. J. Matkowsky, and a. M. C. S. Peskin. *Statistical and Computational Inverse Problems*, volume 160. Springer, 2004.

[68] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering, Transactions of the ASME*, 82(1):35–45, 1960.

[69] N. Kantas, A. Doucet, S. S. Singh, and J. M. MacIejowski. An overview of Sequential Monte Carlo methods for parameter estimation in general state-space models. In *IFAC Proceedings Volumes (IFAC-PapersOnline)*, volume 15, pages 774–785, 2009.

[70] K. T. Kavanagh. Extension of classical experimental techniques for characterizing composite-material behavior - The experimental-analytical method described in this paper is shown to yield material descriptions from specimen shapes previously considered intractable. *Experimental Mechanics*, 12(1):50–56, 1972.

[71] P. Kerfriden, A. Kundu, and S. Claus. Adaptivity in Bayesian inverse finite element problems: Learning and simultaneous control of discretisation and sampling errors. *Materials*, 12(4), 2019.

[72] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *undefined*, 1983.

[73] B. C. Kress. Numerical Optimization. In *Field Guide to Digital Micro-Optics*, page 664. Springer, 2015.

[74] P. Ladevèze. 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.

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

[76] P. Ladevèze and L. Chamoin. The constitutive relation error method: A general verification tool. *SpringerBriefs in Applied Sciences and Technology*, 172:59–94, 2016.

[77] P. Ladevèze and D. Leguillon. Error Estimate Procedure in the Finite Element Method and Applications. *SIAM Journal on Numerical Analysis*, 20(3):485–509, 1983.

[78] P. Ladevèze, D. Nedjar, and M. Reynier. Updating of finite element models using vibration tests. *AIAA Journal*, 32(7):1485–1491, jul 1994.

[79] P. Ladevèze, C. Paillet, and D. Néron. Extended-PGD model reduction for nonlinear solid mechanics problems involving many parameters. In *Computational Methods in Applied Sciences*, volume 46, pages 201–220. Springer, 2018.

[80] P. Ladevèze, G. Puel, A. Deraemaeker, and T. Romeuf. Validation of structural dynamics models containing uncertainties. *Computer Methods in Applied Mechanics and Engineering*, 195(4-6):373–393, 2006.

[81] P. Ladevèze and P. Rougeot. New advances on a posteriori error on constitutive relation in f.e. analysis. *Computer Methods in Applied Mechanics and Engineering*, 150(1-4):239–249, 1997.

[82] T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza. A reduced computational and geometrical framework for inverse problems in hemodynamics. *International Journal for Numerical Methods in Biomedical Engineering*, 29(7):741–776, 2013.

[83] F.-X. Le Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A*, 38 A(2):97–110, 1986.

[84] H. Leclerc, J. Neggers, M. Florent, F. Hild, and S. Roux. Correli 3.0, 2015.

[85] H. Leclerc, J. N. Périé, S. Roux, and F. Hild. Integrated digital image correlation for the identification of mechanical properties. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 5496 LNCS, pages 161–171, 2009.

[86] C. Lieberman, K. Willcox, and O. Ghattas. Parameter and state model reduction for large-scale statistical inverse problems. *SIAM Journal on Scientific Computing*, 32(5):2523–2542, 2010.

[87] X. Liu, P. J. Escamilla-Ambrosio, and N. A. Lieven. Extended Kalman filtering for the detection of damage in linear mechanical structures. *Journal of Sound and Vibration*, 325(4-5):1023–1046, 2009.

[88] M. Loève. *Probability Theory*. 1977.

[89] F. Louf and L. Champaney. Fast validation of stochastic structural models using a PGD reduction scheme. *Finite Elements in Analysis and Design*, 70-71:44–56, 2013.

[90] F. Lu, M. Morzfeld, X. Tu, and A. J. Chorin. Limitations of polynomial chaos expansions in the Bayesian solution of inverse problems. *Journal of Computational Physics*, 282:138–147, 2015.

[91] J. L. Lumley. The structure of inhomogeneous turbulent flows. *Physical Fluid Dynamics*, pages 261–291, 1967.

[92] Y. Maday, A. Manzoni, and A. Quarteroni. An online intrinsic stabilization strategy for the reduced basis approximation of parametrized advection-dominated problems. *Comptes Rendus Mathematique*, 2016.

[93] Y. Maday and O. Mula. A generalized empirical interpolation method: Application of reduced basis techniques to data assimilation. In *Springer INdAM Series*, volume 4, pages 221–235. Springer, 2013.

[94] Y. Maday, A. T. Patera, J. D. Penn, and M. Yano. A parameterized-background data-weak approach to variational data assimilation: Formulation, analysis, and application to acoustics. *International Journal for Numerical Methods in Engineering*, 102(5):933–965, 2015.

[95] Y. Maday and E. M. Ronquist. The Reduced Basis Element Method: Application to a Thermal Fin Problem. *SIAM Journal on Scientific Computing*, 2004.

[96] M. Malesa, D. Szczepanek, M. Kujawińska, A. As Wiercz, and P. Kołakowski. Monitoring of civil engineering structures using digital image correlation technique. In *EPJ Web of Conferences*, volume 6, 2010.

[97] A. Manzoni, S. Pagani, and T. Lassila. Accurate solution of Bayesian inverse uncertainty quantification problems combining reduced basis methods and reduction error models. *SIAM-ASA Journal on Uncertainty Quantification*, 4(1):380–412, 2016.

[98] B. Marchand. *Assimilation de données et recalage rapide de modèles mécaniques complexes*. PhD thesis, École Normale Paris-Saclay, 2017.

[99] B. Marchand, L. Chamoin, and C. Rey. 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*, 107(9):786–810, aug 2016.

[100] B. Marchand, L. Chamoin, and C. Rey. Parameter identification and model updating in the context of nonlinear mechanical behaviors using a unified formulation of the modified Constitutive Relation Error concept. *Computer Methods in Applied Mechanics and Engineering*, 345:1094–1113, mar 2019.

[101] S. Mariani and A. Corigliano. Impact induced composite delamination: State and parameter identification via joint and dual extended Kalman filters. *Computer Methods in Applied Mechanics and Engineering*, 194(50-52):5242–5272, 2005.

[102] S. Mariani and A. Ghisi. Unscented Kalman filtering for nonlinear structural dynamics. *Nonlinear Dynamics*, 49(1-2):131–150, 2007.

[103] Y. Marzouk, T. Moselhy, M. Parno, and A. Spantini. Sampling via measure transport: An introduction. In *Handbook of Uncertainty Quantification*, pages 785–825. Springer, feb 2017.

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

[105] Y. M. Marzouk, H. N. Najm, and L. A. Rahn. Stochastic spectral methods for efficient Bayesian solution of inverse problems. *Journal of Computational Physics*, 224(2):560–586, 2007.

[106] F. Mathieu, H. Leclerc, F. Hild, and S. Roux. Estimation of Elastoplastic Parameters via Weighted FEMU and Integrated-DIC. *Experimental Mechanics*, 55(1):105–119, 2015.

[107] H. G. Matthies, E. Zander, B. V. Rosić, A. Litvinenko, and O. Pajonk. Inverse problems in a Bayesian setting. In *Computational Methods in Applied Sciences*, volume 41, pages 245–286, nov 2016.

[108] H.-M. Nguyen, O. Allix, and P. Feissel. A robust identification strategy for rate-dependent models in dynamics. *Inverse Problems*, 24(6):065006, dec 2008.

[109] M. Nodet. Variational assimilation of Lagrangian data in oceanography. *Inverse Problems*, 22(1):245–263, 2006.

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

[111] J. T. Oden and S. Prudhomme. Estimation of modeling error in computational mechanics. *Journal of Computational Physics*, 182(2):496–515, 2002.

[112] C. Paillet, D. Néron, and P. Ladevèze. A door to model reduction in high-dimensional parameter space. *Comptes Rendus - Mecanique*, 346(7):524–531, 2018.

[113] O. Pajonk, B. V. Rosic, A. Litvinenko, and H. G. Matthies. A deterministic filter for non-Gaussian Bayesian estimation Applications to dynamical system estimation with noisy measurements. *Physica D: Nonlinear Phenomena*, 241(7):775–788, 2012.

[114] I. Papaioannou, C. Papadimitriou, and D. Straub. Sequential importance sampling for structural reliability analysis. *Structural Safety*, 62:66–75, 2016.

[115] M. D. Parno and Y. M. Marzouk. Transport map accelerated markov chain monte carlo. *SIAM-ASA Journal on Uncertainty Quantification*, 6(2):645–682, dec 2018.

[116] Pearson, K. On lines and planes of closets fit to systems of points in the space. *Philosophical Magazine*, 1901.

[117] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162, feb 1964.

[118] E. E. Prudencio, P. T. Bauman, D. Faghihi, K. Ravi-Chandar, and J. T. Oden. A computational framework for dynamic data-driven material damage control, based on Bayesian inference and model selection. *International Journal for Numerical Methods in Engineering*, 102(3-4):379–403, 2015.

[119] E. E. Prudencio, P. T. Bauman, S. V. Williams, D. Faghihi, K. Ravi-Chandar, and J. T. Oden. Real-time inference of stochastic damage in composite materials. *Composites Part B: Engineering*, 67:209–219, 2014.

[120] B. Richard and F. Ragueneau. Continuum damage mechanics based model for quasi brittle materials subjected to cyclic loadings: Formulation, numerical implementation and applications. *Engineering Fracture Mechanics*, 98(1):383–406, 2013.

[121] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer New York, New York, NY, 2004.

[122] G. O. Roberts, A. Gelman, and W. R. Gilks. Weak convergence and optimal scaling of random walk Metropolis algorithms. *Annals of Applied Probability*, 7(1):110–120, 1997.

[123] G. O. Roberts and J. S. Rosenthal. Optimal Scaling for Various Metropolis-Hastings Algorithms. Technical Report 4, 2001.

[124] B. V. Rosić, A. Kučerová, J. Sýkora, O. Pajonk, A. Litvinenko, and H. G. Matthies. Parameter identification in a probabilistic setting. *Engineering Structures*, 2013.

[125] D. A. Ross, J. Lim, R. S. Lin, and M. H. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008.

[126] G. Rozza, D. B. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: Application to transport and continuum mechanics. *Archives of Computational Methods in Engineering*, 15(3):229–275, 2008.

[127] P.-B. Rubio, L. Chamoin, and F. Louf. On-the-fly Bayesian data assimilation using Transport Maps sampling and PGD reduced models. *Springer, to be published*, 2019.

[128] P.-B. Rubio, L. Chamoin, and F. Louf. Real-time Bayesian data assimilation with model enrichment. *Comptes Rendus de l'Académie des Sciences - Mécanique, in preparation*, 2019.

[129] P. B. Rubio, F. Louf, and L. Chamoin. Fast model updating coupling Bayesian inference and PGD model reduction. *Computational Mechanics*, 62(6):1485–1509, apr 2018.

[130] P.-B. Rubio, F. Louf, and L. Chamoin. Bayesian data assimilation with Transport Map sampling and PGD model order reduction. *Journal of Physics : Conference Series*, 2019.

[131] P.-B. Rubio, F. Louf, and L. Chamoin. Transport Map sampling with PGD model reduction for fast dynamical Bayesian data assimilation. *International Journal for Numerical Methods in Engineering*, 2019.

[132] P. Sarkar. Sequential Monte Carlo Methods in Practice. *Technometrics*, 45(1):106–106, 2003.

[133] J. A. Scales and A. Gersztenkorn. Robust methods in inverse theory. *Inverse Problems*, 4(4):1071–1091, 1988.

[134] A. Shahmoradi. Multilevel Bayesian Parameter Estimation in the Presence of Model Inadequacy and Data Uncertainty. *arXiv preprint*, arXiv:1711, nov 2017.

[135] M. Signorini, S. Zlotnik, and P. Díez. Proper generalized decomposition solution of the parameterized Helmholtz problem: application to inverse geophysical problems. *International Journal for Numerical Methods in Engineering*, 109(8):1085–1102, 2017.

[136] S. Sinharay. Assessing convergence of the Markov Chain Monte Carlo Algorithms: A review. *ETS Research Report Series*, (1):1–52, 2003.

[137] F. Soleilhet. *Etudes expérimentales et numériques des matériaux cimentaires sous sollicitations hydro-mécaniques*. PhD thesis, École Normale Supérieure Paris-Saclay, 2018.

[138] A. Spantini, D. Bigoni, and Y. Marzouk. Inference via low-dimensional couplings. *Journal of Machine Learning Research*, 19, mar 2018.

[139] D. Straub, I. Papaioannou, and W. Betz. Bayesian analysis of rare events. *Journal of Computational Physics*, 314:538–556, 2016.

[140] A. M. Stuart. Inverse problems: A Bayesian perspective, 2010.

[141] M. Sutton, W. Wolters, W. Peters, W. Ranson, and S. McNeill. Determination of displacements using an improved digital correlation method. *Image and Vision Computing*, 1(3):133–139, 1983.

[142] O. Talagrand and P. Courtier. Variational Assimilation of Meteorological Observations With the Adjoint Vorticity Equation. I: Theory. *Quarterly Journal of the Royal Meteorological Society*, 113(478):1311–1328, 1987.

[143] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, jan 2005.

[144] V. Tsiolakis, M. Giacomini, R. Sevilla, C. Othmer, and A. Huerta. Nonintrusive proper generalised decomposition for parametrised incompressible flow problems in OpenFOAM. *arXiv preprint*, arXiv:1906, jun 2019.

[145] C. Villani. Optimal Transport: Old and New. *Media*, 2007.

[146] M. Vitse, D. Néron, and P. A. Boucard. Virtual charts of solutions for parametrized nonlinear equations. *Computational Mechanics*, 54(6):1529–1539, dec 2014.

[147] M. Vitse, D. Néron, and P.-A. Boucard. Dealing with a nonlinear material behavior and its variability through PGD models: Application to reinforced concrete structures. *Finite Elements in Analysis and Design*, 153:22–37, jan 2019.

[148] J. Waeytens, B. Rosić, P. E. Charbonnel, E. Merliot, D. Siegert, X. Chapeleau, R. Vidal, V. le Corvec, and L. M. Cottineau. Model updating techniques for damage detection in concrete beam using optical fiber strain measurement device. *Engineering Structures*, 129:2–10, 2016.

[149] E. A. Wan and R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium, AS-SPCC 2000*, 2000.

[150] W.-J. Yan and L. S. Katafygiotis. A novel Bayesian approach for structural model updating utilizing statistical modal information from multiple setups. *Structural Safety*, 52:260–271, 2015.

[151] J. N. Yang, S. Lin, H. Huang, and L. Zhou. An adaptive extended Kalman filter for structural damage identification. *Structural Control and Health Monitoring*, 13(4):849–867, 2006.

[152] X. Zou, M. Conti, P. Díez, and F. Auricchio. A nonintrusive proper generalized decomposition scheme with application in biomechanics. *International Journal for Numerical Methods in Engineering*, 113(2):230–251, 2018.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences mécaniques et énergétiques, matériaux et géosciences (SMEMAG)

Titre : Stratégies numériques innovantes pour l'assimilation de données par inférence bayésienne

Mots clefs : simulation numérique, réduction de modèle, recalage de modèle, inférence bayésienne, quantification d'incertitudes, temps réel

Résumé : Ce travail se place dans le cadre de l'assimilation de données en mécanique des structures. Il vise à développer de nouveaux outils numériques pour l'assimilation de données robuste et en temps réel afin d'être utilisés dans diverses activités d'ingénierie. Une activité cible est la mise en œuvre d'applications DDDAS (Dynamic Data Driven Application System) dans lesquelles un échange continu entre les outils de simulation et les mesures expérimentales est requis dans le but de créer une boucle de contrôle rétroactive sur des systèmes mécaniques connectés. Dans ce contexte, et afin de prendre en compte les différentes sources d'incertitude (erreur de modélisation, bruit de mesure,...), une méthodologie stochastique puissante est considérée dans le cadre général de l'inférence bayésienne. Cependant, un inconvénient bien connu d'une telle approche est la complexité informatique qu'elle engendre et qui rend les simulations en temps réel et l'assimilation séquentielle des données difficiles. Le travail de thèse propose donc de coupler l'inférence bayésienne avec des techniques numériques attrayantes et avancées afin d'envisager l'assimilation stochastique de données de façon séquentielle et en temps réel. Premièrement, la réduction de modèle PGD est introduite pour faciliter le calcul de la fonction de vraisemblance, la propagation des incertitudes dans des modèles complexes et l'échantillonnage de la densité a posteriori. Ensuite, l'échantillonnage par la méthode des Transport Maps est étudiée comme un substitut aux procédures classiques MCMC pour l'échantillonnage de la densité a posteriori. Il est démontré que cette technique conduit à des calculs déterministes, avec des critères de convergence clairs, et qu'elle est particulièrement adaptée à l'assimilation séquentielle de données. Là encore, l'utilisation de la réduction de modèle PGD facilite grandement le processus en utilisant les informations des gradients et hessiens d'une manière simple. Enfin, et pour accroître la robustesse, la correction à la volée du biais du modèle est abordée à l'aide de termes d'enrichissement fondés sur les données. Aussi, la sélection des données les plus pertinentes pour l'objectif d'assimilation est abordée. Cette méthodologie globale est appliquée et illustrée sur plusieurs applications académiques et réelles, comprenant par exemple le recalage en temps réel de modèles pour le contrôle des procédés de soudage, ou l'étude d'essais mécaniques impliquant des structures endommageables en béton instrumentées par mesures de champs.

Title: Development of innovative numerical strategies for Bayesian data assimilation

Keywords: numerical simulation, model reduction, Bayesian inference, uncertainty quantification, real-time simulation

Abstract: The work is placed into the framework of data assimilation in structural mechanics. It aims at developing new numerical tools in order to permit real-time and robust data assimilation that could then be used in various engineering activities. A specific targeted activity is the implementation of DDDAS (Dynamic Data Driven Application System) applications in which a continuous exchange between simulation tools and experimental measurements is envisioned to the end of creating retroactive control loops on mechanical systems. In this context, and in order to take various uncertainty sources (modeling error, measurement noise,..) into account, a powerful and general stochastic methodology with Bayesian inference is considered. However, a well-known drawback of such an approach is the computational complexity which makes real-time simulations and sequential assimilation some difficult tasks. The PhD work thus proposes to couple Bayesian inference with attractive and advanced numerical techniques so that real-time and sequential assimilation can be envisioned. First, PGD model reduction is introduced to facilitate the computation of the likelihood function, uncertainty propagation through complex models, and the sampling of the posterior density. Then, Transport Map sampling is investigated as a substitute to classical MCMC procedures for posterior sampling. It is shown that this technique leads to deterministic computations, with clear convergence criteria, and that it is particularly suited to sequential data assimilation. Here again, the use of PGD model reduction highly facilitates the process by recovering gradient and Hessian information in a straightforward manner. Eventually, and to increase robustness, on-the-fly correction of model bias is addressed using data-based enrichment terms. The overall cost-effective methodology is applied and illustrated on several academic and real-life test cases, including for instance the real-time updating of models for the control of welding processes, or that of mechanical tests involving damageable concrete structures with full-field measurements.

Université Paris–Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France